

**UNIVERSIDAD PONTIFICIA COMILLAS DE MADRID**

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)**

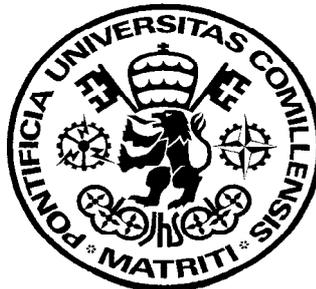
**Departamento de Electrónica y Automática**

**APLICACIÓN DE LA TEORÍA DE LA  
POSIBILIDAD AL ANÁLISIS DE  
PLANIFICABILIDAD DE SISTEMAS DE  
TIEMPO REAL**

Tesis para la obtención del grado de doctor

Director: Prof. Dr. D. José Villar Collado

Autora: D<sup>a</sup>. Asunción Cucala García



**Madrid 2003**



*A mis padres, a Jorge y a Juan*



## Resumen de la tesis

La tesis doctoral que aquí se presenta se ocupa, por un lado del análisis de planificabilidad de sistemas de tiempo real monoprocesador, con planificador expulsor y basado en prioridades, a partir del modelado de los parámetros temporales mediante distribuciones de posibilidad. Por otro lado, se ocupa de la forma en que se incorpora el modelado de la incertidumbre en el análisis de planificabilidad al proceso de diseño de los sistemas de tiempo real.

La aplicación del principio de extensión de la teoría de la posibilidad para extender funciones clásicas a parámetros borrosos está muy restringida debido al alto coste computacional que conlleva, pero gracias a los algoritmos de análisis de planificabilidad borrosa propuestos en esta tesis se pueden calcular los valores de posibilidad y de necesidad de que el sistema sea planificable de forma más eficiente que la aplicación del principio de extensión y con los mismos resultados, de manera que su aplicación en el proceso de diseño de los sistemas se hace viable.

Los algoritmos propuestos están basados en aritmética borrosa y en aritmética de intervalos y permiten calcular la planificabilidad sistemas con prioridades fijas, plazos menores, iguales o mayores que los periodos, recursos compartidos, retardo en la activación de las tareas, servidores aperiódicos y prioridades dinámicas asignadas según el algoritmo de planificación EDF. Mediante dichos algoritmos se calcula la posibilidad y la necesidad de que el sistema sea planificable cuando los tiempos de ejecución, los plazos, los periodos, los retardos en la activación de las tareas, los tiempos de bloqueo, o el periodo y la capacidad del servidor aperiódico se modelan mediante conjuntos borrosos, convexos o no convexos.

El modelado de la incertidumbre existente en las primeras fases de diseño es esencial para tener un modelo realista del sistema. Si no existen datos estadísticos acerca de los parámetros temporales de las tareas es más apropiada la utilización de distribuciones de posibilidad para modelar la incertidumbre, y además la aplicación de los algoritmos de cálculo de planificabilidad borrosa propuestos en esta tesis son más rápidos que la simulación del sistema a partir de funciones de densidad de probabilidad estimadas.

Por ello se ha incorporado el modelado de la incertidumbre mediante distribuciones de posibilidad en la extensión propuesta de la metodología de diseño de sistemas de tiempo real, de manera que es posible obtener más información acerca de la planificabilidad del sistema que si únicamente se modela y analiza la situación más desfavorable, y permite obtener diseños válidos del sistema que serían rechazados si únicamente se analizara dicho peor caso.



## **Agradecimientos**

Quisiera expresar en esta página mi más sincero agradecimiento a todas aquellas personas que de uno u otro modo han contribuido a la realización de esta tesis.

En primer lugar mi especial gratitud a mis padres a quienes les debo todo y nunca podré agradecer lo suficiente, a mi marido Jorge por su amor y la fuerza que me ha infundido, a mi hijo Juan que en este año, su primer año de vida, ha sido mi alegría, y a toda mi familia, en especial a mis hermanos y cuñados.

A mi director de tesis y amigo, José Villar del que he aprendido cada día y con quien ha sido un placer trabajar estos años, por su confianza y ayuda incondicional y por haber sabido guiar mi trabajo a lo largo de todo su desarrollo.

A Fernando de Cuadra, por su apoyo y confianza y por haber sido el inspirador de los trabajos de esta tesis y haber estado involucrado en las primeras fases del trabajo como supervisor y como director.

A todos mis amigos y compañeros del IIT, en especial a Yolanda por todos estos años de trabajo y amistad, de fatigas y alegrías pasadas juntas, a Pedro con el que he compartido tantas charlas sobre nuestras tesis y sobre la vida, a Antonio que ha sido primero compañero, luego supervisor y siempre un amigo, a Pablo que además de cuñado es amigo y compañero y a Alicia por su valiosa ayuda matemática. Y a mis queridos amigos Miguel, Silvia y Begoña. A todos ellos quiero agradecerles la amistad que siempre me han prestado y los ánimos infundidos en los momentos difíciles.

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia en su Programa para la Formación de Personal Investigador.



# Indice general

<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1. Descripción del problema.....	2
1.2. Planteamiento de la tesis .....	3
1.3. Organización de la exposición .....	4
<b>2. SISTEMAS DE TIEMPO REAL</b> .....	<b>9</b>
<b>2.1. Introducción</b> .....	<b>11</b>
2.1.1. Eventos.....	13
2.1.2. Tareas .....	15
2.1.3. Recursos de un sistema .....	16
2.1.4. Restricciones temporales.....	18
2.1.5. Utilización del procesador.....	21
<b>2.2. Modelo del sistema</b> .....	<b>23</b>
<b>2.3. Planificación basada en prioridades fijas</b> .....	<b>24</b>
2.3.1. Prioridades monótonas en frecuencia (RMS).....	26
2.3.2. Prioridades monótonas en plazo (DMS) .....	29
2.3.3. Análisis temporal de algoritmos con prioridades fijas .....	30
<b>2.4. Planificación basada en prioridades fijas, con recursos de datos compartidos</b> .....	<b>32</b>
2.4.1. Tareas que deben sincronizarse para compartir datos comunes .....	32
2.4.2. Cálculo de tiempos de bloqueo con el protocolo PIP.....	37
2.4.3. Límites de utilización .....	39
2.4.4. Algoritmo de análisis temporal con bloqueos y plazos menores que los periodos .....	40
2.4.5. Algoritmo de análisis temporal con bloqueos, plazos menores que los periodos y activación retrasada .....	40
2.4.6. Algoritmo de análisis temporal con bloqueos y plazos mayores que los periodos.....	41

<b>2.5. Sobrecargas .....</b>	<b>43</b>
<b>2.6. Relaciones de precedencia en tareas con prioridades fijas .....</b>	<b>44</b>
2.6.1. Transacciones.....	45
2.6.2. Análisis temporal .....	47
<b>2.7. Planificación basada en prioridades dinámicas: EDF .....</b>	<b>49</b>
2.7.1. Descripción del algoritmo EDF.....	49
2.7.2. Análisis temporal del algoritmo EDF.....	50
<b>2.8. Planificación con tareas esporádicas.....</b>	<b>51</b>
<b>2.9. Planificación con tareas aperiódicas .....</b>	<b>53</b>
2.9.1. Asignación de prioridades fijas en planificación con tareas aperiódicas .....	54
2.9.2. Asignación dinámica de prioridades .....	58
<b>2.10. Incertidumbre en el análisis de sistemas de tiempo real.....</b>	<b>59</b>
<b>3. PLANIFICABILIDAD BORROSA DE SISTEMAS DE TIEMPO REAL .</b>	<b>65</b>
<b>3.1. Conceptos básicos de la Teoría de la Posibilidad.....</b>	<b>68</b>
3.1.1. Conjuntos borrosos .....	70
3.1.2. Teoría de la Evidencia.....	80
<b>3.2. Planificabilidad borrosa.....</b>	<b>85</b>
<b>3.3. Modelado borroso de parámetros .....</b>	<b>91</b>
3.3.1. Recogida de información de los expertos .....	91
3.3.2. Interpretación del modelado borroso de los parámetros .....	95
<b>3.4. Cálculo de la planificabilidad borrosa: principio de extensión y representación gráfica de los cuerpos de evidencia.....</b>	<b>97</b>
3.4.1. Representación gráfica del cuerpo de evidencia conjunto para el cálculo de la distribución de plausibilidad de una función de dos argumentos .....	97
3.4.2. Algoritmo RMS.....	110
3.4.3. Algoritmo DMS .....	120

<b>3.5. Cálculo de planificabilidad borrosa mediante aproximaciones de los tiempos de finalización .....</b>	<b>125</b>
3.5.1. Definición de aproximación válida de un conjunto borroso para el análisis de planificabilidad .....	126
3.5.2. Aproximación válida convexa de los tiempos de finalización borrosos .....	129
<b>4. PRINCIPIO DE EXTENSIÓN DE FUNCIONES MONÓTONAS. APROXIMACIONES.....</b>	<b>139</b>
<b>4.1. Aproximaciones válidas de un conjuntos borroso .....</b>	<b>141</b>
<b>4.2. Aproximación mediante análisis de intervalos.....</b>	<b>142</b>
4.2.1. Funciones continuas .....	142
4.2.2. Funciones discontinuas .....	144
4.2.3. Extensión de funciones monótonas con parámetros modelados como conjuntos borrosos no convexos .....	152
<b>4.3. Aproximación mediante aritmética borrosa .....</b>	<b>154</b>
4.3.1. Extensión de funciones continuas mediante aritmética borrosa.....	154
4.3.2. Extensión de funciones discontinuas mediante aritmética borrosa .....	157
4.3.3. Redondeo borroso hacia arriba.....	160
<b>5. ANÁLISIS BORROSO DE PLANIFICABILIDAD EN SISTEMAS CON PRIORIDADES FIJAS.....</b>	<b>173</b>
<b>5.1. Algoritmo de análisis de planificabilidad con plazos menores que los periodos .....</b>	<b>175</b>
<b>5.2. Extensión del análisis de planificabilidad con plazos menores que los periodos.....</b>	<b>180</b>
5.2.1. Principio de extensión .....	180
5.2.2. Aritmética borrosa.....	184
5.2.3. Aritmética de intervalos .....	193
5.2.4. Extensión del análisis para periodos, tiempos de bloqueo y retraso de la activación borrosos .....	205
<b>5.3. Asignación óptima de prioridades fijas con parámetros borrosos .....</b>	<b>209</b>
<b>5.4. Comparación con la extensión propuesta por Litoiu y Tadei.....</b>	<b>215</b>

<b>5.5. Comparación entre el análisis de planificabilidad posibilista y el probabilista .....</b>	<b>222</b>
<b>5.6. Extensión del análisis de planificabilidad con plazos mayores que los periodos.....</b>	<b>228</b>
5.6.1. Extensión del algoritmo a tiempos de ejecución y plazos borrosos. ....	234
5.6.2. Extensión del algoritmo con tiempos de bloqueo y retraso de la activación.....	244
5.6.3. Extensión del algoritmo con periodos borrosos .....	249
<b>5.7. Extensión del algoritmo de cálculo del tiempo de bloqueo para el protocolo PIP .....</b>	<b>252</b>
<b>5.8. Planificabilidad de sistemas con servidores aperiódicos .....</b>	<b>253</b>
5.8.1. Servidor Esporádico .....	254
5.8.2. Servidor Aplazable.....	256
<b>6. ANÁLISIS BORROSO DE PLANIFICABILIDAD EN SISTEMAS CON PRIORIDADES DINÁMICAS: EDF .....</b>	<b>261</b>
<b>6.1. Plazos iguales a los periodos .....</b>	<b>262</b>
6.1.1. Ecuación de planificabilidad de EDF en función del hiperperiodo.....	264
6.1.2. Extensión del análisis de planificabilidad a tiempos de ejecución borrosos .....	265
6.1.3. Extensión del análisis de planificabilidad a periodos borrosos.....	269
<b>6.2. Plazos menores que los periodos .....</b>	<b>274</b>
6.2.1. Extensión del análisis de planificabilidad a tiempos de ejecución borrosos .....	277
6.2.2. Extensión del análisis de planificabilidad a tiempos de ejecución, periodos y plazos borrosos .....	281
6.2.3. Posibilidad y necesidad total cuando existe orden fuerte en los plazos .....	292
<b>7. METODOLOGÍA DE DISEÑO DE SISTEMAS DE TIEMPO REAL. CASO EJEMPLO.....</b>	<b>303</b>
<b>7.1. Extensión de la metodología de diseño de sistemas de tiempo real .....</b>	<b>304</b>
7.1.1. Descripción del entorno, del sistema, del interfaz y requisitos del sistema .....	306
7.1.2. Diseño y modelado del sistema: tareas y planificación.....	306
7.1.3. Análisis de planificabilidad.....	314
7.1.4. Modificación del diseño.....	331

<b>7.2. Caso ejemplo .....</b>	<b>333</b>
7.2.1. Descripción del entorno, del sistema, del interfaz y requisitos del sistema.....	333
7.2.2. Modelo de las tareas con tiempos crisp y planificación del sistema .....	337
7.2.3. Modificación del diseño: nueva asignación de prioridades.....	352
7.2.4. Modificación del diseño: introducción de incertidumbre en algún plazo .....	356
7.2.5. Modificación del diseño: plazos, tiempos de ejecución y tiempos de utilización de memoria borrosos .....	358
7.2.6. Modificación del modelo: periodo de la tarea de envío de informes de estado borroso	370
7.2.7. Resumen de resultados de planificabilidad .....	374
7.2.8. Modificación del diseño: mejora del modelo a partir del análisis borroso de planificabilidad .....	375
7.2.9. Conclusiones .....	376
<b>8. CONCLUSIONES, APORTACIONES Y FUTUROS DESARROLLOS</b>	<b>379</b>
<b>8.1. Conclusiones.....</b>	<b>380</b>
<b>8.2. Aportaciones.....</b>	<b>384</b>
8.2.1. Conceptos y propiedades matemáticas.....	385
8.2.2. Extensión de la metodología de diseño de sistemas de tiempo real con incertidumbre	386
8.2.3. Análisis de planificabilidad .....	387
8.2.4. Teoría de la evidencia .....	389
<b>8.3. Futuros desarrollos .....</b>	<b>390</b>
<b>A. ANEXO: REPRESENTACIÓN GRÁFICA DEL CUERPO DE EVIDENCIA .....</b>	<b>393</b>
<b>A.1. Teoría de las sombras .....</b>	<b>394</b>
<b>A.2. Representación gráfica .....</b>	<b>401</b>
A.2.1. Distribución conjunta de posibilidad. Teoría de Posibilidad.....	402
A.2.2. Distribución conjunta de plausibilidad. Teoría de la Evidencia.....	405
A.2.3. Principio de extensión .....	421
A.2.4. Interpretación de la relación entre $\alpha_A$ y $\alpha_B$ .....	428

<b>B.</b>	<b>ANEXO: COMPARACIÓN DE DOS CONJUNTOS BORROSOS .....</b>	<b>433</b>
<b>B.1.</b>	<b>Definición de periodos de tiempo antes y después de una fecha .....</b>	<b>434</b>
<b>B.2.</b>	<b>Ordenación de fechas.....</b>	<b>436</b>
B.2.1.	Indices de ordenación.....	437
B.2.2.	Otros índices .....	439
<b>B.3.</b>	<b>Posibilidad y necesidad de que una fecha esté antes que otra.....</b>	<b>443</b>
B.3.1.	Conjuntos convexos .....	443
B.3.2.	Conjuntos no convexos .....	446
<b>C.</b>	<b>ANEXO: PROBABILIDAD Y POSIBILIDAD.....</b>	<b>451</b>
<b>C.1.</b>	<b>Probabilidad y posibilidad .....</b>	<b>452</b>
<b>C.2.</b>	<b>Interpretaciones probabilistas de las distribuciones de posibilidad .....</b>	<b>454</b>
C.2.1.	Cotas de probabilidad.....	454
C.2.2.	Conjuntos aleatorios y Teoría de la Evidencia.....	456
C.2.3.	Función de verosimilitud.....	458
C.2.4.	Interpretación probabilista de las operaciones con conjuntos borrosos .....	460
<b>C.3.</b>	<b>Interpretaciones no probabilistas de las distribuciones de posibilidad.....</b>	<b>461</b>
C.3.1.	Posibilidad como preferencia .....	461
C.3.2.	Posibilidad como similitud.....	462
<b>C.4.</b>	<b>Transformaciones Posibilidad-Probabilidad.....</b>	<b>462</b>
C.4.1.	Transformaciones basadas en los principios de máxima especificidad y razón insuficiente .....	463
C.4.2.	Otras transformaciones.....	467
	<b>BIBLIOGRAFÍA.....</b>	<b>471</b>

## Lista de los símbolos más importantes

$b_i$	tiempo de bloqueo de una tarea $i$
$c_s$	capacidad de un servidor aperiódico
$d_i$	plazo de ejecución relativo o “deadline” relativo de la tarea $i$
$d_{i,j}$	plazo de ejecución absoluto (“deadline” absoluto) de la tarea $i$ en su ejecución $j$
$h$	hiperperiodo
$j_i$	“release jitter” o retardo en la activación de la tarea $i$
$m$	asignación básica de probabilidad de un cuerpo de evidencia
$p_i$	tiempo de ejecución de la tarea $i$
$r_{i,j}$	tiempo listo de la tarea $i$ en su ejecución $j$
$ru_i$	tiempo de utilización del recurso de datos compartido por parte de la tarea $i$
$t_i$	periodo de la tarea $i$
$t_s$	periodo del servidor aperiódico
$wc_i$	peor tiempo de respuesta de la tarea $i$
$wc_{i,j}$	tiempo de finalización de la tarea $i$ en su ejecución $j$
$A$	conjunto borroso, salvo que se indique de forma explícita que es un conjunto clásico
$A_i$	elemento focal obtenido como $\alpha$ -corte del conjunto borroso $A$
$\underline{a}(\alpha)$	límite inferior del $\alpha$ -corte del conjunto borroso $A$ de nivel $\alpha$
$\bar{a}(\alpha)$	límite superior del $\alpha$ -corte del conjunto borroso $A$ de nivel $\alpha$

$(F, m)$  cuerpo de evidencia, donde  $F$  es el conjunto de elementos focales y  $m$  es la asignación básica de probabilidad

$\mu_A$  función de pertenencia del conjunto borroso  $A$

$\pi_A$  distribución de posibilidad asociada al conjunto borroso  $A$

$\alpha/b$  notación para conjuntos borrosos discretos, donde  $\alpha$  es el grado de pertenencia de  $b$  al conjunto borroso

La notación decimal empleada en esta tesis es la notación americana, que emplea el punto en lugar de la coma.

# **1. Introducción**

## 1.1. Descripción del problema

Hoy en día existen en nuestro entorno multitud de sistemas de tiempo real, en contextos tan distintos como la regulación y protección ferroviaria, sistemas de cajeros automáticos, simuladores de vuelo, sistemas de monitorización y diagnóstico de instalaciones industriales, etc. Todos estos sistemas se caracterizan porque tienen que cumplir determinadas restricciones temporales, lo que no es sinónimo de rapidez de respuesta. En los ejemplos anteriores hay sistemas muy lentos como los cajeros automáticos, en comparación con otros más rápidos como el sistema de protección automática de un tren. Simplemente, cada sistema debe cumplir las restricciones temporales especificadas para que su funcionamiento sea el esperado.

Cuando un sistema cumple todas sus restricciones temporales se dice que es planificable. El programa planificador decide qué tarea se debe ejecutar en cada momento, según la política de planificación escogida. Por ejemplo, un tipo de planificación consiste en diseñar el orden en que las tareas van a ser despachadas antes de la ejecución y posteriormente llevarla a cabo en tiempo real. El problema de estas planificaciones es que son muy rígidas y requieren un rediseño completo cuando existe algún cambio, como por ejemplo la incorporación de una nueva tarea. Por otro lado, las planificaciones basadas en prioridades permiten que la decisión acerca de qué tarea debe ser ejecutada se tome en tiempo real, de acuerdo con la prioridad de cada tarea, y son mucho más flexibles frente a la introducción de modificaciones como la adición o eliminación de tareas. Las mismas tareas pertenecientes a un sistema de tiempo real pueden ser planificables o no planificables dependiendo del planificador elegido.

Existe un gran número de ecuaciones y algoritmos propuestos en la literatura para comprobar la planificabilidad de un sistema, supuesto que los parámetros temporales de las tareas y la asignación de prioridades se conocen de forma precisa. Sin embargo, a menudo ésta no es una premisa realista, sobre todo si el sistema está en fase de diseño o desarrollo.

Es interesante, desde la fase de diseño de un sistema y durante todo el proceso de desarrollo, poder comprobar si el diseño es planificable, ya que resultan mucho menos costosas las rectificaciones en dichas fases iniciales, que las debidas a fallos detectados durante la fase de pruebas cuando el sistema ya ha sido desarrollado de forma completa. El inconveniente es que en dichas fases de desarrollo no existe información precisa acerca de los parámetros temporales que describen a las tareas del sistema y hay que introducir incertidumbre en el modelado de los parámetros.

Existen distintas alternativas para representar la incertidumbre. La teoría de la probabilidad es una herramienta idónea para formalizar la incertidumbre cuando la evidencia se basa en series suficientemente grandes de experimentos aleatorios

independientes ([Klir, Yuan, 1995]). Sin embargo, cuando el sistema está en fase de diseño no suele existir este tipo de información.

En 1965 Zadeh ([Zadeh, 1965]) sentó las bases del desarrollo de la teoría de los conjuntos borrosos, que más tarde llevó a la aparición de la teoría de la posibilidad. Zadeh introdujo los conjuntos borrosos como conjuntos con límites imprecisos, de manera que la pertenencia a un conjunto dejó de ser una afirmación o una negación para pasar a ser un grado de pertenencia. La teoría de la posibilidad modela un tipo diferente de incertidumbre que la modelada por la teoría de la probabilidad. Las teorías de conjuntos borrosos y de la posibilidad se han ido mostrando como unas de las mejores herramientas de representación del conocimiento humano, debido a su gran riqueza semántica y flexibilidad para expresar conceptos del lenguaje hablado. Por otro lado, su soporte matemático hace que estas teorías sean uno de los mejores lazos de unión entre las técnicas numéricas de representación del conocimiento y la incertidumbre, y las técnicas de representación del conocimiento humano ([Villar, 1997]).

La teoría de conjuntos borrosos es idónea para formalizar información incompleta expresada en términos de proposiciones borrosas del tipo “el tiempo de ejecución de esta tarea es aproximadamente 10”. El experto que está diseñando el sistema puede considerar que 10 no es el único valor posible, sino que en torno a dicho valor hay otros que también son posibles, aunque en menor grado, por ejemplo entre 9 y 11. Así, se puede modelar el tiempo de ejecución de dicha tarea mediante un conjunto borroso en el que 10 tiene el máximo grado de pertenencia a ese conjunto y en torno a él van decreciendo hasta que en 9 y valores menores, y 11 y valores mayores el grado de pertenencia se hace cero.

El objetivo principal de esta tesis es la extensión del análisis de planificabilidad de sistemas de tiempo real basados en prioridades a parámetros temporales borrosos y el desarrollo de una metodología de diseño de este tipo de sistemas con un modelado adecuado de la incertidumbre mediante conjuntos borrosos, teniendo en cuenta el tipo de información que realmente existe durante la fase de diseño.

## 1.2. Planteamiento de la tesis

La tesis se centra en la extensión de los análisis clásicos de planificabilidad de sistemas de tiempo real, con un solo procesador, basados en prioridades y con tareas expulsables, a parámetros temporales borrosos. Como resultado del análisis borroso de planificabilidad se obtiene un grado en el que el sistema es planificable, en lugar de una negación total o una afirmación total. Esta tesis tiene como objetivo la aplicación rigurosa de la teoría de la posibilidad para el tratamiento de la

incertidumbre, por tanto el grado en el que un sistema es planificable se da en forma de medidas definidas dentro de dicha teoría: las medidas de posibilidad y necesidad.

La teoría de la posibilidad define, además de dichas medidas, la forma de extender una función de argumentos reales a argumentos borrosos mediante la regla denominada principio de extensión. El inconveniente de aplicar dicho principio al caso que nos ocupa es el elevado coste computacional que conlleva y que se puede hacer inviable incluso con un número moderado de tareas en estudio. Por ello se pretende encontrar métodos alternativos más rápidos pero que produzcan los mismos resultados que la aplicación del principio de extensión.

En esta tesis se va a probar que, si los algoritmos de planificación clásicos cumplen determinadas propiedades, es posible extenderlos utilizando algoritmos basados en aritmética de intervalos, o basados en aritmética borrosa, de forma eficiente y con los resultados deseados.

Mediante la metodología de diseño de sistemas de tiempo real basados en prioridades que se propone en esta tesis, se puede representar la incertidumbre en el modelo extrayendo la máxima información posible acerca de la planificabilidad del sistema, lo que permite modificar el diseño para que cumpla las restricciones temporales e incluso para mejorar sus prestaciones funcionales. Esta metodología se adapta mejor al tipo de conocimiento presente en la fase de diseño del sistema que la basada en el análisis temporal del sistema en el peor caso, y nos permite, no sólo detectar incumplimientos de restricciones, sino también validar diseños que con el simple análisis del peor caso serían rechazados.

### **1.3. Organización de la exposición**

El punto de partida de la exposición (capítulo 2) comienza con una breve introducción a los sistemas de tiempo real, enfocada sobre todo a aquellos cuya política de planificación se basa en la asignación de prioridades a las tareas, ya sean prioridades fijas o prioridades dinámicas que van cambiando a lo largo de la ejecución del sistema. Se define el concepto clásico de planificabilidad, que es extendido en el capítulo 3 a planificabilidad borrosa cuando los parámetros temporales del sistema se modelan como conjuntos borrosos.

En dicho capítulo 3 se propone medir la planificabilidad borrosa del sistema mediante la posibilidad y la necesidad de que todas las tareas cumplan sus restricciones temporales, para lo cual es preciso obtener los tiempos de finalización

borrosos de las tareas. Una vez obtenidos, se comparan con las restricciones temporales correspondientes para determinar la posibilidad y necesidad citadas (en el anexo B se describen algunos de los índices de ordenación de fechas basados en la comparación de conjuntos borrosos). Para obtener los tiempos de finalización borrosos se siguen dos métodos: la aplicación del principio de extensión de la teoría de la posibilidad y la combinación de la información incierta a través de sus cuerpos de evidencia. Para la aplicación de este segundo método se ha desarrollado un procedimiento gráfico descrito en el anexo A.

Mediante ejemplos muy sencillos y gracias a la representación gráfica propuesta, se observa que es posible obtener los mismos resultados de planificabilidad borrosa del principio de extensión, aplicando aritmética de intervalos, método que es mucho más rápido. Los tiempos borrosos de finalización de las tareas que se obtienen mediante aritmética de intervalos son sólo aproximaciones de los obtenidos mediante el principio de extensión, pero ese tipo de aproximaciones proporcionan los mismos valores de posibilidad y de necesidad de que las tareas respeten sus restricciones temporales, y por tanto, los mismos valores de planificabilidad borrosa.

Con el fin de generalizar esta conclusión, en el capítulo 4 se establecen las condiciones que debe cumplir cualquier función, continua o no para que su extensión a parámetros borrosos mediante aritmética borrosa o mediante aritmética de intervalos, proporcione este tipo de aproximaciones borrosas.

En el capítulo 5 se comprueba que todos los algoritmos de planificabilidad analizados para sistemas basados en prioridades fijas cumplen las propiedades citadas, por lo que es posible extender los algoritmos clásicos a parámetros borrosos utilizando, tanto aritmética borrosa, como aritmética de intervalos. Así pues se proponen algoritmos de ambos tipos para todos ellos.

Se han encontrado muy pocos estudios en la literatura que aborden el problema de la extensión de los análisis de planificabilidad a parámetros borrosos, reducidos sólo a uno o dos parámetros, y casi siempre con resultados más pesimistas que la aplicación del principio de extensión. Por el contrario, en esta tesis se busca que los resultados de planificabilidad obtenidos por la aplicación de los algoritmos propuestos basados en aritmética de intervalos y en aritmética borrosa coincidan con los que proporciona el principio de extensión.

En el mismo capítulo 5 se presenta, además, una comparación entre los resultados de planificabilidad que se obtienen modelando los parámetros como números borrosos frente a los obtenidos al modelarlos como distribuciones de probabilidad, concluyendo que se obtienen resultados muy similares, y que el camino de la teoría de la posibilidad es significativamente más rápido. Algunas de las transformaciones propuestas en la literatura entre distribuciones de posibilidad y distribuciones de

probabilidad aparecen descritas en el anexo C, así como un estudio comparativo entre ambos conceptos.

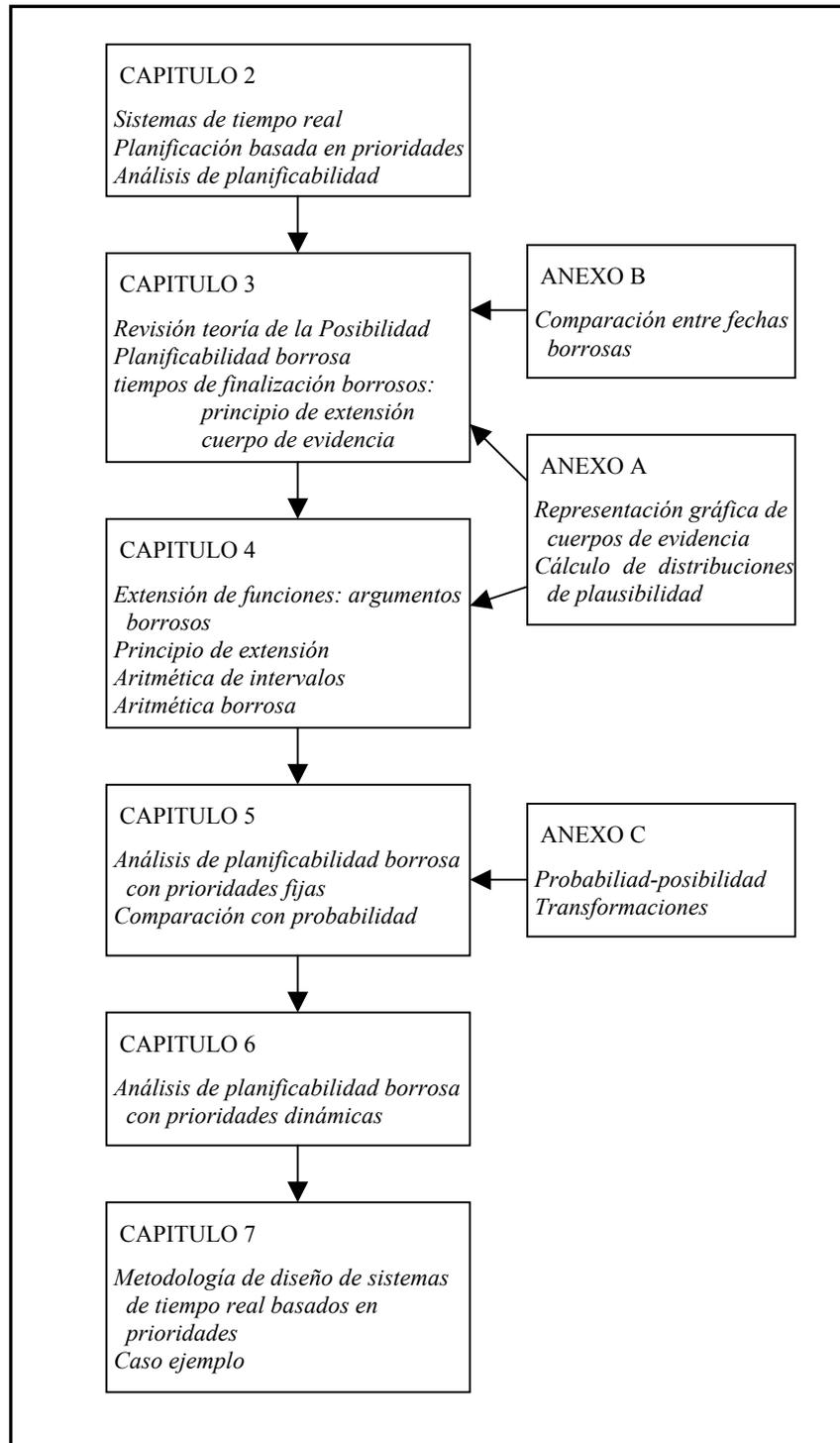
También se extiende en el capítulo 5 el concepto de asignación óptima de prioridades cuando se consideran parámetros borrosos, y se establecen las condiciones que deben cumplirse en las restricciones temporales borrosas para que una determinada asignación sea óptima.

Así como el capítulo 5 se centra en sistemas basados en prioridades fijas, en el capítulo 6 se estudia la planificabilidad de sistemas planificados mediante el algoritmo de prioridades dinámicas EDF (“Earliest Deadline First”). En uno de los casos estudiados en este capítulo se llega a la conclusión de que no se puede extender el análisis de planificabilidad a determinados parámetros borrosos aplicando los dos métodos propuestos: aritmética borrosa y aritmética de intervalos, ya que los resultados difieren de los del principio de extensión, por lo que en este caso sería preciso aplicar el principio de extensión para obtener la planificabilidad borrosa del sistema.

En el capítulo 7 se propone una metodología de diseño de sistemas de tiempo real basados en prioridades, en cuya fase de análisis de planificabilidad del sistema se recopilan los resultados obtenidos en capítulos anteriores. Esta metodología se aplica a un caso ejemplo basado en una parte del sistema de protección automática ferroviario. Mediante dicho ejemplo se muestra las ventajas obtenidas del modelado adecuado de la incertidumbre frente al análisis clásico del peor caso.

La representación propuesta en el anexo A permite calcular de forma gráfica la distribución de plausibilidad conjunta asociada a un cuerpo de evidencia conjunto y la distribución de plausibilidad de una función cualquiera que relacione los dos argumentos de partida. En dicho anexo A se describe la representación basada en la teoría de las sombras o “falling shadows” y se muestran las diferencias con la representación propuesta en esta tesis, así como las aportaciones de la representación propuesta en el cálculo de distribuciones de plausibilidad.

La siguiente figura resume gráficamente el contenido de los capítulos de la tesis:





## **2. Sistemas de Tiempo Real**

En este capítulo se describen los elementos que caracterizan a un sistema de tiempo real y algunas de las políticas de planificación de tareas basadas en prioridades, así como los principales algoritmos de análisis de planificabilidad de los sistemas.

En primer lugar, en el apartado 2.1 se presenta una definición de sistema de tiempo real y se describen algunos elementos básicos de este tipo de sistemas como son los eventos, tareas, los recursos del sistema, el principal de los cuales es el procesador que es asignado a las tareas según una política de planificación, las restricciones temporales y la utilización del procesador.

Una vez presentado en el apartado 2.2 el modelo del sistema que se va a estudiar, en el siguiente apartado se describen dos políticas de planificación de tareas basadas en prioridades fijas, el algoritmo de planificación de prioridades monótonas en frecuencia, y el algoritmo de prioridades monótonas en plazo, así como análisis de planificabilidad encontrados en la literatura para sistemas con planificación basada en prioridades fijas.

Cuando las tareas comparten recursos de datos se deben utilizar protocolos de sincronización para coordinar el acceso a los recursos. En el apartado 2.4 se describen algunos protocolos de sincronización, y como ejemplo se presenta la forma de calcular los tiempos de bloqueo que sufren las tareas cuando el protocolo utilizado es PIP. Además, se describen diversos análisis de planificabilidad, unos basados en la utilización total del procesador, y otros en el cálculo de los tiempos de finalización de las tareas, teniendo en cuenta tiempos de bloqueo, “release jitter” o retrasos de la activación de las tareas, y que los plazos de ejecución puedan ser menores o mayores que los periodos.

En el apartado 2.5 se presenta la forma de modelar las sobrecargas causadas por la manipulación de colas de tareas y por la interrupción de reloj, así como las causadas por la interrupción que se utiliza para atender a tareas esporádicas, y cómo se pueden tener en cuenta en el análisis de planificabilidad del sistema. Así mismo en el apartado 2.6 se muestran tres posibles formas de implementar las relaciones de precedencia entre las tareas con prioridades fijas y el análisis temporal correspondiente.

Tras el estudio de sistemas con planificación basada en prioridades fijas, en el apartado 2.7 se describe el algoritmo de planificación basado en prioridades dinámicas EDF, así como el análisis temporal correspondiente.

A continuación, en los apartados 2.8 y 2.9 se presenta la forma de implementar sistemas con tareas esporádicas y con tareas aperiódicas y por último, en el apartado 2.10 se hace un breve repaso de los estudios encontrados en la literatura donde se introduce la incertidumbre en el análisis de planificabilidad de los sistemas de tiempo real.

## 2.1. Introducción

El comportamiento temporal es importante en todos los sistemas de computación. Sin embargo, lo que distingue a los sistemas de tiempo real es la inclusión del comportamiento temporal en la especificación del sistema, y por tanto en lo que se considera correcto en ese sistema. Por ello, los sistemas de tiempo real son aquellos en los que la corrección del sistema depende no sólo de los resultados lógicos de la computación, sino también del tiempo en el que se producen dichos resultados ([Klein et al., 1993]).

La planificabilidad es un atributo del sistema de tiempo real que describe su capacidad para satisfacer sus requisitos temporales, y que se puede comprobar mediante el análisis temporal. El análisis temporal de planificabilidad es un conjunto de técnicas matemáticas que permiten analizar un sistema y su software off-line, sin necesidad de ejecutar dicho sistema ([Grigg, Audsley, 1999b]). Como consecuencia, se garantizan los requisitos temporales del tiempo de ejecución en el peor caso y se dice que el sistema es planificable. Las ventajas que se obtienen de aplicar las técnicas de análisis temporal son las siguientes:

- Cálculo off-line del comportamiento temporal en el peor caso.
- Cálculo off-line de la utilización de recursos (como la memoria) en el peor caso.
- Análisis de sensibilidad, variando la arquitectura, la asignación del software, etc. sin necesidad de construir el sistema.
- Análisis temporal de alto nivel. Antes de llegar a la fase de codificación dentro del proceso de desarrollo del sistema, el análisis temporal puede predecir si se van a cumplir los requisitos temporales y de utilización de recursos. Este análisis se puede aplicar en las diversas fases del ciclo de vida del sistema.
- Evidencia analítica que puede servir para apoyar la certificación del producto.

Este tipo de análisis es de especial importancia en los sistemas críticos de seguridad donde es necesario demostrar a una autoridad certificadora que se cumplen los requisitos temporales y de utilización de recursos antes de la operación del sistema.

Se pueden citar algunas herramientas de modelado y análisis de aplicaciones de tiempo real como MAST ([González Harbour et al., 2002]), desarrollada en la Universidad de Cantabria, preparada para realizar el análisis de planificabilidad del sistema basado en prioridades, la herramienta Spat (Single Processor Analysis Tool),

desarrollada en la universidad de York, que implementa ecuaciones de análisis de planificabilidad de sistemas con prioridades fijas, el proyecto europeo CARTS (“Computer Aided Architectural Analysis of Real-Time Systems”), para la descripción de arquitecturas software de tiempo real y realizar un análisis temporal de las mismas, desarrollado parcialmente en la Universidad Politécnica de Madrid (DIT) (<http://www.tcpsi.es/carts/>), o la herramienta HRT-HOOD, para el diseño de aplicaciones de tiempo real, que también permite realizar un análisis de planificabilidad de estos sistemas ([Burns, Wellings, 1989]).

A modo de introducción en este apartado se definen y describen algunos conceptos relacionados con los sistemas de tiempo real, como son los eventos, recibidos del entorno o bien emitidos por el propio sistema, o las tareas asociadas a dichos eventos que se encargan de realizar las funciones requeridas al sistema. También se describen los recursos del sistema, de los cuales el principal es la CPU: la forma de asignar este recurso a las tareas es mediante la planificación. Además, se describen dentro de esta introducción las restricciones temporales que caracterizan a un sistema de tiempo real, y el concepto de utilización del procesador, empleado para realizar uno de los posibles análisis temporales del sistema. Otros análisis se basan en el cálculo de los tiempos de finalización de las tareas.

Un sistema de tiempo real típicamente monitoriza y controla ciertos aspectos de su entorno. Algunos ejemplos industriales complejos podrían ser las estaciones espaciales, las fábricas automatizadas, los sistemas de control y comandos, y los aviones. En general suele haber requisitos temporales asociados a la monitorización y control de cada uno de esos aspectos, requisitos temporales para las ejecuciones que se deben realizar con el fin de controlar o monitorizar el entorno. Un escenario típico se da cuando el sistema de tiempo real está esperando la llegada de un estímulo. Cuando éste llega, la computación puede comenzar, y normalmente el trabajo debe terminarse en un tiempo límite (denominado “deadline” o plazo de ejecución) relativo al instante en el que se produjo el estímulo. Este escenario se puede repetir periódicamente durante toda ejecución del sistema.

Se denomina evento a la llegada de un estímulo asociado a un aspecto del entorno. Como el estímulo puede ocurrir repetidamente, la sucesión de eventos se denomina secuencia de eventos (una tarea repetida periódicamente sería la respuesta de una secuencia de eventos periódicos).

### 2.1.1. Eventos

El origen de un evento puede ser el entorno, en cuyo caso se denomina evento del entorno (exógeno), o bien el propio sistema, siendo entonces un evento interno (endógeno). Cuando el paso del tiempo provoca un evento, éste se denomina evento temporal. Por ejemplo, el control de la temperatura de un reactor puede requerir que se lea el sensor de temperatura cada 100 ms. Por tanto, después de 100 ms de una lectura se genera un evento temporal para realizar la siguiente lectura. Si el sistema detecta una temperatura elevada, se genera una alarma (evento interno); además el operador del sistema puede realizar en cualquier momento una petición de informe de estado del sistema (evento externo).

Las llegadas de una secuencia de eventos se caracterizan por su **patrón de llegadas** y por el **modo** del sistema durante el que se producen los eventos. Ambos identifican el momento en el que el evento estará compitiendo por los recursos del sistema.

En muchos sistemas de tiempo real puede ocurrir que las funciones que realiza el sistema cambien a lo largo del tiempo: cambia el modo del sistema y con él las funciones suministradas. Un modo es simplemente un conjunto de eventos. Cada evento pertenece, al menos, a un modo. Por ejemplo, un avión puede tener un conjunto de eventos para el modo “despegue”, otro para “crucero” y otro para el modo “evitar colisión”.

Para una secuencia de eventos es crítico poder caracterizar el patrón de ocurrencia del evento en función del tiempo. Normalmente, el patrón de llegadas se clasifica como periódico, irregular, acotado, en ráfaga (“bursty”), o no acotado ([Klein et al., 1993]).

#### *Patrón de llegadas periódico*

Un patrón de llegadas de eventos es *periódico* cuando, dada una ocurrencia de un evento en  $tm_1$ , el siguiente evento ocurre en  $tm_1+t \pm \Delta tm$ , donde  $t$  es constante y  $\Delta tm$  representa una desviación aleatoria, normalmente pequeña (por ejemplo  $0.01 t$ ), de la llegada periódica. Para eventos periódicos,  $t$  es el periodo del evento. De forma similar, para tareas periódicas (la respuesta asociada a los eventos periódicos),  $t$  es el periodo.

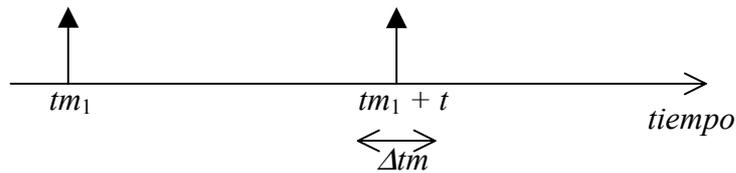


Figura 2.1: Tiempo entre llegadas consecutivas

### Patrón de llegadas irregular

También es posible que  $t$  sea diferente para cada llegada pero conocido de antemano. Cuando ocurre esto, el patrón de llegadas se denomina *irregular*. Un patrón de llegadas irregular se caracteriza por una secuencia de intervalos. Para llegadas periódicas e irregulares es posible predecir de forma determinista (con una pequeña tolerancia) cuándo llegará el siguiente evento.

En el caso de que  $\Delta tm=0$ , se ve en la Figura 2.2 la diferencia entre las llegadas periódicas y las irregulares

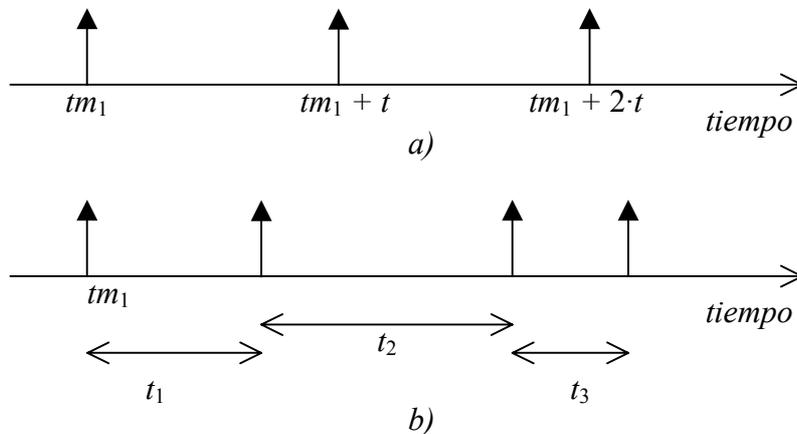


Figura 2.2: a) llegadas periódicas, b) llegadas irregulares

### Patrón de llegadas acotado

Si el siguiente evento no se puede dar hasta después de haber transcurrido un determinado tiempo de la llegada del anterior, el patrón de llegadas se dice que es *acotado*. Una llegada acotada puede deberse a:

- Razones físicas
- Requisitos del sistema
- Restricciones hardware
- Diseño software

En algunos casos hay también un intervalo máximo. La duración del intervalo mínimo, y a veces del intervalo máximo, es lo que caracteriza a la llegada acotada.

Un ejemplo de entrada acotada es la consola de un operador. La mayoría de los paneles de interruptores incluyen un circuito que evita, durante un cierto tiempo, la activación de un segundo interruptor una vez que un primer interruptor ha sido activado. Por tanto, la selección del interruptor por el operador está acotada.

#### *Patrón de llegadas en ráfaga*

Si el siguiente evento se puede producir tan pronto como se desee, pero el número de eventos en un espacio de tiempo está restringido, entonces se dice que el patrón de llegada de eventos es *en ráfaga*. Los patrones de llegada en ráfaga se caracterizan por una densidad de eventos, que consta de un intervalo de ráfaga (longitud de tiempo durante la cual se aplica la restricción en ráfaga), y un tamaño de ráfaga (número de eventos que pueden producirse durante dicho tiempo).

Por ejemplo, un gestor de transacciones puede estar obligado a procesar hasta 50 transacciones en un espacio de 1 segundo. Esas transacciones llegan según un patrón en ráfaga porque pueden llegar 50 al mismo tiempo, o pueden llegar dispersas en el tiempo.

Cuando llegadas consecutivas de eventos pueden estar tan próximas como se desee y las llegadas no están acotadas ni son en ráfaga, decimos que el evento es *no acotado*, porque un número no acotado de eventos se pueden producir en cualquier ventana de tiempo. Cuando los patrones de llegada no acotados se pueden describir estadísticamente, se pueden caracterizar mediante una función de distribución de probabilidad.

### **2.1.2. Tareas**

Al igual que los eventos se pueden clasificar en periódicos y no periódicos, las tareas que se ejecutan asociadas a dichos eventos se pueden también caracterizar por ser

*periódicas* (la llegada del evento asociado a una tarea se produce a intervalos de tiempos iguales a una constante denominada periodo) o *no periódicas*.

También se pueden clasificar como *expulsables* o *no expulsables*. Se dice que una tarea es expulsable si otra tarea puede interrumpir su ejecución en cualquier instante, para luego continuar en el mismo punto. Se dice que una tarea es no expulsable si una vez que ha comenzado tiene que ejecutarse hasta el final. El que una tarea sea expulsable o no, suele venir dado por el entorno de la aplicación.

Las tareas pueden estar sujetas a *restricciones de precedencia*. Se dice que una tarea  $i$  precede a una tarea  $j$  si la tarea  $i$  debe terminar antes de que la tarea  $j$  empiece. Las tareas relacionadas se comunican entre sí para sincronizarse y para intercambiar datos. El grafo de precedencia de un conjunto de tareas es un grafo orientado y acíclico, que puede ser un árbol, un grafo serie-paralelo o de cualquier otro tipo. En un sistema estático se conocen de antemano el conjunto de tareas y el grafo de precedencia. En sistemas dinámicos pueden llegar de forma dinámica nuevas tareas interrelacionadas y sólo se conoce el grafo de precedencia cuando llega el conjunto de tareas.

### 2.1.3. Recursos de un sistema

Para poder ejecutar las tareas que tiene especificadas, un sistema dispone de la CPU, los recursos de datos, y de otros recursos como son el recurso de dispositivo, o el recurso de red. Los recursos deben asignarse con una política de asignación que depende del tipo de recurso de que se trate ([Klein et al., 1993]). Así, por ejemplo la política de asignación de la CPU se denomina planificación.

#### *CPU: Planificación de tareas*

El procesador o CPU es el recurso compartido más común de los sistemas software y la forma de asignar este recurso es mediante una política de planificación. Normalmente, la política de planificación de la CPU la implementa un sistema operativo. Una política de planificación es un conjunto de reglas que determina cuál de las tareas que están listas será asignada a la CPU.

Planificación estática: Tradicionalmente se han venido aplicando soluciones de planificación de tipo estático “static time-driven scheduling” (como el ejecutivo cíclico) en los sistemas de tiempo real en aviación y sistemas críticos de seguridad en general. Este tipo de planificación, denominada off-line por Buttazzo ([Buttazzo,

1997]) consiste en generar la planificación previamente a la activación de las tareas, y guardarla en forma de tabla, para lo cual es necesario tener un conocimiento completo de antemano acerca del número y las características de las tareas. Las planificaciones estáticas tienen poco coste de ejecución, pero tienen desventajas como por ejemplo su escasa flexibilidad, al no poder adaptarse a entornos que cambian o a entornos que no son totalmente predecibles: al añadir tareas nuevas al sistema estático hay que volver a calcular su planificación. Otras desventajas son su ineficiencia debido a que el tiempo de CPU ahorrado por una tarea no se reasigna a ninguna otra, su pesimismo en la estimación del peor caso, y la falta de visibilidad de las relaciones de precedencia existentes entre las tareas.

Planificación dinámica: Por otro lado, los algoritmos de planificación on-line o dinámicos toman las decisiones de planificación en tiempo de ejecución, cada vez que una nueva tarea entra al sistema, o que la tarea que se estaba ejecutando termina. Las planificaciones de tipo dinámico suponen un mayor coste de ejecución pero se adaptan fácilmente a cambios en el entorno.

Un ejemplo de este tipo de planificación son los algoritmos basados en prioridades. En una política de **prioridades fijas** se utilizan las prioridades como mecanismo para decidir cómo asignar la CPU. Las prioridades son fijas en el sentido en que no varían de una ejecución a la siguiente. En [Audsley et al., 1993b] y [Bate, 1998] se compara la planificación estática con la basada en prioridades fijas. Se han mostrado las ventajas que se obtendrían de la aplicación de este tipo de planificación en sistemas de tiempo real como los satélites ([Bailey et al., 1993]) o en aplicaciones de aviación ([Grigg, Audsley, 1999b], [Grigg, Audsley, 1999a]). En [González Harbour et al., 1994] las tareas periódicas se dividen en subtareas que se ejecutan en serie, donde cada subtarea tiene su propia prioridad fija.

Una política de **prioridades dinámicas** también utiliza prioridades pero mediante algún algoritmo que permite cambiar la prioridad de una tarea de una ejecución a la siguiente.

Además de esta clasificación de los algoritmos de planificación en estáticos o dinámicos, y los basados en prioridades (que son dinámicos) en prioridades fijas o prioridades dinámicas, se pueden clasificar también en garantizados o mejor-esfuerzo (“best-effort”) y también en si utilizan o no test de aceptación ([Bernat et al., 1999]).

Garantizado/“best-effort”: Si el algoritmo es de tipo garantizado, se asegura que se cumplen las restricciones temporales. Se dice que un algoritmo de planificación garantiza una nueva tarea si es capaz de encontrar una planificación para todas las tareas previamente garantizadas y para la nueva, de manera que todas las tareas terminen antes de su plazo. Para ello es necesario realizar un análisis de planificabilidad off-line. Si el algoritmo es de tipo “best-effort” o mejor esfuerzo no se tiene garantía alguna sobre el cumplimiento de las restricciones temporales,

aunque el objetivo de este tipo de algoritmos es el de minimizar el número de tareas que no cumplen sus restricciones.

Test de aceptación: En este caso, el planificador se basa en rechazar algunas ejecuciones de las tareas con el fin de no tener una sobrecarga en el número de tareas listas para ejecutar. El test de aceptación puede ser de tipo garantizado o de tipo mejor esfuerzo.

#### *Recurso de datos:*

Un mecanismo software que es utilizado frecuentemente, pero que raras veces es reconocido como recurso, es el recurso de datos. Un recurso de datos puede ser tan pequeño como un bit o un byte y tan grande como una matriz de registros complejos. Cuando los datos se comparten de manera mutuamente excluyente entre tareas concurrentes, una tarea puede provocar retardos en otras. La política de asignación del recurso de datos determina el tamaño del retardo. Algunas políticas de asignación de recursos de datos son: FIFO (“first-in-first-attempt”), cola de prioridades, protocolo básico de herencia de prioridades (PIP), “highest locker” (HL) y “priority ceiling protocol” (PCP) ([Klein et al., 1993]).

#### *Otros recursos:*

Recurso de dispositivo: Otro recurso compartido son los dispositivos de entrada/salida. Algunos ejemplos serían discos, terminales, convertidores analógico-digitales y canales de entrada-salida. Nunca se utiliza este tipo de recurso por dos usuarios de manera concurrente, por lo que hay que determinar qué tarea usará el recurso cuando la tarea actual termine.

Recurso de red: Las redes, tanto de área local (LAN), como de área extensa (WAN), son también recursos compartidos. Los mensajes comparten la red a dos niveles: el adaptador hardware y el medio. El adaptador hardware es compartido por todos los mensajes con la misma dirección física de red. El medio es compartido por todas las direcciones de la red.

### **2.1.4. Restricciones temporales**

Tal y como ya se ha descrito, lo que distingue a los sistemas de tiempo real es la inclusión de requisitos temporales en la especificación del sistema. Los requisitos

temporales de una aplicación incluyen, entre otros, los requisitos computacionales, el “jitter” y los requisitos impuestos por los recursos compartidos ([Grigg, Audsley, 1999b]).

### *Requisitos computacionales*

Los requisitos temporales de las tareas relacionados con su ejecución son los plazos de ejecución o “deadlines”. Cada tarea puede tener un plazo de ejecución que debe cumplir siempre, incluso en el peor caso.

Atendiendo este tipo de restricciones temporales, los sistemas pueden ser estrictos (“hard”), flexibles (“soft”) o firmes (“firm”). En los sistemas de tiempo real flexibles las tareas son realizadas lo más rápido posible pero no están obligadas a terminar antes de un determinado instante. Un requisito de tiempo flexible se caracteriza por un tiempo medio de respuesta requerido. En cambio, en los sistemas de tiempo real estrictos las tareas se deben realizar cumpliendo las restricciones temporales. Típicamente una tarea de tiempo real estricto se caracteriza por sus restricciones temporales, restricciones de precedencia y requisitos de recursos. Un requisito de tiempo firme está compuesto por dos requisitos, uno flexible y otro estricto. Por ejemplo, si una tarea periódica se debe completar en media en un periodo, pero es capaz de tener en un buffer las entradas de cuatro periodos, entonces la tarea tiene un requisito firme: un tiempo de respuesta medio de un periodo y un requisito estricto de cuatro periodos ([Klein et al., 1993]).

La clasificación de las tareas que dan Bernat et al. ([Bernat et al., 1999]) atendiendo a las restricciones temporales es similar a la anterior, añadiendo un nuevo tipo de tarea, las tareas débilmente estrictas. Así, según estos autores los sistemas de tiempo real (o las tareas) pueden ser estrictos, flexibles, firmes o débilmente estrictos. Para las tareas estrictas, es imperativo que se respeten todos los plazos mientras que es admisible que las tareas flexibles no respeten alguno de sus plazos de forma ocasional. Aunque terminen tarde, la finalización de las tareas aporta valor al sistema. Las tareas firmes también pueden no respetar alguno de sus plazos, pero al contrario que las tareas flexibles, las finalizaciones tardías no aportan valor alguno al sistema. El problema es que la palabra “ocasional” (plazos no respetados de forma ocasional) es tan ambigua que resulta de poca utilidad a la hora de especificar un sistema. Cuando se especifica de forma precisa en qué manera se pueden dejar de respetar los plazos, los sistemas (o tareas) se dice que son débilmente estrictos. Un sistema de tiempo real débilmente estricto es un sistema en el que la distribución de plazos que se cumplen y que no se cumplen durante una ventana de tiempo está acotada de forma precisa.

### *Jitter*

Otros requisitos temporales del sistema se deben a la existencia de retardos o “jitter”. Se pueden distinguir tres tipos:

“Release jitter” o retraso de la activación: Es el tiempo que una tarea espera desde que llega el evento asociado hasta que está lista. Si existe “release jitter” una tarea puede sufrir un retardo acotado y especificado en pasar al estado de lista ([Audsley et al., 1993a]). Este retardo se puede producir porque una tarea deba esperar la llegada de un mensaje antes de poder comenzar. Se puede utilizar para representar un sistema distribuido como un conjunto de tareas con relaciones de precedencia [Palencia, 1999]. Cada tarea se asigna de forma estática a un procesador y las tareas de cada procesador se analizan con técnicas que tienen en cuenta la existencia de “release jitter”. Una tarea con restricción de precedencia en un procesador verá retrasado su estado de lista esperando que sus predecesores terminen en otros procesadores (esto es, esperando el mensaje correspondiente). La implementación de un planificador también puede producir “release jitter”. Supongamos que las tareas periódicas y esporádicas de un procesador son planificadas por un planificador que es invocado por una interrupción periódica de reloj. Las tareas pueden experimentar un “release jitter” igual al periodo de reloj (como máximo) si llegan justo después de la interrupción del reloj, ya que entonces deben esperar a la siguiente interrupción para que se ejecute el planificador.

“Input jitter”: El “input jitter” es el intervalo de validez de los datos que van a ser leídos por las tareas invocadas tras la llegada de un evento.

“Output jitter”: El “output jitter” representa el intervalo de tiempo en que los datos de una tarea pueden ser emitidos. El instante más tardío de emisión debe estar antes del plazo de la tarea (si tiene plazo).

### *Recursos compartidos*

Cuando se comparten recursos hace falta un mecanismo que haga de árbitro en los accesos. Hay dos tipos principales de requisitos temporales para recursos compartidos que deben cumplirse en general:

- No deben producirse interbloqueos o “deadlocks”. Un conjunto de tareas está en estado de interbloqueo cuando cada una de ellas está utilizando un recurso y está también esperando a que otro recurso sea liberado por parte de otra tarea de ese conjunto.
- El tiempo que una tarea debe esperar para poder acceder a un recurso debe estar acotado

Más adelante en este capítulo se proporciona una descripción detallada de la forma en que se comparten los recursos.

#### *Otros requisitos temporales*

Requisitos asociados a los cambios de modo: El sistema se puede ejecutar en varios modos, cada uno de los cuales puede tener requisitos temporales diferentes.

Requisitos en las comunicaciones: Cuando se envían datos de un proceso a otro proceso remoto a través de un medio de comunicación, el tiempo que dura dicha comunicación debe estar acotado.

Principio-a-fin (“end-to-end”): Es el requisito temporal que afecta a una transacción de procesos, que puede involucrar a uno o varios procesadores.

Interferencia de múltiple criticidad: Las tareas con un nivel de criticidad menor no deben afectar a las propiedades temporales de las tareas con nivel mayor.

### **2.1.5. Utilización del procesador**

Tal y como ya se ha explicado, el análisis temporal (o análisis de planificabilidad) es un conjunto de técnicas matemáticas que permiten analizar un sistema y su software off-line sin necesidad de ejecutar dicho sistema, con el fin de determinar si cumple sus restricciones temporales. Una de dichas técnicas se basa en el concepto de utilización del procesador, mientras que otras se basan en el cálculo de los tiempos de finalización de las tareas.

Dado un conjunto  $\Gamma$  de  $n_{tar}$  tareas periódicas, se define la utilización total del procesador como la fracción de tiempo de procesador empleada en la ejecución de las tareas ([Liu, Layland, 1973]). Sea  $p_i$  el tiempo de ejecución de la tarea  $i$ , y  $t_i$  el periodo de dicha tarea. Como  $p_i / t_i$  es la fracción de tiempo de procesador empleado en la ejecución de la tarea  $i$ , la utilización total para  $n_{tar}$  tareas viene dada por:

$$(2.1) \quad U = \sum_{i=1}^{n_{tar}} \frac{p_i}{t_i}$$

La utilización total da una medida de la carga computacional en la CPU debido a las tareas periódicas. Si la utilización total es 1, significa que todo el tiempo de la CPU está siendo utilizado. Si la utilización total es mayor que 1, la CPU no es capaz de

ejecutar todas las tareas periódicas, y por tanto se tiene certeza de que el sistema no es planificable. Si se desea un mejor aprovechamiento del tiempo de la CPU, incrementando la utilización total, se puede aumentar los tiempos de ejecución de las tareas o disminuir los periodos, aunque existe un límite máximo de utilización  $U_l(\Gamma, A)$  (que puede ser menor o igual que 1) por debajo del cual el conjunto de tareas es planificable, y por encima del cual deja de ser planificable. Ese límite depende de las tareas y del algoritmo de planificación empleado ([Buttazzo, 1997]). Si el límite es un valor menor que 1 significa que si la utilización total supera ese límite y es también menor que 1, el sistema no es planificable, aunque la CPU no esté todo el tiempo ejecutando tareas periódicas; es decir, la CPU está desocupada durante ciertos periodos de tiempo, y sin embargo no se pueden garantizar todos los plazos de las tareas periódicas. Esto es lo que ocurre por ejemplo con el algoritmo de planificación basado en prioridades fijas RMS, que se describirá más tarde en este capítulo.

Sea  $U_l(\Gamma, A)$  el límite de utilización de un conjunto de tareas para un algoritmo de planificación  $A$  dado. Cuando la utilización  $U$  definida en (2.1) verifica que  $U = U_l(\Gamma, A)$ , se dice que el conjunto de tareas utilizan de forma completa el procesador. En esta situación las tareas son planificables con dicho algoritmo pero un aumento en el tiempo de ejecución de cualquier tarea hará que el sistema sea infactible, es decir, no planificable. Para un algoritmo  $A$ , el menor límite  $U_l(A)$  de utilización del procesador es el mínimo de entre todos los conjuntos de tareas que utilizan de forma completa el procesador:

$$U_l(A) = \min_{\Gamma} U_l(\Gamma, A)$$

$U_l(A)$  define una característica importante del algoritmo de planificación porque permite verificar fácilmente la planificabilidad de un conjunto de tareas. Cualquier conjunto de tareas cuya utilización total se encuentra por debajo de ese límite es planificable mediante el algoritmo  $A$ . Por otro lado, si la utilización total es mayor que ese límite, no se puede decir nada acerca de la planificabilidad del sistema. Si la utilización total de un conjunto de tareas es mayor que 1, el conjunto de tareas no es planificable mediante ningún algoritmo. Para ilustrar este resultado, supongamos que  $w$  es el producto de todos los periodos:  $w = t_1 \cdot t_2 \cdot \dots \cdot t_n$ . Si  $U > 1$ , entonces  $U \cdot w > w$ , que se puede escribir como:

$$\sum_{i=1}^{n_{tar}} \frac{w}{t_i} \cdot p_i > w$$

El factor  $(w/t_i)$  representa el número de veces que la tarea  $i$  se ejecuta en el intervalo de tiempo  $w$ , mientras que  $(w/t_i) \cdot p_i$  es el tiempo total de computación requerido por la tarea  $i$  en el intervalo  $w$ . Por tanto, el sumatorio representa la demanda total de tiempo de computación por parte de todas las tareas durante  $w$ . Si la demanda total

supera el tiempo de procesador disponible, entonces no existe una planificación factible para ese conjunto de tareas ([Buttazzo, 1997]).

## 2.2. Modelo del sistema

Tal y como ya se dijo en la introducción, en esta tesis sólo se va a analizar la planificabilidad de sistemas con un procesador, basados en prioridades y con tareas expulsables. En este apartado se presenta el modelo de sistema de tiempo real que se utilizará en el resto de la exposición.

Una tarea periódica  $i$  quedaría definida por los siguientes parámetros:

- $t_i$  : periodo de la tarea  $i$ .
- $p_i$  : tiempo de ejecución de la tarea  $i$ .
- $d_i$  : plazo de la tarea  $i$ .

En la Figura 2.3 aparecen representados dichos parámetros básicos.

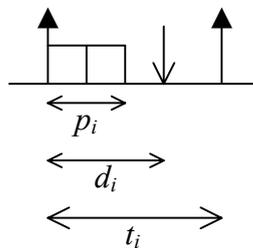


Figura 2.3: Parámetros básicos de una tarea periódica

Teniendo en cuenta las sucesivas ejecuciones de la tarea  $i$ , se pueden definir además los siguientes parámetros, mostrados en la Figura 2.4:

- $r_{i,j}$ : instante listo o “ready” de la tarea  $i$  en su ejecución  $j$ . Al ser una tarea periódica se tiene que  $r_{i,j}=(j-1) \cdot t_i$
- $wc_{i,j}$ : tiempo de finalización de la tarea  $i$  en su ejecución  $j$ .
- $wc_i$ : peor tiempo de respuesta de la tarea  $i$ .

- $e_{i,j}$ : tiempo de respuesta de la tarea  $i$  en su ejecución  $j$ . Es el tiempo de finalización menos el tiempo listo de esa ejecución, es decir,  $e_{i,j} = wc_{i,j} - (j-1) \cdot t_i$
- $d_{i,j}$ : plazo absoluto de la tarea  $i$  en su ejecución  $j$ . Es igual al instante listo de la tarea en esa ejecución más el plazo relativo, es decir,  $d_{i,j} = (j-1) \cdot t_i + d_i$

Para que un sistema sea planificable todos los tiempos de finalización de las tareas deben estar antes del correspondiente plazo absoluto, esto es,  $wc_{i,j} \leq d_{i,j}$ . Cuando las tareas son periódicas es suficiente con comprobar la condición  $wc_{i,j} \leq d_{i,j}$  para las ejecuciones que se encuentran dentro de un hiperperiodo, siendo éste el mínimo común múltiplo de los periodos de las tareas.

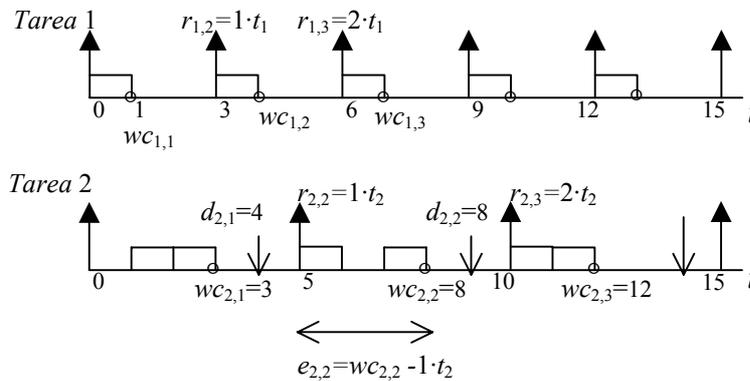


Figura 2.4: Otros parámetros de las tareas periódicas

### 2.3. Planificación basada en prioridades fijas

Los dos algoritmos de planificación más conocidos basados en prioridades fijas son el algoritmo de prioridades monótonas en frecuencia o “Rate Monotonic Scheduling” (RMS) y el algoritmo de prioridades monótonas en plazo o “Deadline Monotonic Scheduling” (DMS). El primero asigna las prioridades a las tareas de acuerdo con el tamaño del periodo, de forma que la tarea con el menor periodo es la más prioritaria y la que tiene el periodo mayor es la menos prioritaria. El segundo algoritmo, DMS, asigna las prioridades de acuerdo con el tamaño del plazo, así la tarea más prioritaria es la que tiene el menor plazo y la menos prioritaria es la de mayor plazo.

Estos algoritmos se aplicaron a un estudio realizado por la British Space Systems Ltd, la Universidad de York y por York Software Engineering Ltd para la Agencia Espacial Europea ([Bailey et al., 1993]). El objetivo era determinar la aplicabilidad de métodos de planificación en aplicaciones de abordo situadas en los satélites (cuyo lenguaje de implantación suele ser Ada). Se demostró que, debido a las características del sistema, una planificación expulsora y basada en prioridades fijas era la más apropiada y que la planificación basada en prioridades monótonas en plazo o “Deadline Monotonic Scheduling” (DMS) era más adecuada que la planificación basada en prioridades monótonas en frecuencia o “Rate Monotonic Scheduling” (RMS). El caso de estudio consistió en el rediseño mediante estas técnicas, del Sistema de Control de Órbita y de Altitud del satélite Olympus, lanzado en 1989. Se mostró con éxito que es posible y práctico usar una solución basada en las tareas en este tipo de sistemas, logrando el mismo nivel de control que con el ejecutivo cíclico empleado hasta entonces. Tiene algunas desventajas como un mayor coste de ejecución o la necesidad de invertir en herramientas de soporte y en formación para extraer el máximo partido a estas técnicas. Pero las ventajas encontradas compensan los problemas citados:

- Está enfocado al diseño y desarrollo de sistemas, pero posee un fundamento matemático.
- Este análisis teórico de planificabilidad permite reducir el tiempo de prueba.
- Este tipo de planificación es más flexible y permite incorporar cambios en la estructura de la aplicación sin los costes que supone el re-desarrollo del ejecutivo cíclico.
- La existencia de herramientas de soporte (como por ejemplo HRT-HOOD) permite que estas técnicas sean aplicadas en desarrollos reales.

Existen grupos de trabajo que están desarrollando la extensión del estándar UML para permitir el análisis de planificabilidad de los sistemas de tiempo real ([OMG., 1999]). De hecho ya existe una propuesta ([OMG, 2001]) a la petición que realizó la organización OMG para la extensión de UML al análisis de planificabilidad y la inclusión de la representación explícita del tiempo. Algunos de los métodos que han servido de entrada a este documento han sido el Rate Monotonic Analysis, el Deadline Monotonic Analysis y el Earliest Deadline First, aunque el documento no sólo está preparado para cubrir estos tipos de análisis de planificabilidad, sino también otras técnicas (UML: [www.omg.org](http://www.omg.org)).

### 2.3.1. Prioridades monótonas en frecuencia (RMS)

En 1973, Liu y Layland ([Liu, Layland, 1973]) escribieron un artículo sobre planificación que planteaba las condiciones matemáticas para el análisis de planificabilidad de un conjunto de tareas periódicas, con plazos iguales a los periodos, expulsables, independientes y planificadas según el algoritmo “Rate Monotonic Scheduling” (RMS) o prioridades monótonas en frecuencia. El algoritmo RMS asigna la prioridad más alta a la tarea con el periodo más pequeño, y el resto se asigna monótonamente de acuerdo al tamaño de los periodos.

La Figura 2.5 muestra un sistema de tiempo real formado por tres tareas  $A$ ,  $B$  y  $C$  periódicas cuyos periodos son  $t_A=3$ ,  $t_B=5$  y  $t_C=15$ . Los tiempos de ejecución son  $p_A=1$ ,  $p_B=2$  y  $p_C=3$ . Los plazos coinciden con los periodos, es decir  $d_A=3$ ,  $d_B=5$  y  $d_C=15$ , y por tanto, una tarea debe finalizar su ejecución  $j$  antes del instante listo siguiente  $r_{i,j+1}$ . El tiempo de finalización de la tarea  $i$  en la ejecución  $n$  se denomina  $wc_{i,n}$  y se ha marcado en el gráfico con un círculo. El tiempo listo de la tarea  $i$  en su ejecución  $n$  se denomina  $r_{i,n}$  y se ha indicado con una flecha.

De acuerdo con el algoritmo RMS, la tarea más prioritaria es la tarea  $A$  porque tiene el menor periodo, mientras que la tarea menos prioritaria es la tarea  $C$ . Se puede observar que durante la segunda ejecución de  $B$ , en el instante 6 la tarea  $A$  vuelve a estar lista. Como las tareas son expulsables y la tarea  $A$  es más prioritaria, se interrumpe la tarea  $B$ , y hasta que no finaliza la tercera ejecución de  $A$  en el instante 7, no se reanuda la ejecución de  $B$ .

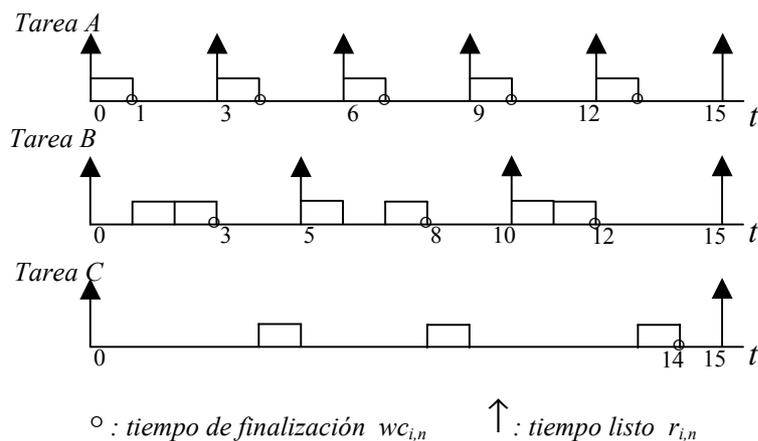


Figura 2.5: Ejemplo de planificación RMS

Cuando los plazos coinciden con los periodos, se dice que este algoritmo es óptimo dentro del conjunto de algoritmos de prioridades fijas, en el sentido de que si existe una asignación de prioridades fija que es planificable, entonces la asignación propuesta por RMS también lo es.

El teorema de Liu y Layland permite comparar la utilización total de la CPU de las tareas periódicas con el límite de planificabilidad para el algoritmo RMS, y dice:

*Un conjunto de  $n$  tareas periódicas e independientes, planificadas según el algoritmo RMS, respetará todos sus plazos si*

$$\frac{p_1}{t_1} + \dots + \frac{p_n}{t_n} \leq U_l(n) = n \cdot \left(2^{1/n} - 1\right)$$

Siendo

$p_i$  el tiempo de ejecución de la tarea  $i$

$t_i$  el periodo de la tarea  $i$

$U_l(n)$  el menor límite de utilización para  $n$  tareas

Nótese que este límite decrece con  $n$ , y que para valores grandes de  $n$  tiende a  $U_l = \ln 2 \cong 0.69$ . Nótese también que la condición de planificabilidad propuesta por el teorema es suficiente, pero no necesaria.

Estos autores definieron también el concepto de **instante crítico**: cuando los plazos son menores o iguales que el periodo, el instante crítico es aquel en que todas las tareas están listas simultáneamente, ya que el peor tiempo de respuesta de una tarea en una ejecución se da cuando está lista en el mismo instante que todas las tareas de prioridad más alta. En otras palabras, el máximo tiempo de respuesta se da cuando una tarea encuentra la mayor competencia por la CPU por parte de las tareas de más prioridad. Este concepto es válido cuando las tareas no se sincronizan y cuando no se pueden ejecutar después del final de su periodo. Si existe un instante crítico es posible ignorar todas las ejecuciones sucesivas de las tareas excepto aquellas que se iniciaron en el instante crítico.

El concepto de instante crítico es ligeramente diferente en el caso de que las tareas se puedan ejecutar después del final de su periodo. En este caso, siendo  $pr_i$  la prioridad de la tarea  $i$ , el peor tiempo de respuesta se da durante un periodo de ocupación de nivel  $pr_i$  que se inició en el mismo instante que todas las tareas más prioritarias (o lo que es lo mismo, en el instante crítico) ([Lehoczky et al., 1991]). Un *periodo de ocupación de nivel  $pr_i$*  es un intervalo de tiempo durante el cual el procesador está ocupado ejecutando tareas de prioridad mayor o igual a  $pr_i$  ([Lehoczky, 1990]).

Resumiendo, cuando cada ejecución de una tarea termina antes del final de su periodo:

- Los tiempos de respuesta se ven afectados sólo por tareas más prioritarias y por el tiempo de ejecución de la propia tarea.
- El mayor tiempo de respuesta es el de la ejecución que comienza en el instante crítico.

Cuando las tareas se ejecutan pasado el final de sus periodos:

- Los tiempos de respuesta se ven afectados por tareas más prioritarias y por el tiempo de ejecución de la propia tarea.
- El mayor tiempo de respuesta se ve afectado también por ejecuciones previas de la misma tarea.
- La ejecución con el mayor tiempo de respuesta puede no ser la que comienza en el instante crítico.
- El peor tiempo de respuesta de la tarea  $i$  se da para alguna ejecución perteneciente al periodo de ocupación de nivel  $pr_i$

En la Figura 2.6, se muestra un ejemplo de un sistema formado por dos tareas  $A$  y  $B$  periódicas de periodos  $t_A=10$  y  $t_B=15$  y prioridades  $pr_A=2$  y  $pr_B=1$ . Sus tiempos de ejecución son  $p_A=4$  y  $p_B=8$  y sus plazos son  $d_A=10$  y  $d_B=16$ . Por tanto la tarea  $A$  es más prioritaria y la tarea  $B$  tiene el plazo mayor que el periodo.

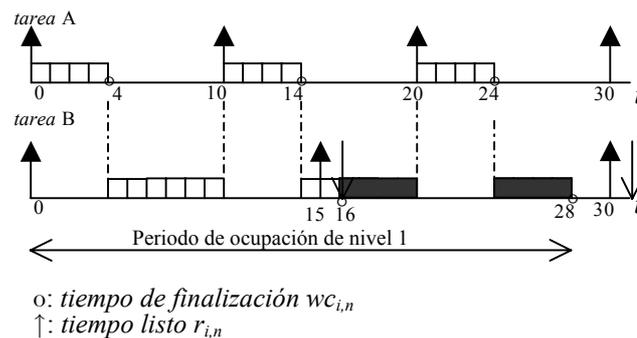


Figura 2.6: Tarea con plazo mayor que el periodo

Se observa en la Figura 2.6 que la tarea  $A$  termina sus ejecuciones antes de su plazo. La tarea  $B$ , en su primera ejecución (en el gráfico la primera ejecución aparece en

blanco y la segunda en gris) termina en el instante 16, que es igual al plazo y que está más allá del periodo. En su segunda ejecución, la tarea *B* termina en el instante 28, antes del plazo y antes del periodo, por lo tanto en esta ejecución termina el periodo de ocupación de nivel 1.

La implementación de RMS en Ada aparece descrita en [Sha, Goodenough, 1989]. En [Shih et al., 1993] se propone un algoritmo RMS modificado capaz de planificar tareas con su plazo mayor que el periodo, útil en el caso de que el periodo de llegada de peticiones sea más rápido que el tiempo en que la tarea debe dar el resultado. Propone además un algoritmo mixto RMS y RMS modificado capaz de tratar con ambos tipos de tareas: tareas urgentes, con plazos iguales al periodo, y tareas con plazos retrasados respecto a sus periodos.

### 2.3.2. Prioridades monótonas en plazo (DMS)

Cuando los plazos no coinciden con los periodos, otra forma de asignar las prioridades de las tareas es según el algoritmo “Deadline Monotonic Scheduling” (DMS) o prioridades monótonas en plazo. Según este algoritmo, la tarea de prioridad más alta es aquella que tiene el menor plazo.

Leung y Whitehead ([Leung, Whitehead, 1982]) probaron que el algoritmo DMS es óptimo para un conjunto de tareas independientes, con prioridades fijas y con plazos que coinciden o que están antes del final del periodo. Como ya se ha indicado, esto significa que si existe una asignación de prioridades fijas planificable, entonces la asignación dada por DMS también lo es.

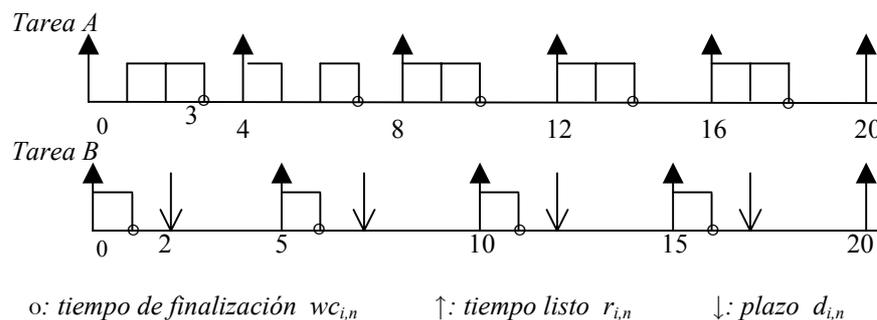


Figura 2.7: Ejemplo de planificación con DMS

En el ejemplo de la Figura 2.7 el sistema está formado por dos tareas  $A$  y  $B$ . Los periodos son  $t_A=4$  y  $t_B=5$ . Los plazos son  $d_A=4$  y  $d_B=2$ . Se puede ver que el plazo de  $A$  coincide con su periodo, pero el de  $B$  no. De acuerdo con DMS, la tarea  $B$  es más prioritaria ya que su plazo es menor. Como existe instante crítico, el peor tiempo de respuesta para todas las tareas se da en la primera ejecución a partir de ese instante. El tiempo de respuesta de la tarea  $A$  en su primera ejecución es  $e_{A,1}=3$  y el de  $B$  es  $e_{B,1}=1$ . Como los plazos son menores o iguales que los periodos, los tiempos de respuesta en la primera ejecución a partir del instante crítico son los peores tiempos de respuesta de las tareas, por lo que  $wc_A=wc_{A,1}=e_{A,1}$  y  $wc_B=wc_{B,1}=e_{B,1}$ . Así, en el ejemplo propuesto las tareas respetan sus plazos.

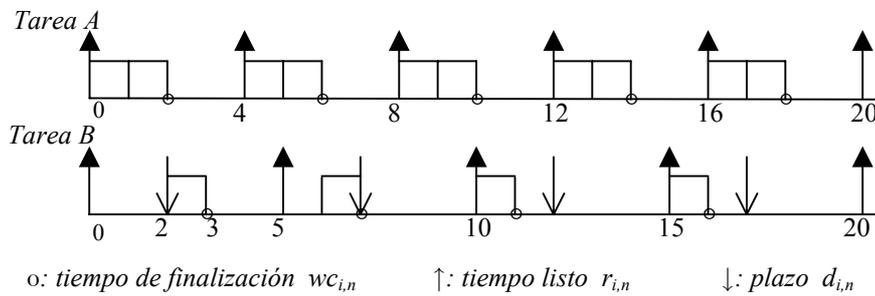


Figura 2.8: El mismo ejemplo planificado mediante RMS

Como puede comprobarse en la Figura 2.8, si este mismo sistema se hubiera planificado mediante el algoritmo RMS, no se habría respetado el plazo de la tarea  $B$ , y el sistema no habría sido planificable.

### 2.3.3. Análisis temporal de algoritmos con prioridades fijas

Joseph y Pandya ([Joseph, Padya, 1986]) fueron los primeros en usar un algoritmo similar a (2.2) para calcular el peor tiempo de respuesta cuando los plazos son menores o iguales que los periodos, que se puede utilizar como condición necesaria y suficiente de planificabilidad. Lehoczky ([Lehoczky et al., 1989]) utilizó la ecuación:

$$s_{i,k}(s_{i,k-1}) = \sum_{j=1}^i p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

para obtener una condición necesaria y suficiente de planificabilidad cuando el plazo coincide con el periodo, evaluando el término de la derecha en los puntos de planificación, o instantes listos, de las tareas más prioritarias. El algoritmo también fue desarrollado de forma independiente por Audsley ([Audsley et al., 1991]). El símbolo  $\lceil \cdot \rceil$  indica la operación redondeo hacia arriba.

En el algoritmo (2.2) se calculan los tiempos de finalización de las tareas en su primera ejecución a partir del instante crítico, ya que en esa ejecución se da el peor tiempo de respuesta  $wc_i$  de cada tarea. Las tareas deben ordenarse según su prioridad fija, de manera que la tarea 1 sea la más prioritaria.

**CRISP ALG**

$wc_0 = 0$

for  $i=1$  to  $n\_tarefas$  do

**BASIC ALG:**

$$\left\{ \begin{array}{l} s_{i,0} = in\_s_i \\ \text{while } (s_{i,k-1} < s_{i,k}(s_{i,k-1})) \quad k = k + 1 \\ wc_i = s_{i,k} \end{array} \right.$$

$$\text{dado, } s_{i,k}(s_{i,k-1}) = \sum_{j=1}^i p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

if  $(wc_i > d_i)$  tarea  $i$  no planificable

end

(2.2)

Existen extensiones, como se verá más tarde, que tienen en cuenta la presencia de tiempos de bloqueo en las tareas cuando hay recursos compartidos, “release jitter” o servidores aperiódicos. Este tipo de análisis se puede aplicar a sistemas complejos de tiempo real como el Sistema de Paso de Mensajes ([Klein et al., 1993]), que es un sistema de un solo procesador que forma parte de un sistema distribuido de paso de mensajes. El mismo autor presenta un caso de estudio de un Subsistema de transmisión Sonar al que aplica este tipo de análisis.

## 2.4. Planificación basada en prioridades fijas, con recursos de datos compartidos

En este apartado se describen algunos enfoques encontrados en la literatura para la sincronización en el acceso de los sistemas a datos compartidos. Cuando las tareas

comparten datos se pueden provocar retrasos, no sólo a las tareas que comparten los recursos sino también a otras tareas, retrasos que hay que tener en cuenta en el análisis de planificabilidad del sistema.

### **2.4.1. Tareas que deben sincronizarse para compartir datos comunes**

Un conjunto de acciones que utilizan un recurso se consideran atómicas en ese recurso cuando debe ser realizadas del comienzo al fin sin liberar dicho recurso. De la definición se deduce que dos acciones atómicas no se pueden realizar simultáneamente en un solo recurso. Nótese que una acción puede ser atómica en un recurso, como un objeto de datos, y simultáneamente utilizar otro recurso como la CPU de manera no atómica. En ese caso, puede haber expulsión en el uso de la CPU pero sin violar el uso atómico del objeto de datos. La parte de la tarea que accede al recurso de datos compartido se denomina sección crítica.

La compartición de un recurso de datos cuando su uso debe ser atómico, requiere elegir los siguientes parámetros ([Klein et al., 1993]):

- Una primitiva de sincronización, necesaria para asegurar que una única tarea utiliza el recurso cada vez.
- Una política de asignación del recurso, que determina las acciones a realizar cuando se hace una petición del recurso estando éste ocupado.
- Una prioridad de ejecución durante el uso del recurso, utilizada para asignar la CPU durante la acción atómica, y que también afecta al tiempo que deben esperar las demás peticiones en cola.

La elección combinada de una primitiva de sincronización, una política de asignación y una prioridad de ejecución de la acción atómica, se denomina protocolo de sincronización.

Algunas combinaciones de soluciones no son posibles. Por ejemplo, algunos mecanismos para implementar el uso de recursos compartidos pueden crear colas y otros no. Una primitiva que utiliza manipulación de prioridades normalmente no usa colas e implementa todo el protocolo de sincronización en un solo mecanismo.

A continuación se citan algunas situaciones de tiempo real, todas ellas de modo único. Se puede consultar [Tindell et al., 1992], [Sha et al., 1989] y [Sha et al., 1991] donde los autores estudian situaciones de tiempo real con más de un modo.

Los tiempos de bloqueo de una tarea aparecen en sistemas cuyas tareas deben sincronizarse para compartir datos comunes, y deben ser tenidos en cuenta en el análisis de planificabilidad del sistema. Por ejemplo, una tarea de prioridad más alta puede tener que esperar a que una de prioridad más baja termine de utilizar un recurso compartido (que puede ser una estructura de datos o un dispositivo de entrada/salida). Esto es lo que se denomina inversión de prioridades. El cálculo de los tiempos de bloqueo depende del protocolo de sincronización utilizado. Para algunas implementaciones hay dos formas de tener en cuenta el bloqueo producido por la sincronización:

- Representar la variación en las prioridades tal y como ocurriría, sin calcular retardos por bloqueo.
- Ignorar la variación en las prioridades y calcular el retardo por bloqueo.

Se pueden distinguir dos casos: que se comparta un único recurso de datos, o que se compartan múltiples recursos.

### **Recurso único**

En este caso varias tareas comparten un solo recurso, cada una de ellas haciendo uso del recurso una sola vez en cada ejecución. En esta situación se puede dar una inversión de prioridades no acotada. La característica de no acotada significa que no hay un límite de tiempo de bloqueo a priori, antes de conocer el conjunto específico de tareas, debido a que puede haber expulsión múltiple. La inversión de prioridades no acotada se considera intolerable ya que compromete la integridad de todo el sistema y se puede dar cuando una tarea de prioridad alta y otra de prioridad baja comparten un recurso de uso atómico, y cuando también hay tareas de prioridad media que no lo utilizan. Las tareas de prioridad media pueden prolongar el tiempo en que el recurso está ocupado por la tarea de prioridad baja ([Klein et al., 1993]).

En la siguiente figura se muestra un ejemplo de inversión de prioridades no acotada. Se representan tres tareas, donde la tarea 1 es la más prioritaria y la 3 la menos prioritaria, y ambas comparten un recurso. En este caso la inversión de prioridades es no acotada porque la tarea 1 se ve bloqueada, no sólo por la tarea 3 con la que comparte el recurso, sino también por la tarea 2 que, debido a su prioridad, interrumpe a la tarea 3 y por tanto retrasa a la tarea 1.

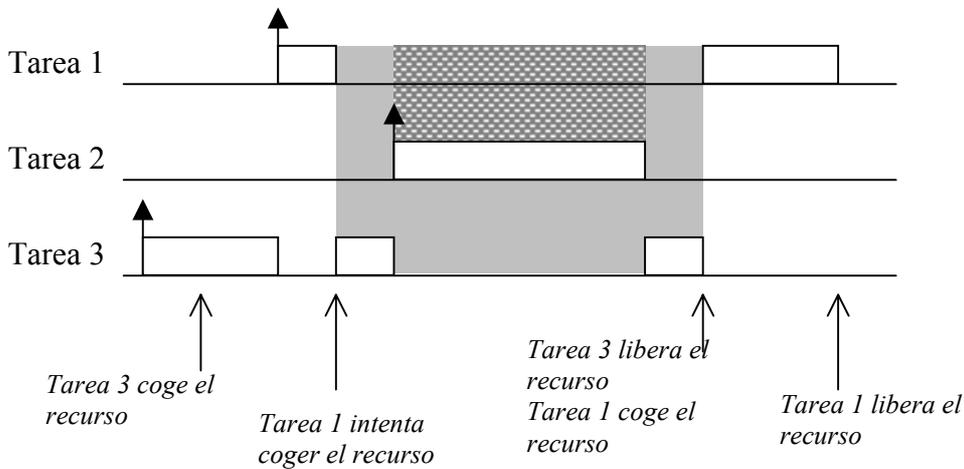


Figura 2.9: Inversión de prioridades no acotada

Tres posibles protocolos de sincronización son el semáforo FIFO, el protocolo de máscara de interrupciones y el protocolo básico de herencia de prioridades.

#### *Semáforo FIFO:*

Las principales características de este protocolo de sincronización son:

- Se utiliza un semáforo para implementar la sección crítica.
- En el caso de que el recurso esté ocupado, se encola la petición en una cola “first-in-first-out” (FIFO).
- La tarea que usa el recurso no ajusta su prioridad de ejecución en la sección crítica.

En el caso de utilizar un semáforo FIFO se pueden producir inversiones de prioridades no acotadas, tal y como se puede ver en la figura anterior. Esta implementación es muy común, especialmente en aplicaciones que no son de tiempo real.

*Protocolo de máscara de interrupciones (IMP)*

Con este protocolo cada tarea se asegura el uso exclusivo del recurso enmascarando todas las interrupciones antes de su utilización y desenmascarándolas al liberar el recurso. Cuando se enmascaran las interrupciones ninguna otra tarea puede expulsar a la tarea que está usando el recurso y así se garantiza su uso exclusivo. Además, este protocolo asegura que, como mucho, una tarea puede estar bloqueada el tiempo correspondiente a la duración de una sección crítica. Enmascarar las interrupciones es lo mismo que poner la prioridad de la tarea al nivel de interrupción hardware (máxima prioridad).

Este protocolo es sencillo de implementar. Tareas que no comparten un recurso pueden sufrir también retardos por bloqueo.

*Protocolo básico de herencia de prioridades (PIP)*

En este protocolo ([Sha et al., 1990]) se utiliza un semáforo para implementar la sección crítica. Cuando el recurso está ocupado, la petición se encola de acuerdo con su prioridad. Al encolar una petición, la prioridad de la tarea que está utilizando el recurso se eleva temporalmente al valor de la prioridad de la tarea encolada (se dice que hereda su prioridad). Esta herencia es la que evita que haya expulsión en la sección crítica por parte de otra tarea, lo que produciría una inversión de prioridades no acotada. Al liberar el semáforo la tarea que está al comienzo de la cola accede al recurso.

Debido al uso de una cola de prioridades, una tarea en una ejecución puede ser bloqueada como mucho una vez por cada recurso que utiliza. Al igual que en el protocolo IMP, tareas que no comparten un recurso pueden sufrir también retardos por bloqueo. Comparado con el IMP, el PIP mejora el comportamiento medio sin empeorar el peor caso.

La descripción de su implementación en Ada puede encontrarse en [Borger, Rajkumar, 1989].

**Recursos múltiples**

En este caso, varias tareas comparten varios recursos, haciendo uso de más de un recurso en cada ejecución, por lo que se pueden dar interbloqueos y retardos por bloqueos múltiples, cada uno de los cuales puede tener inversión de prioridades no acotada. Un conjunto de tareas está en estado de interbloqueo cuando cada una de ellas está utilizando un recurso y está también esperando a que otro recurso sea

liberado por parte de otra tarea de ese conjunto. El bloqueo en cadena o bloqueo múltiple es aquel que se produce cada vez que se solicita un nuevo recurso. La elección del protocolo determina la posibilidad de que se dé interbloqueo o bloqueo en cadena ([Klein et al., 1993]).

Tres de las posibles implementaciones en este caso son las siguientes:

- Protocolo de protección de prioridad o prioridad maximal (“highest locker”, HL)
- Protocolo básico de herencia de prioridades (PIP)
- Protocolo de techo de prioridad o “Priority Ceiling” (PCP)

*Protocolo de protección de prioridad o prioridad maximal (“highest locker”, HL)*

Este protocolo realiza una manipulación de prioridades para implementar las secciones críticas. Todos los recursos tienen definida una prioridad máxima que es la mayor prioridad de todas las tareas que pueden utilizar ese recurso. Cuando una tarea necesita un recurso en tiempo de ejecución, fija su propia prioridad a un valor mayor que el de la prioridad máxima del recurso. De esta forma ninguna otra tarea puede acceder al recurso mientras está en uso. En este protocolo se evita el interbloqueo siempre y cuando ninguna tarea se suspenda a sí misma mientras está haciendo uso de un recurso compartido. El HLP evita el bloqueo en cadena, y es sencillo de implementar.

*Protocolo básico de herencia de prioridades (PIP)*

Este protocolo, que se ha descrito antes, permite el bloqueo en cadena y no evita el interbloqueo.

*Protocolo de techo de prioridad o “Priority Ceiling Protocol” (PCP)*

En el protocolo PCP ([Goodenough, Sha, 1988], [Sha et al., 1990]) se utiliza un semáforo para implementar la sección crítica. Al igual que en el protocolo HL, cada recurso tiene su propia prioridad máxima (“priority ceiling”). Para ello:

- Se mantiene una variable del sistema operativo con el valor de la mayor prioridad máxima de todos los semáforos cerrados. Esta variable se denomina el techo actual del sistema.

- A no ser que una tarea posea el semáforo que está fijando el techo actual del sistema, la tarea sólo puede coger un semáforo si su prioridad es mayor que el techo actual del sistema.
- Si una tarea no puede coger un semáforo, la tarea que lo tiene hereda la prioridad de la tarea bloqueada.

Mediante este protocolo, una tarea en una ejecución no puede ser bloqueada durante más de una sección crítica.

En [Baker, 1993] se proponen algunas extensiones del protocolo PCP que tienen en cuenta, entre otras cosas, el uso de prioridades dinámicas (tal y como hace el algoritmo EDF, descrito en el apartado 2.7).

### 2.4.2. Cálculo de tiempos de bloqueo con el protocolo PIP

Debido a la existencia de recursos compartidos, las tareas pueden sufrir bloqueos que dependen de la asignación de prioridades y del protocolo de sincronización utilizado. En este apartado se va a describir como ejemplo un método para obtener los retardos por bloqueo de las tareas en el caso de que la asignación del recurso compartido se realice según el protocolo PIP (protocolo básico de herencia de prioridades), descrito en el apartado anterior.

Sea  $b$  una variable que almacena el valor actual de bloqueo para las tareas de más prioridad,  $ru_i$  el tiempo de utilización del recurso por parte de la tarea  $i$ ,  $pc_r$  la prioridad máxima del recurso  $r$ , y  $pr_i$  la prioridad de la tarea  $i$ .

Para calcular el tiempo de bloqueo  $b_i$  de cada tarea  $i$ , hay que seguir los siguientes pasos:

1. Se ordenan las tareas de acuerdo con su prioridad, de forma que la tarea 1 es la más prioritaria.
2. Inicialización de  $b$ :  $b = 0$
3. Comienzo con la tarea menos prioritaria:  $i = n$
4.  $b_i = b$
5. Si  $ru_i > b$ , entonces  $b = ru_i$

6. Si  $i = 1$ , termina el algoritmo. Si  $pr_{i-1} > pc_r$ , entonces  $b = 0$
7. Iterar:  $i = i-1$  y continúa en el paso 4.

Supongamos que un sistema está formado por seis tareas periódicas ordenadas de mayor a menor prioridad y con valores de prioridad  $pr_1=60$ ,  $pr_2=50$ ,  $pr_3=40$ ,  $pr_4=30$ ,  $pr_5=20$ ,  $pr_6=10$ . En la siguiente tabla se presentan los parámetros temporales que describen a las tareas, los periodos  $t_i$ , los tiempos de ejecución  $p_i$ , y los tiempos de utilización del recurso  $ru_i$ .

$pr_1=60$	$pr_2=50$	$pr_3=40$	$pr_4=30$	$pr_5=20$	$pr_6=10$
$t_1=40$	$t_2=100$	$t_3=150$	$t_4=350$	$t_5=480$	$t_6=500$
$p_1=2$	$p_2=20$	$p_3=20$	$p_4=50$	$p_5=34$	$p_6=38$
$ru_1 = 0$	$ru_2 = 5$	$ru_3 = 0$	$ru_4 = 12$	$ru_5 = 0$	$ru_6 = 10$

Aplicando el algoritmo anterior, se tiene:

$$b=0, b_6=0$$

$$\text{Como } ru_6 > b, \quad b = ru_6 = 10$$

$$\text{Como } i \text{ es distinto de 1 y no se cumple que } pr_{i-1} > pc_r \text{ (20 > 50), } i = i-1 = 5$$

$$b_5=b=10$$

$$\text{No se cumple que } ru_5 > b, \text{ (0 > 10)}$$

$$\text{Como } i \text{ es distinto de 1 y no se cumple que } pr_{i-1} > pc_r \text{ (30 > 50), } i = i-1 = 4$$

$$b_4=b=10$$

$$\text{Como } ru_4 > b, \quad b = ru_4 = 12$$

$$\text{Como } i \text{ es distinto de 1 y no se cumple que } pr_{i-1} > pc_r \text{ (40 > 50), } i = i-1 = 3$$

$$b_3=b=12$$

$$\text{No se cumple que } ru_3 > b, \text{ (0 > 12)}$$

$$\text{Como } i \text{ es distinto de 1 y no se cumple que } pr_{i-1} > pc_r \text{ (50 > 50), } i = i-1 = 2$$

$$b_2=b=12$$

$$\text{No se cumple que } ru_2 > b, \text{ (5 > 12)}$$

$$\text{Como } i \text{ es distinto de 1 y se cumple que } pr_{i-1} > pc_r \text{ (60 > 50), } b = 0 \text{ y } i = i-1 = 1$$

$$b_1=b=0$$

$$\text{No se cumple que } ru_1 > b, \text{ (0 > 0)}$$

$$\text{Como } i = 1, \text{ termina el algoritmo.}$$

Resumiendo, los tiempos de bloqueo que resultan son  $b_1=0$ ,  $b_2=12$ ,  $b_3=12$ ,  $b_4=10$ ,  $b_5=10$  y  $b_6=0$

Como se puede comprobar, la tarea 3 y la 5, aunque no utilizan el recurso de datos, también sufren retardos por bloqueo.

Con esta información y con los plazos de las tareas se pueden obtener los límites de utilización, o bien realizar un análisis de planificabilidad del sistema aplicando un algoritmo de análisis temporal, como se describe en los siguientes apartados.

### 2.4.3. Límites de utilización

Sha, Rajkumar y Lehoczky ([Sha et al., 1990]) añadieron un término de tiempo de bloqueo a la desigualdad planteada por Liu y Layland ([Liu, Layland, 1973]) en los límites de utilización de la CPU para la planificación, que permite tener una condición suficiente pero no necesaria para que las tareas sean planificables.

Siendo los plazos iguales a los periodos,

- El límite de utilización es  $U(n) = n \cdot \left(2^{1/n} - 1\right) \rightarrow \ln 2$
- La utilización total de las tareas es  $\sum_{i=1}^n \frac{p_i}{t_i}$
- El término de bloqueo es  $b_{tot} = \max_{i=1}^n \left(\frac{b_i}{t_i}\right)$

donde  $b_i$  es el tiempo de bloqueo de cada tarea. La condición suficiente de planificabilidad queda como:

$$U_{tot} + b_{tot} \leq U(n)$$

Existe también un límite de utilización para el caso en el que los plazos son menores que los periodos. Para más detalles consúltese [Lehoczky et al., 1991], [Klein et al., 1993].

#### 2.4.4. Algoritmo de análisis temporal con bloqueos y plazos menores que los periodos

Para obtener una condición necesaria y suficiente de planificabilidad de un sistema se pueden calcular los peores tiempos de respuesta de las tareas y compararlos con sus plazos. Este análisis tiene en cuenta la existencia de tiempos de bloqueo debido a la existencia de recursos de datos compartidos.

Este algoritmo, válido para plazos menores que los periodos, es una extensión del algoritmo (2.2) donde  $s_{i,k}$  se calcula iterativamente como:

$$s_{i,k} = b_i + p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

y el tiempo de finalización  $wc_i$  de la tarea  $i$  es  $s_{i,k}$  cuando  $s_{i,k} = s_{i,k-1}$ . Este algoritmo es aplicable tanto a tareas periódicas como esporádicas (tareas no periódicas con tiempos entre llegadas acotados).

#### 2.4.5. Algoritmo de análisis temporal con bloqueos, plazos menores que los periodos y activación retrasada

Si una tarea  $i$  tiene retraso de la activación o “release jitter”  $j_i$ , puede que esté lista en un instante posterior a la llegada del evento correspondiente pero antes de un tiempo acotado. Esto puede ocurrir en el caso de que el reconocimiento la llegada de una tarea consuma un tiempo no nulo, o bien porque las tareas esperen la llegada de un mensaje. En [Audsley et al., 1993a] y [Tindell et al., 1994] se extiende el algoritmo presentado en (2.2), para el caso de que las tareas tengan “release jitter”. Entonces, el cálculo iterativo de  $s_{i,k}$  se realiza aplicando:

$$s_{i,k} = b_i + p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1} + j_j}{t_j} \right\rceil$$

El tiempo de finalización de la tarea  $i$  se calcula como  $wc_i = s_{i,k} + j_i$ , cuando  $s_{i,k} = s_{i,k-1}$

Este análisis es aplicable tanto a tareas periódicas como esporádicas. Además se puede extender también el análisis de planificabilidad para tareas esporádicamente periódicas. Estas son tareas que llegan periódicamente un número de veces, luego se interrumpen las llegadas, para reaparecer de nuevo más tarde. Este comportamiento

aparece en sistemas reales cuando al llegar un evento se inicia una tarea y luego, durante un intervalo de tiempo corto monitoriza o controla alguna parte del sistema.

### 2.4.6. Algoritmo de análisis temporal con bloqueos y plazos mayores que los periodos

El algoritmo que se muestra en (2.3) fue desarrollado por Lehoczky ([Lehoczky, 1990]) y, de forma independiente, también por Tindell ([Tindell, 1992], [Tindell et al., 1994]) y es aplicable tanto a tareas periódicas como esporádicas. Con este algoritmo se pueden calcular los peores tiempos de cada tarea considerando cualquier plazo, es decir, siendo éstos menores, iguales o mayores que el periodo correspondiente. Además, este algoritmo tiene en cuenta la existencia de tiempos de bloqueo en las tareas, causadas por el acceso de una tarea menos prioritaria a un recurso compartido.

```

for  $i=1$  to  $n\_tareas$  do
 $s_{i,0} = b_i + \sum_{j=1}^i p_j$  ;  $n = 1$ 
loop
while ( $s_{i,k} \neq s_{i,k-1}$ )
{if ( $s_{i,k} > t_i(n-1) + d_i$ ) tarea  $i$  no planificable
 $k = k + 1$ }
 $e_{i,n} = s_{i,k} - t_i(n-1)$ 
if ( $e_{i,n} > t_i$ )
 $n = n + 1$  ;  $s_{i,k} = s_{i,k} + p_i$ 
else
 $wc_i = \max(e_{i,x}) \quad \forall x \in \mathbb{N}, 1 \leq x \leq n$ 
Siguiente tarea
end
endloop
end
(2.3)  $s_{i,k}(s_{i,k-1}) = b_i + n \cdot p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$ 

```

En (2.3) se supone que las tareas se encuentran ordenadas de acuerdo a su prioridad, de forma que la tarea 1 es la más prioritaria. El parámetro  $n$  indica la ejecución de la tarea  $i$  correspondiente al tiempo de respuesta  $e_{i,n}$  que se está calculando en cada momento. Se siguen calculando tiempos de finalización de ejecuciones de la tarea

mientras sus tiempos de respuesta  $e_{i,n}$  son mayores que el periodo  $t_i$ , es decir, mientras dure el periodo de ocupación.

El peor tiempo de respuesta de la tarea  $i$  es el máximo de todos los tiempos de respuesta de  $i$ , calculados dentro del periodo de ocupación.

En [Tindell et al., 1994], [Burns, Wellings, 1996] se extiende el algoritmo anterior al caso de que las tareas tengan “release jitter”, de forma que la ecuación iterativa del algoritmo de planificabilidad queda:

$$(2.4) s_{i,k}(s_{i,k-1}) = b_i + n \cdot p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1} + j_j}{t_j} \right\rceil$$

Cuando  $s_{i,k} = s_{i,k-1}$ , el tiempo de respuesta es  $e_{i,n} = s_{i,k} - t_i(n-1) + j_i$

### Plazos internos

Es posible que el modelo necesite tener en cuenta los tiempos de cambio de contexto entre tareas y asignárselos a la tarea que entra en ejecución, o bien a la tarea que deja de ser ejecutada. Con tiempos de cambio de contexto realistas (distintos de cero) es posible que el plazo de la tarea no se encuentre al final del cambio de contexto. Es más, puede ocurrir que el plazo, relacionado con el último evento observable de la tarea no se encuentre al final de la ejecución de la misma, sino que exista una serie de acciones internas después del evento de salida.

En la Figura 2.10 se muestra una representación de la ejecución de una tarea (sin considerar las posibles expulsiones de tareas más prioritarias). El bloque  $a$  es el cambio de contexto inicial para comenzar la ejecución de la tarea, el bloque  $b$  es el tiempo de ejecución de la tarea hasta el último evento observable, el bloque  $c$  es el conjunto de acciones internas posteriores a dicho evento y el bloque  $d$  es el cambio de contexto al final de la ejecución de esta tarea. El plazo de la tarea se encuentra al final del bloque  $b$ .

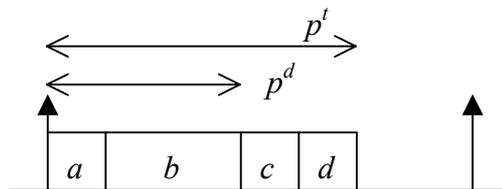


Figura 2.10: Fases de ejecución de una tarea considerando cambios de contexto

Sea  $p^d$  el tiempo de ejecución requerido antes del plazo interno (bloques  $a$  y  $b$ ) y el tiempo total  $p^t$  de ejecución de la tarea en un periodo (todos los bloques). Como no se requiere finalizar el tiempo total de ejecución dentro del periodo  $t$ , la ecuación que se debe utilizar y extender es la correspondiente a plazos mayores que los periodos (2.4) La ecuación iterativa para el cálculo de  $s_{i,k}$  en este caso es:

$$s_{i,k}(s_{i,k-1}) = b_i + (n-1) \cdot p_i^t + p_i^d + \sum_{j=1}^{i-1} p_j^t \cdot \left[ \frac{s_{i,k-1} + j}{t_j} \right]$$

Y la condición de planificabilidad es que los tiempos de respuesta estén antes del plazo interno correspondiente ([Burns, Wellings, 1996]).

## 2.5. Sobrecargas

Tal y como se ha mostrado en la descripción de los plazos internos, a menudo hay que tener en cuenta los tiempos de cambio de contexto y la manipulación de colas, ya que no es posible despreciarlos en un modelo realista ([Burns, Wellings, 1996]). Incluso si el cambio de contexto lo realiza otro procesador en paralelo, habrá un tiempo asociado que debe ser tenido en cuenta. Cuando se utiliza un “kernel”, si se desconocen los tiempos de las operaciones normalmente se toman tiempos de sobrecarga grandes para mayor seguridad. Además, el manejador de interrupciones de reloj debe ocuparse también de la cola de retardo. Si no hay tareas en dicha cola, la sobrecarga puede ser de sólo unos pocos microsegundos. Pero si, por ejemplo, hay 20 tareas periódicas con lanzamiento en el mismo instante, la sobrecarga de llevar todas las tareas de la cola de retardo a la cola de ejecución puede ser de cientos de microsegundos.

Los cambios de contexto se pueden tener en cuenta añadiendo el tiempo asociado al tiempo de ejecución de la tarea que causa el cambio de contexto. En la mayoría de los entornos de ejecución, el cambio de contexto lo realiza una sección de código no expulsable que, por tanto, puede producir bloqueos. Por ejemplo, si ocurre una interrupción de reloj durante un cambio de contexto, ésta se ve retrasada, y si dicha interrupción lanzaba una tarea de prioridad alta, la tarea se ve retrasada igualmente. Como ya se indicó, el tiempo de bloqueo depende del protocolo utilizado. Al considerar los retardos, el nuevo tiempo de bloqueo es el máximo entre el calculado para el protocolo en uso y el mayor tiempo de ejecución no expulsable del “kernel”.

Para tener en cuenta en el análisis de planificabilidad el retardo por la manipulación de la cola que realiza el manejador de interrupciones de reloj, hay que considerar las sobrecargas causadas por cada tarea periódica. Este retardo se puede modelar mediante dos parámetros: la sobrecarga  $p^{clk}$  de cada interrupción, supuesto que no se mueve ninguna tarea de la cola, y el coste  $p^{cola}$  de mover una tarea de la cola de retardo a la cola de ejecución. Entonces, la ecuación iterativa para el cálculo del tiempo de finalización de las tareas queda como sigue:

$$s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j \in hp(i)} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil + p^{clk} \cdot \left\lceil \frac{s_{i,k-1}}{t^{clk}} \right\rceil + \sum_{f \in pts} p^{cola} \cdot \left\lceil \frac{s_{i,k-1}}{t_f} \right\rceil$$

donde  $hp(i)$  es el conjunto de tareas más prioritarias que  $i$ , y  $pts$  es el conjunto de todas las tareas periódicas.

Para tareas esporádicas (lanzadas por una interrupción), hace falta tener en cuenta el tiempo de ejecución del manejador de interrupciones. En la mayor parte de los sistemas este manejador se ejecuta con una prioridad más alta que la tarea lanzada, pudiendo ser de prioridad más alta que cualquier tarea. Para ello se añade un término más a la ecuación anterior

$$s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j \in hp(i)} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil + p^{clk} \cdot \left\lceil \frac{s_{i,k-1}}{t^{clk}} \right\rceil + \sum_{f \in pts} p^{cola} \cdot \left\lceil \frac{s_{i,k-1}}{t_f} \right\rceil + \sum_{g \in sts} p^{int} \cdot \left\lceil \frac{s_{i,k-1}}{t_g} \right\rceil$$

donde  $sts$  es el conjunto de tareas esporádicas lanzadas por interrupciones,  $p^{int}$  es el tiempo de interrupción (suponiendo que es constante para todas las interrupciones),  $pts$  es el conjunto de tareas periódicas, y  $hp(i)$  es el conjunto de tareas más prioritarias que  $i$ .

## 2.6. Relaciones de precedencia en tareas con prioridades fijas

Las relaciones de precedencia se pueden modelar mediante transacciones ([Burns, Wellings, 1996]) que se implementan a su vez mediante tareas y recursos de datos compartidos.

### 2.6.1. Transacciones

Una transacción sencilla, y no distribuida, se puede implementar mediante una sola tarea, pero normalmente las transacciones están formadas por una serie de tareas relacionadas según un determinado orden de precedencia. Además, las tareas y los recursos de datos compartidos se pueden encontrar distribuidos en diferentes nodos del sistema. El mecanismo de transacción sirve para enlazar las actividades de entrada y salida, que tienen plazos asociados, reflejando propiedades principio-a-fin. Además al igual que las tareas las transacciones pueden ser periódicas o esporádicas.

Consideremos por ejemplo una transacción periódica no distribuida que consta de tres tareas  $\tau_1, \tau_2, \tau_3$  y que el orden de precedencia entre ellas es  $\tau_1, \tau_2, \tau_3$  que se representa mediante el operador de precedencia  $\prec$ . Para la ejecución  $i$ -ésima de cada tarea se tiene:

$$(2.5) (\tau_1, i) \prec (\tau_2, i) \prec (\tau_3, i) \forall i$$

Por tanto, la  $i$ -ésima ejecución de la tarea  $\tau_1$  ha de terminar antes de que comience la tarea  $\tau_2$ , y lo mismo para las tareas  $\tau_2$  y  $\tau_3$ . En las siguientes figuras se muestran tres formas de implementar esta transacción. En las tres implementaciones  $\tau_1$  es una tarea periódica, ya que la transacción es periódica. En la Figura 2.11 las tareas  $\tau_2$  y  $\tau_3$  también son periódicas y tienen el mismo periodo que  $\tau_1$  pero se lanzan con retardo. El flujo de datos entre tareas se realiza a través de recursos de datos compartidos. El tiempo de respuesta  $e_i$  de cada tarea se mide desde que se lanza cada una y los retardos, u “offsets” se representan como  $o_2$  y  $o_3$ .

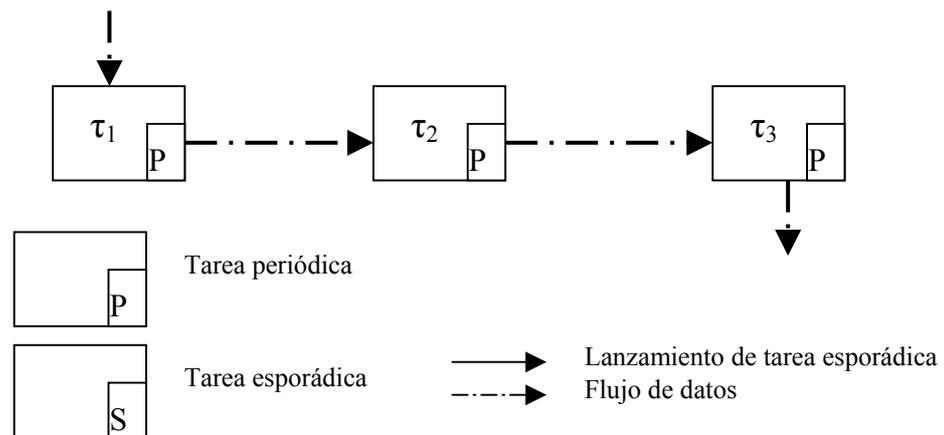


Figura 2.11: Primera implementación de la transacción

Para que se satisfaga la relación de precedencia, la planificación debe asegurar:

$$e_1 < o_2 \quad \text{y} \quad e_2 + o_2 < o_3$$

Y para que se cumpla el plazo  $d$  de la transacción, se debe verificar:

$$o_3 + e_3 < d$$

En la Figura 2.12,  $\tau_2$  y  $\tau_3$  son tareas esporádicas lanzadas por  $\tau_1$  y  $\tau_2$  respectivamente.  $\tau_1$  envía un evento a  $\tau_2$  como su última acción. En este caso el envío de un evento a la tarea esporádica puede llevar asociado también un flujo de datos entre ambas.

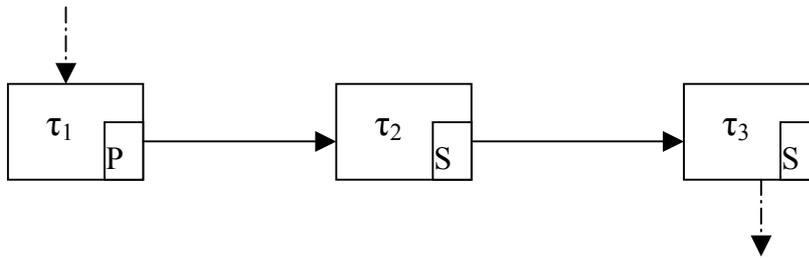


Figura 2.12: Segunda implementación de la transacción

En esta segunda implementación la relación (2.5) se satisface por definición. En este caso para cumplir el plazo de la transacción la condición que se debe cumplir es:

$$(2.6) \quad e_1 + e_2 + e_3 < d$$

Esta segunda implementación tiene la ventaja de que el tiempo de respuesta total es posiblemente menor que el de la primera implementación. Esto se debe a que  $o_2$  puede ser mayor que  $wc_1$ . No solamente el peor tiempo de respuesta puede ser menor sino también el mejor y el medio, de forma que la respuesta puede llegar a ser muy rápida. Debido a retardos en la activación de tareas periódicas (“release jitter”) puede ocurrir que ciertas aplicaciones se vuelvan inestables si la salida se obtiene demasiado pronto. Por tanto una transacción puede tener asociada tanto un plazo como un “jitter” de salida máximo, es decir la salida se debe producir dentro del intervalo  $[d - jo, d]$  donde  $jo$ , que es una constante, es el “jitter” de salida.

Supongamos que el tiempo mínimo de respuesta de todas las tareas es 0. Entonces la segunda implementación tiene como límites del tiempo de respuesta total  $[0, e_1+e_2+e_3]$ , mientras que la primera implementación tiene unos límites más ajustados  $[o_3, o_3+e_3]$ , que pueden ser más adecuados a los requisitos.

La tercera implementación (Figura 2.13) combina las mejores características de las dos anteriores obteniendo un tiempo de respuesta total bajo y unos límites de respuesta más ajustados. En este caso,  $\tau_2$  es esporádica y  $\tau_3$  es periódica y los límites de la respuesta son los mismos que en la implementación 2,  $[o_3, o_3+e_3]$ , sólo que  $o_3$  puede tener un valor menor:

$$e_1 + e_2 < o_3$$

Las ventajas de esta última implementación aumentan con el número de tareas de la transacción: sólo la primera y la última son tareas periódicas.

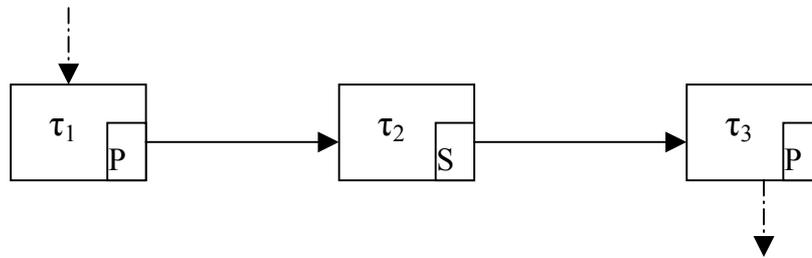


Figura 2.13: Tercera implementación de la transacción.

## 2.6.2. Análisis temporal

Hasta ahora, los métodos de análisis han permitido calcular los peores tiempos de respuesta para tareas individuales. Sin embargo, tal y como se ha visto, los requisitos temporales se refieren a características principio-a-fin asociadas a transacciones que se ejecutan en el sistema. Aunque algunas transacciones están formadas por una sola tarea, normalmente no suele ser así. A continuación se presenta la forma de verificar requisitos temporales a nivel de sistema, mediante una serie de ejemplos. Se supondrá que se conocen los peores tiempos de respuesta de las tareas. [Burns, Wellings, 1996]

### *Un bucle de control*

El ejemplo más sencillo consiste en una tarea periódica que lee una entrada del entorno y produce una salida de control. El requisito básico para esta tarea es enviar

la salida dentro de un tiempo acotado, que es el plazo. En este caso la verificación consiste simplemente en comprobar que  $e \leq d$

*Respuesta a un evento mediante una tarea esporádica*

También se puede poner un tiempo límite o plazo a la respuesta del sistema a un evento externo, manifestado como una interrupción (condición de lanzamiento de una tarea esporádica). También en este caso basta con comprobar que  $e \leq d$ .

*Respuesta a un evento mediante una tarea periódica.*

El evento externo puede ser resultado de un sondeo o “polling” (de forma periódica se comprueba la llegada de nuevos eventos). En el peor caso, la llegada del evento se producirá justo después del sondeo periódico y la comprobación se realizará en el siguiente periodo  $t$ . Por tanto hay que verificar que  $t + e \leq d$ .

*Cadena de precedencia en el mismo procesador.*

En la Figura 2.12 se vio una transacción formada por tres tareas, las dos últimas esporádicas lanzadas por el predecesor correspondiente. Una forma de modelar esta cadena en un procesador es suponer que las tres se lanzan en el mismo instante, pero que se ejecutan en el orden correcto porque las tareas predecesoras tienen prioridades más altas. Entonces, el tiempo total de respuesta principio-a-fin es igual al tiempo de respuesta de la tarea esporádica final, con lo que se debe cumplir que  $e_3 \leq d$ , o lo que es lo mismo  $wc_3 \leq d$ . Esta ecuación difiere de la (2.6) presentada anteriormente ya que aquí el valor de  $e_3$  (y por tanto de  $wc_3$ ) se mide relativo al comienzo de la transacción e incluye a  $e_1$  y  $e_2$ .

*Cadena de precedencia distribuida.*

Supongamos que en el ejemplo anterior la comunicación entre la segunda y la tercera tarea utiliza un enlace de comunicación entre procesadores independientes. Cuando una tarea lanza una tarea esporádica local se considera apropiado incorporar dentro del tiempo de respuesta de la primera el tiempo asociado al proceso de lanzar la tarea esporádica. Pero cuando se lanza de forma remota no es así: la primera tarea envía el mensaje de activación pero el software del sistema realiza la transmisión por la red y la activación de la tarea remota. Suponiendo que  $m_2$  es el peor tiempo de retardo de comunicación cuando la tarea 2 del ejemplo lanza a la tarea 3, la condición que hay que verificar es  $e_2 + m_2 + e_3 \leq d$ . El tiempo de respuesta de la tarea 3 se calcula de acuerdo con la prioridad que tiene en su procesador, mientras que las tareas 1 y 2 están juntas, en otro procesador. Para calcular el tiempo de respuesta de la tarea 3 hará falta tener en cuenta el “release jitter” de dicha tarea. Si se considera que la tarea 3 se puede lanzar tan cerca de la 1 como se quiera, entonces  $j_3 = e_2 + m_2$ . Se

puede reducir este valor de “jitter” si se conocen los tiempos mínimos de ejecución y de comunicación.

*Cadena de precedencia usando offsets.*

En la Figura 2.11 se vio otra implementación de las relaciones de precedencia, y la comprobación a realizar era  $o_3 + e_3 \leq d$ . Cuando las tareas interactúan de forma asíncrona (a través de los recursos de datos compartidos) es preciso conocer la antigüedad que tienen los datos cuando el receptor los lee.

Cuando el recurso de datos es remoto hay que tener en cuenta el coste de la comunicación en la antigüedad de los datos. Para implementar las relaciones de precedencia se pueden usar los “offsets” sólo si los procesadores están sincronizados. Si denominamos  $\Delta$  al desfase máximo entre los dos relojes, y se tiene la tarea 3 en un nodo diferente, igual que antes, entonces el “offset” necesario es  $o_3 \geq o_2 + e_2 + m_2 + \Delta$ .

En [Gilles, 1993] el autor propone algoritmos de planificación para sistemas paralelos con restricciones de precedencia entre las tareas, distinguiendo entre restricciones AND (una tarea no puede comenzar hasta que todas sus predecesoras han terminado) y restricciones OR (una tarea está lista para su ejecución cuando algunas de sus predecesoras han finalizado).

## 2.7. Planificación basada en prioridades dinámicas: EDF

Los algoritmos de planificación basados en prioridades dinámicas permiten un mayor aprovechamiento del tiempo de la CPU que los basados en prioridades fijas. En algunos casos el límite de utilización es 1, con lo que es posible utilizar la CPU al 100% respetando todos los plazos. Por otro lado tiene la desventaja de que su comportamiento es imprevisible en caso de sobrecarga.

### 2.7.1. Descripción del algoritmo EDF

El algoritmo Earliest Deadline First (EDF), propuesto por primera vez por Horn ([Horn, 1974]), es un algoritmo de planificación dinámica expulsor que selecciona tareas de acuerdo con su plazo absoluto. La tarea con el plazo más cercano se ejecuta

con prioridad más alta. EDF no hace suposición alguna sobre la periodicidad de las tareas, por lo que se puede usar para planificar tanto tareas periódicas como aperiódicas.

El algoritmo EDF es óptimo en el sentido de que si existe una asignación de prioridades planificable, el algoritmo EDF es capaz de encontrarla. Además el algoritmo EDF es óptimo en el sentido de que minimiza el retraso máximo, o lo que es lo mismo, maximiza la distancia mínima de los tiempos de finalización a los plazos ( $wc_i - d_i$ ) para la planificación de tareas tanto periódicas como aperiódicas, con lo que no existe la condición de que el plazo sea igual al periodo ([Buttazzo, 1997]).

A continuación se muestra un ejemplo para ilustrar el algoritmo de planificación EDF. En la Figura 2.14 aparece representado el hiperperiodo de dos tareas  $A$  y  $B$ . El hiperperiodo se define como el mínimo común múltiplo de los periodos de las tareas. Los periodos de las tareas  $A$  y  $B$  son  $t_A=5$  y  $t_B=7$ , los tiempos de ejecución son  $p_A=2$  y  $p_B=4$  y los plazos coinciden con los periodos. Se puede observar, que en el instante inicial, la tarea  $A$  es la más prioritaria y cuando termina, en 2, se ejecuta la tarea  $B$ . Cuando la tarea  $A$  vuelve a estar lista, en 5, la tarea  $B$  aún no ha terminado. El plazo de la tarea  $B$  en ese instante se encuentra en 7 y el de la tarea  $A$  (en esa segunda ejecución) se encuentra en 10. Como el plazo de la tarea  $B$  está antes, esta tarea tiene prioridad más alta que la tarea  $A$  en su segunda ejecución, por tanto hasta que no termine la tarea  $B$ , no puede comenzar la tarea  $A$ .

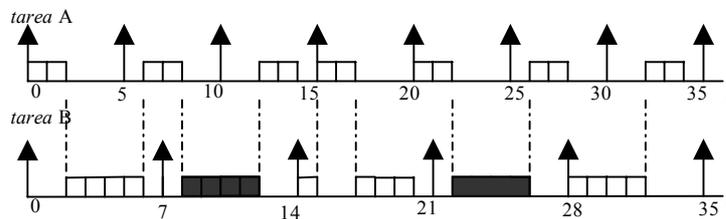


Figura 2.14: Ejemplo de planificación mediante el algoritmo EDF

## 2.7.2. Análisis temporal del algoritmo EDF

Cuando las tareas son periódicas se pueden encontrar en la literatura diversos test para determinar si el sistema es planificable. Se pueden distinguir dos situaciones diferentes: que el plazo sea igual al periodo o que el plazo sea menor que el periodo.

Cuando para todas las tareas periódicas el plazo es igual al periodo, se dice que el sistema es planificable con EDF si y sólo si la utilización total es menor o igual a 1.

$$\sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} \leq 1$$

En este caso el límite de utilización es 1, por lo que las tareas pueden utilizar el procesador hasta un 100% y ser planificables.

Cuando los plazos no coinciden con los periodos entonces  $\sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} \leq 1$  es una condición necesaria pero no suficiente. En este caso el análisis se puede realizar usando un criterio de demanda del procesador ([Buttazzo, 1997]), que queda expresado mediante el siguiente teorema:

Si  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ , entonces un conjunto de tareas periódicas con plazos menores que los periodos es planificable si y sólo si

$$\forall l \in \mathcal{D} \quad l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

donde  $b_p$  es el periodo de ocupación,  $h$  es el hiperperiodo y  $\lfloor \cdot \rfloor$  es la operación redondeo hacia abajo.

## 2.8. Planificación con tareas esporádicas

Las tareas no periódicas se pueden subdividir en dos categorías: aperiódicas y esporádicas ([Audsley, 1990]). Recordemos que las tareas aperiódicas son aquellas cuya frecuencia de lanzamiento no está acotada. Debido a esta característica, sus requisitos temporales son flexibles, ya que si puede haber un número arbitrario de estas tareas activas, es imposible garantizar los plazos. Las tareas esporádicas tienen frecuencias no constantes, pero acotadas. Es posible garantizar sus plazos analizando el sistema para la frecuencia máxima. Por tanto estas tareas son no periódicas con requisitos temporales estrictos.

La llegada de tareas esporádicas se puede implementar mediante un *manejador de interrupciones* ([Klein et al., 1993]). Se utilizan interrupciones para detectar los eventos y el servicio a estas tareas se hace a nivel de prioridad hardware, que es más alta que el resto de prioridades. No hay una política determinada para limitar el tiempo de este tipo de ejecuciones. En este caso las respuestas son rápidas pero se puede comprometer la planificabilidad de tareas estrictas con prioridad más baja.

Además, esta implementación puede ser peligrosa si se exceden los límites de frecuencia de llegadas.

Otra forma de tratar las tareas esporádicas es mediante el uso de *prioridades software* ([Klein et al., 1993]). En este caso se utilizan también interrupciones para detectar los eventos, pero el servicio a las tareas aperiódicas se realiza principalmente a nivel de prioridades software (una pequeña parte del tiempo es para el manejador de interrupciones cuya prioridad es hardware y el resto es para la tarea aperiódica que tiene una prioridad software menor). Para el caso en el que se ejecuta la respuesta a un solo nivel de prioridad, la asignación óptima es la correspondiente a prioridades monótonas en plazo. Se pueden usar las prioridades software cuando el manejador de interrupciones resulta inapropiado debido a los tiempos de respuesta. Al igual que con el manejador de interrupciones, esta implementación puede resultar peligrosa si se exceden los límites de frecuencia de llegadas.

También se puede detectar la llegada de eventos esporádicos mediante sondeo o “polling”. En este caso el servicio a tareas esporádicas se realiza a nivel de prioridad software y el sondeo limita la frecuencia de las llegadas. Este método es necesario cuando la llegada de eventos no produce interrupciones. Los tiempos de respuesta dependen del periodo de sondeo, lo que a su vez afecta a la sobrecarga. El sondeo reduce el ancho de banda de procesamiento, ya que si no hay una tarea esporádica activa cuando se ejecuta el proceso de sondeo se desperdicia el tiempo de procesamiento asociado al mismo. Además, si llega una tarea esporádica cuando se ha acabado el tiempo de procesamiento del sondeo o llega demasiado tarde para ser atendida por el sondeo, entonces esta tarea debe esperar al siguiente periodo de sondeo ([Klein et al., 1993], [Audsley, 1990]).

Algoritmos que preservan el ancho de banda y que se han propuesto para usar conjuntamente con RMS son el servidor aplazable o “deferrable server” y el servidor esporádico o “sporadic server” ([Lehoczky et al., 1987], [Sprunt et al., 1989]). Como se verá en el siguiente apartado, estos algoritmos también se utilizan para atender a tareas aperiódicas flexibles.

En [Audsley et al., 1993a] los autores proponen un análisis de planificabilidad (necesario y suficiente) para sistemas con tareas periódicas y esporádicas planificadas según un planificador de prioridades fijas y expulsor. Este análisis tiene en cuenta el problema del “release jitter”, los tiempos de bloqueo que aparecen cuando las tareas comparten recursos de datos, y también la posibilidad de que los plazos sean menores que los periodos. Las tareas esporádicas tienen un periodo variable pero acotado, y en el análisis de planificabilidad se estudia el peor caso que es el límite inferior del periodo de la tarea esporádica. En principio los autores consideran sólo la planificación de tareas en un único procesador, pero las técnicas

propuestas se pueden aplicar también a sistemas distribuidos con asignación estática de tareas. Extraen como resultado del análisis la conclusión de que no es necesaria la utilización de servidores aperiódicos, y el análisis a realizar es el descrito en (2.2). Los autores aplicaron este análisis a un caso de estudio de un sistema de aviación planteado por Locke ([Locke et al., 1991]). En [Tindell et al., 1994] el análisis se extiende al caso de tener plazos mayores que los periodos.

## 2.9. Planificación con tareas aperiódicas

Muchas aplicaciones de tiempo real requieren la ejecución de actividades con diferentes restricciones temporales. Así, las actividades relacionadas con la adquisición de datos de los sensores se deben ejecutar periódicamente para poder reconstruir de forma fiable las señales externas, mientras que otras actividades no necesitan una ejecución periódica sino que son disparadas de manera aperiódica cuando se verifica una determinada condición.

Como ya se ha explicado, si una tarea tiene un plazo crítico, entonces la tarea se denomina estricta. Por otra parte, si el no cumplimiento del plazo no tiene consecuencias serias para el sistema, se dice que la tarea es flexible.

En las aplicaciones de control críticas las tareas periódicas suelen ser estrictas y las no periódicas estrictas o flexibles. Las tareas no periódicas estrictas sólo se pueden garantizar si existe un límite inferior en el tiempo entre activaciones consecutivas de las tareas. Cuando una tarea no periódica tiene dicho límite se denomina esporádica, y para garantizar su plazo se puede tratar como una tarea periódica.

El principal problema en la planificación de conjuntos híbridos de tareas es garantizar los plazos de las tareas periódicas y esporádicas y obtener una buena respuesta de las tareas aperiódicas flexibles. En este caso, los objetivos de la planificación son ([Ghazalie, Baker, 1995]):

- Garantizar el cumplimiento de los plazos estrictos
- Lograr tiempos medios de respuesta rápidos para las tareas aperiódicas con plazos flexibles
- Lograr un nivel alto de utilización del procesador

Este problema ha sido tratado en la literatura en dos vertientes: la asignación estática de prioridades y la asignación dinámica ([Buttazzo, Sensini, 1997]).

### **2.9.1. Asignación de prioridades fijas en planificación con tareas aperiódicas**

Se pueden encontrar en la literatura varias propuestas para la planificación de tareas híbridas, basadas en prioridades fijas. La más sencilla y menos efectiva consiste en ejecutar las tareas flexibles con una prioridad menor que las tareas estrictas. Esto relega a las tareas flexibles a una ejecución en “background”, esto es, con una prioridad más baja que las otras tareas, lo que será una desventaja si su comportamiento está íntimamente ligado al de las tareas estrictas por aumentar su precisión, utilidad o nivel de confianza ([Davis et al., 1993]).

Otra propuesta consiste en ejecutar las tareas flexibles con una prioridad más alta bajo el control de una tarea servidora pseudo estricta, como por ejemplo un servidor de sondeos (“polling”). Un servidor de sondeos es una tarea periódica con prioridad fija (normalmente la más alta) y una capacidad de ejecución. La capacidad del servidor se calcula off-line y normalmente se fija al máximo valor posible, de forma que las tareas estrictas, incluido el servidor, sean planificables. En tiempo de ejecución el servidor de sondeos se lanza periódicamente y su capacidad se utiliza para dar servicio a las tareas flexibles. Una vez agotada dicha capacidad, la ejecución se suspende hasta que vuelve a ser rellenada la siguiente vez que se lanza el servidor.

El servidor de sondeo suele mejorar significativamente los tiempos de respuesta de las tareas flexibles frente al procesamiento en “background”. Sin embargo, si las tareas flexibles listas para la ejecución exceden la capacidad del servidor, algunas de ellas deben esperar a la siguiente ejecución pudiendo obtener tiempos de respuesta más largos. Por otro lado, si no hay tareas flexibles listas cuando se lanza al servidor se pierde su capacidad.

#### *Servidores que preservan el ancho de banda:*

Esta última desventaja se evita en los algoritmos Priority Exchange Server o servidor de intercambio de prioridades ([Lehoczky et al., 1987]), Deferrable Server o servidor aplazable ([Strosnider et al., 1995]), y Sporadic Server o servidor esporádico ([Sprunt et al., 1989]). El más sencillo es el servidor denominado aplazable o Deferrable Server (DS). Los otros dos, aunque de mayor complejidad, fueron diseñados para lograr respuestas mejores. En algunos trabajos se afirma que el servidor esporádico o Sporadic Server (SS) es el mejor y, de hecho, éste es el único que tiene soporte por parte del estándar de POSIX (Portable Operating System

Interface) P1003.1 (<http://www.pasc.org>). Sin embargo, en [Bernat, Burns, 1999] se afirma que el servidor aplazable tiene respuestas tan buenas como el servidor esporádico y que se debería incluir en el estándar.

Los tres se basan en una asignación de prioridades fijas de acuerdo con el algoritmo RMS y en principios similares a los del servidor de sondeo. Sin embargo son capaces de preservar la capacidad en el caso de que no haya tareas flexibles pendientes. Debido a esta propiedad se dice que estos algoritmos preservan el ancho de banda. Los tres algoritmos difieren en la forma de preservar y rellenar la capacidad del servidor, y en el análisis de planificabilidad necesario para determinar la capacidad máxima.

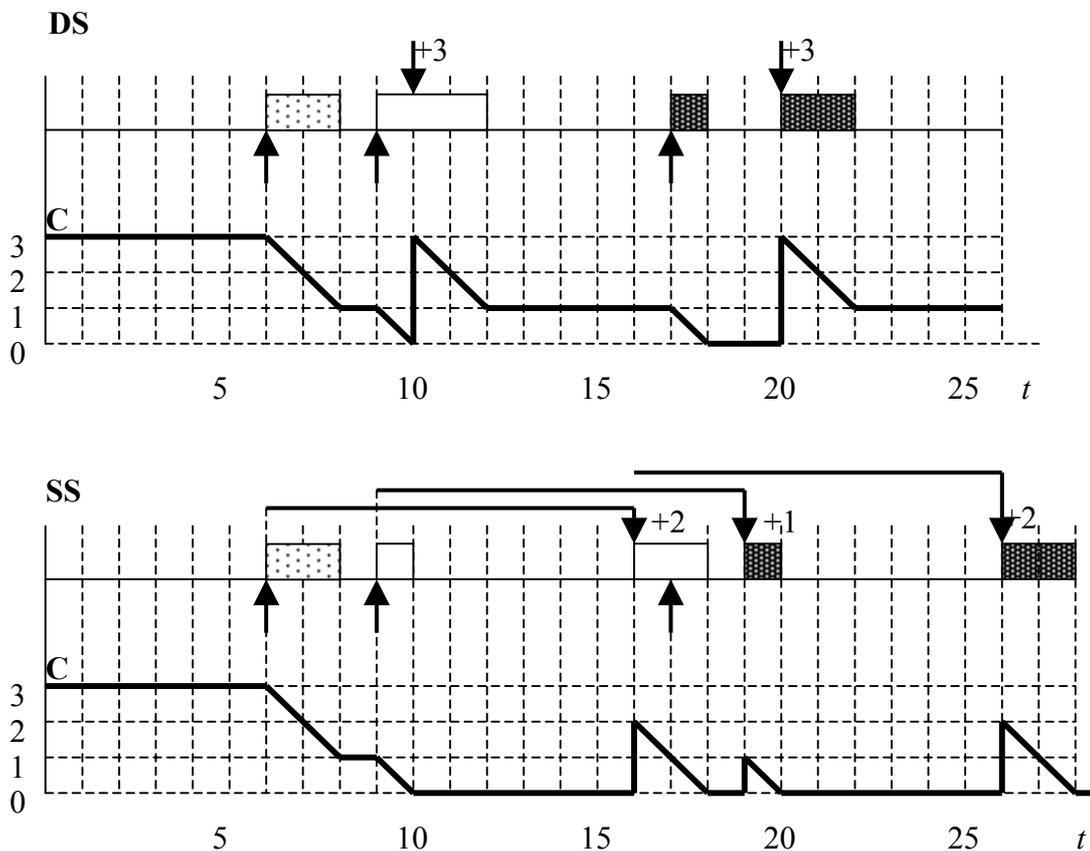


Figura 2.15: Operación de los servidores DS y SS para tres llegadas aperiódicas

La Figura 2.15 reproduce un gráfico de [Bernat, Burns, 1999] que muestra la diferencia en la operación de los servidores DS y SS, dadas tres tareas con llegadas

aperiódicas en los instantes 6, 9 y 17. Los tiempos de ejecución de cada tarea son 2, 3 y 3 respectivamente. El periodo de relleno  $t_s$  es 10 para ambos servidores y su capacidad máxima es 3. El gráfico muestra la ejecución de las tareas, la evolución de la capacidad  $c_s$  del servidor y los instantes en que se produce el relleno de la capacidad.

En la Figura 2.15 se ve que el relleno de la capacidad del DS se realiza en instantes fijos, múltiplos del periodo de relleno, y se rellena hasta la capacidad máxima. El relleno del SS se produce un periodo de relleno después de la llegada de cualquier tarea aperiódica, y se rellena la cantidad de tiempo que esa tarea ha consumido. Si la tarea necesita más tiempo que la capacidad disponible, se ejecutará hasta agotar la capacidad y después se suspenderá la tarea hasta que se rellene la capacidad del servidor.

El análisis de planificabilidad de los sistemas que poseen servidores DS o SS fue tratado por Lehoczky et al. ([Lehoczky et al., 1987]), definiendo los límites de utilización de dichos servidores; recordemos que estos límites de utilización son condiciones suficientes pero no necesarias de planificabilidad. Una conclusión interesante del servidor SS es que se puede modelar como una tarea periódica ([Sprunt et al., 1989]) y, por tanto, cualquier tipo de análisis de planificabilidad aplicable a tareas periódicas se puede aplicar también a un conjunto de tareas con un servidor SS.

Para obtener una condición necesaria y suficiente de planificabilidad se pueden calcular los peores tiempos de respuesta de las tareas. En [Ghazalie, Baker, 1995] se formula el método de cálculo del peor tiempo de respuesta de las tareas para sistemas con servidor SS, y en [Lehoczky et al., 1987] el correspondiente a sistemas con un servidor DS.

Servidor aplazable (DS): El cálculo del peor tiempo de respuesta se realiza iterativamente, hasta que  $s_{i,k} = s_{i,k-1}$  siendo

$$s_{i,k} = p_i + \sum_{j=1}^{i-1} \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil \cdot p_j + \left( 1 + \left\lceil \frac{s_{i,k-1} - c_s}{t_s} \right\rceil \right) \cdot c_s$$

donde el último sumando corresponde a la interferencia del servidor DS en tareas menos prioritarias

Servidor esporádico (SS): el peor tiempo de respuesta se calcula iterativamente como:

$$s_{i,k} = p_i + \sum_{j=1}^{i-1} \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil \cdot p_j + \left\lceil \frac{s_{i,k-1}}{t_s} \right\rceil \cdot c_s$$

En general, los tres servidores ofrecen una respuesta mejor que el servidor de sondeo, pero siguen teniendo desventajas. No son capaces de utilizar el tiempo sobrante que puede existir de no encontrarse en el peor caso, es decir, son incapaces de reclamar la capacidad ganada cuando por ejemplo los tiempos de ejecución de las tareas estrictas son mejores que los que se tienen en el peor caso ([Davis et al., 1993]). Además, se degradan de tal forma que en condiciones de sobrecarga se comportan igual que el servidor de sondeo.

*Algoritmos de Slack Stealing o extracción de holgura:*

El algoritmo de extracción de holgura de Lehoczky y Ramos-Thuel ([Lehoczky, Ramos-Thuel, 1992]) no tiene las desventajas citadas anteriormente. Es óptimo en el sentido de que minimiza los tiempos de respuesta de las tareas aperiódicas flexibles, entre todos los algoritmos que cumplen los plazos de las tareas estrictas. El algoritmo “Slack Stealing” o extracción de holgura atiende peticiones aperiódicas haciendo que el tiempo sobrante de procesado esté disponible lo antes posible. De esta forma, roba tiempo sobrante a las tareas periódicas estrictas. La clave del problema consiste en encontrar la manera de determinar el máximo tiempo sobrante que se puede robar sin comprometer las restricciones temporales de las tareas estrictas. En [Lehoczky, Ramos-Thuel, 1992] esto se hace analizando la planificación de las tareas periódicas en un hiperperiodo (mínimo común múltiplo de los periodos) y obteniendo el tiempo sobrante entre un plazo y el de la siguiente ejecución de la tarea. En tiempo real, cuando se detecta que hay un tiempo disponible a todos los niveles de prioridad, se pueden ejecutar las tareas aperiódicas flexibles al máximo nivel de prioridad.

Desafortunadamente, la necesidad de trazar y analizar el plan en un hiperperiodo restringe la aplicabilidad del algoritmo Slack Stealer. Solamente se puede robar tiempo sobrante de tareas estrictas periódicas y sin “jitter”, ni sincronización ([Audsley et al., 1993a]). Además con hiperperiodos grandes la aplicación del algoritmo se hace impracticable, lo que puede limitar mucho el número de tareas estrictas.

En [Davis et al., 1993], se extiende el análisis dado por Audsley y sus colaboradores del algoritmo de planificación tipo DMS ([Audsley et al., 1991]), para determinar el tiempo sobrante que se puede robar de las tareas esporádicas estrictas y de las periódicas estrictas. Además se describe el análisis de planificabilidad aplicable a estos sistemas y se llega a la conclusión de que este algoritmo dinámico de

extracción de holgura es óptimo. Gracias al cálculo dinámico del tiempo sobrante, este algoritmo se puede aplicar a tareas esporádicas, tareas con “release jitter” y con sincronización. Sin embargo la sobrecarga de la ejecución dinámica de este algoritmo óptimo es tan grande que lo hace impracticable, por lo que también se han propuesto algoritmos aproximados con respuesta próxima al óptimo.

Estos algoritmos de extracción de holgura ([Lehoczky, Ramos-Thuel, 1992], [Davis et al., 1993]) se comportan de forma que las tareas flexibles puedan hacer uso del tiempo sobrante de forma inmediata (algoritmos ávidos o “greedy”) con el fin de minimizar sus tiempos de respuesta, y se ha visto que son óptimos entre todos los algoritmos que tienen este comportamiento. Sin embargo, para minimizar el tiempo de respuesta de una tarea aperiódica, a veces es necesario retrasarla incluso aunque exista tiempo sobrante en ese instante. En [Tia, 1995] y [Tia et al., 1996] se proponen algoritmos basados en extracción de holgura pero no exhiben el comportamiento ávido de los anteriores.

### **2.9.2. Asignación dinámica de prioridades**

Aunque RMS es un algoritmo óptimo, en general no logra la completa utilización del procesador. En el peor caso, la utilización máxima es del 69 por ciento ([Liu, Layland, 1973]). Por tanto, para aquellas aplicaciones con una carga de tareas periódicas elevada, se puede incrementar la utilización del procesador mediante algoritmos de planificación dinámica, tales como el Earliest Deadline First (EDF) ([Liu, Layland, 1973]) o el algoritmo Least Slack (que no se ha descrito pero se puede encontrar en [Ramamritham, 1996]). Ambos algoritmos son óptimos y logran una completa utilización del procesador, aunque el algoritmo EDF lo hace con menor sobrecarga.

La planificación de tareas aperiódicas mediante el algoritmo EDF ha sido estudiado en [Chetto, Chetto, 1989] y [Chetto et al., 1990], donde se proponen tests de aceptación para garantizar la planificabilidad de las tareas esporádicas individuales, o bien grupos de tareas aperiódicas con relación de precedencia. Estos tests de aceptación son óptimos desde el punto de vista de la utilización pero presentan una sobrecarga demasiado grande como para que sean aplicables en sistemas reales.

En [Ghazalie, Baker, 1995] Ghazalie y Baker proponen servidores para la planificación de tareas aperiódicas mediante el algoritmo EDF. Estos servidores son una versión dinámica de los ya citados DS y SS, llamados respectivamente Deadline Deferrable Server y Deadline Sporadic Server. Este último fue modificado para obtener un algoritmo más sencillo llamado Deadline Exchange Server. Además los autores analizan el impacto de los servidores en la planificabilidad de otras tareas.

Spuri y Buttazzo ([Spuri, Buttazzo, 1994], [Spuri, Buttazzo, 1996]) propusieron cinco algoritmos para el servicio de tareas aperiódicas bajo EDF. Uno de esos algoritmos, denominado Total Bandwidth Server (TBS) se extendió para manejar tareas aperiódicas estrictas y situaciones de sobrecarga ([Spuri et al., 1995]). Otra extensión del TBS, propone un método general para asignar plazos a peticiones aperiódicas flexibles ([Buttazzo, Sensini, 1997])

Los servidores citados anteriormente no consideran la compartición de recursos entre tareas periódicas y aperiódicas. Se han propuesto controles de concurrencia bajo RMS y bajo EDF pero no consideran la presencia de tareas aperiódicas. En [Buttazzo, Sensini, 1997] y [Lipari, Buttazzo, 2000] se estudia la planificabilidad de conjuntos híbridos de tareas estrictas periódicas y flexibles aperiódicas que interaccionan a través de recursos compartidos. Se tiene en cuenta, además, la posibilidad de que los plazos sean menores que los periodos.

En [Tia, 1995] se proponen algoritmos para planificar peticiones esporádicas y aperiódicas (AD: “aperiodic dynamic” y SD: “sporadic dynamic”) en sistemas con prioridades dinámicas asignadas según EDF, basados en el tiempo sobrante de procesado (“slack time”)

## 2.10. Incertidumbre en el análisis de sistemas de tiempo real

A menudo, el conocimiento acerca de los parámetros que describen a las tareas es incierto y se hace necesario tenerlo en cuenta en el análisis de planificabilidad del sistema. En la literatura se encuentran estudios que introducen la incertidumbre en el modelado de los tiempos de ejecución mediante intervalos, distribuciones de probabilidad y conjuntos borrosos (que se describirán en el capítulo 3 de esta tesis), pero existen pocos estudios en los que se tiene en cuenta la incertidumbre en los plazos.

Así, en [Sun, Liu, 1996] y [Sun et al., 1997], se introduce la imprecisión en los tiempos de ejecución mediante intervalos, y en [Tia et al., 1995] se introduce la incertidumbre mediante distribuciones de probabilidad. En [Terrier, Chen, 1994] se plantea un problema simplificado de EDF estático con plazos iguales a los periodos, donde se modelan los tiempos de ejecución como números borrosos. En [Aldarmi, Burns, 1999a] los autores analizan bases de datos de tiempo real, en las que la incertidumbre es inherente en los tiempos de ejecución y se modela mediante una cota del porcentaje de error. En [Litoiu, Tadei, 1997] se modelan los plazos como conjuntos borrosos y se propone un algoritmo de búsqueda de asignación de

prioridades óptima en este caso. En [Litoiu, Tadei, 2001b] se da una solución, que según el estudio presentado en el apartado 5.4 de esta tesis consideramos pesimista, al análisis de planificabilidad borrosa con tiempos de ejecución y plazos borrosos para sistemas con algoritmos de planificación basados en prioridades fijas.

En [Sun, Liu, 1996] los autores proponen tres algoritmos de cálculo de cotas para los tiempos de finalización de las tareas planificadas mediante prioridades fijas, cuando las tareas tienen tiempos listos arbitrarios y tiempos de ejecución variables y acotados entre un tiempo mínimo y un tiempo máximo de ejecución. El primero calcula cotas superiores de los tiempos de respuesta efectivos de las tareas en cada ejecución y se denomina ERT. El segundo se basa en el análisis de la ejecución crítica y se denomina CJ y el tercero aplica de forma iterativa el algoritmo CJ para lograr cotas más ajustadas y se denomina ITR. Cada uno de los algoritmos logra cotas más ajustadas que el anterior, pero a costa de una mayor complejidad. Por esta razón, el algoritmo ERT se puede utilizar en tiempo real para determinar si el planificador puede aceptar una nueva tarea, asegurando la ejecución de todas las tareas, mientras que el algoritmo ITR es adecuado para su aplicación off-line en el análisis de planificabilidad del sistema. En [Sun et al., 1997], además se tiene en cuenta que las tareas pueden compartir recursos de datos.

En [Tia et al., 1995] se describe la forma de extender los algoritmos de planificación y los análisis de planificabilidad desarrollados para tareas periódicas, al caso de tener tareas denominadas por los autores como semi-periódicas. Al igual que las tareas periódicas, las semi-periódicas se lanzan a intervalos regulares de tiempo, sin embargo sus tiempos de ejecución varían mucho. Esto se puede deber a la diferencia en la cantidad de datos de entrada que deban ser procesados en cada periodo, o a que en distintas llamadas de la tarea se ejecuten diferentes ramas condicionales cuya complejidad pueda variar.

Los autores modelan los tiempos de ejecución de las tareas, que consideran variables aleatorias, mediante funciones de densidad de probabilidad que se conocen a priori. En la Figura 2.16 se muestra la función del tiempo de ejecución de una tarea semi-periódica. El tiempo de ejecución máximo es significativamente mayor que el tiempo medio, con lo que si se modelaran este tipo de tareas mediante el modelo periódico que toma en cuenta sólo el caso más desfavorable, la utilización podría ser mayor que 1.

Si la utilización total máxima es mayor que 1 no se puede garantizar que estas tareas semi-periódicas sean planificables, a pesar de que su utilización total media puede ser muy pequeña. Se propone por tanto dar una garantía de planificabilidad de tipo

probabilista, mediante dos métodos: análisis probabilista de demanda de tiempo (“probabilistic time-demand analysis”) y método de la transformación de tareas (“transform-task method”).

El método de análisis probabilista de demanda de tiempo es una modificación del test de planificabilidad existente para tareas periódicas, y el resultado que proporciona es una cota de la probabilidad de que cada tarea cumpla su plazo. Este método sólo es aplicable a tareas con plazos menores o iguales que los periodos y tiene un alto coste computacional.

El método de la transformación de tareas transforma cada tarea semi-periódica en una tarea periódica de tiempo de ejecución  $p_p$  (que es un parámetro de diseño y se muestra en la Figura 2.16) seguida de una tarea esporádica. Si el tiempo de ejecución de la tarea semi-periódica supera  $p_p$  se lanza una tarea esporádica. Por tanto, la probabilidad de que se lance una tarea esporádica en cada ejecución de la tarea semi-periódica es el área marcada en la función de densidad de probabilidad que se muestra en la Figura 2.16. Los autores emplean dos formas de planificación diferentes: en la primera las tareas periódicas se planifican mediante prioridades fijas y las esporádicas mediante un servidor esporádico, y en la segunda las tareas periódicas se planifican de acuerdo con el algoritmo EDF y las tareas esporádicas mediante el algoritmo de extracción de holgura.

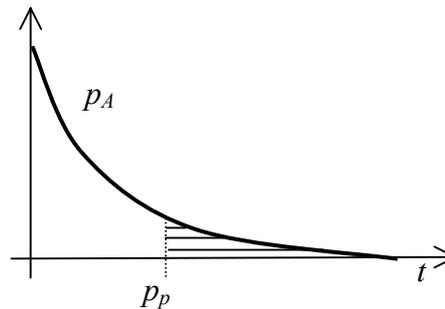


Figura 2.16: Función de densidad de probabilidad del tiempo de ejecución de una tarea semi-periódica

Este segundo método proporciona garantía absoluta de planificabilidad a aquellas tareas cuyo tiempo de ejecución es menor que  $p_p$  y garantía probabilista a tareas con tiempos de ejecución mayores. Concretamente, para las tareas esporádicas se obtiene una cota superior de la probabilidad de que cada tarea supere su plazo, que para el servidor esporádico es computacionalmente costosa de obtener y para el algoritmo de extracción de holgura es como mucho el área indicada en la Figura 2.16, cuando el plazo es mayor o igual al periodo.

En [Aldarmi, Burns, 1999a] los autores analizan bases de datos de tiempo real, en las que la incertidumbre es inherente. En estos sistemas el conocimiento impreciso acerca de los tiempos de ejecución se modela mediante una cota del porcentaje de error y de esta forma se propone un nuevo esquema de planificación. Es decir, para una transacción con tiempo de ejecución estimado  $p$  de 100 y un error en la estimación del 25% ( $\beta=0.25$ ), su tiempo mínimo de ejecución es  $p \cdot (1-\beta)=75$ . Si la estimación es precisa  $\beta=0$ . Este modelado tan sencillo es suficiente para lograr los propósitos de los autores: tomar una decisión de forma dinámica sobre la necesidad de abortar transacciones, de acuerdo con su tardanza. Sin embargo este modelo no es lo suficientemente flexible para analizar la planificabilidad del sistema.

En [Terrier, Chen, 1994] y [Terrier et al., 1995] se plantea el problema de que en la práctica, a menudo las restricciones temporales del sistema de tiempo real y las características de las tareas se conocen de forma incierta. Por ello los autores proponen aplicar el cálculo borroso a la planificación de las tareas con el fin de tener una representación más realista del conocimiento y algoritmos de planificación más flexibles.

El parámetro más obvio en el que se puede tener en cuenta la incertidumbre es en el tiempo de ejecución de una tarea, que es una de las medidas que la caracteriza. Aunque a menudo se asume que este tiempo es conocido a priori de forma precisa y cierta, muchas veces esta suposición es poco realista, si no completamente falsa, especialmente en las fases de diseño.

Muchas veces se puede considerar que las restricciones temporales del sistema son precisas, pero se puede tener en cuenta, de forma más general, que estén expresadas usando términos lingüísticos como “periodo de aproximadamente dos segundos”, “tiempo de respuesta menor que aproximadamente 0.1 segundos”, etc. En general, cuando se dice “periodo de aproximadamente dos segundos” se admiten pequeñas fluctuaciones en torno a ese valor, por ejemplo “2 segundos  $\pm$  0.1%”. Incluso según el tipo de tarea esas variaciones admitidas como válidas pueden ser mucho mayores, como puede ser el caso de una tarea que se ocupa de mostrar unos datos por pantalla cada 10 segundos y puede admitir grandes perturbaciones en el periodo mientras el valor medio permanezca en 10 segundos. En tal caso, una buena solución para modelar el periodo es mediante el uso de periodos borrosos.

Los autores, en [Terrier, Chen, 1994] y [Terrier et al., 1995], emplean el algoritmo de planificación basado en prioridades dinámicas EDF. Suponen que el plazo de cada tarea es igual al periodo, con lo que pueden usar la condición de planificabilidad:

$$\sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} \leq 1$$

extendiendo el primer término a tiempos de ejecución borrosos, y empleando aritmética borrosa para operar con números borrosos. Los autores no resuelven completamente la extensión a números borrosos de las restricciones temporales. Para medir la planificabilidad del sistema, emplean dos medidas que dependen de la distribución de posibilidad del primer término: la posibilidad y la necesidad (ver el capítulo 3):

$$\Pi\left(\sum_{i=1}^{n_{tar}} \frac{P_i}{t_i} \leq 1\right) \quad N\left(\sum_{i=1}^{n_{tar}} \frac{P_i}{t_i} \leq 1\right)$$

Basado en estas medidas, proponen una planificación estática (definida off-line) generada a partir del análisis de un hiperperiodo del sistema si estuviera planificado según EDF. De dicha planificación pueden concluir las medidas de planificabilidad del sistema y cuáles son los valores de tiempos de ejecución de cada tarea máximos para que se cumplan dichas medidas de planificabilidad.

En [Litoiu, Tadei, 2001b] se propone una extensión del análisis de planificabilidad de sistemas con prioridades fijas al caso de tener los tiempos de ejecución y los plazos modelados como números borrosos. En el modelo presentado por estos autores se introduce una función de coste denominada satisfacción de planificabilidad para medir el cumplimiento de los plazos. En general hay que estudiar el cumplimiento de todos los plazos situados dentro de un hiperperiodo, es decir, dar una medida del cumplimiento de cada plazo  $d_{ij}$  de la tarea  $i$  en su ejecución  $j$ , por parte de los tiempos de finalización correspondientes  $wc_{ij}$ .

Si los plazos son borrosos y los tiempos de ejecución son números reales, la satisfacción de planificabilidad  $S_{di}(wc_{ij})$  de una tarea  $i$  en su ejecución  $j$  depende del tiempo de finalización de la tarea  $i$  en la ejecución  $j$  que es un número real, y de la función de pertenencia del plazo borroso  $D_{ij}$

Por otro lado, si los plazos son números reales y los tiempos de ejecución son borrosos la satisfacción de planificabilidad  $S_{wci}(d_{ij})$  para la tarea  $i$  en su ejecución  $j$  depende del plazo y de la función de pertenencia del tiempo de finalización borroso  $WC_{ij}$ .

Cuando tanto los tiempos de ejecución como los plazos son borrosos la satisfacción de planificabilidad  $S_{ij}$  de una tarea  $i$  en su iteración  $j$  se obtiene como el punto de corte de  $S_{di}$  y  $S_{wci}$

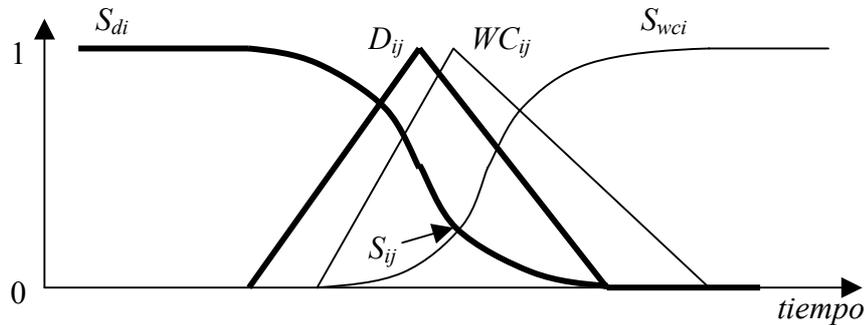


Figura 2.17: Satisfacciones  $S_{di}$ ,  $S_{wci}$  y  $S_{ij}$ .

Para obtener los tiempos de finalización borrosos en el caso de que exista un instante crítico, los autores ([Litoiu, Tadei, 2001b]) proponen una extensión del algoritmo (2.2) basada en aritmética borrosa, pero introduciendo una simplificación en la operación de redondeo hacia arriba. Debido a dicha simplificación, el resultado obtenido en la medida de planificabilidad del sistema es más pesimista que la que se obtendría de haber aplicado el principio de extensión de la teoría de la posibilidad. En el capítulo 5 se presenta más en detalle la extensión propuesta por dichos autores y se compara con la solución propuesta en esta tesis.

### **3. Planificabilidad borrosa de sistemas de tiempo real**

Tal y como ya se ha descrito en el capítulo 2, los sistemas de tiempo real se caracterizan porque tienen que respetar restricciones temporales. Constan de una serie de tareas que compiten por la misma CPU y en el caso más sencillo los parámetros que pueden definir cada tarea son el plazo, el tiempo listo, el periodo y el tiempo de ejecución ([Terrier, Chen, 1994]).

El planificador tiene que decidir dinámicamente qué tarea se tiene que ejecutar. Dependiendo del algoritmo de planificación, el sistema podrá respetar o no sus restricciones temporales (plazos). Si el sistema respeta todos los plazos, se dice que es planificable. En esta tesis el análisis se centra en la planificación basada en prioridades (que como ya se dijo en el capítulo 2 permite una planificación más flexible): cuando hay más de una tarea lista, la que tiene mayor prioridad es la que logra el control de la CPU. Recordemos, que como ya se dijo en el capítulo 2, los algoritmos de planificación más conocidos y utilizados basados en prioridades son el RMS, cuyas prioridades fijas se asignan de acuerdo con el tamaño del periodo, el DMS, cuyas prioridades fijas se asignan de acuerdo con el tamaño del plazo relativo, y el EDF, cuyas prioridades dinámicas se asignan en cada instante según sea el plazo absoluto más cercano.

En determinadas condiciones dependientes del algoritmo de planificación escogido, existen ecuaciones para determinar si un sistema es planificable. Por ejemplo para el algoritmo EDF cuando los plazos coinciden con los periodos y hay un instante crítico ([Liu, Layland, 1973]), la condición necesaria y suficiente de planificabilidad es que la utilización total del procesador sea menor o igual a 1 (ver el capítulo 2). En los casos en los que sólo se dispone de condiciones suficientes, pero no necesarias, si éstas no se cumplen es preciso simular el sistema para verificar su planificabilidad. Si todas las tareas son periódicas, aunque su ejecución sea indefinida, es suficiente simular una ventana de tiempo correspondiente al mínimo común múltiplo de los periodos de las tareas, denominado hiperperiodo.

En el proceso de diseño de un sistema, resulta interesante tener información acerca de la planificabilidad del mismo, especialmente en las primeras fases, ya que cuanto antes se encuentre un problema potencial más fácil resultará corregirlo. Como en estas fases iniciales del desarrollo los diseñadores sólo tienen una idea aproximada sobre las tareas, un modelado de su conocimiento basado en incertidumbre (probabilidad, conjuntos borrosos, ...) puede dar más información que un estudio determinista basado normalmente en el análisis del peor caso. Por otro lado, en las primeras fases de desarrollo no suele haber información probabilista disponible y normalmente no se cumplen los fuertes requisitos que se precisan para realizar un análisis probabilista.

La teoría de conjuntos borrosos y la teoría de la posibilidad proporcionan una base teórica para modelar el conocimiento de diversos expertos y combinarlo para obtener conjuntos borrosos que describan adecuadamente a las tareas. La incertidumbre

presente en las fases de diseño del sistema es la que mejor representan los conjuntos borrosos: expresiones del tipo “el tiempo de ejecución de la tarea es aproximadamente 7” se puede representar modelando el tiempo de ejecución de la tarea mediante un número borroso cuyo valor más posible es 7.

Además, gracias a los algoritmos propuestos en esta tesis, los cálculos borrosos se pueden realizar de forma más sencilla y rápida que en teoría de la probabilidad, y el modelado es muy flexible: es posible modelar un parámetro de una tarea con un conjunto borroso con cualquier función de pertenencia sin que se añada complejidad significativa al análisis como se muestra en esta tesis. Cuando los parámetros que describen a las tareas son inciertos, la respuesta sobre la planificabilidad del sistema deja de ser “sí”, o “no” y pasa a tomar un valor incierto. El análisis de planificabilidad borrosa proporciona valores de posibilidad ( $\Pi$ ) y de necesidad ( $N$ ) (o certidumbre) de que el sistema sea planificable que podrían interpretarse como cotas superiores e inferiores de probabilidad respectivamente ([Dubois, Prade, 1989a], [Dubois, Prade, 1996], [Walley, 1996]) en el sentido de que  $\text{Prob} \in [N, \Pi]$  (ver anexo C). Además, el análisis de planificabilidad borrosa puede ayudar a realizar un análisis de sensibilidad de la planificabilidad ante variaciones de los parámetros de entrada.

En este capítulo se va a extender el análisis de planificabilidad de sistemas de tiempo real a tareas modeladas con parámetros borrosos. Se va a considerar únicamente el análisis de tareas periódicas, con prioridades fijas y expulsables, y con tiempos de ejecución y plazos borrosos. En siguientes capítulos se analizarán también sistemas con prioridades dinámicas, tareas aperiódicas, la utilización de recursos compartidos y la existencia de “release jitter” o retraso de la activación de la tarea, así como el modelado de otros parámetros con conjuntos borrosos (periodos, tiempos de bloqueo, etc.).

Tras un repaso de los conceptos básicos de la teoría de la posibilidad que van a ser utilizados en esta tesis, se propone en el apartado 3.2 una definición de planificabilidad borrosa de un sistema de tiempo real definido mediante parámetros borrosos. Los pasos a seguir en ese proceso son el modelado borroso de los parámetros, descrito en el apartado 3.3, el cálculo de los tiempos de finalización borrosos y el cálculo de la planificabilidad borrosa a partir de la comparación de los tiempos de finalización con los correspondientes plazos, presentados en el apartado 3.4.

La utilidad del modelado de un concepto mediante un conjunto borroso depende de la definición adecuada de la función de pertenencia. Los métodos propuestos en la literatura para obtener la función de pertenencia son principalmente empíricos y suelen consistir en experimentos realizados sobre una población con el fin de obtener las percepciones subjetivas de los grados de pertenencia a un concepto. Se puede hacer preguntando directamente a los sujetos los grados de pertenencia, pidiendo una

respuesta del tipo verdadero o falso sobre la pertenencia a un conjunto y analizando los datos estadísticos, o traduciendo los tiempos de respuesta de esta pregunta a grados de pertenencia. Una vez recogidos estos datos, existen diversas formas de derivar el conjunto borroso. En el apartado 3.3 se muestran, como ejemplo, dos formas de obtención de conjuntos borrosos.

A continuación, en el apartado 3.4 se calculan los tiempos de finalización borrosos en sistemas planificados mediante RMS y mediante DMS, aplicando el principio de extensión de la teoría de la posibilidad y aplicando la representación gráfica de los cuerpos de evidencia conjuntos propuesta en esta tesis (ver anexo A).

El método gráfico propuesto en el anexo A se utiliza como herramienta de ayuda en el análisis de estos sistemas, más que como método sistemático de cálculo. Por otro lado, la aplicación directa del principio de extensión puede ser inviable debido a su coste computacional, por lo que en el apartado 3.5 se propone aplicar aritmética de intervalos para acelerar el cálculo. Los tiempos de finalización borrosos obtenidos de esta forma son aproximaciones válidas de los proporcionados por el principio de extensión ya que la planificabilidad borrosa resultante coincide.

### **3.1. Conceptos básicos de la Teoría de la Posibilidad**

Uno de los paradigmas en la ciencia y en las matemáticas que ha cambiado en el último siglo es el de la incertidumbre. En el campo de la ciencia se ha pasado de forma gradual del punto de vista tradicional en el que la incertidumbre en su sentido más amplio (imprecisión, no especificidad, vaguedad, inconsistencia...) era indeseable y había que evitarla, a un punto de vista tolerante que reconoce que la incertidumbre se debe modelar para tener una representación adecuada de la realidad y de la que se puede obtener gran utilidad ([Klir, Yuan, 1995]).

La incertidumbre es una característica del conocimiento humano siempre presente en mayor o menor medida. En su acepción más amplia incluye imprecisión, vaguedad e incertidumbre (entendida en su acepción más restringida) ([Dubois, Prade, 1988a]).

La verdad se entiende generalmente como la conformidad, compatibilidad o parecido entre una afirmación y lo que efectivamente se conoce de la realidad. La existencia de parecidos parciales genera la necesidad de manejar grados de verdad parciales, que se expresan normalmente con una cifra en el intervalo  $[0,1]$ . Nótese que tanto el conocimiento disponible como la afirmación que se contrasta pueden ser precisos, imprecisos, vagos y/o inciertos ([Villar, 1997]).

Los grados de verdad parciales dan lugar a la vaguedad de los conceptos. La vaguedad de una propiedad se debe a la posibilidad de que ésta pueda cumplirse de forma gradual, y está presente en la mayoría de los conceptos o propiedades en el lenguaje hablado. Por ejemplo en la afirmación “*ha llovido mucho*”, el concepto *mucha lluvia* tiene un amplio contenido de vaguedad, resulta imposible determinar con precisión cuando se considera que ha llovido mucho, y en general existirá una zona de transición progresiva entre los conceptos *poca lluvia* y *mucha lluvia*, que además dependerá de la zona geográfica considerada. Utilizar un límite rígido no permite describir con suficiente fidelidad estos conceptos, ya que en la vida real las transiciones no suelen ser bruscas sino graduales, a no ser que se dé una catástrofe.

La incertidumbre de una afirmación permite modelar el desconocimiento que se tiene sobre un grado de verdad. Independientemente del grado de verdad de una afirmación, su incertidumbre será tanto menor cuanto mejor se conozca su grado de verdad y viceversa. Así cuando se conoce con total exactitud podemos decir que se tiene una total certidumbre, o que la certidumbre es 1, mientras que un total desconocimiento lleva a una total incertidumbre o certidumbre 0. Existen varias formas de expresar la incertidumbre en el lenguaje hablado. Una muy común es hablar de probabilidades como por ejemplo “*es probable que mañana llueva*”. Otras formas alternativas pueden apreciarse en las afirmaciones “*es posible que mañana llueva*”, y “*es cierto que mañana lloverá*”, y en todas ellas hay diferencias semánticas que se caracterizan por distintas formas de entender la incertidumbre. Mientras que la primera refleja una creencia basada en que cuando el cielo está oscuro es frecuente que llueva, la segunda puede reflejar falta de información (no se conoce el aspecto del cielo) que impide afirmar o contradecir la predicción de lluvia, y la última refleja un total convencimiento o seguridad en lo que se afirma. Lo probable se asocia con lo frecuente, lo posible con lo factible, viable o practicable, y lo necesario con lo fiable, seguro o conocido ([Villar, 1997]).

La imprecisión por último es una característica de la información que hace que ésta no sea suficientemente específica. Por ejemplo, afirmar “*la presión atmosférica está en [0 atm, 10 atm]*” es una afirmación con total certidumbre, pero tan imprecisa que carece de información. Una información más precisa es por ejemplo “*la presión atmosférica está en [0 atm, 1.1 atm]*”. En este caso la información, aunque sigue siendo imprecisa lo es menos, y tiene mayor contenido informativo porque es menos probable.

La mayoría de las técnicas de representación de incertidumbre están enfocadas a su representación en el sentido más restringido, es decir a evaluar en qué medida se conoce el grado de verdad de una proposición. Este es el enfoque que se adopta tanto desde el punto de vista probabilístico, como desde el punto de vista de la teoría de la evidencia.

La teoría de la **probabilidad** tiene la gran desventaja de que no se puede utilizar para representar la ausencia total de información, o desconocimiento. En efecto, el mayor desconocimiento que puede ser modelado sobre la ocurrencia de un suceso de un conjunto de posibles sucesos viene dado por la máxima incertidumbre probabilística que se materializa en la equiprobabilidad, es decir la probabilidad de un suceso y del suceso negado es 0.5 (equiprobables), lo cual tiene un gran contenido informativo. Por otro lado, en entornos reales donde la información suele ser incompleta, resulta difícil estimar las probabilidades de los sucesos, por lo que se recurre a las llamadas probabilidades subjetivas, debiendo cuidarse la posible aparición de inconsistencias. Por último, aunque tiene la ventaja de tener un sólido fundamento teórico, requiere demasiadas hipótesis para que muchas de las herramientas tengan validez y sean operativas (como por ejemplo la hipótesis de independencia) lo que, o bien dificulta considerablemente su aplicación práctica, o bien reduce sustancialmente su validez teórica cuando se aplica ([Oberuggenberger, 2002]). Los enfoques basados en teoría de la evidencia sí permiten modelar el desconocimiento sobre la ocurrencia de un suceso, presentan un marco teórico muy sólido, y además incluyen (bajo ciertas hipótesis) a las teorías de la probabilidad y de la posibilidad como casos particulares. Todos estos enfoques sin embargo sólo permiten modelar la incertidumbre pero no la vaguedad de las proposiciones, es decir la existencia de verdades parciales conocidas con precisión.

En cambio, la teoría de **conjuntos borrosos** ([Zadeh, 1965]), permite manejar en único marco teórico la incertidumbre y la vaguedad, mediante la utilización de conjuntos borrosos y su combinación con la teoría de la posibilidad, y permite modelar el desconocimiento total de forma similar a como se hace en teoría de la evidencia.

La **teoría de la posibilidad** es, por un lado un caso particular de la teoría de la evidencia y por otro lado es una aplicación importante de la teoría de conjuntos borrosos, donde el valor que toma una variable se modela mediante un conjunto borroso que permite expresar incertidumbre.

En este apartado se presentan algunas definiciones básicas de conjuntos borrosos, de teoría de la evidencia y de sus conexiones con la teoría de la posibilidad.

### 3.1.1. Conjuntos borrosos

Un paso importante en la evolución del concepto de incertidumbre fue la publicación de un artículo de Zadeh ([Zadeh, 1965]), donde el autor presentó la teoría de conjuntos borrosos, esto es conjuntos con límites imprecisos. La pertenencia a un conjunto deja de ser una afirmación o una negación para ser una cuestión de grado.

La proposición “ $x$  es miembro de  $A$ ” no es necesariamente cierta o falsa, sino que puede ser cierta en un determinado grado, el grado en que  $x$  es miembro de  $A$ . Lo más común es que los grados de pertenencia a conjuntos borrosos (grados de verdad de proposiciones) se encuentren en el intervalo unidad  $[0,1]$ . Los extremos 0 y 1 representan respectivamente la negación total y la afirmación total de la pertenencia a un conjunto borroso dado, así como la falsedad y verdad de la proposición asociada.

La capacidad de los conjuntos borrosos de representar grados entre la pertenencia y la no pertenencia resulta de gran utilidad en la representación de incertidumbre en las medidas, y de conceptos vagos expresados en lenguaje natural. Por ejemplo, en lugar de describir el tiempo climatológico como un tanto por ciento en que el cielo está cubierto por nubes, se puede decir que es un día *soleado*. Se puede aceptar que un cielo cubierto al 10 o al 20% es soleado, pero no hay un límite claro a partir del cual pasa de ser soleado a no ser soleado. Se puede introducir vaguedad en el término *soleado* permitiendo una transición gradual de los tantos por ciento de cobertura que se consideran soleados a los que no los son. El conjunto borroso es una generalización del conjunto clásico o “crisp”.

La función característica  $\mu_A$  de un conjunto clásico  $A$  asigna el valor 0 o 1 a cada punto del conjunto universal  $X$  ( $\mu_A : X \rightarrow \{0,1\}$ ), discriminando de esta forma qué puntos son del conjunto clásico que se define y qué puntos no lo son. La función característica  $\mu_A$  del conjunto clásico  $A$  se define como:

$$\mu_A(x) = \begin{cases} 1 & \text{ssi } x \in A \\ 0 & \text{ssi } x \notin A \end{cases}$$

Esta función se puede generalizar de manera que los valores asignados a los elementos del conjunto universal estén dentro del rango  $[0,1]$  indicando el grado de pertenencia de esos elementos al conjunto en cuestión. Esa función se denomina *función de pertenencia* y el conjunto definido por esa función de pertenencia es un conjunto borroso (ver Figura 3.1). En efecto, la función de pertenencia de un conjunto borroso  $A$  asigna un valor del intervalo  $[0,1]$  a cada punto del conjunto universal  $X$ ,  $\mu_A : X \rightarrow [0,1]$ , donde  $[0,1]$  es el intervalo de números reales desde 0 hasta 1, ambos incluidos.

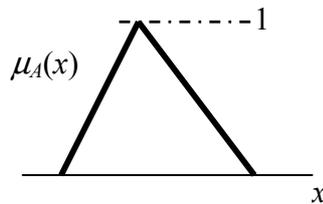


Figura 3.1: Función de pertenencia de un conjunto borroso  $A$

Es importante destacar que los grados de pertenencia no son probabilidades. Una diferencia que se puede extraer de forma inmediata es que la suma de probabilidades en un conjunto universal finito debe ser 1, mientras que no existe un requisito semejante para los grados de pertenencia. En el anexo C se analiza esta cuestión más en detalle.

A continuación se definen algunos conceptos básicos de conjuntos borrosos que serán de utilidad en el desarrollo de esta tesis.

El *sopORTE*  $]A[$  de un conjunto borroso  $A$  en el conjunto universal  $X$  es el conjunto “crisp” o clásico que contiene todos los elementos de  $X$  cuyo grado de pertenencia en  $A$  es distinto de cero, esto es:

$$]A[ = \{ x \in X \mid \mu_A(x) > 0 \}$$

La *altura* de un conjunto borroso es el mayor grado de pertenencia que tiene ese conjunto. Se dice que un conjunto borroso es normalizado si al menos uno de sus elementos tiene grado de pertenencia 1.

El *núcleo* de un conjunto borroso  $A$  en el conjunto universal  $X$  es el conjunto clásico que contiene todos los elementos de  $X$  cuyo grado de pertenencia en  $A$  es la altura del mismo.

Un  $\alpha$ -*corte* de un conjunto borroso  $A$  es un conjunto clásico  $A_\alpha$  que contiene todos los elementos del conjunto universal  $X$  con un grado de pertenencia en  $A$  mayor o igual que el valor especificado en  $\alpha$ :

$$A_\alpha = \{ x \in X \mid \mu_A(x) \geq \alpha \}.$$

Se puede observar que el conjunto de todos los  $\alpha$ -cortes de cualquier conjunto borroso en  $X$  es una familia de subconjuntos anidados de  $X$ . Se dice que los conjuntos clásicos  $A_1, A_2, \dots, A_n$  están anidados cuando verifican  $A_1 \subseteq A_2 \subseteq \dots \subseteq A_n$ .

La función de pertenencia  $\mu_A(a): X \rightarrow [0,1]$  de un conjunto borroso  $A$  se puede expresar también en función de sus  $\alpha$ -cortes, donde el supremo se puede interpretar como una unión ([Dubois, Prade, 1989a], [Dubois, Prade, 1986b]), como:

$$\mu_A(x) = \sup \{ \alpha \in (0,1] \mid x \in A_\alpha \}$$

Un conjunto clásico o “crisp”  $A$  en  $\mathbb{R}^n$  es compacto si, para cualquier par de puntos  $r$  y  $s$  en  $A$ , todos los puntos del segmento que une  $r$  y  $s$  pertenecen también al conjunto  $A$ . Un conjunto borroso es *convexo* si y sólo si cada uno de sus  $\alpha$ -cortes es un

conjunto clásico compacto. En la Figura 3.2 se muestra un ejemplo de conjunto borroso no convexo.

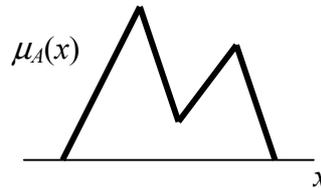


Figura 3.2: Conjunto borroso no convexo

Un *número borroso* es un conjunto borroso normalizado, convexo, definido en  $\mathbb{R}$ , cuya función de pertenencia es continua a trozos.

Dos conjuntos borrosos  $A$  y  $B$  se dice que son *iguales* si  $\mu_A(x) = \mu_B(x)$  para todos los elementos  $x \in X$ , y se expresa como  $A = B$ . Por otro lado, dos conjuntos borrosos  $A$  y  $B$  se dice que no son iguales si se verifica  $\mu_A(x) \neq \mu_B(x)$  para al menos un elemento  $x \in X$  y se expresa como  $A \neq B$ .

Se dice que un número borroso  $A$  es *fuertemente mayor* que un número borroso  $B$  ( $A >_s B$ ) cuando  $A \neq B$  y los lados cumplen:

$$f_A(x) \leq f_B(x)$$

$$g_A(x) \geq g_B(x)$$

siendo  $f_A$  la función que define el lado izquierdo de un número borroso  $A$  y  $g_A$  la función que define el lado derecho de  $A$  ([Heilpern, 1997]).

El lado izquierdo y el lado derecho de un número borroso  $A$  se definen como:

$$f_A(x) = \mu_A(x) \quad \forall x \leq x_i \quad \text{donde } x_i = \min(x) \text{ con } \mu_A(x)=1$$

$$g_A(x) = \mu_A(x) \quad \forall x \geq x_s \quad \text{donde } x_s = \max(x) \text{ con } \mu_A(x)=1$$

En la Figura 3.3 se muestran dos ejemplos:  $A$  es fuertemente mayor que  $B$ : ( $A >_s B$ ) y  $A'$  no fuertemente mayor que  $B'$  ( $A' \not>_s B'$ )

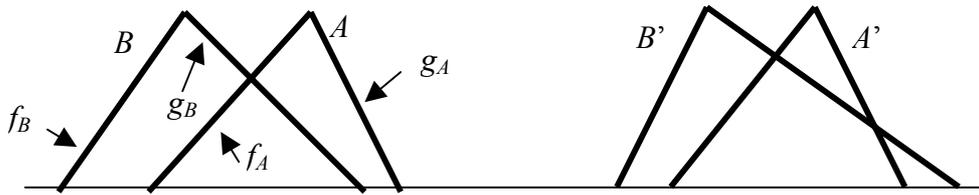


Figura 3.3: Ejemplo de  $(A >_s B)$  y  $(A' >_s B')$

**Teoría de la posibilidad.** Tal y como ya se ha dicho, la teoría de la posibilidad es una aplicación importante de la teoría de conjuntos borrosos, donde el valor que toma una variable se modela mediante un conjunto borroso.

Sea una variable  $x$  definida sobre un universo de discurso  $X$ . Una distribución de posibilidad  $\pi$  asociada a la variable  $x$  es una función  $\pi: X \rightarrow [0,1]$ , que expresa el conocimiento que se tiene de lo factible, admisible, creíble, posible, etc., que es que la variable  $x$  tome un determinado valor de  $X$  ([Dubois, Prade, 1993a]). Por tanto para cada valor  $x_0$  de  $X$  el grado de posibilidad de que  $x$  tome dicho valor  $x_0$  viene dado por  $\pi(x_0)$ .

Cuando el conocimiento que se tiene sobre los valores que la variable  $x$  puede tomar se expresa mediante un predicado vago del tipo “ $x$  es  $A$ ” donde  $A$  es un conjunto borroso, entonces en este contexto la distribución de posibilidad coincide con  $A$ , de forma que la posibilidad de que  $x$  tome el valor  $x_0$  viene dada por la compatibilidad del valor  $x_0$  con el concepto expresado por  $A$  ([Zadeh, 1978]), es decir por la pertenencia de  $x_0$  a  $A$ :

$$\pi(x_0) = \mu_A(x_0)$$

Nótese que en lenguaje natural se usan constantemente predicados vagos expresados a través de propiedades que no admiten una transición brusca del cumplimiento al no cumplimiento de la misma. Por ello una forma natural de modelar estos predicados es mediante el uso de conjuntos borrosos. Cuando estos predicados se utilizan para describir el valor que toma una variable entonces se interpretan como distribuciones de posibilidad.

Dada una distribución de posibilidad  $\pi(x)$  se puede calcular la medida de posibilidad de un conjunto clásico  $B$  como:

$$\Pi(B) = \sup_{x \in B} \pi(x), \text{ para cada } B \in \mathcal{P}(X)$$

donde  $\mathcal{P}(X)$  es el conjunto potencia de subconjuntos clásicos de  $X$ .

### Operaciones con conjuntos borrosos

Se puede encontrar una revisión exhaustiva de las operaciones con conjuntos borrosos en [Villar, 1997].

La *negación* de un conjunto borroso  $A$  es otro conjunto borroso  $\neg A$  tal que  $\neg A = \Gamma(A)$ , donde  $\Gamma$  es un operador de negación. Se dice que una función  $\Gamma(x)$  es un operador de negación, o simplemente una negación cuando cumpla:

$$\Gamma(0)=1 \text{ y } \Gamma(1)=0$$

$$\Gamma(x) \text{ no creciente con } x$$

La negación más comúnmente utilizada es  $\Gamma(x)=1-x$ .

### Intersección y unión de conjuntos borrosos

Otras dos operaciones básicas de conjuntos borrosos son la intersección y la unión. La intersección y la unión de dos conjuntos borrosos  $A$  y  $B$  son dos conjuntos borrosos  $A \cap B$  y  $A \cup B$ , definidos en  $X$ :

$$(A \cap B)(x) = AC(A(x), B(x))$$

$$(A \cup B)(x) = AD(A(x), B(x))$$

donde  $AC$  es un operador de agregación conjuntivo y  $AD$  es un operador de agregación disyuntivo.

Se dice que una función  $AC(x,y): [0,1] \times [0,1] \rightarrow [0,1]$  es un operador de agregación conjuntivo cuando cumpla las siguientes propiedades:

$$AC(0,1)=AC(1,0)=AC(0,0)=0$$

$$AC(1,1)=1$$

$$AC(x,y) \text{ no decreciente ni con } x \text{ ni con } y$$

Del mismo modo se dice que una función  $AD(x,y): [0,1] \times [0,1] \rightarrow [0,1]$  es un operador de agregación disyuntivo cuando verifique:

$$AD(0,1)=AD(1,0)=AD(1,1)=1$$

$$AD(0,0)=0$$

$$AD(x,y) \text{ no decreciente ni con } x \text{ ni con } y$$

Se dice que un operador conjuntivo  $AC(x,y)$  y uno disyuntivo  $AD(x,y)$  son duales respecto a una negación  $\Gamma$  cuando se verifiquen las propiedades:

$$AC(x,y) = \Gamma \{AD(\Gamma(x), \Gamma(y))\}$$

$$AD(x,y) = \Gamma \{AC(\Gamma(x), \Gamma(y))\}$$

Entre los operadores conjuntivos y disyuntivos destacan por su extendido uso y sus buenas propiedades matemáticas las conocidas t-normas y t-conormas.

Una t-norma  $T(x,y): [0,1] \times [0,1] \rightarrow [0,1]$  es un operador de agregación conjuntivo que cumple las propiedades de conmutatividad, asociatividad y las condiciones de contorno  $T(x,1)=x$ ,  $T(x,0)=0$  ([Alsina et al., 1983], [Alsina, Trillas, 1984], [Dubois, Prade, 1985]).

Por otro lado, una t-conorma  $S(x,y): [0,1] \times [0,1] \rightarrow [0,1]$  es un operador de agregación disyuntivo que cumple las propiedades de conmutatividad, asociatividad y además  $S(x,1)=1$ ,  $S(x,0)=x$ .

A partir de una t-norma cualquiera y de una negación se puede obtener una t-conorma. La mayor t-norma, o lo que es lo mismo, la menos específica es *min* (operación mínimo) y la menor t-conorma es *max* (operación máximo). Además de éstas, otras t-normas y t-conormas ampliamente conocidas y utilizadas son los operadores probabilísticos y los operadores de Lukasiewicz.

La t-norma probabilística o producto es  $T(x,y)=x \cdot y$ , y la t-conorma probabilística o suma probabilística se calcula como  $S(x,y)=x+y-x \cdot y$

La t-norma de Lukasiewicz se define como  $T(x,y)=\max(0, x+y-1)$  y la t-conorma es  $S(x,y)=\min(1, x+y)$ .

### *Principio de extensión*

Las operaciones vistas hasta ahora eran generalizaciones de las operaciones clásicas entre conjuntos. El principio de extensión es por el contrario un procedimiento general que permite extender funciones  $f: X_1 \times X_2 \times \dots \times X_n \rightarrow X_{n+1}$  a funciones con argumentos borrosos  $F: \mathfrak{S}(X_1) \times \mathfrak{S}(X_2) \times \dots \times \mathfrak{S}(X_n) \rightarrow \mathfrak{S}(X_{n+1})$ , donde  $\mathfrak{S}(X_1)$  representa el conjunto de todos los conjuntos borrosos que pueden definirse en  $X$ .

El principio de extensión se enuncia del siguiente modo. Dada una función  $f: X_1 \times X_2 \times \dots \times X_n \rightarrow X_{n+1}$ , la extensión  $F$  de  $f$  viene dada por:

$$F: \mathfrak{S}(X_1) \times \mathfrak{S}(X_2) \times \dots \times \mathfrak{S}(X_n) \rightarrow \mathfrak{S}(X_{n+1})$$

$$\mu_Y(y) = F(A, B, \dots, M)(y) = \sup \{ T(\mu_A(a), \mu_B(b), \dots, \mu_M(m)) \mid (a, b, \dots, m) \in X_1 \times X_2 \times \dots \times X_n \text{ e } y=f(a, b, \dots, m) \}$$

$T(\mu_A(a), \mu_B(b), \dots, \mu_M(m))$  representa el grado de pertenencia de  $(a, b, \dots, m)$  a  $(A, B, \dots, M)$  por lo que el principio de extensión puede entenderse como ([Villar, 1997]) la expresión matemática de “*existe al menos un conjunto de valores  $(a, b, \dots, m)$  de  $X_1 \times X_2 \times \dots \times X_n$  con grado de pertenencia conjunto  $T(\mu_A(a), \mu_B(b), \dots, \mu_M(m))$  a  $(A, B, \dots, M)$ , tal que  $y$  es función de  $(a, b, \dots, m)$  según  $f$ , es decir  $y=f(a, b, \dots, m)$ ”.*

Dicho de otro modo, si la imagen  $y$  por la función  $f$  tiene más de un antecedente  $(a, b, \dots, m)$ , entonces el máximo grado de pertenencia de esos antecedentes al antecedente borroso  $(A, B, \dots, M)$  es el grado de pertenencia de  $y$  a la imagen borrosa de  $(A, B, \dots, M)$ , que denominamos  $Y$ , por la función  $F$ . El grado de pertenencia de cada tupla  $(a, b, \dots, m)$  a  $Y$  se obtiene como una conjunción de los grados de pertenencia individuales de cada  $a, b, \dots, m$  al correspondiente conjunto borroso  $A, B, \dots, M$  ([Villar, 1997]).

La forma más común de aplicar el principio de extensión es mediante la t-norma *min* ([Lootsma, 1997]), que se aplica cuando no se tiene información adicional acerca de la relación entre los conjuntos borrosos a combinar, por ser la menos específica (principio de mínima especificidad, [Dubois, Prade, 1991a]).

$$\mu_Y(y) = \sup_{y=f(a,b,\dots,m)} \min \{ \mu_A(a), \mu_B(b), \dots, \mu_M(m) \}$$

### Aritmética borrosa / aritmética de intervalos

La aritmética borrosa es tanto una generalización de la aritmética de números reales como de la aritmética de intervalos, y permite extender, mediante el principio de extensión, las operaciones ordinarias entre números reales a los números borrosos. Por ejemplo, las cuatro operaciones típicas de suma, resta, producto y cociente pueden definirse del siguiente modo:

Suma:  $\mu_{A+B}(c) = \mu_C(c) = \sup_{c=a+b} T(\mu_A(a), \mu_B(b))$

Resta:  $\mu_{A-B}(c) = \mu_C(c) = \sup_{c=a-b} T(\mu_A(a), \mu_B(b))$

Producto:  $\mu_{A \cdot B}(c) = \mu_C(c) = \sup_{c=a \cdot b} T(\mu_A(a), \mu_B(b))$

Cociente:  $\mu_{A \div B}(c) = \mu_C(c) = \sup_{c=a \div b} T(\mu_A(a), \mu_B(b))$

donde  $A$ ,  $B$  y  $C$  son conjuntos borrosos definidos en  $\mathbb{R}$  y  $a$ ,  $b$  y  $c$  pertenecen a  $\mathbb{R}$ .

Cuando la t-norma empleada para extender funciones aritméticas a argumentos borrosos es *min*, esas operaciones se pueden expresar de forma más sencilla [Lootsma, 1997]. Supongamos que se tienen dos números borrosos triangulares  $A$  y  $B$  definidos por sus vértices como  $A=(a_l, a_m, a_u)$  y  $B=(b_l, b_m, b_u)$ , tal y como se muestra en la Figura 3.4. La función de pertenencia se define por tramos y por ejemplo para el conjunto  $A$  viene dada por:

$$\mu_A(x) = \begin{cases} \frac{x - a_l}{a_m - a_l} & \text{si } a_l \leq x \leq a_m \\ \frac{x - a_u}{a_m - a_u} & \text{si } a_m \leq x \leq a_u \end{cases}$$

La función de pertenencia de  $B$  se define de forma similar.

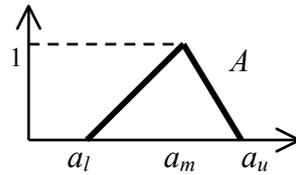


Figura 3.4: Número borroso triangular definido por sus vértices

La suma de números borrosos triangulares conserva la forma y es un número borroso triangular  $R$  de vértices  $R=(a_l + b_l, a_m + b_m, a_u + b_u)$ . Esto se cumple porque el  $\alpha$ -corte de  $R$  es igual a la suma de los  $\alpha$ -cortes de  $A$  y  $B$ , con lo que expresando la operación en función de los  $\alpha$ -cortes se tiene que, siendo el  $\alpha$ -corte de  $A$   $[a_l + \alpha(a_m - a_l), a_u - \alpha(a_u - a_m)]$  y el de  $B$   $[b_l + \alpha(b_m - b_l), b_u - \alpha(b_u - b_m)]$ , el  $\alpha$ -corte de  $R$  se puede calcular como  $[a_l + \alpha(a_m - a_l) + b_l + \alpha(b_m - b_l), a_u - \alpha(a_u - a_m) + b_u - \alpha(b_u - b_m)]$

A partir de la expresión en  $\alpha$ -cortes de  $R$  se puede obtener su función de pertenencia como:

$$\mu_R(x) = \begin{cases} \frac{x - (a_l + b_l)}{(a_m + b_m) - (a_l + b_l)} & \text{si } r_l \leq x \leq r_m \\ \frac{x - (a_u + b_u)}{(a_m + b_m) - (a_u + b_u)} & \text{si } r_m \leq x \leq r_u \end{cases}$$

donde  $r_l = a_l + b_l$ ,  $r_m = a_m + b_m$ , y  $r_u = a_u + b_u$

De forma similar se pueden calcular otras operaciones de aritmética borrosa como son la diferencia, que también conserva la forma de los números borrosos de partida, la multiplicación y la división. Sin embargo éstas últimas no conservan la forma de los números borrosos.

En efecto con la multiplicación, si  $a_l$  y  $b_l$  son mayores que cero, el  $\alpha$ -corte del producto  $R$  es el intervalo:

$$[ (a_l + \alpha(a_m - a_l)) \cdot (b_l + \alpha(b_m - b_l)), (a_u - \alpha(a_u - a_m)) \cdot (b_u - \alpha(b_u - b_m)) ]$$

Si se aproxima por un número triangular  $R = (r_l, r_m, r_u) = (a_l \cdot b_l, a_m \cdot b_m, a_u \cdot b_u)$ , el  $\alpha$ -corte de esa aproximación es  $[ (a_l \cdot b_l + \alpha(a_m \cdot b_m - a_l \cdot b_l)), (a_u \cdot b_u - \alpha(a_u \cdot b_u - a_m \cdot b_m)) ]$  y la función de pertenencia de la aproximación es:

$$\mu_R(x) = \begin{cases} \frac{x - (a_l \cdot b_l)}{(a_m \cdot b_m) - (a_l \cdot b_l)} & \text{si } r_l \leq x \leq r_m \\ \frac{x - (a_u \cdot b_u)}{(a_m \cdot b_m) - (a_u \cdot b_u)} & \text{si } r_m \leq x \leq r_u \end{cases}$$

Estas operaciones de aritmética borrosa se pueden expresar en forma de aritmética de intervalos, precisamente como operaciones entre  $\alpha$ -cortes.

Así, en la operación suma el  $\alpha$ -corte de  $R$  es  $[ a_l + \alpha(a_m - a_l) + b_l + \alpha(b_m - b_l), a_u - \alpha(a_u - a_m) + b_u - \alpha(b_u - b_m) ]$ , donde el límite inferior de dicho intervalo es la suma de los límites inferiores de los  $\alpha$ -cortes de  $A$  y  $B$ , y el límite superior de dicho intervalo es la suma de los límites superiores de los  $\alpha$ -cortes de  $A$  y  $B$ . Es decir,

$$[ \underline{r}(\alpha), \overline{r}(\alpha) ] = [ \underline{a}(\alpha) + \underline{b}(\alpha), \overline{a}(\alpha) + \overline{b}(\alpha) ]$$

donde  $\underline{r}(\alpha)$  es el límite inferior del  $\alpha$ -corte de  $R$  y  $\overline{r}(\alpha)$  es el límite superior del  $\alpha$ -corte de  $R$ , y  $\underline{a}(\alpha) = a_l + \alpha(a_m - a_l)$ ,  $\underline{b}(\alpha) = b_l + \alpha(b_m - b_l)$ ,  $\overline{a}(\alpha) = a_u - \alpha(a_u - a_m)$  y  $\overline{b}(\alpha) = b_u - \alpha(b_u - b_m)$

### 3.1.2. Teoría de la Evidencia

En la base de la mayoría de las teorías de representación de la incertidumbre están las medidas borrosas, también llamadas medidas de incertidumbre ([Klir, Folger, 1988], [Alvarez, Castillo, 1994]). Una medida borrosa asigna un valor a cada conjunto clásico del universo de discurso indicando el grado de evidencia o creencia de que un elemento particular pertenece al conjunto.

Por ejemplo, sea una persona de la que no se puede decir con certidumbre si su edad se encuentra en la veintena, en la treintena, en la cuarentena o en la cincuenta. Los conjuntos citados son conjuntos clásicos y el conjunto clásico con el mayor valor asignado en esta medida borrosa es nuestra mejor opción para estimar la edad de esa persona. El siguiente mayor valor proporciona la siguiente opción mejor, y así sucesivamente. Si se tiene certidumbre absoluta se asignaría 1 a un solo conjunto clásico, y 0 al resto.

Una medida borrosa  $g$  dado un universo de discurso  $X$  es una función que verifica las siguientes propiedades:

$$g: \mathcal{P}(X) \rightarrow [0,1]$$

$$g(\emptyset)=0$$

$$g(X)=1$$

$$\text{si } P_1 \subset P_2 \text{ entonces } g(P_1) \leq g(P_2) \quad \forall P_1, P_2 \in \mathcal{P}(X)$$

donde  $\mathcal{P}(X)$  es el conjunto potencia de subconjuntos clásicos de  $X$ .

Para garantizar la continuidad es necesario añadir una propiedad adicional de forma que si  $P_1 \subset P_2 \dots \subset P_i \subset \dots \subset P_n$  entonces se verifique:

$$\lim_{i \rightarrow \infty} g(P_i) = g(\lim P_i)$$

Algunos autores extienden la definición al conjunto potencia de subconjuntos borrosos en  $X$ .

Las medidas borrosas más comunes son las medidas de probabilidad, de plausibilidad y credibilidad, y de posibilidad y necesidad, todas ellas definidas dentro de la teoría de la evidencia.

En la Figura 3.5 se puede observar que las medidas de posibilidad son un caso particular de las medidas de plausibilidad, las medidas de necesidad son un caso

particular de las de credibilidad y las medidas de probabilidad son un caso particular tanto de las medidas de plausibilidad como de las de credibilidad.

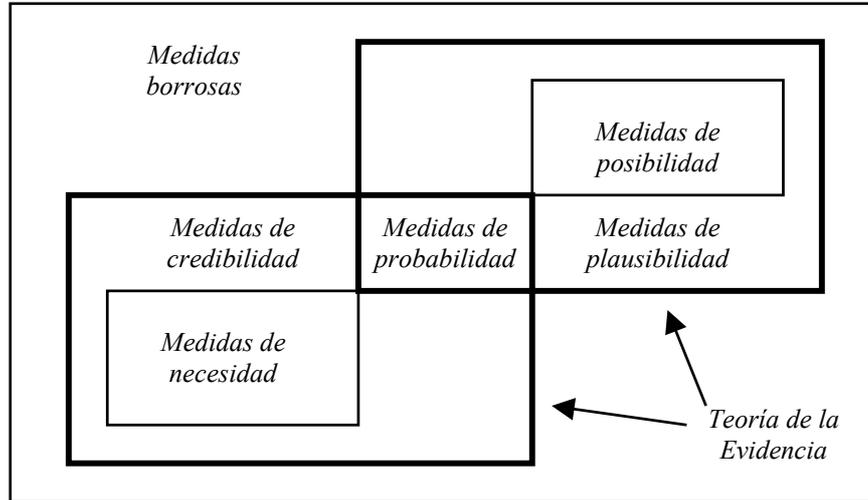


Figura 3.5: Principales tipos de medidas borrosas

Las medidas de plausibilidad  $Pl$  y credibilidad (“belief”)  $Bel$  son medidas borrosas definidas como ([Shafer, 1987], [Klir, Folger, 1988]):

$$Pl: \mathcal{P}(X) \rightarrow [0,1]$$

$$Pl(A_1 \cap A_2 \cap \dots \cap A_n) \leq \sum_i Pl(A_i) - \sum_{i < j} Pl(A_i \cup A_j) + \dots + (-1)^{n+1} Pl(A_1 \cup A_2 \cup \dots \cup A_n)$$

$$Bel: \mathcal{P}(X) \rightarrow [0,1]$$

$$Bel(A_1 \cup A_2 \cup \dots \cup A_n) \geq \sum_i Bel(A_i) - \sum_{i < j} Bel(A_i \cap A_j) + \dots + (-1)^{n+1} Bel(A_1 \cap A_2 \cap \dots \cap A_n)$$

donde  $\mathcal{P}(X)$  es el conjunto potencia de subconjuntos clásicos de  $X$ .

A cada medida de credibilidad  $Bel(A)$  se le puede asociar una medida de plausibilidad  $Pl(A)$  definida como:

$$Pl(A) = 1 - Bel(\neg A)$$

y se dice que son duales.

Se denomina cuerpo de evidencia al par  $(F, m)$ , donde  $F$  es el conjunto de elementos focales y  $m$  es la asignación básica de probabilidad asociada. Los elementos focales son conjuntos clásicos  $A_i$  que pertenecen a  $\mathcal{P}(X)$  tales que  $m(A_i) > 0$ . La asignación básica es una función  $m: \mathcal{P}(X) \rightarrow [0, 1]$ , que verifica  $m(\emptyset) = 0$  y además:

$$(3.1) \quad \sum_{A_i \in \mathcal{P}(X)} m(A_i) = 1$$

La asignación básica de probabilidad es una función de densidad de probabilidad en  $\mathcal{P}(X)$ , si  $X$  es continuo. Esto permite fundamentar la teoría de la evidencia desde la teoría de la probabilidad, considerando conjuntos en lugar de elementos, lo que confiere un sólido fundamento teórico.

Dado un cuerpo de evidencia  $(F, m)$ , se pueden definir las medidas de plausibilidad y credibilidad también como ([Klir, Folger, 1988], [Smets, 1988]):

$$(3.2) \quad Pl(B) = \sum_{A_i \cap B \neq \emptyset} m(A_i)$$

$$Bel(B) = \sum_{A_i \subseteq B} m(A_i)$$

La medida  $m(A_i)$  se interpreta como la creencia que se tiene únicamente en el conjunto  $A_i$ , y es su probabilidad en relación con todos los demás conjuntos de  $\mathcal{P}(X)$ ,  $Bel(B)$  se interpreta como la creencia total que se tiene en  $B$  y en cualquier subconjunto de  $B$ , y por último  $Pl(B)$  se interpreta como la creencia en  $B$ , en cualquier subconjunto de  $B$  y en cualquier otro conjunto que tenga una intersección no nula con  $B$ . Esto hace que siempre se verifique:

$$Pl(B) \geq Bel(B)$$

En la teoría de la evidencia la ignorancia o total desconocimiento sobre la verdad de una afirmación se expresa mediante una asignación básica de probabilidad dada por  $m(X) = 1$  y  $m(B) = 0$  para cualquier conjunto  $B$  distinto de  $X$ . La plausibilidad y credibilidad para cualquier conjunto  $B$  son en este caso  $Pl(B) = 1$  y  $Bel(B) = 0$ .

Cuando la función de credibilidad se define sólo sobre conjuntos unitarios, los elementos focales del cuerpo de evidencia cuya asignación básica es distinta de cero son conjuntos unitarios (átomos), y la medida que se obtiene es una medida de probabilidad  $p: X \rightarrow [0, 1]$ ,  $Bel = Pl = p$ .

Cuando los elementos focales están anidados, se dice que el cuerpo de evidencia es consonante, la medida de plausibilidad se denomina medida de posibilidad  $\Pi$ , y verifica:

$$Pl(A \cup B) = \max \{ Pl(A), Pl(B) \} \equiv \Pi(A \cup B) = \max \{ \Pi(A), \Pi(B) \} \quad \forall A, B \in \mathcal{P}(X)$$

Del mismo modo, la medida de credibilidad se denomina medida de necesidad  $N$ , y cumple:

$$Bel(A \cap B) = \min \{ Bel(A), Bel(B) \} \equiv N(A \cap B) = \min \{ N(A), N(B) \} \quad \forall A, B \in \mathcal{P}(X)$$

Dado un cuerpo de evidencia  $(F, m)$  se puede interpretar la función de contorno  $Pl(\{x\})$  como la función de pertenencia  $\mu(x)$  de un conjunto borroso ([Dubois, Prade, 1990]):

$$(3.3) \quad \mu(x) = Pl(\{x\}) = \sum_{A_i \cap \{x\} \neq \emptyset} m(A_i) = \sum_{x \in A_i} m(A_i)$$

Si el cuerpo de evidencia es consonante, entonces la función de distribución de plausibilidad  $Pl(\{x\})$  es una función de distribución de posibilidad  $\Pi(\{x\})$ , que denotaremos como  $\pi(x)$  tal que  $\pi : X \rightarrow [0, 1]$ . Cada medida de posibilidad  $\Pi$  de  $B$  dado  $\pi(x)$  se puede calcular a partir de la distribución de posibilidad como ([Dubois, Prade, 1992]):

$$\Pi(B) = \sup_{x \in B} \pi(x) \quad \forall B \in \mathcal{P}(X)$$

También se puede calcular la medida de necesidad  $N$  de  $B$  a partir de  $\pi(x)$  como:

$$N(B) = \inf_{x \in -B} (1 - \pi(x)) \quad \forall B \in \mathcal{P}(X)$$

Existe también una estrecha relación entre la representación en  $\alpha$ -cortes de un conjunto borroso y el cuerpo de evidencia consonante asociado al mismo. Tal y como se ha visto antes, la función de pertenencia de un conjunto borroso  $A$  se puede representar también en términos de sus  $\alpha$ -cortes. Siendo  $A_i$  el  $\alpha$ -corte de nivel  $\alpha^i$ , o  $\alpha^i$ -corte,  $A_i$  se puede interpretar como un elemento focal del cuerpo de evidencia consonante de  $A$   $(F_A, m_A)$  que subyace ([Dubois, Prade, 1986a], [Dubois, Prade, 1989a]), donde:

$$(3.4) \quad m_A(A_i) = \alpha^i - \alpha^{i+1}$$

cumpléndose  $\alpha^1 > \alpha^2 > \dots > \alpha^n$  con  $\alpha^{n+1} = 0$  y  $\alpha^1 = 1$

En este contexto, la función de pertenencia  $\mu(x)$  se podría utilizar para obtener un conjunto clásico  $A_i$  y realizar simulaciones numéricas mediante el método de Montecarlo ([Dubois, Prade, 1986a], [Chanas, Nowakowski, 1988]). Para ello, se genera un número aleatorio según una distribución uniforme entre 0 y 1. Ese valor se lleva a la función de pertenencia del conjunto borroso y se obtiene el  $\alpha$ -corte correspondiente, esto es, se obtiene un elemento focal del cuerpo de evidencia consonante, tal y como se indica en la Figura 3.6.

Los autores consideran que, dado un elemento focal, la probabilidad de cada elemento del intervalo es la misma, es decir, que se tiene una función de probabilidad uniformemente distribuida en el intervalo, de manera que generando un número aleatorio según dicha distribución obtienen un valor para realizar la simulación numérica.

En esta tesis se realizan simulaciones directamente con los intervalos obtenidos, mediante simulación por intervalos.

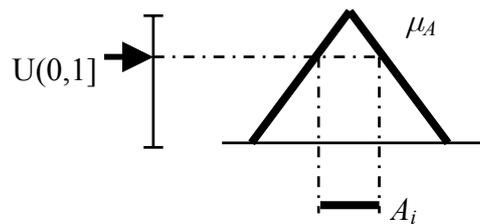


Figura 3.6: Obtención de un elemento focal  $A_i$  a partir de  $\mu_A$

Supongamos que se construye un conjunto borroso  $A$  a partir de una serie de observaciones imprecisas o intervalos  $A_i$ , que supondremos consonantes. Entonces, se puede obtener un cuerpo de evidencia  $(F_A, m_A)$  a partir de las observaciones, donde la asignación básica  $m_A(A_i)$  es la probabilidad del intervalo  $A_i$  ([Dubois, Prade, 1986a], [Klir, Folger, 1988], [Dubois, Prade, 1991b]). A cada intervalo  $A_i$  se asocia un valor  $\alpha_A^i$  que se interpreta como el nivel  $\alpha^i$  del  $\alpha$ -corte  $A_i$ , y que se puede calcular a partir del cuerpo de evidencia como:

$$\alpha_A^i = \sum_{j=n}^i m_A(A_j).$$

Tal y como se describe en el anexo A, cuando se obtienen simultáneamente dos observaciones imprecisas, esto es, dos intervalos  $A_i$  y  $B_j$ , se puede calcular una asignación básica conjunta como la distribución de probabilidad del par  $(A_i, B_j)$ . Los elementos focales  $C_i$  del cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$  son el producto

cartesiano de las observaciones  $A_i$  y  $B_j$ , y la asignación básica conjunta  $m_C(C_i)$  está definida en el producto Cartesiano  $X \times Y$ ,  $m_C(C_i) = m_{A \times B}(A_j \times B_k)$  tal que  $m_{A \times B}: \mathcal{P}(X \times Y) \rightarrow [0,1]$ . Los cuerpos de evidencia  $(F_A, m_A)$  y  $(F_B, m_B)$  son las proyecciones del cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$  en  $X$  e  $Y$  respectivamente, también denominados cuerpos de evidencia marginales ([Klir, Folger, 1988]).

La distribución de plausibilidad conjunta de cualquier par  $(a,b)$  se puede obtener aplicando la definición de medida de plausibilidad  $Pl(B) = \sum_{A_i \cap B \neq \emptyset} m(A_i)$  ([Wang et al., 1991]) como:

$$Pl(\{(a,b)\}) = \sum_{(a,b) \in C_i} m_{A \times B}(C_i)$$

Si el cuerpo de evidencia conjunto es consonante, esta distribución de plausibilidad se denomina distribución de posibilidad.

Suponiendo que los cuerpos de evidencia marginales son consonantes, los elementos focales  $A_j$  y  $B_k$  asociados a  $C_i$  tal que  $C_i = A_j \times B_k$  son los  $\alpha$ -cortes de los conjuntos borrosos  $A$  y  $B$  correspondientes a  $\alpha_A^j$  y  $\alpha_B^k$ . Por tanto, cada elemento focal conjunto  $C_i$  se puede caracterizar mediante una pareja  $(\alpha_A^j, \alpha_B^k)$  y la asignación básica conjunta se puede expresar en términos de correlación existente entre  $\alpha_A$  y  $\alpha_B$ , con lo que  $m_{A \times B}(\alpha_A, \alpha_B)$  está definida en  $[0,1]^2$ . Se puede interpretar como la probabilidad  $P(A_j \times B_k)$  del elemento focal conjunto o la probabilidad del par  $(\alpha_A^j, \alpha_B^k)$ .

Dado un cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$ , se puede extender una función de números reales  $r=f(a,b)$  de forma que la distribución de plausibilidad resultante de la extensión de  $f$  es:

$$Pl(r) = \sum_{\substack{(a_1, b_1) \in C_i \\ OR (a_2, b_2) \in C_i \\ \dots OR (a_n, b_n) \in C_i \\ \forall (a_j, b_j) \mid r=f(a_j, b_j)}} m_{A \times B}(C_i)$$

### 3.2. Planificabilidad borrosa

Una vez repasados los conceptos básicos de teoría de la posibilidad que se van a emplear en esta tesis, en este apartado se modela la incertidumbre existente en el

conocimiento de los parámetros temporales del sistema mediante distribuciones de posibilidad, con el fin de obtener una medida de planificabilidad del sistema.

Un sistema de tiempo real donde todos sus parámetros son números reales, se dice que es planificable si y sólo si se respetan todos los plazos. Para ello, para todas las tareas  $i=1..n\_tar$  y en todas las ejecuciones  $j$  se debe cumplir que el tiempo de finalización  $wc_{ij}$  de la tarea  $i$  en la ejecución  $j$  esté antes que el plazo absoluto correspondiente  $d_{ij}$ , donde  $d_{ij} = (j-1) \cdot t_i + d_i$  con  $j \in \mathbb{N}$ , esto es  $wc_{ij} \leq d_{ij}$

Nótese que, como ya se ha dicho en el capítulo 2, es suficiente con comprobar que se respetan los plazos en un hiperperiodo, por lo que los tiempos de finalización a calcular son los correspondientes a ejecuciones de las tareas dentro del hiperperiodo.

Cuando un sistema incluye tareas caracterizadas por parámetros borrosos interpretados como distribuciones de posibilidad tales como tiempos de ejecución borrosos (para modelar por ejemplo que “el tiempo de ejecución de una tarea es aproximadamente 7”), su planificabilidad se puede cuantificar en términos de grados de posibilidad y de necesidad. La posibilidad y necesidad de que el sistema sea planificable se puede calcular como el mínimo de todos los valores de posibilidad y necesidad de que los tiempos de finalización de las tareas estén antes del plazo correspondiente. Si los tiempos de ejecución son números borrosos  $P_i$ , los tiempos de finalización  $WC_{ij}$  también lo son, con lo que la posibilidad  $\Pi_{planif}$  y la necesidad  $N_{planif}$  de que el sistema sea planificable se pueden definir como:

$$(3.5) \quad \Pi_{planif} = \min(\Pi(WC_{ij} < d_{ij}))$$

$$N_{planif} = \min(N(WC_{ij} < d_{ij})) \quad \forall i \in \mathbb{N} \mid 1 \leq i \leq n\_tareas$$

Si los plazos son también números borrosos, entonces la comparación entre el tiempo de finalización y el plazo es una comparación entre conjuntos borrosos, y en (3.5)  $d_{ij}$  se puede sustituir por  $D_{ij}$  con  $D_{ij} = (j-1) \cdot t_i + D_i$ .

En esta tesis se extienden también análisis de planificabilidad clásicos basados en el cálculo de la utilización total del procesador (ver apartado 6.1) y el basado en el criterio de demanda del procesador (ver apartado 6.2), en lugar del basado en el cálculo del tiempo de finalización, en cuyo caso la planificabilidad del sistema se mide como la posibilidad y la necesidad de que se verifiquen las desigualdades correspondientes.

De acuerdo con Dubois y Prade ([Dubois, Prade, 1983], [Dubois, Prade, 1989b]) la posibilidad  $\Pi(A \leq B)$  y necesidad  $N(A \leq B)$  de que un conjunto borroso  $B$  con función de pertenencia  $\mu_B$  sea mayor o igual que otro conjunto borroso  $A$  con función de pertenencia  $\mu_A$ , se pueden definir como (ver anexo B):

$$\Pi(A \leq B) = \sup_{s \leq t} \min(\mu_A(s), \mu_B(t))$$

$$N(A \leq B) = 1 - \Pi(A > B) = 1 - \sup_{s > t} \min(\mu_A(s), \mu_B(t))$$

Si para cada valor de  $x \in \mathbb{R}$ ,  $A$  o bien  $B$  es continuo en  $x$ , entonces se cumple que  $\Pi(A \leq B) = \Pi(A < B)$  y  $N(A \leq B) = N(A < B)$  (ver anexo B). En este análisis suponemos que esta condición siempre se verifica.

En la Figura 3.7 se muestran dos ejemplos de comparación entre el tiempo de finalización borroso y el plazo borroso. En la Figura 3.7 a) la posibilidad de que la tarea  $i$  en su ejecución  $j$  termine antes de su plazo  $D_{ij}$  es 1, esto es  $\Pi(WC_{ij} \leq D_{ij}) = 1$ . La necesidad de que la tarea  $i$  en su ejecución  $j$  termine antes de su plazo se puede calcular como uno menos la posibilidad de que la tarea  $i$  en su ejecución  $j$  termine después de su plazo, esto es,  $N(WC_{ij} \leq D_{ij}) = 1 - \Pi(WC_{ij} > D_{ij}) = 1 - \alpha$ .

En la Figura 3.7 a), si  $\alpha$  es pequeño y la tarea  $i$  en su ejecución  $j$  es la que proporciona el mínimo valor de posibilidad y de necesidad de terminar antes de su plazo, entonces es razonable afirmar que es “casi seguro” que el sistema es planificable.

Por otro lado, en la Figura 3.7 b) se tiene que  $\Pi(WC_{ij} \leq D_{ij}) = \alpha$  y  $N(WC_{ij} \leq D_{ij}) = 1 - \Pi(WC_{ij} > D_{ij}) = 1 - 1 = 0$ . Si  $\alpha$  es pequeño y la tarea  $i$  en su ejecución  $j$  es la que proporciona el mínimo valor de posibilidad y de necesidad de terminar antes de su plazo, entonces es razonable afirmar que es “casi seguro” que el sistema no es planificable.

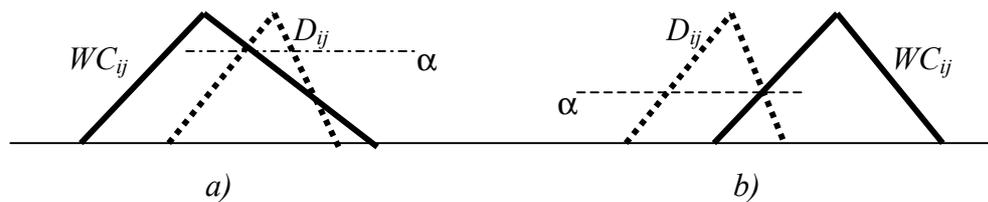


Figura 3.7: Dos ejemplos de comparación entre  $WC_{ij}$  and  $D_{ij}$

El análisis de planificabilidad borrosa que se propone sigue el esquema de la Figura 3.8. Cuando los parámetros temporales se modelan mediante conjuntos borrosos el siguiente paso es el cálculo de los tiempos de finalización borrosos. Como los tiempos de finalización de las tareas son función de los tiempos de ejecución, su

cálculo requiere la extensión o generalización de dicha función a tiempos de ejecución borrosos.

En teoría de la posibilidad la forma general de extender una función a parámetros borrosos es la aplicación del principio de extensión, tal y como ya se ha descrito en el apartado anterior. En el caso de que la t-norma utilizada sea el *min*, la extensión de la función  $wc_{ij}=f(p_h)$  con  $h=1..n\_tar$  a tiempos de ejecución borrosos  $P_h$ , da como resultado un conjunto borroso  $WC_{ij}$  cuya función de pertenencia se calcula como:

$$\mu_{WC_{ij}}(wc_{ij}) = \sup_{wc_{ij}=f(p_h)} \min\{\mu_{P_h}(p_h)\} \text{ con } h=1..n\_tar$$

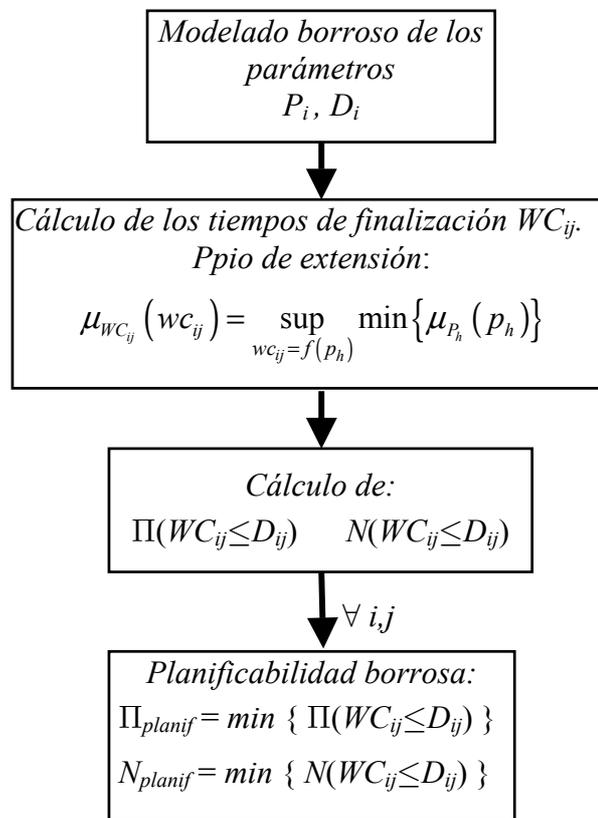


Figura 3.8: Análisis de planificabilidad borrosa

Una vez obtenido el tiempo de finalización borroso (ver Figura 3.8) se compara con el plazo correspondiente y se obtiene un valor de posibilidad y un valor de necesidad de que el tiempo de finalización esté antes del plazo. En general este plazo podría ser numérico (“crisp”) o borroso. Se repite el cálculo anterior para todas las tareas en

todas las ejecuciones dentro del hiperperiodo y el mínimo de todas las posibilidades (respectivamente necesidades) es la posibilidad (respectivamente necesidad) de que el sistema sea planificable.

Dado que cada conjunto borroso tiene un cuerpo de evidencia consonante asociado, otra forma de extender una función a parámetros borrosos es mediante el cuerpo de evidencia conjunto. Como ya se ha descrito, a partir de un cuerpo de evidencia conjunto se puede obtener una distribución conjunta de plausibilidad o de posibilidad. En el caso de tratarse de una función de sólo dos parámetros la extensión se puede obtener gráficamente con el método propuesto en el anexo A.

Si se expresa la asignación básica en función de  $\alpha$ , cuando dicha asignación básica conjunta  $m_{A \times B}(\alpha_A, \alpha_B)$  se encuentra concentrada en la diagonal principal de  $[0,1]^2$  (correlación positiva perfecta), el cuerpo de evidencia conjunto es consonante, por lo que la extensión de una función  $f$  por teoría de la evidencia es una distribución de posibilidad que recordemos que se obtiene como:

$$\pi(c) = \sum_{\substack{(a_1, b_1) \in C_i \\ \text{OR } (a_2, b_2) \in C_i \\ \dots \text{OR } (a_n, b_n) \in C_i \\ \forall (a_j, b_j) \mid c=f(a_j, b_j)}} m_{A \times B}(C_i)$$

En este caso, la distribución de posibilidad coincide con la que se obtiene de aplicar el principio de extensión de la teoría de la posibilidad, con la t-norma *min* (ver anexo A):

$$\pi(c) = \sup \min (\pi(a_j), \pi(b_j)) \quad \forall (a_j, b_j) \mid c=f(a_j, b_j)$$

Normalmente, en las operaciones de conjuntos borrosos si no se dispone de información adicional se utiliza la t-norma *min* ya que es la menos específica (al ser la mayor de todas), y es la utilizada en esta tesis para la extensión del análisis de planificabilidad. Por tanto, se podrá obtener el mismo tiempo de finalización borroso, bien aplicando el principio de extensión, bien considerando correlación positiva perfecta en el cuerpo de evidencia conjunto.

En la Figura 3.9 se muestra un ejemplo donde el tiempo de finalización de una tarea en una ejecución es una función de sólo dos tiempos de ejecución. En este caso, se puede obtener el tiempo de finalización borroso mediante la aplicación de la representación gráfica propuesta en esta tesis, por lo que el análisis propuesto de planificabilidad borrosa se puede realizar aplicando tanto el principio de extensión, como la representación gráfica.

Una vez definida la planificabilidad borrosa y enunciados los pasos a seguir para realizar el análisis, en los siguientes apartados se describen algunos métodos de obtención de los modelos borrosos de los parámetros y su posible interpretación, así como los requisitos que se exigen a dichos modelos para poder aplicar el análisis de planificabilidad borrosa propuesto en esta tesis. Posteriormente se presentará la obtención de los tiempos de finalización borrosos mediante el principio de extensión y mediante la aplicación de la representación gráfica de los cuerpos de evidencia.

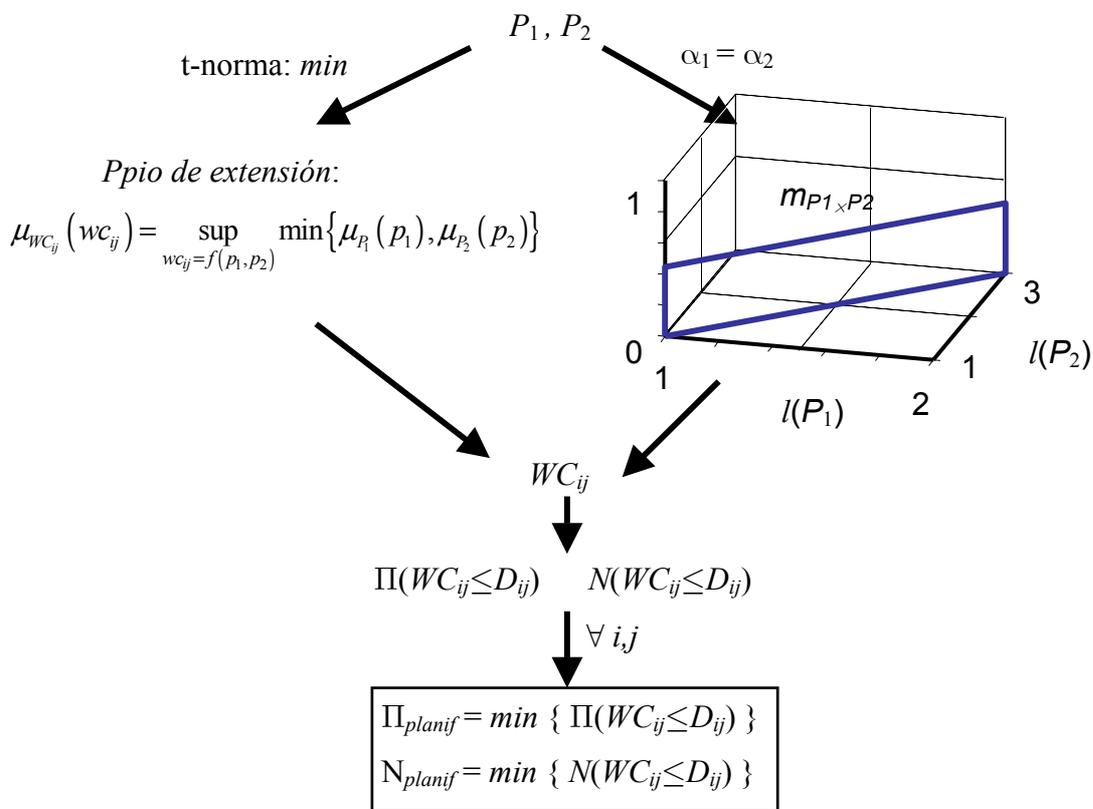


Figura 3.9: Análisis de planificabilidad borrosa, con dos formas de obtención de  $WC_{ij}$

### 3.3. Modelado borroso de parámetros

#### 3.3.1. Recogida de información de los expertos

En este apartado se describen algunos de los métodos existentes en la literatura, utilizados para recoger el conocimiento de los expertos y obtener un modelo borroso del parámetro acerca del cual se obtiene la información. Además se plantean las dos restricciones que hay que exigir al modelado borroso de los parámetros para que el análisis de planificabilidad que se propone en esta tesis sea válido.

##### Métodos

Existen en la literatura diversos métodos de obtención de la información de los expertos y su modelado con conjuntos borrosos. Aunque no es el objeto de esta tesis el profundizar en dichos métodos, éste sería el primer paso del proceso para el cálculo de la planificabilidad borrosa de un sistema de tiempo real. En [Dubois, Prade, 2000] se da un método para obtener la distribución de posibilidad asociada a un conjunto borroso a partir del conocimiento de un experto y en [Dubois, Prade, 1986a] se propone otro procedimiento que se basa en interrogar a un grupo de expertos.

El primero de los métodos ([Dubois, Prade, 2000]) se basa en la relación de las distribuciones de posibilidad con los intervalos de confianza de las distribuciones de probabilidad. Dada una función de densidad de probabilidad unimodal definida en la recta real, un conjunto de intervalos de confianza anidados pueden formar una distribución de posibilidad. Para un intervalo cualquiera  $[a, b]$ , la posibilidad de  $a$  es igual a la de  $b$  y es 1 menos la probabilidad de dicho intervalo, esto es,  $\pi(a)=\pi(b)=1-P([a, b])$ . Si se eligen los intervalos de confianza centrados en la moda de  $P$  y se obtienen haciendo cortes horizontales a la función de densidad de probabilidad, la distribución de posibilidad  $\pi$  obtenida es la más específica compatible con  $P$ , tal y como se describe en el anexo C. De acuerdo con esta transformación, la distribución de posibilidad asociada a una función de densidad uniforme es triangular.

Basado en esta interpretación se propone un método de obtención de una distribución de posibilidad interrogando a un solo experto. En fiabilidad se supone que un experto puede suministrar intervalos anidados con niveles de confianza asociados  $\lambda_1, \dots, \lambda_m$ , por ejemplo, dando la mejor cota inferior de la proporción de casos en que  $x$  pertenece al intervalo  $i$ . En el ejemplo de la Figura 3.10 tomado de [Dubois, Prade, 2000], hay tres intervalos de confianza asociados a los niveles  $\lambda_1=0.05$ ,  $\lambda_2=0.5$  y  $\lambda_3=0.95$ , y el intervalo con nivel de confianza 0 es toda la recta real debido a la

incertidumbre residual, ya que la probabilidad de que la variable esté fuera de  $[a_1, b_1]$  es 0.05.

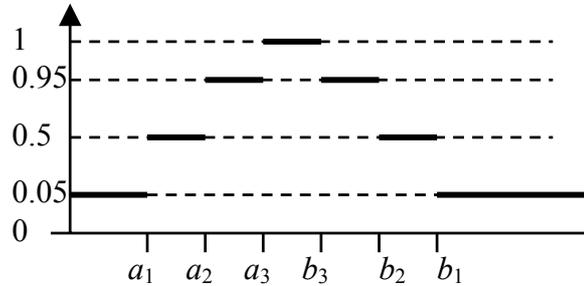


Figura 3.10: Distribución de posibilidad obtenida de un experto

En el segundo método ([Dubois, Prade, 1986a]) se obtienen las distribuciones de posibilidad a partir de experimentos aleatorios con salidas imprecisas, pidiendo a un grupo de personas que den un intervalo para definir el parámetro a modelar. Un caso sencillo consiste en modelar un parámetro que por su propia naturaleza proporcione intervalos anidados, por ejemplo el modelado de “alto”. Se entiende que “alto” significa mayor que un cierto umbral. Cada persona interrogada da un valor  $s_i$  a dicho umbral y suponiendo que no hay alturas mayores de 2.5 metros, el resultado es un conjunto de intervalos  $S_i = [s_i, 2.5]$  anidados entre sí. Si se dispone de un gran número de respuestas, se puede construir un histograma con los valores de probabilidad de cada intervalo y estas probabilidades constituyen una asignación básica de probabilidad con elementos focales anidados  $m([s_i, 2.5]) = P(S_i)$  (ver el apartado anterior). Por tanto, el conjunto borroso asociado a dicha asignación básica se obtiene como:

$$\forall s \quad \mu_{\text{alto}}(s) = Pl(\{s\}) = \sum_{s_i \leq s} P(S_i)$$

En un caso más general, cada persona proporciona un intervalo  $[a, b]$ . De forma estricta, para poder obtener un conjunto borroso sería necesario que los intervalos  $[a_k, b_k]$  estuvieran anidados, pero en la práctica esta condición es demasiado estricta. Una condición más realista y menos estricta es que todos ellos se solapen, esto es,

$$\exists [a, b] \neq \emptyset \mid \bigcap_{i=1..q} [a_i, b_i] = [a, b]$$

Se define entonces un conjunto de intervalos anidados tal que  $[a, b] \subseteq I_1 \subset I_2 \subset \dots \subset I_n = [A, B]$ , donde  $[A, B] = \bigcup_{i=1..q} [a_i, b_i]$ .

Cada salida del experimento  $[a_k, b_k]$  se asigna de forma única al intervalo  $I_i$  más pequeño que la contiene. Si se tiene un número suficientemente grande de respuestas, a partir del número de salidas asignadas a  $I_i$  se obtiene  $m(I_i)$ , la asignación básica de probabilidad a partir de la cual se puede calcular la distribución de posibilidad.

$$\mu_F(x) = Pl(\{x\}) = \sum_{x \in I_i} P(I_i) = \Pi(\{x\})$$

### **Tipos de conjuntos borrosos permitidos para realizar el análisis de planificabilidad borrosa**

A continuación se analiza si hay alguna restricción, en el marco de esta tesis, en los tipos de conjuntos borrosos permitidos para modelar los parámetros del sistema, concretamente en los aspectos de convexidad, continuidad o tipo de soporte de los conjuntos borrosos. Estas cuestiones se discuten en el apartado 3.5, en el capítulo 4 y en el anexo B, pero se presentan aquí de forma resumida las conclusiones con el fin de que quede explícito el tipo de modelos permitidos en esta descripción general del proceso de análisis de planificabilidad borrosa.

#### *Convexidad*

En esta tesis se realiza el estudio partiendo del modelado de parámetros como números borrosos. Sin embargo, los análisis borrosos que se van a proponer son igualmente válidos, si se modelan como conjuntos borrosos no convexos.

En el anexo B se prueba que las medidas de posibilidad y necesidad que se obtienen al comparar un número con un conjunto borroso no convexo coinciden con las que se obtienen al comparar ese número con cualquier otro conjunto borroso que tenga los mismos límites de sus  $\alpha$ -cortes. Dichos conjuntos borrosos se denominan en esta tesis aproximaciones válidas. La aproximación válida convexa  $A^*$  de un conjunto borroso no convexo  $A$  es aquella que tiene los mismos límites de los  $\alpha$ -cortes que  $A$  y es convexa (ver Figura 3.11).

Por tanto, es posible modelar el plazo como un conjunto borroso no convexo y tomar su aproximación convexa para trabajar con ese número borroso, realizando las comparaciones de los tiempos de finalización borrosos con las aproximaciones, de forma que el resultado de posibilidad y necesidad es el mismo que se hubiera obtenido de trabajar con los plazos no convexos.

En el capítulo 4, se analiza la extensión de funciones monótonas con parámetros modelados como conjuntos borrosos no convexos y se demuestra que si la función es

monótona, es suficiente operar con los límites de los  $\alpha$ -cortes para obtener una aproximación válida del resultado. Por tanto, es equivalente trabajar con el conjunto no convexo que con su aproximación convexa. Esto es aplicable, por ejemplo al caso de modelar los tiempos de ejecución como conjuntos no convexos.

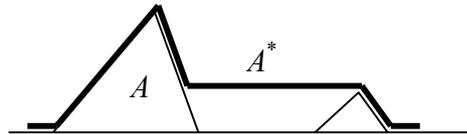


Figura 3.11: Conjunto borroso no convexo  $A$  y su aproximación válida convexa  $A^*$

### Continuidad

En el análisis borroso de planificabilidad que se propone en esta tesis se requiere que el número borroso que modela el parámetro sea continuo en  $\alpha$  para que se verifique  $\Pi(A \leq B) = \Pi(A < B)$ .

Tal y como se muestra en el anexo B, para que se cumpla  $\Pi(A \leq B) = \Pi(A < B)$  es suficiente con que una de las dos distribuciones de posibilidad  $\pi_A(s)$ ,  $\pi_B(t)$  sea continua, ver Figura 3.12.

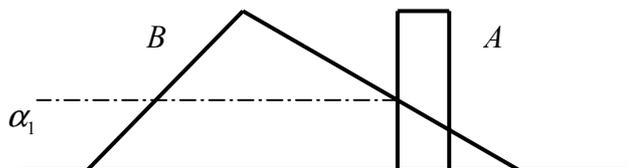


Figura 3.12: Comparación de dos números borrosos, siendo  $B$  continuo

Si el conjunto borroso no cumple esta condición de continuidad, se requiere que la discontinuidad sea sustituida por una recta con pendiente muy pronunciada.

### *Soporte acotado*

En el análisis que se presenta en esta tesis se requiere que el soporte del conjunto borroso sea acotado. De no ser así, no sería posible garantizar la planificabilidad de tareas estrictas.

Se puede observar, por tanto, que no se imponen restricciones fuertes al modelado de parámetros borrosos para permitir la aplicación del análisis borroso de planificabilidad. Sólo se requiere que el soporte de los conjuntos borrosos sea acotado y que o bien el plazo, o bien el tiempo de ejecución de la misma tarea sean continuos en  $\alpha$ . Sin embargo no se exige ninguna condición respecto a la convexidad de los parámetros borrosos.

### **3.3.2. Interpretación del modelado borroso de los parámetros**

En general, el modelado borroso de los parámetros puede responder a un conocimiento incierto de los mismos, como por ejemplo el tiempo de ejecución de una tarea cuando aún no está implementada. Si se dispone de la información necesaria, este tipo de modelado proporciona más información que dar un único valor para el peor caso, como sería en este ejemplo el máximo tiempo de ejecución de la tarea.

El modelado borroso de un plazo puede tener relación con el tipo de tarea a la que corresponde, estricta o flexible.

Si una tarea es estricta, debe haber total seguridad de que va a terminar su ejecución antes del plazo correspondiente. Si el modelo de alguno de sus parámetros es borroso esto se traduce en que la posibilidad y la necesidad de que la tarea termine antes de su plazo ha de ser 1. Esta condición es independiente de cómo esté modelado el plazo y de cómo sea el tiempo de finalización, numérico o borroso. En principio en este caso es más adecuado modelar el plazo como un número clásico, pero si el plazo depende de otro parámetro que es borroso, podría ser modelado también como un conjunto borroso. Si la tarea es estricta siempre hay que exigir que la posibilidad y la necesidad de que la tarea termine antes de su plazo sea 1, ya que el tiempo de finalización puede ser borroso.

Si una tarea es flexible, se permite que en algunas ejecuciones no cumplan su plazo. A menudo se trata de tareas con llegadas no acotadas, pero se podría modelar también tareas esporádicas y periódicas a las que se permite cierta flexibilidad en el

cumplimiento de sus plazos. Este tipo de tareas se pueden modelar en términos borrosos permitiendo que en ocasiones la posibilidad o la necesidad de que la tarea finalice antes de su plazo sea distinta de 1. De nuevo, esta condición es independiente de cómo esté modelado el plazo y de cómo sea el tiempo de finalización. Sería más apropiado modelar el plazo como un número borroso, aunque también sería válido en el caso de tener un número clásico.

Tanto el plazo borroso como la posibilidad y la necesidad de que la tarea finalice antes de su plazo pueden modelar el valor que la tarea aporta al sistema: cuanto más cerca de 1 estén la posibilidad y la necesidad, más valor ([Aldarmi, Burns, 1999a], [Aldarmi, Burns, 1999b]), bien por precisión del resultado, bien por ser un resultado más completo ([Liu et al., 1991], [Shih, Liu, 1995]). Así, una tarea flexible aporta un valor completo al sistema cuando finaliza antes de su plazo, mientras que si finaliza después sólo aporta una parte del valor completo, de acuerdo con la Figura 3.13.

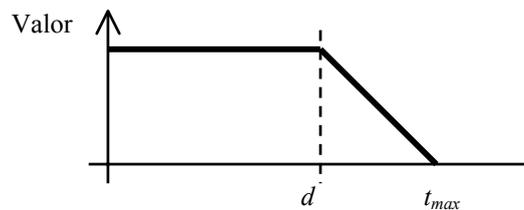


Figura 3.13: Función de valor de una tarea

De forma similar se puede modelar el concepto de valor de una tarea mediante un plazo borroso, de manera que cuando la posibilidad y la necesidad de que la tarea finalice antes de su plazo es 1, el valor es uno y cuando ambos son cero, el valor es cero. Un plazo borroso que cumple esta condición es aquel cuyo soporte es  $[d, t_{max}]$ , donde  $d$  y  $t_{max}$  se muestran en la figura anterior.

El modelado borroso del periodo puede reflejar falta de información, o bien el deseo de introducir cierta flexibilidad en el requisito de tiempo entre eventos consecutivos. Además, puede ser una forma de modelar el periodo de tareas esporádicas, si se tiene más información que el tiempo mínimo entre activaciones consecutivas.

### 3.4. Cálculo de la planificabilidad borrosa: principio de extensión y representación gráfica de los cuerpos de evidencia

En este apartado se revisa en primer lugar la representación gráfica propuesta en el anexo A y después se estudian dos ejemplos de cálculo de tiempo de finalización con sistemas planificados con el algoritmo RMS y un ejemplo con el algoritmo DMS, mediante el método gráfico de combinación de los cuerpos de evidencia y mediante la aplicación del principio de extensión. Aunque la representación gráfica propuesta es válida sólo para funciones con dos argumentos borrosos, sirve para obtener conclusiones extensibles a más de dos argumentos. Las conclusiones obtenidas gráficamente han sido comprobadas también mediante simulación para ejemplos en los que no se puede aplicar la representación gráfica, por ejemplo para calcular el tiempo de finalización de una tarea que depende de los tiempos de ejecución de tres tareas.

#### 3.4.1. Representación gráfica del cuerpo de evidencia conjunto para el cálculo de la distribución de plausibilidad de una función de dos argumentos

Tal y como se dijo en el apartado 3.1.2, se puede calcular la distribución de plausibilidad conjunta de cualquier par  $(a, b)$  según (3.6), conocida la asignación básica conjunta  $m_C(C_i)$  definida en el producto cartesiano  $X \times Y$ ,  $m_C(C_i) = m_{A \times B}(A_j \times B_k)$  tal que  $m_{A \times B}: \mathcal{P}(X \times Y) \rightarrow [0, 1]$  y  $(F_A, m_A)$ , de forma que  $(F_B, m_B)$  son las proyecciones del cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$  en  $X$  e  $Y$  respectivamente, también denominados cuerpos de evidencia marginales.

$$(3.6) \quad Pl(\{(a, b)\}) = \sum_{(a, b) \in C_i} m_C(C_i)$$

Los elementos focales conjuntos  $C_i$  se obtienen como el producto cartesiano de los elementos focales de los cuerpos de evidencia marginales, esto es,  $C_i = A_j \times B_k$ .

Si el cuerpo de evidencia conjunto es consonante, esta medida de plausibilidad es una medida de posibilidad. Suponiendo que los cuerpos de evidencia marginales son consonantes, los elementos focales  $A_j$  y  $B_k$  asociados a  $C_i$  tal que  $C_i = A_j \times B_k$  son los  $\alpha$ -cortes de los conjuntos borrosos  $A$  y  $B$  correspondientes a  $\alpha_A^j$  y  $\alpha_B^k$ . Por tanto, cada elemento focal conjunto  $C_i$  se puede caracterizar mediante una pareja  $(\alpha_A^j, \alpha_B^k)$

y la asignación básica conjunta se puede expresar en términos de correlación existente entre  $\alpha_A$  y  $\alpha_B$ , con lo que  $m_{A \times B}(\alpha_A, \alpha_B)$  está definida en  $[0,1]^2$ . Se puede interpretar como la probabilidad  $P(A_j \times B_k)$  del elemento focal conjunto o la probabilidad del par  $(\alpha_A^j, \alpha_B^k)$ .

El método gráfico propuesto consiste en representar, para dos números borrosos triangulares  $A$  y  $B$ :

- La asignación básica conjunta en función de los límites inferiores  $l(A_j)$  y  $l(B_k)$  de los elementos focales (o  $\alpha$ -cortes)  $A_j$  y  $B_k$  en los ejes  $x$  e  $y$ , en lugar de los correspondientes  $\alpha_A^j$  y  $\alpha_B^k$ . Cada par  $(l(A_j), l(B_k))$  del gráfico representa un elemento focal conjunto  $C_i$ . El dominio de los límites inferiores de los  $\alpha$ -cortes de un número borroso  $A$  lo denominamos  $U_{l(A)}$ , por lo que esta representación se realiza en el dominio de  $U_{l(A)} \times U_{l(B)}$
- Los elementos focales conjuntos  $C_i = A_j \times B_k$  en el dominio de los soportes de  $A$  y  $B$ , esto es  $]A[ \times ]B[$ , donde  $]A[$  es el soporte de  $A$ .

Recordemos que dado un cuerpo de evidencia, se puede calcular la distribución de plausibilidad y esa distribución se puede interpretar como la función de pertenencia de un conjunto borroso:  $\mu(x) = Pl(\{x\}) = \sum_{A_i \cap \{x\} \neq \emptyset} m(A_i) = \sum_{x \in A_i} m(A_i)$ . Si la asignación

básica es continua y los elementos focales son intervalos convexos de la recta real, entonces se expresa como:

$$\mu(x) = Pl(\{x\}) = \int_{x \in A_i} m(A_i) \cdot d\delta_{A_i}$$

donde  $\delta_{A_i}$  es una distancia entre intervalos compactos de la recta real. Esa distancia se define como:

$$\delta(A, B) = \sup \{ ds(a_{inf}, b_{inf}), ds(a_{sup}, b_{sup}) \}$$

donde  $a_{inf}$  es un número real que se define como el límite inferior del intervalo  $a_{inf} = \inf_{x \in A} x$  y  $a_{sup}$  es un número real que se define como el límite superior del intervalo  $a_{sup} = \sup_{x \in A} x$ . Así,  $ds(a_{inf}, b_{inf})$  es la distancia euclídea entre dos números  $a_{inf}$  y  $b_{inf}$ .

Si el cuerpo de evidencia es consonante, suponiendo que se ordenan los elementos focales de forma que  $A_1 \subset A_2 \subset \dots \subset A_n$ , entonces la función de pertenencia se calcula como:

$$\mu(x) = \pi(x) = \int_{A_n}^{A_j} m(A_i) \cdot d\delta_{A_i}$$

siendo los elementos focales que van de  $A_n$  a  $A_j$  todos aquellos que contienen a  $x$ .

Si se escoge como representante de cada elemento focal  $A_i$  su límite inferior  $l(A_i)$ , se puede expresar la asignación básica en función de dichos representantes  $m(l(A_i))$  y calcular así la función de pertenencia:

$$(3.7) \quad \mu(x) = \pi(x) = \int_{A_n}^{A_j} m(A_i) \cdot d(A_i) = \int_{l(A_n)}^{l(A_j)} m(l(A_i)) \cdot d(l(A_i))$$

Por tanto, dada una función de pertenencia de un conjunto borroso, o una distribución de posibilidad, se puede calcular la asignación básica del cuerpo de evidencia consonante asociado derivando la distribución de posibilidad como:

$$m(l(A_i)) = \frac{d(\pi(x))}{d(l(A_i))}$$

y los elementos focales de dicho cuerpo de evidencia son los  $\alpha$ -cortes del conjunto borroso de forma que  $A_1 \subset A_2 \subset \dots \subset A_n$  y  $A_1$  es el  $\alpha$ -corte con  $\alpha=1$ . Cuando  $A$  es un número borroso triangular  $l(A_j)$  está uniformemente distribuido en  $U_{l(A_i)}$ . Es decir, la asignación básica de los elementos focales representada en función de  $l(A_j)$  es una distribución uniforme, tal y como se ve en la Figura 3.14, ya que la derivada de la distribución de posibilidad en el dominio de los límites inferiores es una constante. El área completa bajo  $m_A(l(A_i))$  es 1.

Así, en la Figura 3.14, la posibilidad de  $x$  es el área marcada entre  $l(A_n)$  y  $l(A_j)$ , de acuerdo con (3.7), ya que los elementos focales del cuerpo de evidencia consonante asociado (o  $\alpha$ -cortes del conjunto borroso) que van desde  $A_n$  hasta  $A_j$  contienen a  $x$ . De la misma forma, se podrían haber escogido como representantes de los elementos focales los límites superiores de dichos intervalos y la representación sería similar.

La distribución de plausibilidad conjunta depende de la asignación básica conjunta y de su distribución en el dominio  $U_{l(A_i)} \times U_{l(B_i)}$ . En el anexo A se han analizado tres tipos distintos de distribución, la correlación positiva perfecta, la independencia y la correlación negativa perfecta.

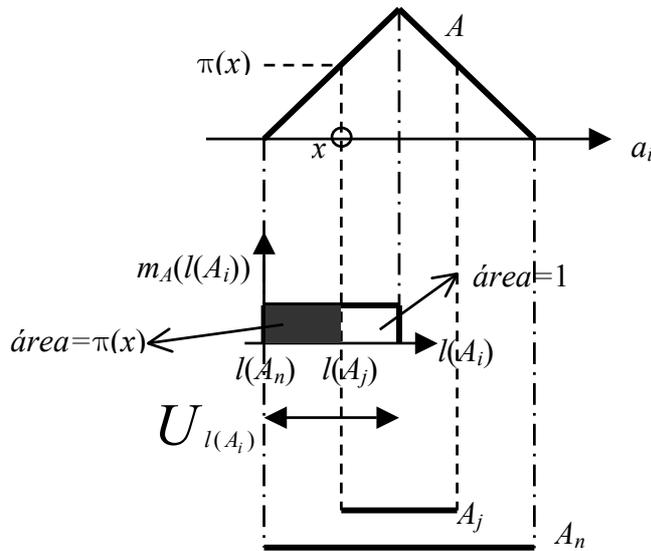


Figura 3.14: Número borroso triangular y asignación básica del cuerpo de evidencia consonante asociado

Cuando existe *correlación positiva perfecta* entre  $\alpha_A$  y  $\alpha_B$ , la asignación básica conjunta  $m_{A \times B}(\alpha_A, \alpha_B)$  se distribuye uniformemente en la diagonal principal  $\alpha_A = \alpha_B$  del cuadrado unidad  $[0,1]^2$ . Si los cuerpos de evidencia marginales son consonantes y las distribuciones de posibilidad asociadas a dichos cuerpos de evidencia son triangulares, las asignaciones básicas  $m_A(l(A_i))$  y  $m_B(l(B_i))$  son distribuciones uniformes en  $l(A_i)$  y  $l(B_i)$  respectivamente. Además, la diagonal principal  $\alpha_A = \alpha_B$  del cuadrado unidad  $[0,1]^2$  se corresponde con la diagonal principal de  $U_{l(A_i)} \times U_{l(B_i)}$ , y la distribución en la diagonal principal de  $U_{l(A_i)} \times U_{l(B_i)}$  cuyas proyecciones son uniformes es a su vez una distribución uniforme. Ver Figura 3.15.

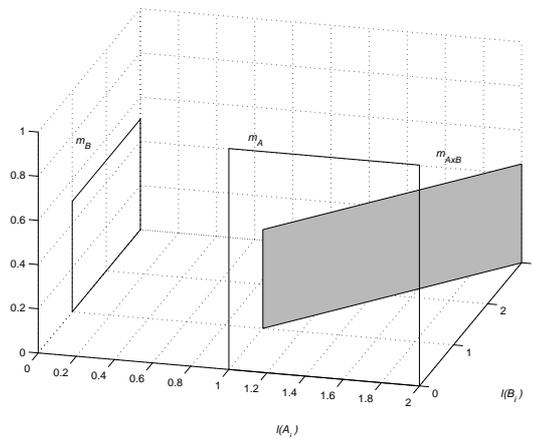


Figura 3.15: Asignación básica conjunta cuando  $\alpha_A = \alpha_B$

La Figura 3.16 muestra los elementos focales conjuntos completos, cuyo límite inferior izquierdo se desplaza a lo largo de la diagonal. Los elementos focales conjuntos son productos Cartesianos  $C_i=A_j \times B_k$  y en la Figura 3.16 se dibujan en el dominio de los soportes  $]A[\times]B[$ . El límite inferior izquierdo del elemento focal conjunto es el representante de dicho elemento focal en el gráfico de límites inferiores de la Figura 3.15. Se observa en la Figura 3.16 que los elementos focales conjuntos están anidados y, por tanto, las medidas de plausibilidad son medidas de posibilidad. Como los elementos focales conjuntos están anidados, se pueden numerar de forma que  $C_1 \subset C_2 \subset \dots \subset C_n$ . Para variables discretas, la posibilidad de  $(a_1, b_1)$  se calcula sumando la asignación básica de cada elemento focal conjunto que contenga el par  $(a_1, b_1)$ . En este ejemplo  $(a_1, b_1)$  vale  $(1.8, 2.6)$  y se puede ver en la Figura 3.16 que todos los elementos focales conjuntos desde  $C_i$  hasta  $C_n$  contienen el par  $(a_1, b_1)$ , por tanto la posibilidad de  $(a_1, b_1)$  expresado en continuo es:

$$\pi((a_1, b_1)) = \int_{(a_1, b_1) \in C_i} m_{A \times B}(C_i) \cdot d\delta_{C_i}$$

donde  $\delta_{C_i}$  es la distancia entre los elementos focales definidos anteriormente. La distancia  $\delta_{C_i}$  se define como:

$$\delta(C_j, C_s) = \sup \left\{ d(C_{sup}^j, C_{sup}^s), d(C_{inf}^j, C_{inf}^s), d(C_{der}^j, C_{der}^s), d(C_{izq}^j, C_{izq}^s) \right\}$$

donde  $d(I_1, I_2)$  es la distancia entre dos intervalos definida posteriormente y  $C_{izq}^j$  es un intervalo correspondiente al lado izquierdo del elemento focal rectangular  $C_j$ ,  $C_{der}^j$  es un intervalo correspondiente al lado derecho de  $C_j$ ,  $C_{inf}^j$  es un intervalo correspondiente al lado inferior de  $C_j$ , y  $C_{sup}^j$  es un intervalo correspondiente al lado superior de  $C_j$ , esto es:

$$C_{izq}^j = \{(x, y) \mid x = \inf(x_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{der}^j = \{(x, y) \mid x = \sup(x_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{inf}^j = \{(x, y) \mid y = \inf(y_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{sup}^j = \{(x, y) \mid y = \sup(y_0)\}_{(x_0, y_0) \in C_j}$$

Además se define la distancia entre dos intervalos como:

$$d(I_1, I_2) = \sup \left\{ \inf_{\substack{a \in I_1 \\ b \in I_2}} ds(a, b) \right\}$$

donde  $ds(a, b)$  es la distancia euclídea entre dos números reales.

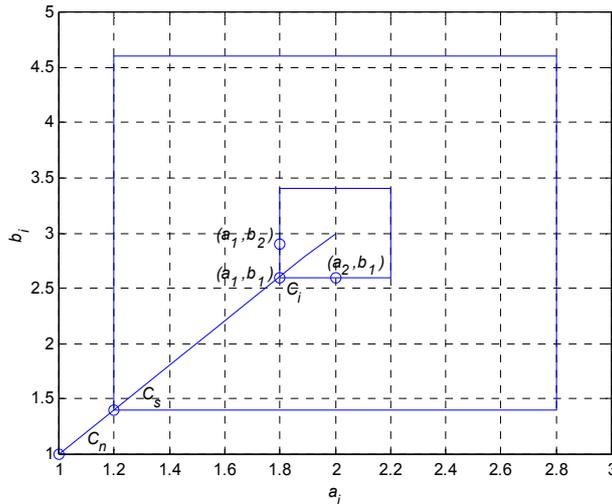


Figura 3.16: Elementos focales del cuerpo de evidencia conjunto cuando  $\alpha_A = \alpha_B$

Tanto la asignación básica conjunta  $m_{A \times B}$  como la posibilidad  $\pi((a_1, b_1))$  del par  $(a_1, b_1)$  aparecen en la Figura 3.17. En continuo, la posibilidad  $\pi((a_1, b_1))$  es el área sombreada, donde la posición del punto  $(a_1, b_1)$  y del par  $(I(A_r), I(B_s))$  que representa el elemento focal conjunto  $C_i$  coincide, ambos están en (1.8, 2.6). Como dicho punto pertenece a la recta definida por  $\alpha_A = \alpha_B$ , se cumple que  $\pi((a_1, b_1)) = \pi_A(a_1) = \pi_B(b_1)$  (que en este ejemplo vale 0.8).

Para un punto  $(a_2, b_1)$  del cuadrante inferior izquierdo situado por debajo de la diagonal correspondiente a  $\alpha_A = \alpha_B$ , se tiene que  $\alpha_A > \alpha_B$ , o lo que es lo mismo,  $\pi_A(a_2) > \pi_B(b_1)$ . Los elementos focales conjuntos que contienen al par  $(a_2, b_1)$  son los mismos que contienen a  $(a_1, b_1)$ , con lo que, para  $\pi_A(a_2) > \pi_B(b_1)$ , se tiene que  $\pi((a_2, b_1)) = \pi((a_1, b_1)) = \pi_B(b_1)$

Por otro lado, para un punto  $(a_1, b_2)$  del cuadrante inferior izquierdo situado por encima de la diagonal correspondiente a  $\alpha_A = \alpha_B$ , se tiene que  $\alpha_B > \alpha_A$ , o lo que es lo mismo,  $\pi_A(a_1) < \pi_B(b_2)$ . Los elementos focales conjuntos que contienen al par  $(a_1, b_2)$

son los mismos que contienen a  $(a_1, b_1)$ , con lo que para  $\pi_A(a_1) < \pi_B(b_2)$ , se tiene que  $\pi((a_1, b_2)) = \pi((a_1, b_1)) = \pi_A(a_1)$

Por tanto, se concluye que  $\pi((a_i, b_j)) = \min(\pi_A(a_i), \pi_B(b_j))$

En el ejemplo de la Figura 3.16, se tiene:

$$\pi((a_2, b_1)) = \min(\pi_A(2), \pi_B(2.6)) = \min(1, 0.8) = 0.8$$

$$\pi((a_1, b_2)) = \min(\pi_A(1.8), \pi_B(2.9)) = \min(0.8, 0.95) = 0.8$$

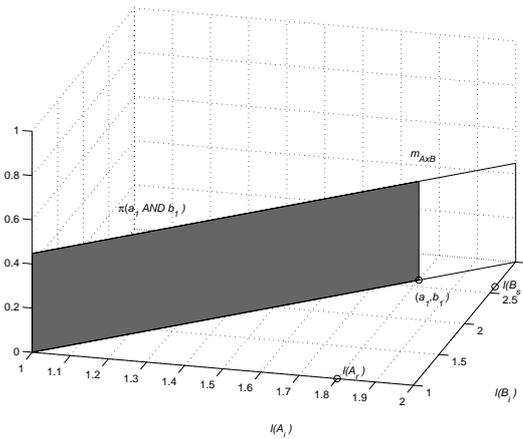


Figura 3.17: Posibilidad de  $(a_1, b_1)$  cuando  $\alpha_A = \alpha_B$

Mediante esta representación se obtiene el resultado ya conocido de que las distribuciones conjuntas de posibilidad obtenidas por teoría de la evidencia con  $\alpha_A = \alpha_B$  por un lado, y por teoría de la posibilidad con la t-norma *min* por otro lado, coinciden.

El segundo tipo de correlación analizado es la relación de *independencia*. Cuando  $\alpha_A$  y  $\alpha_B$  son independientes están uniformemente distribuidas en el cuadrado unidad  $[0,1]^2$ , o lo que es lo mismo,  $(I(A_j), I(B_k))$  está uniformemente distribuida en todo el dominio  $U_{I(A_i)} \times U_{I(B_i)}$

En la Figura 3.18 se muestra la asignación básica conjunta de los cuerpos de evidencia de dos conjuntos borrosos *A* y de *B*, cuando  $\alpha_A$  y  $\alpha_B$  son independientes.

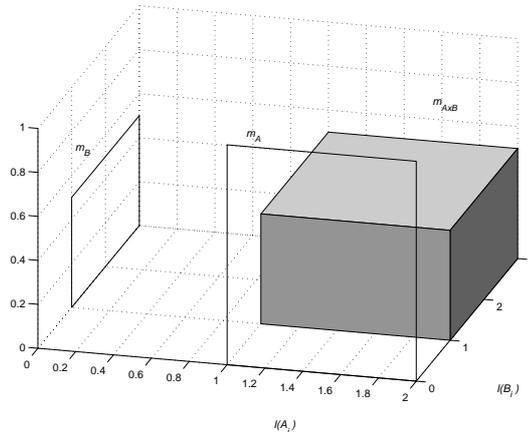


Figura 3.18: Asignación básica conjunta cuando  $\alpha_A$  y  $\alpha_B$  son independientes

La asignación básica conjunta coincide con el resultado de combinar los cuerpos de evidencia de  $A$  y de  $B$  o cuerpos de evidencia marginales  $(F_A, m_A)$  y  $(F_B, m_B)$  mediante la regla de Dempster [Klir, Folger, 1988]. En este caso se dice que los cuerpos de evidencia marginales son no interactivos, ya que los elementos focales conjuntos son productos cartesianos de los elementos focales de cuerpos de evidencia marginales  $A_i \times B_i$  y la asignación básica conjunta se calcula como el producto de las asignaciones marginales.  $m_{A \times B}(A_i \times B_i) = m_A(A_i) \cdot m_B(B_i)$

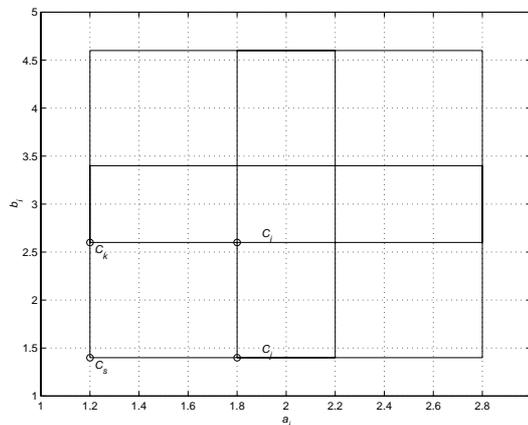


Figura 3.19: Elementos focales conjuntos cuando  $\alpha_A$  y  $\alpha_B$  son independientes

En la Figura 3.19 se muestran cuatro elementos focales  $C_i, C_j, C_k, C_s$  del cuerpo de evidencia conjunto. Como se puede ver, no están anidados por lo que las medidas de plausibilidad asociadas a este cuerpo de evidencia no son medidas de posibilidad.

Los límites inferiores izquierdos de los elementos focales conjuntos que en la Figura 3.19 se marcan con un punto, son los representantes de los elementos focales,  $(I(A_i), I(B_j))$  en la Figura 3.18.

Para calcular la plausibilidad de  $(a_1, b_1)$  hay que considerar todos los elementos focales conjuntos que contienen el par  $(a_1, b_1)$ . Los elementos focales conjuntos representados por los puntos situados en la base del volumen de la Figura 3.20, contienen  $(a_1, b_1)$ , siendo en este ejemplo  $(a_1, b_1) = (1.3, 2.2)$ . Expresado para variables continuas, la plausibilidad de ese punto es el volumen marcado en la Figura 3.20, y se puede calcular como:

$$\begin{aligned}
 Pl((a_1, b_1)) &= \int_{C_i \cap (a_1, b_1) \neq \emptyset} m_{A \times B}(C_i) \cdot d\delta_{C_i} = \int_{A_i=A_j}^{A_n} \int_{B_i=B_k}^{B_n} m_A(A_i) \cdot m_B(B_i) \cdot d\delta_{A_i} \cdot d\delta_{B_i} = \\
 &= \int_{A_i=A_j}^{A_n} m_A(A_i) \cdot d\delta_{A_i} \cdot \int_{B_i=B_k}^{B_n} m_B(B_i) \cdot d\delta_{B_i} = \pi_A(a_1) \cdot \pi_B(b_1)
 \end{aligned}$$

donde  $\delta_{C_i}$  es la medida definida entre elementos focales conjuntos como:

$$\delta(C_j, C_s) = \sup \left\{ d(C_{sup}^j, C_{sup}^s), d(C_{inf}^j, C_{inf}^s), d(C_{der}^j, C_{der}^s), d(C_{izq}^j, C_{izq}^s) \right\}$$

y donde  $\delta_{A_i}$  (respectivamente  $\delta_{B_i}$ ) se refiere a la distancia entre elementos focales de  $A$  (respectivamente  $B$ ) definida anteriormente en este apartado.

Vemos que la plausibilidad de  $(a_1, b_1)$  es el producto de posibilidades  $\pi_A(a_1)$ ,  $\pi_B(b_1)$ , y por tanto, la distribución conjunta de plausibilidad en este caso coincide con la distribución conjunta de posibilidad obtenida por teoría de la posibilidad cuando la t-norma utilizada es el producto.

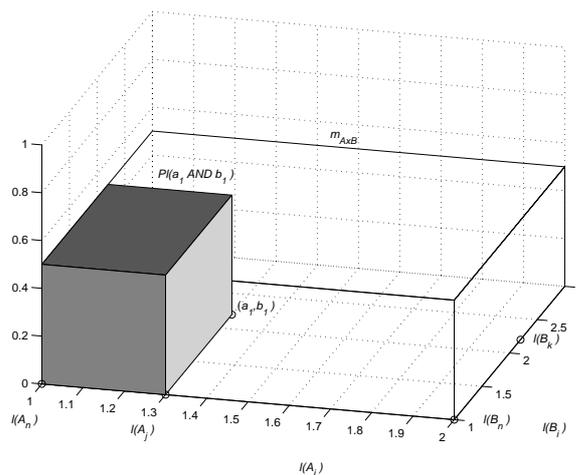


Figura 3.20: Plausibilidad de  $(a_1, b_1)$  cuando  $\alpha_A$  y  $\alpha_B$  son independientes

Cuando existe *correlación negativa perfecta* entre  $\alpha_A$  y  $\alpha_B$  la asignación básica conjunta es una distribución uniforme en la anti-diagonal del cuadrado unidad, esto es, en la recta  $\alpha_A=1-\alpha_B$ . La anti-diagonal del cuadrado unidad se corresponde con la anti-diagonal de  $U_{I(A_i)} \times U_{I(B_i)}$ .

En la Figura 3.21 se muestra la asignación básica conjunta de los cuerpos de evidencia de  $A$  y de  $B$ , cuando existe correlación negativa perfecta entre  $\alpha_A$  y  $\alpha_B$ .

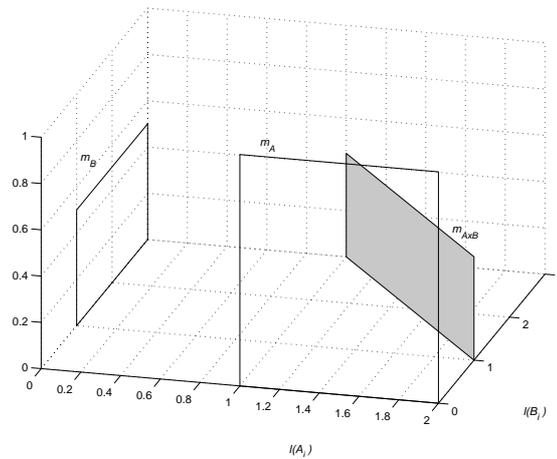


Figura 3.21: Asignación básica conjunta cuando  $\alpha_A = 1 - \alpha_B$

En este caso los elementos focales conjuntos no están anidados, tal y como se observa en la Figura 3.22, donde se muestran dos elementos focales conjuntos  $C_k$  y  $C_j$  (nótese que los vértices de los elementos focales se desplazan a lo largo de las líneas discontinuas). Al igual que en el caso anterior, el cuerpo de evidencia no es consonante y, por tanto, las medidas de plausibilidad no son medidas de posibilidad.

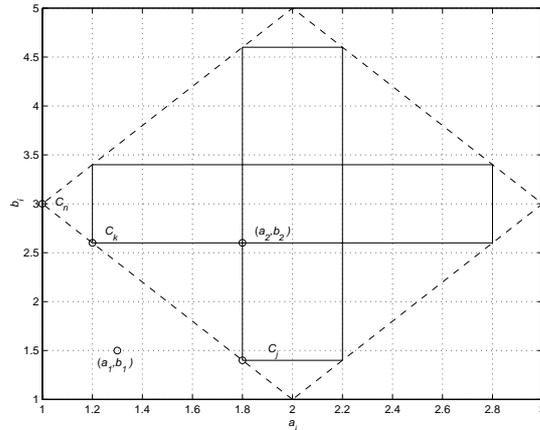


Figura 3.22: Elementos focales del cuerpo de evidencia conjunto cuando  $\alpha_A=1-\alpha_B$

En la Figura 3.22 se puede ver que la plausibilidad de  $(a_1, b_1)$ , con  $(a_1, b_1)=(1.3, 1.5)$  es 0, ya que ningún elemento focal del cuerpo de evidencia conjunto contiene dicho punto. Para calcular la plausibilidad de  $(a_2, b_2)=(1.8, 2.6)$ , se observa que todos los elementos focales conjuntos que van de  $C_j$  a  $C_k$  contienen dicho punto, con lo que expresado en variables continuas se tiene:

$$Pl((a_2, b_2)) = \int_{C_i=C_k}^{C_j} m_{AxB}(C_i) \cdot d\delta_{C_i}$$

que puede interpretarse como el área sombreada en la Figura 3.23.

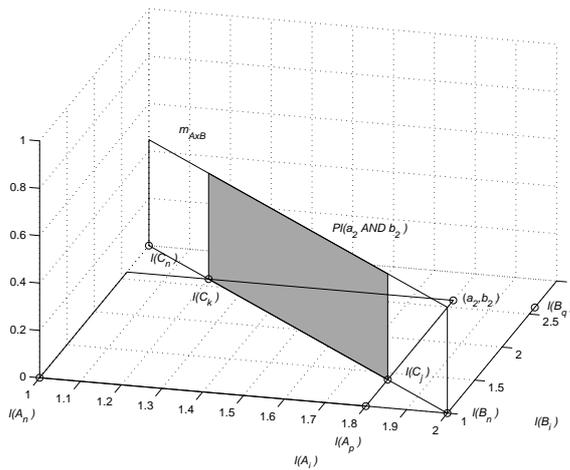


Figura 3.23: Plausibilidad de  $(a_2, b_2)$  cuando  $\alpha_A=1-\alpha_B$

Se puede expresar la medida de plausibilidad en términos de los cuerpos de evidencia de partida como

$$\begin{aligned} Pl((a_2, b_2)) &= \int_{C_i=C_k}^{C_j} m_{A \times B}(C_i) \cdot d\delta_{C_i} = \int_{C_i=C_n}^{C_j} m_{A \times B}(C_i) \cdot d\delta_{C_i} - \int_{C_i=C_n}^{C_k} m_{A \times B}(C_i) \cdot d\delta_{C_i} = \\ &= \int_{A_i=A_p}^{A_n} m_A(A_i) \cdot d\delta_{A_i} - \int_{B_i=B_1}^{B_q} m_B(B_i) \cdot d\delta_{B_i} = \int_{A_i=A_p}^{A_n} m_A(A_i) \cdot d\delta_{A_i} - \left( 1 - \int_{B_i=B_q}^{B_n} m_B(B_i) \cdot d\delta_{B_i} \right) = \\ &= \pi_A(a_2) - 1 + \pi_B(b_2) \end{aligned}$$

La plausibilidad del par anterior  $(a_1, b_1)$  es nula y  $\pi_A(a_1) + \pi_B(b_1) - 1 < 0$ , por tanto en general, para cualquier par  $(a_i, b_j)$ , se cumple:

$$Pl((a_i, b_j)) = \max(0, \pi_A(a_i) + \pi_B(b_j) - 1)$$

que es precisamente la t-norma de Lukasiewicz. La distribución conjunta de plausibilidad obtenida en teoría de la evidencia con  $\alpha_A = 1 - \alpha_B$  coincide con la distribución conjunta de posibilidad con la t-norma de Lukasiewicz.

### Extensión de una función por teoría de la evidencia

Dado un cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$ , se puede extender una función de números reales  $r = f(a, b)$  de forma que la distribución de plausibilidad resultante de la extensión de  $f$  es:

$$(3.8) \quad Pl(r) = \sum_{\substack{(a_1, b_1) \in C_i \\ OR (a_2, b_2) \in C_i \\ \dots OR (a_n, b_n) \in C_i \\ \forall (a_j, b_j) \mid r = f(a_j, b_j)}} m_{A \times B}(C_i)$$

En el anexo A se muestra de forma gráfica que en general las distribuciones de plausibilidad que se obtienen al aplicar (3.8) en teoría de la evidencia son menos específicas que las distribuciones de posibilidad obtenidas al aplicar el principio de extensión de la teoría de la posibilidad. Únicamente, cuando la correlación es positiva perfecta en el cuerpo de evidencia conjunto, la distribución de plausibilidad de la imagen de la función es una distribución de posibilidad y coincide con la que se obtiene mediante el principio de extensión empleando la t-norma *min*. A continuación se describe en detalle la representación gráfica de la extensión de una función cuando existe correlación positiva perfecta. La representación para las otras dos correlaciones analizadas se puede consultar en el anexo A.

Dada una función  $r=f(a,b)$  la posibilidad de cada curva de nivel  $f(a,b)=cte$  se calcula sumando (o bien integrando, para variables continuas) la asignación básica conjunta de todos los elementos focales que contengan al menos un punto de dicha curva según (3.8).

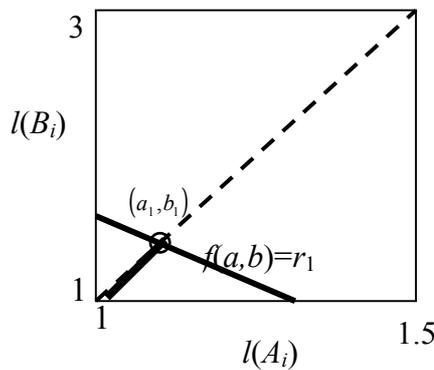


Figura 3.24: Elementos focales conjuntos que contienen algún punto de  $f(a,b)=r_1$  con  $\alpha_A=\alpha_B$

En la Figura 3.24 se marcan con línea gruesa los representantes de dichos elementos focales cuando la correlación entre  $\alpha_A$  y  $\alpha_B$  es positiva perfecta, esto es  $\alpha_A=\alpha_B$  (en línea discontinua). En el ejemplo de la Figura 3.24, la posibilidad de  $f(a,b)=r_1$  coincide con la posibilidad del punto  $(a_1,b_1)$  definido por el corte entre la diagonal principal y  $f(a,b)=r_1$ . Es decir, la posibilidad de  $f(a,b)=r_1$  es la integral de la asignación básica conjunta en el tramo grueso de la diagonal principal.

En la Figura 3.25 se representan, tanto los elementos focales conjuntos  $C_i$  como las curvas de nivel de la función  $f(a,b)$  a extender. Los elementos focales conjuntos son rectángulos con los vértices situados a lo largo de las diagonales. Los pares  $(a,b)$  con el mismo  $\alpha$ , es decir con el mismo valor de posibilidad  $\pi_A(a)=\pi_B(b)$  dados los conjuntos borrosos  $A$  y  $B$  son los puntos situados en dichas diagonales.

La posibilidad de cada curva de nivel  $f(a,b)=cte$  es la suma de la asignación básica de los elementos focales conjuntos  $C_i$  que contienen al menos un punto de la curva. Esto es, en la Figura 3.25 todos los elementos focales desde  $C_n$  hasta  $C_i$  contienen al menos un punto de  $f(a,b)=r_1$ . Estos elementos focales conjuntos son los que contienen al par  $(a_1,b_1)$  que es el punto de la recta  $f(a,b)=r_1$  que cumple que  $a$  y  $b$  tienen el mismo valor de posibilidad  $\pi_{A \times B}((a_1,b_1))=\pi_A(a_1)=\pi_B(b_1)=\alpha_i$ . Para calcular la posibilidad de  $f(a,b)=r_2$  (ver Figura 3.25) hay que integrar la asignación básica conjunta para los mismos elementos focales de  $C_n$  a  $C_i$ , con lo que  $\pi_{A \times B}((a_1,b_1))=\pi_{A \times B}((a_2,b_2))=\pi_A(a_1)=\pi_B(b_1)=\pi_A(a_2)=\pi_B(b_2)=\alpha_i$ .

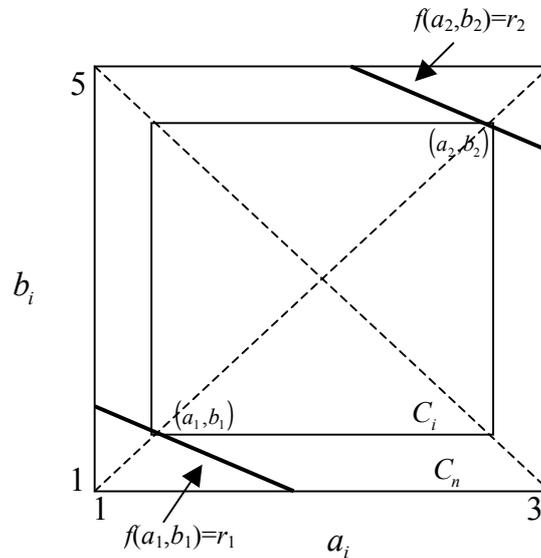


Figura 3.25: Representación de la función  $r=f(a,b)$  así como de los elementos focales conjuntos

Cuando la correlación es positiva perfecta el cálculo gráfico de la posibilidad es sencillo por lo que se propone trabajar directamente con la representación de los elementos focales conjuntos y de la función  $f(a,b)$  en el dominio de los soportes de los conjuntos borrosos  $]A[\times]B[$ , prescindiendo de la representación de la asignación básica conjunta en el dominio de los límites inferiores de los  $\alpha$ -cortes asociada. Por tanto los pasos a seguir son:

1. Representación gráfica de la función real  $r=f(a,b)$ . Se representan las curvas de nivel de la función en el dominio de los soportes  $]A[\times]B[$
2. Representación gráfica de los elementos focales conjuntos sobre la representación de las curvas de nivel de la función, y obtención del valor de posibilidad de cada curva de nivel.

### 3.4.2. Algoritmo RMS

Tal y como se describió en el capítulo 2, el algoritmo de planificación RMS basado en prioridades fijas asigna las prioridades a las tareas de acuerdo con el tamaño del periodo. En este apartado el tiempo de finalización borroso de una tarea se obtiene

mediante la aplicación de dos métodos, el principio de extensión y la representación gráfica propuesta en el anexo A.

En un ejemplo sencillo, de un sistema formado por tres tareas, primero se calculan los tiempos de finalización cuando todos los parámetros son numéricos, y a continuación se extiende a tiempos de ejecución borrosos.

### Tiempos de ejecución numéricos

Sea un sistema de tiempo real formado por tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos 3, 5 y 15 respectivamente y tiempos de ejecución numéricos 1, 2 y 3 respectivamente. Con el algoritmo RMS, la tarea  $A$  es la de mayor prioridad ya que tiene el menor periodo, mientras que la  $C$  es la de menor prioridad. La Figura 3.26 muestra el gráfico de planificación, llamado también cronograma [Burns, Wellings, 1989].

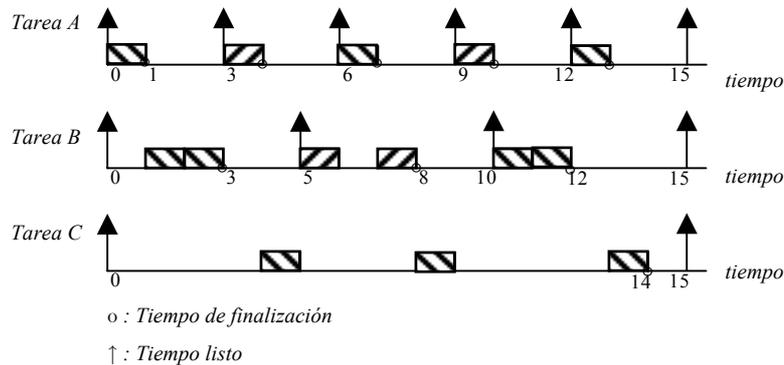


Figura 3.26: Gráfico de planificación de un sistema con algoritmo RMS

Este análisis se plantea de forma general, calculando todos los tiempos de finalización en un hiperperiodo, para ilustrar el proceso propuesto en la Figura 3.9. Nótese que en un caso como este, donde existe un instante crítico, el peor caso en el tiempo de finalización de cada tarea se produce en la primera ejecución (ver capítulo 2). Por tanto, sería suficiente comparar los tiempos finales en la primera ejecución con los plazos correspondientes para determinar si el sistema es planificable.

Las tres tareas están listas por primera vez en el instante crítico. Si nos fijamos en los tiempos de finalización de  $A$ , esta tarea podrá respetar cualquier plazo  $d_A$  situado

entre 1 y 3 (su periodo). Dado que los tiempos de finalización de  $B$  están en 3, 8 y 12, si el valor del plazo relativo  $d_B$  está entre 3 y 5 lo podrá respetar. Finalmente, el plazo  $d_C$  de  $C$  debería situarse entre 14 y 15.

En este caso el algoritmo RMS es una función discontinua  $rms: \mathbb{R}^3 \rightarrow \mathbb{R}^9$  de los tiempos de ejecución de las tareas ( $p_1, p_2$  y  $p_3$ ) a los tiempos de finalización ( $wc_{ij}$ ) en un hiperperiodo, que se puede observar en la Figura 3.26 que son 9.

### Tiempos de ejecución borrosos

Supongamos ahora unos tiempos de ejecución borrosos  $P_A, P_B$  y  $P_C$  con funciones de pertenencia triangulares definidas por sus vértices  $P_A=(0.9;1;1.1)$ ,  $P_B=(1.9;2;2.1)$  y  $P_C=(2.9;3;3.1)$ .

El primer método que se propone para obtener las distribuciones de posibilidad es la aplicación del principio de extensión con la t-norma *min*. Sea el tiempo de finalización de una tarea  $i$  en su ejecución  $j$ ,  $wc_{ij}$ . Se puede aplicar el principio de extensión para obtener el conjunto borroso  $WC_{ij}$  cuando se consideran tiempos de ejecución borrosos:

$$\mu_{WC_{ij}}(wc_{ij}) = \sup_{wc_{ij}=rms(p_A, p_B, p_C)} \min\{\mu_{P_A}(p_A), \mu_{P_B}(p_B), \mu_{P_C}(p_C)\}$$

donde en este caso la función  $rms$  es el algoritmo RMS que da como única salida el tiempo de finalización  $wc_{ij}$  es decir, se define esta función como  $rms: \mathbb{R}^3 \rightarrow \mathbb{R}^1$ , en lugar de obtener todos los tiempos de finalización en un hiperperiodo.

Para obtener la función de pertenencia de  $WC_{ij}$  se puede aproximar el principio de extensión muestreando los soportes de  $P_A, P_B$  y  $P_C$  con  $n_A, n_B$  y  $n_C$  número de muestras, y simulando todas las combinaciones ( $p_A, p_B, p_C$ ) [Lootsma, 1997]. Esto supone realizar  $n_A \cdot n_B \cdot n_C$  simulaciones con números reales.

Por ejemplo, el tiempo de finalización de la tarea  $B$  en su tercera ejecución  $wc_{B3}$ , sólo depende de los tiempos de ejecución de las tareas  $A$  y  $B$  es decir  $rms: \mathbb{R}^2 \rightarrow \mathbb{R}^1$  y la Figura 3.27 muestra la distribución de posibilidad que se obtiene para el conjunto borroso  $WC_{B3}$ .

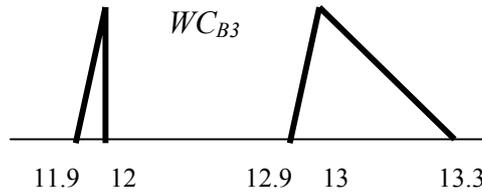


Figura 3.27: Distribución de posibilidad del tiempo de finalización de la tarea B en su tercera ejecución

Se puede observar que  $WC_{B3}$  es un conjunto borroso no convexo. El “hueco” se debe al hecho de que cuando  $A$  se está ejecutando (entre 12 y 12.9) es imposible que  $B$  se ejecute, y por tanto que finalice. Como  $A$  comienza su quinta ejecución exactamente en el instante 12, la posibilidad de que  $B$  finalice en 12 cae verticalmente a 0. El tiempo de finalización de la tarea  $A$  en su quinta ejecución es borroso, y por eso la distribución de posibilidad  $WC_{B3}$  crece linealmente de 12.9 a 13.

El segundo procedimiento para calcular la distribución de posibilidad del tiempo de finalización de la tarea  $B$  en su tercera ejecución es mediante la representación gráfica del cuerpo de evidencia conjunto con correlación positiva perfecta, esto es siendo la distribución de la asignación básica conjunta una uniforme en la diagonal principal. Es posible aplicar este segundo método porque, como ya se ha dicho, la función a extender en este caso es una función de dos argumentos  $w_{CB3} = rms(p_A, p_B)$ . En la aplicación del método gráfico hay que seguir dos pasos:

1. Representación gráfica de la función real. En este caso  $w_{CB3} = rms(p_A, p_B) = cte$ .
2. Representación gráfica de los elementos focales conjuntos, y obtención del valor de posibilidad de cada curva de nivel  $w_{CB3} = cte$ .

1. Representación de la función real.

La función real a representar es  $w_{CB3} = rms(p_A, p_B)$ , donde  $rms: \mathbb{R}^2 \rightarrow \mathbb{R}^1$ . Como la tarea  $C$  es la de menor prioridad,  $w_{CB3}$  no depende de su tiempo de ejecución, sino que únicamente depende de los tiempos de ejecución de  $A$  y de  $B$ . Al ser una función de dos argumentos se puede aplicar la representación gráfica propuesta en el anexo A. En la Figura 3.28 se muestran algunas de las curvas de nivel correspondientes a  $w_{CB3} = cte$ .

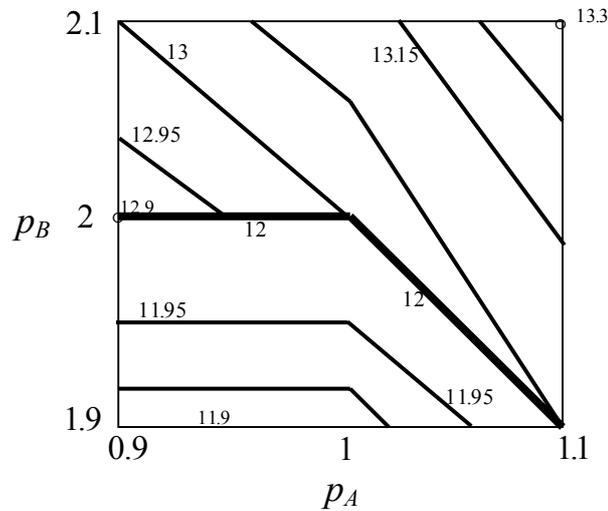


Figura 3.28: Representación de  $wc_{B3} = cte$

Cuando  $p_A \leq 1$  y  $p_B \leq 2$ , se puede ver en la Figura 3.26 que el tiempo de finalización de la tarea  $B$  en su tercera ejecución sólo depende del tiempo de ejecución de la tarea  $B$ , es decir, para tiempo de ejecución  $p_B$  constante, el tiempo de finalización  $wc_{B3}$  es también constante. Esto es debido a que en ese caso:

- el tiempo de finalización de  $A$  en su cuarta ejecución no afecta a la tercera ejecución de  $B$  porque la cuarta ejecución de  $A$  termina antes de que  $B$  esté lista para su tercera ejecución (en el instante 10), y además,
- el tiempo de finalización de  $A$  en su quinta ejecución no afecta a la tercera ejecución de  $B$  porque  $B$  termina su tercera ejecución antes de que  $A$  esté lista para su quinta ejecución (en el instante 12).

El menor valor que toma  $wc_{B3}$  en esta zona  $p_A \leq 1$  y  $p_B \leq 2$  es para  $p_B = 1.9$  y viene dado por  $wc_{B3} = 10 + 1.9 = 11.9$ . El mayor valor es  $wc_{B3} = 12$ , para  $p_B = 2$ .

Si  $wc_{B3} \leq 12$  y  $p_A \geq 1$  entonces el tiempo de finalización de la tarea  $A$  en su cuarta ejecución interfiere en el tiempo de finalización de  $B$   $wc_{B3}$  analizado. Esto ocurre porque al ser  $p_A \geq 1$ , la tarea  $A$  termina su cuarta ejecución después de que  $B$  está lista en su tercera ejecución (instante 10), retrasando la ejecución y la finalización de la tarea  $B$ . Por tanto, en esa zona  $wc_{B3} = cte$  para  $p_A + p_B = cte'$  siendo  $cte \neq cte'$ . El tiempo de finalización de  $B$ ,  $wc_{B3}$ , vale 12 para la recta  $p_A + p_B = 3$ , con  $p_A \geq 1$  y  $p_B \leq 2$ . La línea formada por  $wc_{B3} = 12$  es la línea de discontinuidad de la función, tal y como se puede ver en la Figura 3.28.

Si  $p_A \leq 1$  y  $p_B > 2$ , la cuarta ejecución de  $A$  no afecta a  $w_{CB3}$ , porque  $A$  termina antes del instante 10. Además, como  $p_B > 2$  su tiempo de finalización es mayor que 12, con lo que la tarea  $A$  en su quinta ejecución interrumpe a  $B$  y retrasa su finalización. Por tanto, en esa zona se cumple que  $w_{CB3} = cte$  para  $p_A + p_B = cte$ . El menor valor que toma  $w_{CB3}$  es, en el límite cuando  $p_B$  tiende a 2 y  $p_A = 0.9$ ,  $w_{CB3} = 12.9$ , ( $w_{CB3} = 10 + p_B + p_A$ ). El mayor valor de  $w_{CB3}$  en esta zona es 13.1, para  $p_A = 1$  y  $p_B = 2.1$ .

Si  $p_A \geq 1$  y  $w_{CB3} > 12$ , se tiene que:

- La cuarta ejecución de  $A$  retrasa la finalización de  $B$   $w_{CB3}$ , ya que al ser  $p_A \geq 1$ , la tarea  $A$  termina su ejecución después del instante 10.
- Como  $w_{CB3} > 12$ , la quinta ejecución de  $A$  interrumpe a  $B$ , retrasando su finalización.

Como hay dos ejecuciones de  $A$  que afectan a  $w_{CB3}$ , en esta zona se cumple  $w_{CB3} = cte$  para  $2 \cdot p_A + p_B = cte$ , tal y como se ve en la Figura 3.28, y  $w_{CB3}$  varía desde 13 hasta 13.3 (en  $p_A = 1.1$  y  $p_B = 2.1$ ).

## 2. Representación de los elementos focales y obtención de la distribución de posibilidad.

Una vez representadas las curvas de nivel del tiempo de finalización  $w_{CB3}$  en función de los tiempos de ejecución de las tareas  $A$  y  $B$ , se obtiene de forma gráfica la distribución de posibilidad del tiempo de finalización borroso  $W_{CB3}$ , supuesto que la asignación básica conjunta expresada en función de los límites inferiores de los  $\alpha$ -cortes está uniformemente distribuida en la diagonal principal de dominio de los límites inferiores  $U_{I(PA)} \times U_{I(PB)}$ .

La Figura 3.29 muestra la función  $rms$  y los elementos focales conjuntos  $R_i$  del cuerpo de evidencia conjunto  $(F_{P_A \times P_B}, m_{P_A \times P_B})$  cuando se supone que entre  $\alpha_A$  y  $\alpha_B$  hay una correlación positiva perfecta, por lo que los elementos focales están anidados y situados a lo largo de las diagonales definidas por  $\alpha_A = \alpha_B$ .

Para  $p_B = 1.9$  y  $p_A$  entre 0.9 y 1,  $w_{CB3}$  toma el valor 11.9. Para calcular la posibilidad de  $w_{CB3} = 11.9$  hay que integrar la asignación básica de los elementos focales conjuntos que contienen al menos un punto de esa línea:

$$(3.9) \pi_{w_{CB3}}(w_{CB3}) = \int_{w_{CB3} \cap R_j \neq \emptyset} m_{P_A \times P_B}(R_j) \cdot dR_j$$

Como el único elemento focal que contiene algún punto es el externo  $R_n$ , la posibilidad es 0. A partir de  $w_{CB3}=11.9$ , al aumentar el valor de  $w_{CB3}$ , la posibilidad resultante aumenta porque cada vez más elementos focales conjuntos contienen puntos de las líneas  $w_{CB3}=cte$ . Por ejemplo, para  $w_{CB3}=11.95$ , todos los elementos focales conjuntos desde  $R_n$  hasta  $R_i$  contienen algún punto de esa línea, con lo que la posibilidad de ese punto se calcula según (3.10), dando una posibilidad de 0.5 (ya que se integra la asignación básica uniforme en la mitad de la diagonal correspondiente al dominio de los límites inferiores).

$$(3.10) \pi_{w_{CB3}}(11.95) = \int_{R_j=R_n}^{R_i} m_{P_A \times P_B}(R_j) \cdot dR_j$$

Para  $w_{CB3}=12$ , se puede observar que todos los elementos focales conjuntos contienen algún punto de esa línea, con lo que la posibilidad de  $w_{CB3}=12$  es 1. Después se produce una discontinuidad en  $w_{CB3}$  que pasa de valer 12 a 12.9 en  $(p_A, p_B)=(0.9, 2)$ . Como el único elemento focal conjunto que contiene ese punto es el externo  $R_n$ , la posibilidad de  $w_{CB3}=12.9$  es cero. Desde  $w_{CB3}=12.9$  hasta  $w_{CB3}=13$ , a medida que crece  $w_{CB3}$  también crece el valor de posibilidad, porque cada vez más elementos focales contienen puntos de las líneas  $w_{CB3}=cte$ .

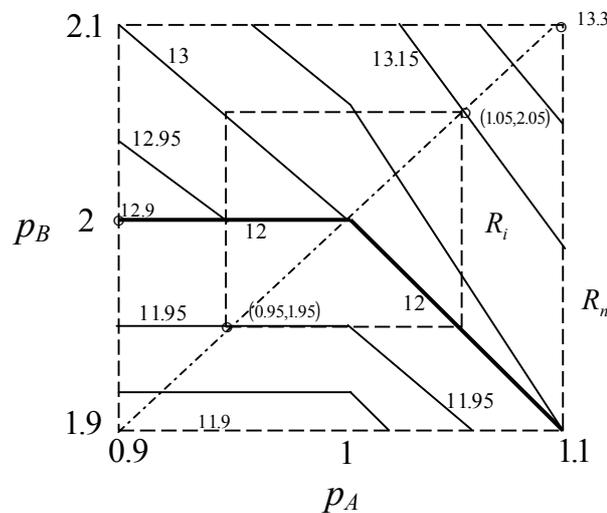


Figura 3.29: Representación de la función rms aplicada a los tiempos de ejecución de A y de B

En el caso de  $w_{CB3}=12.95$  todos los elementos focales desde  $R_n$  hasta  $R_i$  contienen algún punto de esa línea. Estos son los mismos elementos focales que contienen a la línea  $w_{CB3}=11.95$ , con lo que la posibilidad de 12.95 es igual a la posibilidad de 11.95 (3.10). De nuevo en  $w_{CB3}=13$  todos los elementos focales contienen al menos un punto de la línea correspondiente, con lo que la posibilidad es 1. Desde  $w_{CB3}=13$

hasta  $w_{CB_3}=13.3$  la posibilidad vuelve a decrecer. Por ejemplo, para  $w_{CB_3}=13.15$  los elementos focales que la contienen son de nuevo todos los que están entre  $R_n$  y  $R_i$  y por tanto la posibilidad de  $w_{CB_3}=13.15$  es la misma que la de 12.95 y que la de 11.95. En  $w_{CB_3}=13.3$ , situado en  $(p_A, p_B)=(1.1, 2.1)$ , la posibilidad vuelve a ser cero, ya que sólo  $R_n$  contiene ese punto.

La distribución de posibilidad resultante (ver Figura 3.27) coincide con la que se obtiene mediante el principio de extensión.

Una vez obtenida la distribución de posibilidad de  $w_{CB_3}$ , de acuerdo con el método de análisis de planificabilidad borrosa propuesto en el apartado 3.2 (Figura 3.8) se compara con su plazo de ejecución, sea numérico o borroso. Por ejemplo, si  $d_{B_3} = (j-1) \cdot t_B + d_B$ , se tiene que:

- Si el plazo  $d_{B_3}$  está después de 13.3, la posibilidad de que  $B$  termine antes de su plazo es  $\Pi(WC_{B_3} \leq d_{B_3}) = 1$ , y la necesidad es  $N(WC_{B_3} \leq d_{B_3}) = 1$ .
- Si el plazo  $d_{B_3}$  está entre 13 y 13.3 entonces  $\Pi(WC_{B_3} \leq d_{B_3}) = 1$  y  $N(WC_{B_3} \leq d_{B_3}) = 1 - \pi_{w_{CB_3}}(d_{B_3})$
- Si el plazo está entre 12 y 13:  $\Pi(WC_{B_3} \leq d_{B_3}) = 1$ ,  $N(WC_{B_3} \leq d_{B_3}) = 0$
- Si el plazo está entre 11.9 y 12:  $\Pi(WC_{B_3} \leq d_{B_3}) = \pi_{w_{CB_3}}(d_{B_3})$ ,  $N(WC_{B_3} \leq d_{B_3}) = 0$
- Si el plazo está antes de 11.9:  $\Pi(WC_{B_3} \leq d_{B_3}) = 0$ ,  $N(WC_{B_3} \leq d_{B_3}) = 0$

Para obtener la planificabilidad del sistema a partir de la planificabilidad de las tareas hay que calcular las distribuciones de posibilidad de todos los tiempos de finalización y compararlas con sus plazos. El mínimo de todas las posibilidades (respectivamente necesidades) es el valor de posibilidad (respectivamente necesidad) de que el sistema sea planificable. Para las demás ejecuciones de  $B$  se puede hacer de forma similar. Para los tiempos de finalización de la tarea  $A$ , el cálculo es muy sencillo ya que al ser la más prioritaria se tiene que  $WC_{A_j} = (j-1) \cdot t_A + P_A$  con  $1 \leq j \leq 5$ . Para el tiempo de finalización de la tarea  $C$ , hay que aplicar el principio de extensión ya que el método gráfico se complica demasiado para funciones de tres argumentos: el tiempo de finalización de  $C$  depende de los tiempos de ejecución de  $A$ , de  $B$  y de  $C$ , y la representación debería hacerse en tres dimensiones. Recordemos que este método gráfico se propone únicamente como herramienta de estudio, y no como método sistemático de cálculo de la planificabilidad.

En el ejemplo anterior todos los tiempos de ejecución borrosos eran números borrosos triangulares simétricos. La representación gráfica propuesta también permite la obtención de distribuciones de posibilidad a partir de números borrosos triangulares no simétricos, por ello se presenta a continuación otro ejemplo basado en el anterior, donde el tiempo de ejecución de la tarea  $B$ ,  $P_B$ , no es simétrico. Así pues, en este caso el sistema está formado por tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos: 3, 5 y 15 respectivamente, tiempos de ejecución borrosos  $P_A$ ,  $P_B$  y  $P_C$  con funciones de pertenencia triangulares definidos por sus vértices  $P_A=(0.9;1;1.1)$ ,  $P_B=(1.8;1.9;2.1)$  y  $P_C=(2.9;3;3.1)$ .

En la Figura 3.30 aparecen representadas las curvas de nivel  $w_{CB_3} = \text{cte}$  del tiempo de finalización de la tarea  $B$  en su tercera ejecución en función de los tiempos de ejecución de la tarea  $A$  y de la tarea  $B$ .

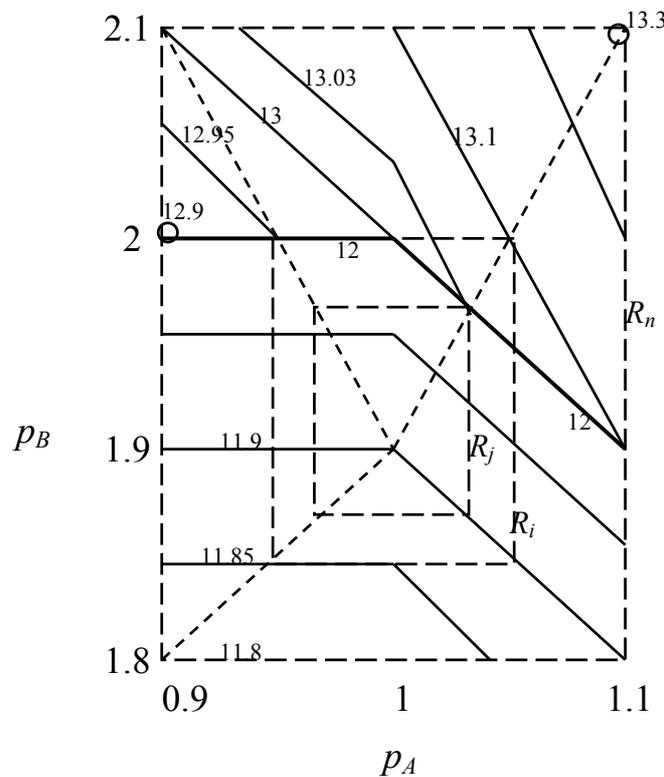


Figura 3.30: Representación de la función rms aplicada a los tiempos de ejecución de A y B. Segundo ejemplo

Esta representación de la Figura 3.30 es similar a la del ejemplo anterior, salvo que se ha ampliado el dominio de  $p_B$  y por tanto también el de  $w_{CB_3}$ . Así, cuando  $p_B=1.8$  y  $p_A \leq 1$  entonces  $w_{CB_3}=11.8$ . Sin embargo los elementos focales conjuntos sí han

cambiado, ya que en este caso son rectángulos anidados cuyos vértices se desplazan a lo largo de las diagonales correspondientes a  $\alpha_A = \alpha_B$ , esto es, la diagonal principal del dominio de los límites inferiores  $[0.9, 1] \times [1.8, 1.9]$  y la diagonal principal del dominio de los límites superiores  $[1, 1.1] \times [1.9, 2.1]$ . En la Figura 3.30 se muestran tres de los elementos focales conjuntos,  $R_n$ ,  $R_i$  y  $R_j$ .

Se puede observar en la Figura 3.30 que el único elemento focal conjunto que contiene algún punto de  $w_{CB_3} = 11.8$  es el externo  $R_n$  por lo que la posibilidad es 0. A partir de ese valor la posibilidad va creciendo hasta que para  $w_{CB_3} = 11.9$  la posibilidad es 1, ya que todos los elementos focales contienen algún punto de dicha línea. Para calcular la posibilidad en un valor intermedio, por ejemplo  $w_{CB_3} = 11.85$ , hay que integrar la asignación básica conjunta en todos los elementos focales desde  $R_n$  hasta  $R_i$  (dando una posibilidad de 0.5, ya que es la mitad de la diagonal correspondiente al dominio de los límites inferiores). Desde  $w_{CB_3} = 11.9$  hasta  $w_{CB_3} = 12$ , la posibilidad va decreciendo. Para calcular la posibilidad de  $w_{CB_3} = 12$ , hay que integrar la asignación básica conjunta en todos los elementos focales desde  $R_n$  hasta  $R_j$  (posibilidad 0.66, correspondiente a  $2/3$  de la diagonal de los límites inferiores).

Para  $w_{CB_3} = 12.9$  la posibilidad vuelve a ser 0 y va creciendo con  $w_{CB_3}$  hasta llegar a  $w_{CB_3} = 12.95$ . Los elementos focales que contienen algún punto de  $w_{CB_3} = 12.95$  son los mismos que contienen algún punto de  $w_{CB_3} = 11.85$ , esto es, desde  $R_n$  hasta  $R_i$ , con lo que la posibilidad de ambos es la misma. Todos los valores de  $w_{CB_3}$  desde 12.95 hasta 13 tienen el mismo valor de posibilidad ya que en todos los casos son los elementos focales desde  $R_n$  hasta  $R_i$  los que intervienen. A partir de  $w_{CB_3} = 13$  la posibilidad vuelve a crecer hasta  $w_{CB_3} = 13.03$ , en cuyo cálculo intervienen los elementos focales desde  $R_n$  hasta  $R_j$ , igual que para  $w_{CB_3} = 12$ , con lo que sus valores de posibilidad coinciden. A partir de  $w_{CB_3} = 13.03$  la posibilidad decrece hasta 0 en  $w_{CB_3} = 13.3$ . Para  $w_{CB_3} = 13.1$ , los elementos focales que intervienen son los situados entre  $R_n$  y  $R_i$ , con lo que la posibilidad de  $w_{CB_3} = 13.1$  es igual que la posibilidad de 11.85 y que la de 12.95.

En la Figura 3.31 se muestra la distribución de posibilidad resultante para  $WC_{B_3}$ .

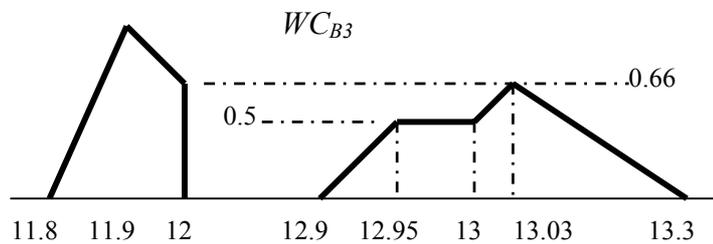


Figura 3.31: Distribución de posibilidad del tiempo de finalización de la tarea B en su tercera ejecución. Segundo ejemplo.

Una vez obtenida la distribución de posibilidad de  $w_{CB3}$  se compara con el plazo correspondiente, sea numérico o borroso. Por ejemplo, si  $d_{B3}$  es numérico, con  $d_{B3} = (j-1) \cdot t_B + d_B$ , y  $j=3$  se tiene que:

- Si el plazo  $d_{B3}$  está después de 13.3, la posibilidad de que  $B$  termine antes de su plazo es  $\Pi(WC_{B3} \leq d_{B3}) = 1$ , y la necesidad es  $N(WC_{B3} \leq d_{B3}) = 1$ .
- Si el plazo  $d_{B3}$  está entre 13.03 y 13.3 entonces  $\Pi(WC_{B3} \leq d_{B3}) = 1$  y  $N(WC_{B3} \leq d_{B3}) = 1 - \pi_{w_{CB3}}(d_{B3})$
- Si el plazo está entre 12 y 13.03:  
 $\Pi(WC_{B3} \leq d_{B3}) = 1$ ,  $N(WC_{B3} \leq d_{B3}) = 1 - 0.66 = 0.33$
- Si el plazo está entre 11.9 y 12:  
 $\Pi(WC_{B3} \leq d_{B3}) = 1$ ,  $N(WC_{B3} \leq d_{B3}) = 1 - \pi_{w_{CB3}}(d_{B3})$
- Si el plazo está entre 11.8 y 11.9:  
 $\Pi(WC_{B3} \leq d_{B3}) = \pi_{w_{CB3}}(d_{B3})$ ,  $N(WC_{B3} \leq d_{B3}) = 0$
- Si el plazo está antes de 11.8:  $\Pi(WC_{B3} \leq d_{B3}) = 0$ ,  $N(WC_{B3} \leq d_{B3}) = 0$

Para obtener la planificabilidad del sistema hay que calcular las distribuciones de posibilidad de todos los tiempos de finalización y compararlas con los plazos correspondientes. El mínimo de todas las posibilidades (respectivamente necesidades) es el valor de posibilidad (respectivamente necesidad) de que el sistema sea planificable.

### 3.4.3. Algoritmo DMS

Como ya se ha descrito en el capítulo 2, el algoritmo de planificación DMS basado en prioridades fijas asigna las prioridades atendiendo al tamaño del plazo, de forma que a menor plazo mayor prioridad. Si los plazos son numéricos la ordenación de las tareas por prioridades es unívoca salvo si existen tareas con plazos iguales. En ese caso el orden relativo de prioridad asignado a las tareas con el mismo plazo no afecta a la planificabilidad del sistema.

En este apartado se presenta un ejemplo de planificación con el algoritmo DMS resuelto primero con parámetros numéricos, que se extiende a tiempos de ejecución borrosos aplicando la representación gráfica propuesta.

**Tiempos de ejecución y plazos numéricos**

Supongamos que se tiene un sistema compuesto por tres tareas periódicas  $A$ ,  $B$  y  $C$ , con periodos  $t_A=3$ ,  $t_B=5$  y  $t_C=15$ , tiempos de ejecución  $p_A=1$ ,  $p_B=1$  y  $p_C=3$ , y plazos  $d_A=3$ ,  $d_B=2$  y  $d_C=11$ . Por tanto la tarea más prioritaria de acuerdo con el algoritmo DMS es la tarea  $B$ , ya que tiene el menor plazo, y la menos prioritaria es la tarea  $C$ . En la Figura 3.32 se muestra el gráfico de planificación de este ejemplo.

Nótese que el algoritmo DMS es una función discontinua definida de los tiempos de ejecución de las tareas a los tiempos de finalización. En este caso  $DMS: \mathbb{R}^3 \rightarrow \mathbb{R}^9$

Como puede comprobarse, todas las tareas terminan su ejecución antes del plazo correspondiente, con lo que este sistema es planificable.

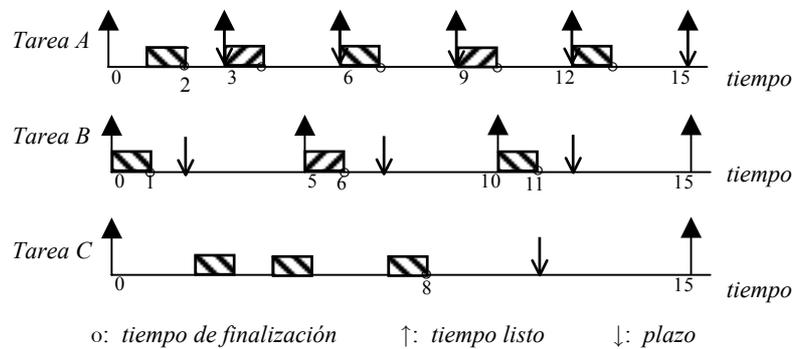


Figura 3.32: Gráfico de planificación de un sistema con algoritmo DMS

Nótese que al igual que ocurría con el algoritmo RMS, cuando existe un instante crítico el peor tiempo de respuesta de cada tarea se produce en la primera ejecución. Por tanto, sería suficiente comparar los tiempos de finalización en la primera ejecución con sus plazos para determinar si el sistema es planificable. En general es preciso analizar el sistema en un hiperperiodo y en el siguiente punto se muestra como ejemplo el cálculo del tiempo de finalización de la tarea  $A$  en su cuarta ejecución.

### Tiempos de ejecución borrosos y plazos numéricos

En este caso, siendo los plazos numéricos, la mayor prioridad es la de la tarea cuyo plazo es menor. El aspecto del gráfico sería similar al del RMS debido a que de nuevo se tienen sumas de números con discontinuidades. Se ha comprobado, como era de esperar, que de nuevo coinciden las distribuciones de posibilidad obtenidas por teoría de la posibilidad con la t-norma del mínimo y las distribuciones obtenidas por teoría de la evidencia con correlación positiva perfecta. El procedimiento para el análisis de planificabilidad es el mismo que el que se ha descrito para el algoritmo RMS.

Como ejemplo, supongamos tres tareas periódicas  $A$ ,  $B$  y  $C$ , con periodos 3, 5 y 15. Los tiempos de ejecución son números borrosos triangulares  $P_A=(0.9;1;1.1)$ ,  $P_B=(0.9;1;1.1)$  y  $P_C=(2.9;3;3.1)$ . Los plazos son 3, 2 y 11. Por tanto, la tarea más prioritaria de acuerdo con el algoritmo DMS es la tarea  $B$  y la menos prioritaria, la tarea  $C$ .

Si se define  $dms$  como la función  $dms: \mathbb{R}^3 \rightarrow \mathbb{R}^1$  que da como resultado el tiempo de finalización de la tarea  $A$  en su cuarta ejecución  $w_{C_{A4}}$ , dicha función se puede extender a tiempos de ejecución borrosos por dos caminos ya que únicamente depende de los tiempos de ejecución de  $A$  y de  $B$ , mediante el principio de extensión con la t-norma  $min$ , o aplicando la representación gráfica del cuerpo de evidencia conjunto con correlación positiva perfecta.

#### 1. Representación de la función real.

En la Figura 3.33 se muestra la representación de la función real  $w_{C_{A4}} = f(p_A, p_B)$ , para distintos valores de  $w_{C_{A4}} = cte$ . El tiempo de ejecución de la tarea  $C$  no afecta a la tarea  $A$  ya que  $C$  es menos prioritaria.

Cuando el tiempo de ejecución de la tarea  $A$  es menor o igual que 1,  $p_A \leq 1$ , el tiempo de finalización de la tarea  $A$  en su cuarta ejecución  $w_{C_{A4}}$  es independiente del tiempo de ejecución de la tarea  $B$ , ya que  $A$  termina antes de que  $B$  comience su tercera ejecución (ver Figura 3.32). Por tanto, el menor valor de  $w_{C_{A4}}$  se da para el menor valor de  $p_A$  que es 0.9, resultando  $w_{C_{A4}} = 9 + 0.9 = 9.9$ . Y el mayor valor de  $w_{C_{A4}}$  se da para  $p_A = 1$ , resultando  $w_{C_{A4}} = 9 + 1 = 10$ . La recta  $w_{C_{A4}} = 10$  es la recta de discontinuidad de la función.

Cuando el tiempo de ejecución de  $A$  es mayor que 1,  $p_A > 1$ , la tarea  $A$  es interrumpida por la tarea  $B$  en su tercera ejecución en el instante 10, retrasando así su finalización,

por lo que  $w_{C_{A4}}$  es mayor. Por tanto, en esta zona  $w_{C_{A4}}$  depende de  $p_A$  y  $p_B$ , y se cumple que  $w_{C_{A4}}=cte$  para  $p_A+p_B=cte$ . El primer valor de  $w_{C_{A4}}$ , en el punto de discontinuidad  $(p_A,p_B)=(1, 0.9)$  es  $w_{C_{A4}} =10.9$ . A partir de ese valor, el tiempo de finalización va creciendo hasta  $w_{C_{A4}} =11.2$  en  $(p_A , p_B)=(1.1, 1.1)$ . Ver Figura 3.33.

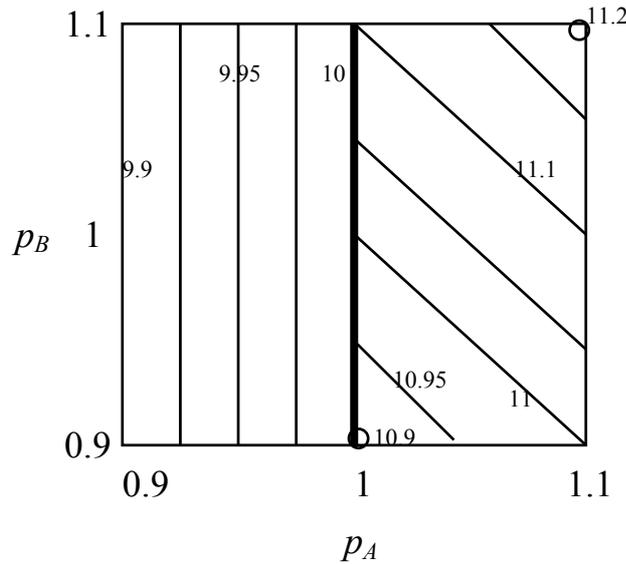


Figura 3.33: Representación de la función real  $w_{C_{A4}} = cte$

2. Representación de los elementos focales y obtención de la distribución de posibilidad.

En la Figura 3.34 se muestra la representación gráfica propuesta en esta tesis para obtener la distribución de posibilidad de  $WC_{A4}$  a partir de los cuerpos de evidencia de  $P_A$  y  $P_B$ . El valor de posibilidad de  $w_{C_{A4}} =9.9$  es 0, ya que únicamente el elemento focal conjunto externo  $R_n$  contiene algún punto de dicha línea, tal y como se ve en la Figura 3.34. A partir de ahí, al crecer  $w_{C_{A4}}$  también crece la posibilidad asociada. Por ejemplo, para  $w_{C_{A4}} =9.95$ , todos los elementos focales desde  $R_n$  hasta  $R_i$  contienen algún punto de dicha línea, con lo que la posibilidad de  $w_{C_{A4}} =9.95$  es la integral de la asignación básica conjunta en dichos elementos focales (posibilidad de 0.5, correspondiente a la mitad de la diagonal).

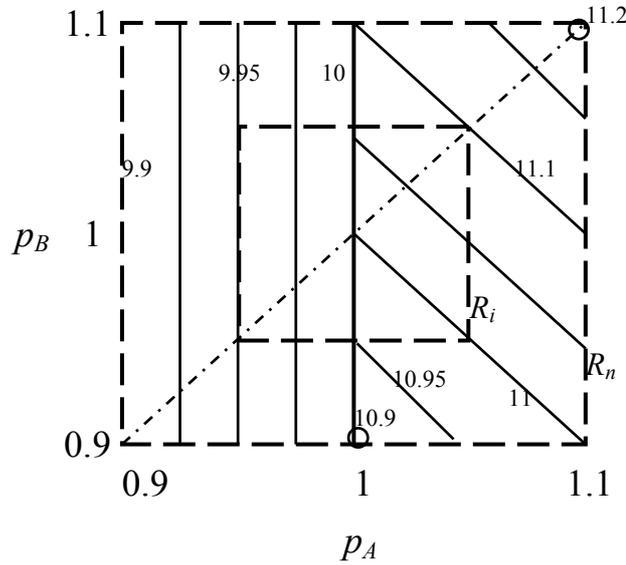


Figura 3.34: Representación de la función dms aplicada a los tiempos de ejecución de A y B

Cuando  $w_{C_{A4}}$  vale 10, la posibilidad se hace 1 ya que todos los elementos focales conjuntos contienen algún punto de  $w_{C_{A4}} = 10$ . El siguiente valor de  $w_{C_{A4}}$  10.9 en el límite, vuelve a tener posibilidad 0 y a partir de ese valor la posibilidad crece con  $w_{C_{A4}}$  hasta llegar a  $w_{C_{A4}} = 11$ , cuya posibilidad es 1. Para un valor situado entre  $w_{C_{A4}} = 10$  y  $w_{C_{A4}} = 11$ , por ejemplo para  $w_{C_{A4}} = 10.95$ , los elementos focales que contienen algún punto de esa línea son los mismos que los de  $w_{C_{A4}} = 9.95$ , desde  $R_n$  hasta  $R_i$ , con lo que la posibilidad de ambos es la misma. A partir de  $w_{C_{A4}} = 11$  la posibilidad decrece, así para  $w_{C_{A4}} = 11.1$ , los elementos focales que contienen alguno de sus puntos son los situados entre  $R_n$  y  $R_i$ , con lo que la posibilidad de  $w_{C_{A4}} = 11.1$  es la misma que la de 9.95 y que la de 10.95. Finalmente, la posibilidad de  $w_{C_{A4}}$  sigue decreciendo hasta llegar a  $w_{C_{A4}} = 11.2$  cuya posibilidad vuelve a ser 0. En la Figura 3.35 se muestra la distribución de posibilidad resultante del tiempo de finalización de A en su cuarta ejecución.

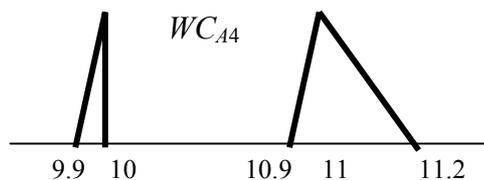


Figura 3.35: Distribución de posibilidad del tiempo de finalización de la tarea A en su cuarta ejecución

Al comparar con el plazo correspondiente, en este caso  $d_{A4}=3 \cdot t_A+d_A=12$ , se tiene que  $\Pi(WC_{A4} \leq d_{A4})=1$  y  $N(WC_{A4} \leq d_{A4})=1$ . Para calcular la planificabilidad del sistema hay que obtener todos los tiempos borrosos y compararlos con sus plazos escogiendo los valores mínimos de posibilidad y necesidad resultantes.

Si los plazos son números borrosos hay que comparar los tiempos de finalización borrosos con los plazos borrosos. La única dificultad puede surgir en la asignación de las prioridades fijas de la tareas, que según DMS depende del tamaño de los plazos. Si los plazos son borrosos, puede haber más de una ordenación de los plazos con posibilidad distinta de cero y con ello más de una asignación de prioridades posible. Tal y como se verá en el capítulo 5, si existe un orden fuerte entre los plazos la asignación dada por dicho orden fuerte es óptima en el sentido de que maximiza la posibilidad y la necesidad de que el sistema sea planificable. Además se puede analizar el sistema teniendo en cuenta todas las asignaciones posibles, es decir, si  $\Pi(D_1 \leq D_2) \neq 0$  y  $\Pi(D_2 \leq D_1) \neq 0$  se puede calcular la posibilidad y la necesidad de que el sistema sea planificable teniendo en cuenta ambas asignaciones, cada una con su posibilidad asociada, denominadas en esta tesis posibilidad y necesidad total. Esta cuestión se presenta en detalle en el capítulo 6.

### 3.5. Cálculo de planificabilidad borrosa mediante aproximaciones de los tiempos de finalización

La representación gráfica propuesta es una herramienta útil para obtener conclusiones acerca del cálculo de los tiempos de finalización borrosos, pero no es un método sistemático de cálculo, ya que se complica cuando los argumentos borrosos no son números triangulares (la asignación básica en función de los límites inferiores de los  $\alpha$ -cortes deja de ser una distribución uniforme), o cuando la función depende de tres argumentos (en cuyo caso habría que representar la función en tres dimensiones).

Por otro lado, como ya se ha mencionado, para aplicar el principio de extensión muestreando los soportes de los conjuntos borrosos hace falta realizar muchas simulaciones de la ejecución del sistema en un hiperperiodo. Por ejemplo, para tres tiempos de ejecución borrosos  $P_A$ ,  $P_B$  y  $P_C$  con  $n_A$ ,  $n_B$  y  $n_C$  número de muestras respectivamente, el número de simulaciones sería  $n_A \times n_B \times n_C$  y depende, por tanto, del número de tareas.

Supongamos que se busca un resultado de planificabilidad con una determinada precisión en  $\alpha$ , es decir, una precisión determinada  $r$  en los valores de posibilidad de las distribuciones de posibilidad de los tiempos de finalización. Para ello se divide el intervalo  $(0,1]$  en  $1/r$  intervalos, de forma que el número de muestras de cada tiempo de ejecución es  $n_A = (2/r)-1 = n_B = n_C = n$ . Para aplicar el principio de extensión hacen falta  $n^3$  simulaciones. En general, para  $n\_tar$  número de tareas las simulaciones necesarias para aplicar el principio de extensión con una resolución  $r$  en  $\alpha$  es  $n^{n\_tar}$ , siendo  $n=(2/r)-1$ .

La aplicación del principio de extensión se puede hacer inviable debido al gran número de simulaciones que son necesarias. Por tanto interesa buscar un método más rápido para realizar el análisis de planificabilidad.

### 3.5.1. Definición de aproximación válida de un conjunto borroso para el análisis de planificabilidad

El resultado  $WC_{ij}$  de la extensión de  $wc_{ij}=f(p_h)$  con  $h=1..n\_tar$  a argumentos borrosos mediante la aplicación del principio de extensión con la t-norma mínimo, es lo que a partir de ahora se denominará resultado “exacto” de la extensión y se tomará como referencia en las extensiones borrosas que se propongan del análisis de planificabilidad. Como en muchas ocasiones la aplicación del principio de extensión se hace inviable debido a su coste computacional, uno de los objetivos de esta tesis es encontrar aproximaciones  $WC_{ij}^*$  del conjunto borroso  $WC_{ij}$  obtenido mediante el principio de extensión, que sean válidas para el cálculo de la planificabilidad borrosa del sistema. Además, esas aproximaciones válidas deben ser más rápidas de obtener que la aplicación del principio de extensión.

Para ello, en primer lugar es necesario definir lo que se consideran aproximaciones válidas del tiempo de finalización de una tarea  $i$  en su ejecución  $j$   $WC_{ij}$ . Se dice que un tiempo de finalización borroso  $WC_{ij}^*$  es una aproximación válida de otro tiempo de finalización borroso  $WC_{ij}$  cuando la posibilidad y la necesidad de que  $WC_{ij}^*$  sea menor o igual que el plazo de la tarea  $i$   $D_i$  es igual que la posibilidad y la necesidad de que el tiempo de finalización  $WC_{ij}$  sea menor o igual que dicho plazo. Si el tiempo de finalización borroso  $WC_{ij}$  es exacto, la posibilidad y necesidad de que el sistema sea planificable utilizando aproximaciones válidas  $WC_{ij}^*$  para todos los tiempos de finalización borrosos exactos en un hiperperiodo son la posibilidad y necesidad exactas de planificabilidad del sistema. Es decir:

$WC_{ij}^*$  es una aproximación válida de  $WC_{ij}$  si:

$$\Pi( WC_{ij}^* \leq D_{ij} ) = \Pi( WC_{ij} \leq D_{ij} ) \text{ y } N( WC_{ij}^* \leq D_{ij} ) = N( WC_{ij} \leq D_{ij} )$$

con lo que se cumple que  $\Pi_{planif} = \min\{\Pi( WC_{ij}^* \leq D_{ij} )\} = \min\{\Pi( WC_{ij} \leq D_{ij} )\}$  y  $N_{planif} = \min\{N( WC_{ij}^* \leq D_{ij} )\} = \min\{N( WC_{ij} \leq D_{ij} )\}$

Tal y como ya se ha visto, los conjuntos borrosos  $WC_{ij}$  son, en general, no convexos (ver Figura 3.27), debido a que el algoritmo de planificación es una función discontinua.

De acuerdo con lo presentado en el anexo B, se puede afirmar que la comparación de un conjunto borroso no convexo con otro número, clásico o borroso, da el mismo resultado de posibilidad y necesidad que la comparación de una aproximación, si se define la aproximación como un conjunto borroso con los mismos límites inferiores y superiores para cada  $\alpha$ -corte que el conjunto de partida. Por tanto, cualquier conjunto borroso  $WC_{ij}^*$  que tenga los límites exactos de los  $\alpha$ -cortes, es una aproximación válida de  $WC_{ij}$  en este contexto.

Recordemos que la función de pertenencia  $\mu_{WC_{ij}}(wc_{ij}): \mathbb{R} \rightarrow [0,1]$  del tiempo de finalización borroso  $WC_{ij}$  se puede representar en términos de sus  $\alpha$ -cortes ([Dubois, Prade, 1989a], [Dubois, Prade, 1986b]):

$$\mu_{WC_{ij}}(wc_{ij}) = \sup \{ \alpha \in (0,1] \mid wc_{ij} \in WC_{ij \alpha} \}$$

donde  $WC_{ij \alpha}$  son los  $\alpha$ -cortes de  $WC_{ij}$ , esto es:

$$WC_{ij \alpha} = \{ wc_{ij} \mid \mu_{WC_{ij}}(wc_{ij}) \geq \alpha \}.$$

El límite inferior del  $\alpha$ -corte de  $WC_{ij}$  se define como:

$$\underline{wc}_{ij}(\alpha) = \inf_{x \mid \mu_{WC_{ij}}(x) \geq \alpha} x$$

Y el límite superior del  $\alpha$ -corte de  $WC_{ij}$  se define como:

$$\overline{wc}_{ij}(\alpha) = \sup_{x \mid \mu_{WC_{ij}}(x) \geq \alpha} x$$

Si el tiempo de finalización es un conjunto borroso convexo, entonces el intervalo  $[\underline{wc}_{ij}(\alpha), \overline{wc}_{ij}(\alpha)] = [\underline{wc}_{ij}, \overline{wc}_{ij}](\alpha)$  es el  $\alpha$ -corte de  $WC_{ij}$ . Como en general  $WC_{ij}$  no es convexo, el  $\alpha$ -corte de  $WC_{ij}$  está incluido en el citado intervalo (ver Figura 3.36).

En la Figura 3.36 se muestran los límites inferiores y superiores de los  $\alpha$ -cortes de  $WC_{ij}$ .

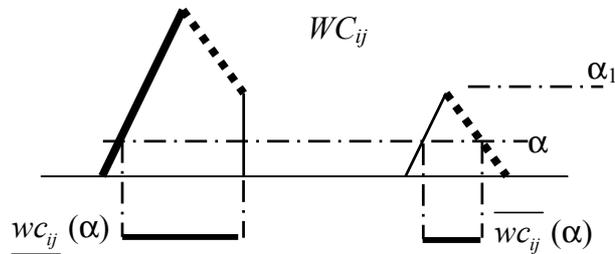


Figura 3.36: Límites inferiores (-) y superiores (...) de los  $\alpha$ -cortes de  $WC_{ij}$

Así, cualquier conjunto borroso con los límites de los  $\alpha$ -cortes exactos  $\underline{wc}_{ij}(\alpha)$  y  $\overline{wc}_{ij}(\alpha)$ , es una aproximación válida de  $WC_{ij}$ . En la Figura 3.37 se muestra un ejemplo de aproximación válida  $WC_{ij}^*$  en línea continua.

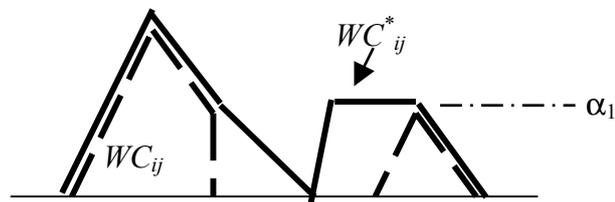


Figura 3.37: Aproximación válida  $WC_{ij}^*$  de  $WC_{ij}$

Una de las posibles aproximaciones válidas de  $WC_{ij}$  es el conjunto borroso convexo definido por los límites de los  $\alpha$ -cortes exactos  $\underline{wc}_{ij}(\alpha)$  y  $\overline{wc}_{ij}(\alpha)$ , tal que el  $\alpha$ -corte de la aproximación  $WC_{ij}^*$  es:

$$\left[ \underline{wc}_{ij}^*(\alpha), \overline{wc}_{ij}^*(\alpha) \right] = \left[ \underline{wc}_{ij}^*, \overline{wc}_{ij}^* \right](\alpha) = \left[ \underline{wc}_{ij}(\alpha), \overline{wc}_{ij}(\alpha) \right]$$

Esta aproximación convexa se muestra en la Figura 3.38.

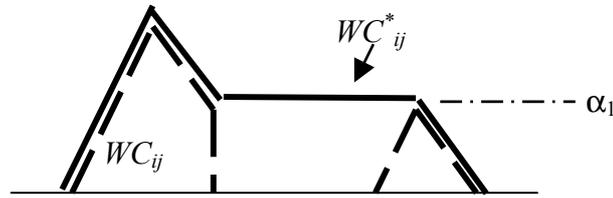


Figura 3.38: Aproximación válida convexa  $WC_{ij}^*$  de  $WC_{ij}$

### 3.5.2. Aproximación válida convexa de los tiempos de finalización borrosos

Tras definir el concepto de aproximación válida de un conjunto borroso en el contexto del análisis de planificabilidad de un sistema, en este apartado se muestra que es posible calcular una aproximación válida del tiempo de finalización borroso de una tarea aplicando cálculo de intervalos y que es más rápida de obtener que los tiempos borrosos exactos calculados mediante la aplicación del principio de extensión.

En efecto, sea el sistema del ejemplo anterior formado por tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos: 3, 5 y 15 respectivamente. Los tiempos de ejecución son números borrosos  $P_A$ ,  $P_B$  y  $P_C$  de forma triangular definidos por sus vértices:  $P_A=(0.9;1;1.1)$ ,  $P_B=(1.9;2;2.1)$  y  $P_C=(2.9;3;3.1)$ .

Como se vio previamente, la distribución de posibilidad exacta del tiempo de finalización de la tarea  $B$  en su tercera ejecución  $WC_{B3}$  se puede obtener mediante la representación gráfica de la combinación de los cuerpos de evidencia, con correlación positiva perfecta entre las  $\alpha$ . Esta distribución de posibilidad aparece en la Figura 3.27. Por otro lado, en la Figura 3.39 se utiliza la representación gráfica propuesta en esta tesis con el fin de obtener una aproximación válida del tiempo de finalización borroso exacto.

Con el fin de ilustrar mejor la relación existente entre los conjuntos borrosos  $P_A$  y  $P_B$ , la correlación entre  $\alpha_A$  y  $\alpha_B$  y los elementos focales del cuerpo de evidencia conjunto, en la Figura 3.39 la representación gráfica propuesta, aplicada en este caso a la obtención de  $WC_{B3}$ , se ha completado con la representación de la correlación positiva perfecta en el cuadrado unidad  $[0,1]^2$  y con los tiempos de ejecución borrosos  $P_A$  y  $P_B$ .

Así, la correlación positiva perfecta entre  $\alpha_A$  y  $\alpha_B$  viene representada en la Figura 3.39, en el dominio de  $\alpha_A$  y  $\alpha_B$ ,  $[0,1]^2$ , como la diagonal principal  $\alpha_A=\alpha_B$ .

Proyectando cada valor de  $\alpha$  en los conjuntos borrosos  $P_A$  y  $P_B$ , se obtienen dos valores de  $p_A$  y dos valores de  $p_B$  cuya posibilidad es  $\alpha$ . Esos son precisamente los límites inferiores y superiores de cada  $\alpha$ -corte de  $P_A$  y  $P_B$ . Al proyectar dichos límites de los  $\alpha$ -cortes de  $P_A$  y  $P_B$  en el dominio de los soportes  $]P_A[\times]P_B[$  (ver la Figura 3.39), los pares resultantes  $(p_A, p_B)$  se distribuyen a lo largo de las diagonales de  $]P_A[\times]P_B[$ .

Concretamente, los pares formados por los límites inferiores  $(\underline{p}_A(\alpha), \underline{p}_B(\alpha))$ , con el mismo valor de  $\alpha$ , se sitúan en la diagonal principal del dominio de los límites inferiores, que en el ejemplo es  $[0.9, 1] \times [1.9, 2]$ . Los pares formados por los límites superiores  $(\overline{p}_A(\alpha), \overline{p}_B(\alpha))$ , con el mismo valor de  $\alpha$ , se sitúan en la diagonal principal del dominio de los límites superiores, que en el ejemplo es  $[1, 1.1] \times [2, 2.1]$ .

Además, en la Figura 3.39 se representa la función  $wc_{B3} = rms(p_A, p_B)$  en el dominio de los soportes  $]P_A[\times]P_B[$ . Tal y como se describió anteriormente,  $wc_{B3}$  crece de 11.9 a 12, donde existe una discontinuidad, y el siguiente valor se sitúa en  $(p_A, p_B) = (0.9, 2)$  y es  $wc_{B3} = 12.9$ . A partir de ese valor,  $wc_{B3}$  continúa creciendo hasta 13.3 en  $(p_A, p_B) = (1.1, 2.1)$ .

La posibilidad de todos los valores de  $wc_{B3} \in [11.9, 12]$  se puede calcular como la posibilidad de  $(p_A, p_B)$ , que es el punto de corte entre la línea  $wc_{B3} = cte$  y  $\alpha_A = \alpha_B$  (diagonal principal). Esto se cumple porque los elementos focales conjuntos que contienen algún punto de la línea  $wc_{B3} = cte$  son los mismos elementos focales que contienen al citado punto  $(p_A, p_B)$ . Y la posibilidad de ese punto  $(p_A, p_B)$  es precisamente  $\pi_{wc_{B3}}(wc_{B3}) = \alpha_A = \alpha_B$ . Ese valor de  $\alpha$  es la posibilidad  $\pi_{p_A}(p_A)$  de un tiempo de ejecución  $p_A$  de la tarea  $A$ , y la posibilidad  $\pi_{p_B}(p_B)$  de un tiempo de ejecución  $p_B$  de  $B$ . Por tanto se tiene:

$$\pi_{wc_{B3}}(wc_{B3}) = \pi_{P_A \times P_B}((p_A, p_B)) = \pi_{p_A}(p_A) = \pi_{p_B}(p_B)$$

A su vez,  $p_A$  es el límite inferior del  $\alpha$ -corte de  $P_A$  de valor  $\alpha_A$ ,  $p_A = \underline{p}_A(\alpha_A)$ . Y también se verifica para el tiempo de ejecución de  $B$  y el tiempo de finalización de  $B$  en su tercera ejecución,  $p_B = \underline{p}_B(\alpha_B)$  y  $wc_{B3} = \underline{wc}_{B3}(\alpha)$ , con  $\alpha = \alpha_A = \alpha_B$

Por tanto, el límite inferior del  $\alpha$ -corte de  $wc_{B3}$  se puede calcular operando mediante la función  $rms$  el límite inferior del  $\alpha$ -corte de  $P_A$  con el límite inferior del  $\alpha$ -corte de  $P_B$ , esto es,  $\underline{wc}_{B3}(\alpha) = rms(\underline{p}_A(\alpha), \underline{p}_B(\alpha))$

Con los límites superiores se puede obtener de forma similar la posibilidad de todos los valores de  $WC_{B3} \in [13, 13.3]$ , como la posibilidad del punto de corte  $(p_A, p_B)$  entre la línea  $wc_{B3} = \text{cte}$  y  $\alpha_A = \alpha_B$  (diagonal principal del dominio de los límites superiores  $[1,1.1] \times [2, 2.1]$ ). Y la posibilidad de ese punto  $(p_A, p_B)$  es  $\pi_{WC_{B3}}(wc_{B3}) = \alpha_A = \alpha_B$ . Ese valor de  $\alpha$  es la posibilidad  $\pi_{P_A}(p_A)$  de un tiempo de ejecución  $p_A$  de la tarea  $A$ , y la posibilidad  $\pi_{P_B}(p_B)$  de un un tiempo de ejecución  $p_B$  de la tarea  $B$ . Por tanto se tiene:

$$\pi_{WC_{B3}}(wc_{B3}) = \pi_{P_A \times P_B}((p_A, p_B)) = \pi_{P_A}(p_A) = \pi_{P_B}(p_B)$$

A su vez,  $p_A$  es el límite superior del  $\alpha$ -corte de  $P_A$  de valor  $\alpha_A$   $p_A = \overline{p_A}(\alpha_A)$ . Y además,  $p_B = \overline{p_B}(\alpha_B)$  y  $wc_{B3} = \overline{wc_{B3}}(\alpha)$ , con  $\alpha = \alpha_A = \alpha_B$

Por tanto, el límite superior del  $\alpha$ -corte del tiempo de finalización se puede obtener directamente operando los límites superiores de los  $\alpha$ -cortes de  $P_A$  y de  $P_B$ , esto es,

$$\overline{wc_{B3}}(\alpha) = \text{rms}(\overline{p_A}(\alpha), \overline{p_B}(\alpha))$$

Tomemos como ejemplo, en la Figura 3.39, el cálculo del valor de posibilidad de  $wc_{B3} = 11.95$ , y de  $wc_{B3} = 13.15$ . Para calcular la posibilidad de  $wc_{B3} = 11.95$  hay que integrar la asignación básica conjunta de todos los elementos focales desde  $R_n$  hasta  $R_i$  ya que contienen al menos un punto de la línea  $wc_{B3} = 11.95$ . Este valor de posibilidad coincide con el del par  $(p_A, p_B) = (0.95, 1.95)$  situado en la diagonal principal del dominio de los límites inferiores, con lo que

$$\pi_{WC_{B3}}(11.95) = \pi_{P_A}(0.95) = \pi_{P_B}(1.95) = 0.5$$

donde 11.95, 0.95 y 1.95 son los límites inferiores de los  $\alpha$ -cortes de  $WC_{B3}$ ,  $P_A$  y  $P_B$  con  $\alpha = 0.5$ . Es decir,  $\underline{wc_{B3}}(0.5) = 11.95$ ,  $\underline{p_A}(0.5) = 0.95$  y  $\underline{p_B}(0.5) = 1.95$ . Este procedimiento se puede extender para obtener la posibilidad  $\alpha$  de todo  $wc_{B3} \in [11.9, 12]$ .

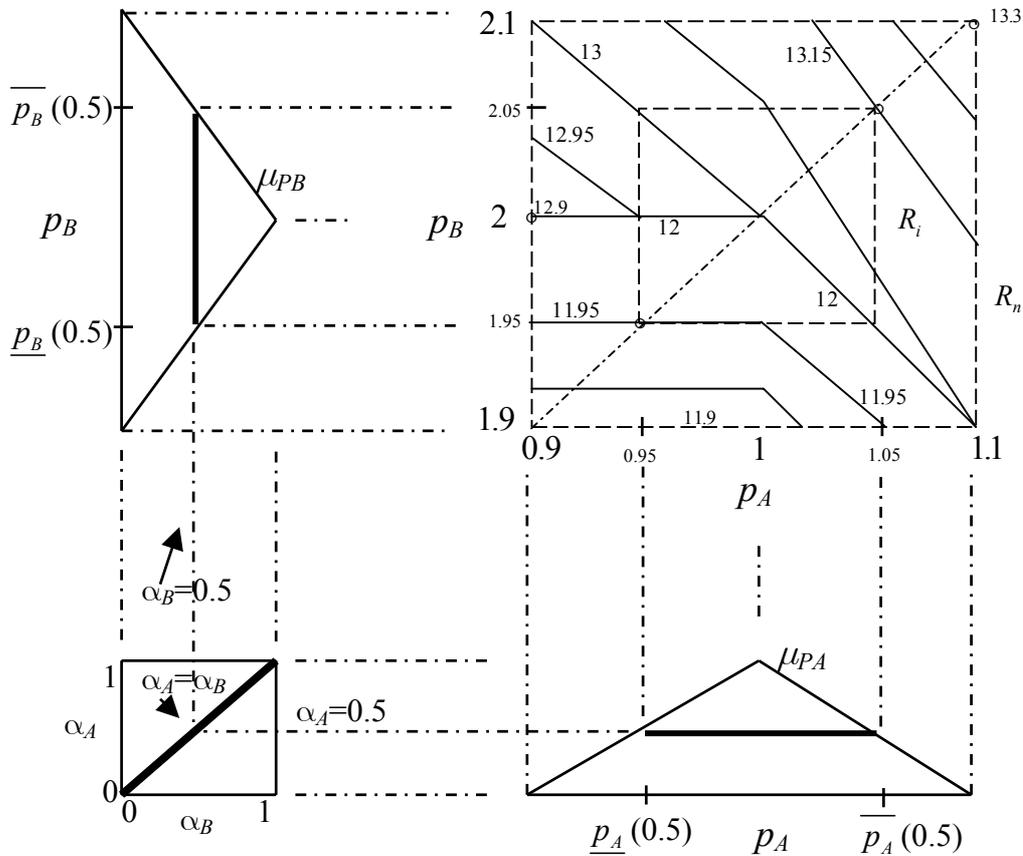


Figura 3.39: Obtención de  $WC_{B3}^*$  mediante operación de intervalos a partir de la representación propuesta

Como los mismos elementos focales conjuntos de  $R_n$  a  $R_i$  contienen también al menos un punto de la recta  $wc_{B3} = 13.15$ , se tiene que:

$$\pi_{WC_{B3}}(13.15) = \pi_{WC_{B3}}(11.95)$$

Esta posibilidad coincide con la del par  $(p_A, p_B) = (1.05, 2.05)$  situado en la diagonal principal del dominio de los límites superiores, con lo que

$$\pi_{WC_{B3}}(13.15) = \pi_{WC_{B3}}(11.95) = \pi_{p_A}(1.05) = \pi_{p_B}(2.05) = 0.5$$

donde 13.15, 1.05 y 2.05 son los límites superiores de los  $\alpha$ -cortes de  $WC_{B3}$ ,  $p_A$  y  $p_B$  con  $\alpha = 0.5$ . Es decir,  $\overline{wc_{B3}}(0.5) = 13.15$ ,  $\underline{p_A}(0.5) = 1.05$  y  $\overline{p_B}(0.5) = 2.05$ . Así, se puede obtener la posibilidad  $\alpha$  de todo  $wc_{B3} \in [13, 13.3]$ .

Los resultados anteriores muestran que no es necesario calcular todas las combinaciones de  $(p_A, p_B)$  para calcular los límites inferiores (respectivamente superiores) de la distribución de posibilidad de  $WC_{B3}$ , sino que basta con operar los límites inferiores entre sí (respectivamente superiores) de los  $\alpha$ -cortes de  $P_A, P_B$  con el mismo valor de  $\alpha$ . Sin embargo, de esta forma no se obtiene la parte no convexa de la distribución de  $WC_{B3}$  (ver Figura 3.27), ya que para ello habría que tener en cuenta que no todos los puntos del intervalo  $[\underline{wc}_{B3}, \overline{wc}_{B3}](\alpha)$  pertenecen al  $\alpha$ -corte de  $WC_{B3}$ . En nuestro ejemplo, para  $\alpha=0.5$  es posible deducir gráficamente que los subintervalos válidos dentro de  $[11.95, 13.15]$  son  $[11.95, 12]$  y  $[12.95, 13.15]$ . Sin esta corrección, se obtiene un valor de posibilidad de 1 para los  $wc_{B3}$  situados entre 12 y 13, valor, que como hemos visto antes, no es el que se obtiene mediante el principio de extensión.

En conclusión, la aplicación del procedimiento descrito ha permitido mostrar que es posible obtener una aproximación válida convexa de los tiempos de finalización, a partir de los límites de los  $\alpha$ -cortes de los tiempos de ejecución. La obtención de estas aproximaciones  $WC_{ij}^*$  mediante análisis de intervalos es mucho más rápida que la obtención de los tiempos borrosos exactos. La aproximación válida convexa  $WC_{B3}^*$  del tiempo de finalización borroso de  $B$  en su tercera ejecución cuyos límites de los  $\alpha$ -cortes son exactos,

$$[\underline{wc}_{B3}^*(\alpha), \overline{wc}_{B3}^*(\alpha)] = [\underline{wc}_{B3}(\alpha), \overline{wc}_{B3}(\alpha)]$$

se puede calcular mediante análisis de intervalos como:

$$\underline{wc}_{B3}(\alpha) = rms(\underline{p}_A(\alpha), \underline{p}_B(\alpha)) \quad \overline{wc}_{B3}(\alpha) = rms(\overline{p}_A(\alpha), \overline{p}_B(\alpha))$$

La obtención de  $WC_{B3}^*$  mediante cálculo de intervalos se ilustra en la Figura 3.40.

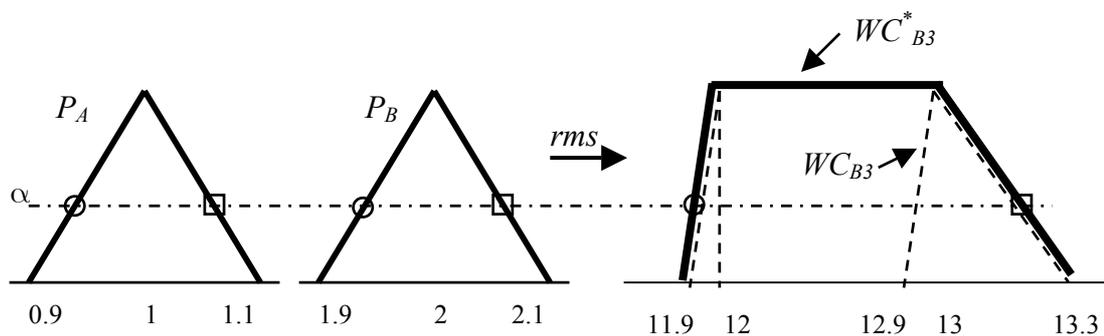


Figura 3.40: Obtención de  $WC_{B3}^*$  mediante operación de intervalos

Como se trata de un aproximación válida, la comparación de  $WC_{B3}^*$  con el plazo de ejecución da el mismo resultado que la comparación del tiempo borroso exacto  $WC_{B3}$  con ese plazo, independientemente de si el plazo es un número clásico (“crisp”) o borroso (ver anexo B). En el caso de que el plazo sea numérico  $d_{B3}$ , es fácil comprobar que el resultado coincide con lo obtenido en el primer ejemplo del apartado 3.4.2, donde se calculaba el tiempo borroso exacto  $WC_{B3}$  y se comparaba con el plazo numérico. Así, la posibilidad y necesidad de que la tarea respete su plazo de ejecución es:

- Si el plazo  $d_{B3}$  está después de 13.3, la posibilidad de que  $B$  termine antes de su plazo es  $\Pi(WC_{B3}^* \leq d_{B3}) = 1$ , y la necesidad es  $N(WC_{B3}^* \leq d_{B3}) = 1$ .
- Si el plazo  $d_{B3}$  está entre 13 y 13.3 entonces  
 $\Pi(WC_{B3}^* \leq d_{B3}) = 1$  y  $N(WC_{B3}^* \leq d_{B3}) = 1 - \pi_{WC_{B3}^*}(d_{B3}) = 1 - \pi_{WC_{B3}}(d_{B3})$
- Si el plazo está entre 12 y 13:  $\Pi(WC_{B3}^* \leq d_{B3}) = 1$ ,  $N(WC_{B3}^* \leq d_{B3}) = 0$
- Si el plazo está entre 11.9 y 12:  
 $\Pi(WC_{B3}^* \leq d_{B3}) = \pi_{WC_{B3}^*}(d_{B3}) = \pi_{WC_{B3}}(d_{B3})$ ,  $N(WC_{B3}^* \leq d_{B3}) = 0$
- Si el plazo está antes de 11.9:  $\Pi(WC_{B3}^* \leq d_{B3}) = 0$ ,  $N(WC_{B3}^* \leq d_{B3}) = 0$

Recordemos que para calcular la planificabilidad del sistema a partir de muestras del soporte de los conjuntos borrosos, si se utilizan aproximaciones válidas sólo se precisa realizar  $n$  simulaciones del sistema frente a las  $n^{n\_tar}$  necesarias para aplicar el principio de extensión.

Además, si el objetivo es encontrar el plazo de ejecución que puede ser respetado con un determinado valor de posibilidad  $\Pi_1$ , hace falta realizar una única simulación del sistema con los límites inferiores  $\underline{p}_i(\alpha)$ ,  $\forall i=1..n\_tar$ , con  $\alpha=\Pi_1$ . El resultado de esa simulación son los tiempos de finalización  $wc_{ij}$  de todas las tareas  $i$  en todas sus iteraciones  $j$  (que son los límites inferiores de los  $\alpha$ -cortes de  $WC_{ij}$ ). Cogiendo el peor caso para cada tarea, se obtendrá el plazo de cada tarea que puede ser respetado con posibilidad  $\Pi_1$ . Esto es, para cada tarea  $i$ ,  $d_i = \max\{wc_{ij} - (j-1) \cdot t_i\} \forall j$ .

Del mismo modo, sólo se requiere una simulación para hallar el valor del plazo que puede ser respetado con una necesidad  $N_1$ : obteniendo todos los tiempos de finalización de las tareas  $wc_{ij}$  para los límites superiores  $\overline{p}_i(\alpha)$ ,  $\forall i=1..n\_tar$ , con  $\alpha=1-N_1$ . De nuevo, para cada tarea  $i$ ,  $d_i = \max\{wc_{ij} - (j-1) \cdot t_i\} \forall j$ .

Como conclusión de este capítulo la Figura 3.41 muestra el proceso de análisis de planificabilidad borrosa mediante aproximaciones válidas de los tiempos de finalización borrosos en comparación con la aplicación del principio de extensión.

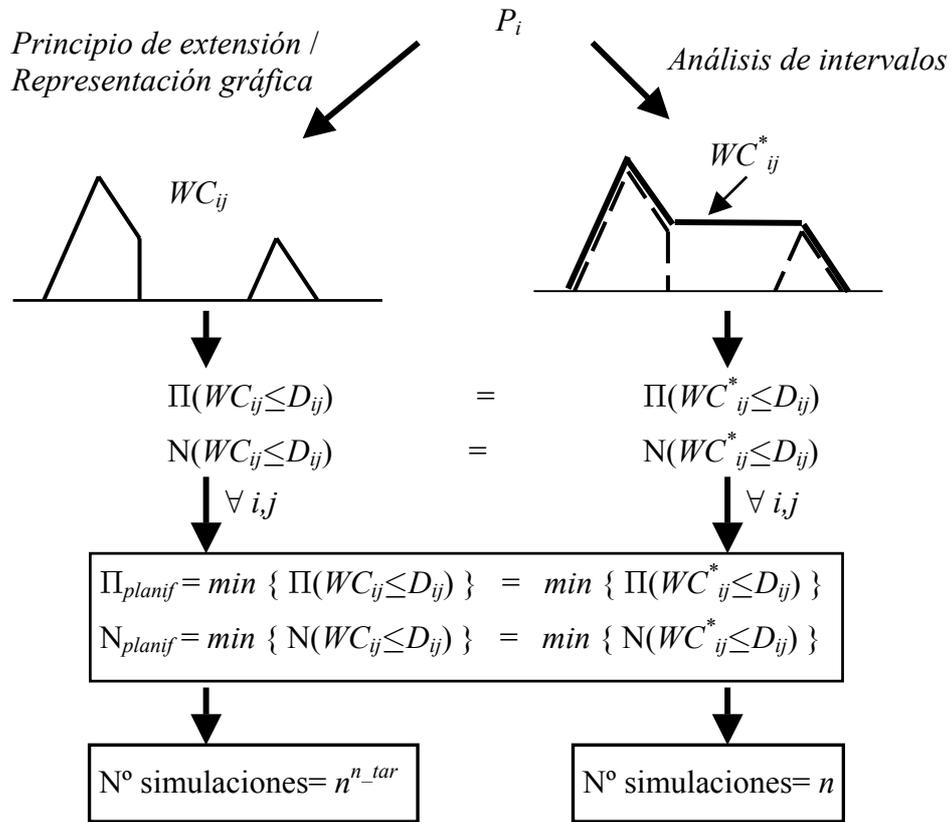


Figura 3.41: Proceso del análisis borroso de planificabilidad exacto o mediante tiempos de finalización aproximados

La representación del tiempo de finalización de una tarea en función de los tiempos de ejecución permite suponer que se pueden generalizar los resultados de operar con los intervalos, por el carácter no decreciente de la función en los parámetros. Debido a la naturaleza del problema, los tiempos de finalización de las tareas se pueden expresar como sumas de tiempos con discontinuidades. El siguiente objetivo es generalizar estos resultados, por lo que en el próximo capítulo se presenta la condición que debe cumplir cualquier función, continua o discontinua, para poder obtener una aproximación válida de su extensión borrosa mediante análisis de intervalos. Además, se propone otra aproximación válida para ese tipo de funciones, la que se obtiene mediante aritmética borrosa.

La representación gráfica propuesta en esta tesis, que se ha utilizado para concluir que se pueden obtener aproximaciones válidas convexas mediante cálculo de intervalos, sólo permite la representación de funciones de dos argumentos y sólo resulta fácil de interpretar cuando los argumentos se modelan como números borrosos triangulares. Para comprobar este resultado en casos no representables se realizaron simulaciones por intervalos con correlación positiva perfecta entre  $\alpha$  y simulaciones aplicando el principio de extensión de la teoría de la posibilidad, con la t-norma mínimo en los casos:

1. Tiempo de finalización que depende de tres tiempos de ejecución
2. Tiempos de ejecución borrosos no triangulares

### 1. Tiempo de finalización que depende de tres tiempos de ejecución

Sea el sistema del ejemplo anterior formado por tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos 3, 5 y 15 respectivamente y tiempos de ejecución son borrosos triangulares  $P_A=(0.9;1;1.1)$ ,  $P_B=(1.9;2;2.1)$  y  $P_C=(2.9;3;3.1)$ .

En la Figura 3.42 se muestran superpuestos los resultados del tiempo de finalización de la tarea  $C$  en su primera ejecución por ambos caminos, el principio de extensión  $WC_{C1}$  y simulación por intervalos  $WC_{C1}^*$ , y tal y como se puede observar, coinciden.

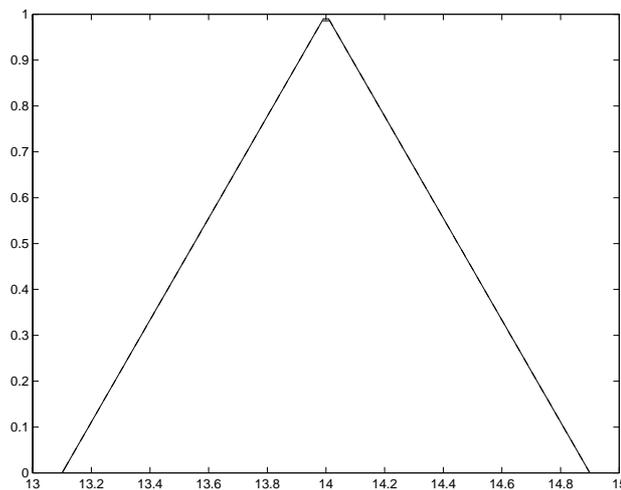


Figura 3.42:  $WC_{C1}$  obtenido mediante simulación del principio de extensión (-) y  $WC_{C1}^*$  obtenido mediante análisis de intervalos (...)

2. Tiempos de ejecución borrosos no triangulares

Sea un sistema formado por tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos 3, 5 y 15 respectivamente y tiempos de ejecución son borrosos lineales a tramos definidos por los vértices  $P_A=(0/0.9; 1/0.95; 1/1; 0.3/1.05; 0/1.1)$ ,  $P_B=(0/1.75; 0.5/1.85; 1/1.9; 0.5/2.05; 0/2.1)$  y  $P_C=(0/2.9; 0.2/2.97; 1/3; 0.1/3.02; 0/3.1)$ , donde el primer elemento  $\alpha$  de cada par  $\alpha/b$  es el grado de pertenencia del segundo elemento  $b$  al conjunto borroso.

En la Figura 3.43 se muestra el tiempo de finalización borroso exacto  $WC_{B3}$  y la aproximación convexa obtenida mediante simulación por intervalos  $WC_{B3}^*$ . Nótese que los límites de los  $\alpha$ -cortes coinciden.

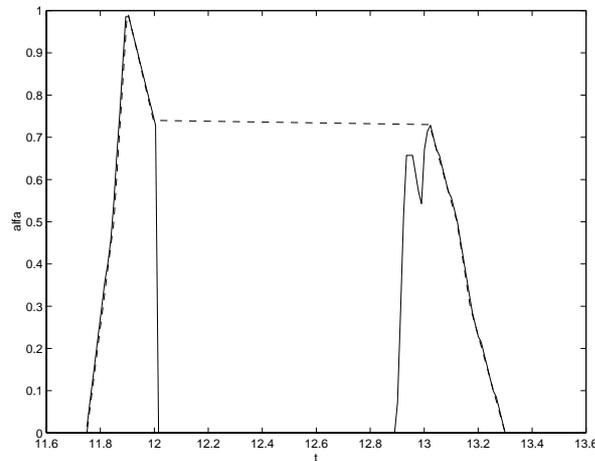


Figura 3.43:  $WC_{B3}$  obtenido mediante simulación del principio de extensión (-) y  $WC_{B3}^*$  obtenido mediante análisis de intervalos (- -), con tiempos de ejecución no triangulares

También se han realizado simulaciones con el algoritmo de planificación DMS para comprobar que las aproximaciones que se obtienen mediante simulación por intervalos tienen los mismos límites inferiores y superiores de los  $\alpha$ -cortes que el resultado de aplicar el principio de extensión mediante simulación. Así, para un sistema con tres tareas periódicas  $A$ ,  $B$  y  $C$ , con periodos 3, 5 y 15, tiempos de ejecución borrosos triangulares  $P_A=(0.9;1;1.1)$ ,  $P_B=(0.9;1;1.1)$  y  $P_C=(2.9;3;3.1)$  y plazos 3, 2 y 11, en la Figura 3.44 se muestra el tiempo de finalización borroso de  $A$  en su cuarta ejecución, por ambos métodos. Se puede ver que la distribución obtenida mediante simulación del principio de extensión coincide con la previamente obtenida gráficamente en la Figura 3.35, y que los límites de los  $\alpha$ -cortes de la aproximación coinciden con los del tiempo borroso resultante de aplicar el principio de extensión.

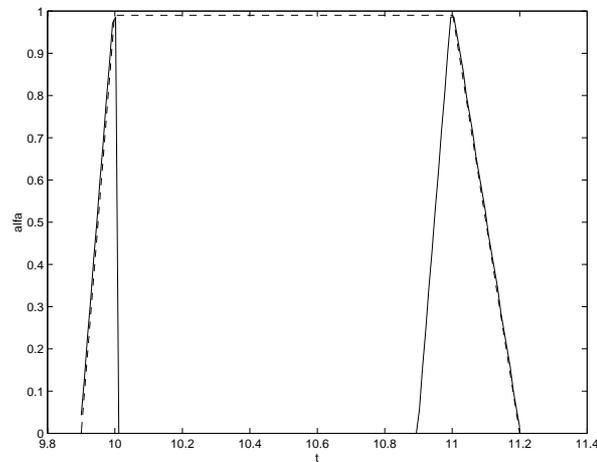


Figura 3.44:  $WC_{A4}$  obtenido mediante simulación del principio de extensión (-) y  $WC_{A4}^*$  obtenido mediante análisis de intervalos (...)

En conclusión, gracias a la representación gráfica propuesta para la obtención de los tiempos de finalización borrosos a partir de los tiempos de ejecución y a las simulaciones realizadas, se ha comprobado que en todos los casos analizados ha sido posible obtener aproximaciones de los tiempos de finalización borrosos mediante la aplicación de simulación por intervalos. El tiempo de computación de estas aproximaciones puede ser mucho menor que el de la aplicación del principio de extensión, y los resultados de planificabilidad borrosa coinciden.

## **4. Principio de extensión de funciones monótonas. Aproximaciones.**

Para extender una función de parámetros reales a parámetros borrosos se puede aplicar el principio de extensión y de esa forma el resultado de la función es un conjunto borroso. El resultado de aplicar el principio de extensión se toma como referencia y se denomina *exacto*. El problema es que en muchas ocasiones la aplicación del principio de extensión se hace inviable debido a su coste computacional, por lo que se buscan caminos alternativos para la extensión de funciones que sean más rápidos de obtener. En esta tesis se desea extender el análisis de planificabilidad a parámetros borrosos, para lo cual uno de los objetivos es la obtención de aproximaciones válidas rápidas de calcular de los tiempos de finalización, en lugar de los tiempos de finalización borrosos exactos. Las aproximaciones válidas de los tiempos de finalización proporcionan los valores exactos de posibilidad y de necesidad de que el sistema sea planificable.

En el capítulo anterior se mostró mediante la utilización de la representación gráfica propuesta en esta tesis, que en determinados casos es posible obtener aproximaciones válidas convexas de los tiempos de finalización, aplicando cálculo de intervalos. Un objetivo de este capítulo es enunciar las condiciones que debe cumplir cualquier función para que se pueda obtener una aproximación convexa de la imagen borrosa exacta, mediante cálculo de intervalos. Además, se enuncian las condiciones que debe cumplir una función para que se pueda obtener una aproximación no convexa de la imagen borrosa exacta mediante aritmética borrosa.

El análisis que se realiza en este capítulo es general, para cualquier función monótona continua o discontinua, sin centrarse en el cálculo de los tiempos de finalización de las tareas en los sistemas de tiempo real. Únicamente, se restringe el análisis a funciones de parámetros positivos, que son con las que se trabaja en sistemas de tiempo real.

En el apartado 4.1 se revisa la definición dada en el capítulo 3 de aproximación válida de un conjunto borroso, que puede ser convexa o no convexa.

En el apartado 4.2 de este capítulo se enuncia la propiedad que debe cumplir una función para poder obtener una aproximación válida convexa mediante aritmética de intervalos. Se prueba que si la función a extender es monótona, continua o discontinua, en sus parámetros (números reales), las aproximaciones válidas borrosas convexas citadas son muy rápidas de obtener mediante aritmética de intervalos.

En el apartado 4.3 se presentan las propiedades que debe cumplir una función para que el conjunto borroso que se obtiene al extender la función mediante aritmética borrosa sea una aproximación válida del conjunto borroso resultante exacto. En general, las aproximaciones obtenidas mediante aritmética borrosa son no convexas (ver Figura 4.1, aproximación  $A^{**}$ ). Si una función es monótona en cada variable y cada función parcial de la que está formada es igualmente monótona en la misma variable, la aproximación que se obtiene al extender la función mediante aritmética

borrosa es una aproximación válida. En ese apartado se ha hecho especial hincapié en la extensión borrosa de la operación matemática redondeo hacia arriba, empleada en el próximo capítulo para realizar el análisis de planificabilidad de sistemas de tiempo real.

### 4.1. Aproximaciones válidas de un conjuntos borroso

En el capítulo anterior se definió lo que en este contexto se consideran aproximaciones válidas de un conjunto borroso, que, en general, puede ser no convexo. En esta tesis, se trata de extender el algoritmo de obtención de tiempos finales de las tareas a parámetros borrosos. Los tiempos finales resultantes son conjuntos borrosos, en general no convexos, que hay que comparar con los plazos correspondientes, numéricos o borrosos. El resultado de dicha comparación dará un valor de posibilidad y un valor de necesidad de que la tarea termine su ejecución antes del plazo. Si en lugar de obtener el tiempo de finalización borroso exacto que proporciona el principio de extensión, se obtiene una aproximación cuya comparación con el plazo da la misma posibilidad y necesidad que el tiempo final exacto, la planificabilidad borrosa expresada en términos de posibilidad y necesidad será también exacta. Estas aproximaciones de los tiempos de finalización borrosos son las que se consideran válidas en el contexto de esta tesis.

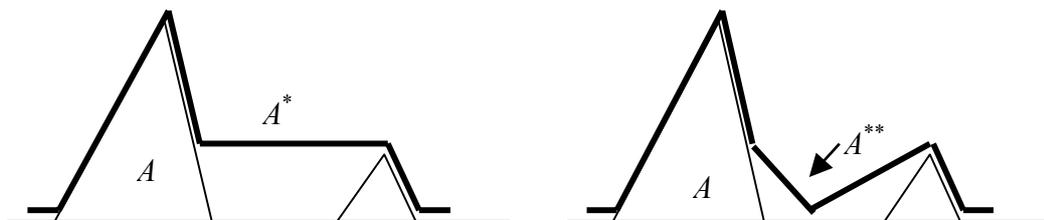


Figura 4.1: Conjunto borroso  $A$  y sus aproximaciones válidas convexas  $A^*$  y no convexas  $A^{**}$

Tal y como se presenta en el anexo B, la comparación de un conjunto borroso no convexo con otro número, clásico o borroso, da el mismo resultado de posibilidad y necesidad que la comparación de la aproximación convexa del primero con el segundo, definida la aproximación como un número borroso (convexo) con los mismos límites inferiores y superiores para cada  $\alpha$ -corte (ver Figura 4.1, aproximación convexa  $A^*$ ). Además de dicha aproximación convexa, cualquier otro conjunto borroso que tenga los mismos límites de los  $\alpha$ -cortes es una aproximación

válida (ver Figura 4.1, aproximación no convexa  $A^{**}$ ). Estas aproximaciones de los tiempos de finalización son distintas que los tiempos borrosos exactos obtenidos mediante el principio de extensión, pero el valor de planificabilidad borrosa obtenido con dichas aproximaciones es exacto.

## 4.2. Aproximación mediante análisis de intervalos

En [Dubois, Prade, 1980] se describe y demuestra la manera de extender la operación de dos números reales a números borrosos cuando la operación es continua y no decreciente en ambos, o no creciente en ambos, o no decreciente en uno y no creciente en el otro. Este mismo estudio se plantea en [Nakamura, 1993], [Klir, 1997] mediante los  $\alpha$ -cortes de los conjuntos borrosos, obteniéndose los mismos resultados. En el siguiente apartado se describen los resultados para el caso de funciones continuas de variables reales positivas.

### 4.2.1. Funciones continuas

Sea el  $\alpha$ -corte  $[\underline{a}, \bar{a}](\alpha) = [\underline{a}(\alpha), \bar{a}(\alpha)]$  de un número borroso  $A$  con función de pertenencia  $\mu_A$ , donde el límite inferior del  $\alpha$ -corte se expresa como ([Nakamura, 1993]):

$$(4.1) \quad \underline{a}(\alpha) = \inf_{x/\mu_A(x) \geq \alpha} x$$

y el límite superior se expresa como:

$$(4.2) \quad \bar{a}(\alpha) = \sup_{x/\mu_A(x) \geq \alpha} x$$

En la Figura 4.2 se muestra un  $\alpha$ -corte de  $A$ .

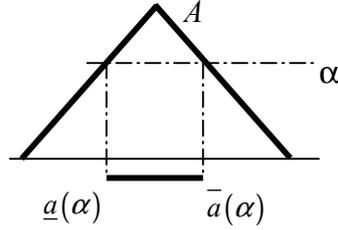


Figura 4.2: Representación de un  $\alpha$ -corte del número borroso  $A$

Si el número borroso es continuo entonces se verifica  $\mu_A(\underline{a}(\alpha)) = \mu_A(\overline{a}(\alpha)) = \alpha$

Por otro lado, en [Dubois, Prade, 1980], [Nakamura, 1993] y [Klir, 1997] se demuestra que es posible extender las funciones continuas y monótonas a parámetros borrosos aplicando aritmética de intervalos con los mismos resultados que el principio de extensión. Dada una función continua y monótona  $f(a,b)$  tal que  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  se pueden distinguir los siguientes casos:

1.  $f(a,b)$  no decreciente ni en  $a$  ni en  $b$
2.  $f(a,b)$  no decreciente en  $a$  y no creciente en  $b$
3.  $f(a,b)$  no creciente ni en  $a$  ni en  $b$

1. Sea  $f(a,b)$  una función continua en  $a$  y  $b$  y no decreciente ni en  $a$  ni en  $b$ . El resultado de extender la función  $f$  a números borrosos  $A$  y  $B$  es un número borroso  $C$  cuya función de pertenencia  $\mu_C$  se puede obtener mediante la aplicación del principio de extensión (4.3) o bien mediante aritmética de intervalos con el mismo resultado.

$$(4.3) \quad \forall \omega \in \mathbb{R} > 0, \quad \pi_C(\omega) = \mu_C(\omega) = \sup_{\omega=f(a,b)} \min\{\mu_A(a), \mu_B(b)\}$$

El número borroso  $C$  se puede representar a partir de sus  $\alpha$ -cortes  $[\underline{c}(\alpha), \overline{c}(\alpha)]$  que se pueden calcular a partir de los  $\alpha$ -cortes de  $A$  y  $B$  como:

$$[\underline{c}, \overline{c}](\alpha) = [\underline{c}(\alpha), \overline{c}(\alpha)] = [f(\underline{a}, \underline{b}), f(\overline{a}, \overline{b})]$$

2. Si la función  $f$  es continua no decreciente en  $a$  y no creciente en  $b$ , los  $\alpha$ -cortes de  $C$  se pueden calcular mediante aritmética de intervalos como:

$$[\underline{c}, \bar{c}](\alpha) = [\underline{c}(\alpha), \bar{c}(\alpha)] = [f(\underline{a}, \bar{b}), f(\bar{a}, \underline{b})]$$

3. Del mismo modo, cuando la función  $f$  es continua no creciente en  $a$  y en  $b$ , los  $\alpha$ -cortes de  $C$  se pueden calcular mediante aritmética de intervalos a partir de los  $\alpha$ -cortes de  $A$  y  $B$  como:

$$[\underline{c}, \bar{c}](\alpha) = [\underline{c}(\alpha), \bar{c}(\alpha)] = [f(\bar{a}, \bar{b}), f(\underline{a}, \underline{b})]$$

Sin embargo, los autores únicamente tratan el caso de funciones continuas. En este capítulo se pretende extender las conclusiones anteriores al caso de tener funciones no continuas, para poder aplicarlas a los algoritmos de planificabilidad de los sistemas de tiempo real presentados en el capítulo 2. Concretamente, se pretende obtener aproximaciones válidas de los tiempos de finalización mediante la aplicación de aritmética de intervalos. Es decir, una aproximación convexa del tiempo de finalización borroso exacto con los mismos límites inferiores y superiores de los  $\alpha$ -cortes que el exacto.

#### 4.2.2. Funciones discontinuas

Supongamos ahora que  $f(a,b)$  es una función:

- Discontinua en  $a$  y/o en  $b$  (número infinito numerable de discontinuidades)
- No decreciente en  $a$
- No decreciente en  $b$

Si  $a$  y  $b$  se modelan mediante números borrosos  $A$  y  $B$  respectivamente definidos en el dominio de los números reales positivos, podemos extender la función  $f$  a una función de argumentos borrosos mediante el principio de extensión (4.3). El resultado de dicha función es un conjunto borroso  $C$  que puede ser convexo o no convexo.

Al ser  $f(a,b)$  no decreciente ni en  $a$  ni en  $b$ , si  $f(a,b)$  permanece constante e igual a  $\omega_0$ , es decir, si  $f(a,b) = f(a',b') = \omega_0$ , tal que  $a, b, a', b' \in \mathbb{R}$  se cumple que:

$$\text{Si } a' > a \Rightarrow b' \leq b$$

$$\text{Si } a' < a \Rightarrow b' \geq b$$

Por otro lado, al crecer  $\alpha$  el límite inferior del  $\alpha$ -corte correspondiente crece o permanece igual, y el límite superior decrece o permanece igual (ver Figura 4.3). Esto es, siendo  $\alpha_1, \alpha_2 \in [0,1]$ :

$$0 \leq \alpha_1 \leq \alpha_2 \leq 1 \text{ si y sólo si}$$

$$(4.4) \quad \begin{aligned} \underline{a}(\alpha_1) &\leq \underline{a}(\alpha_2) \\ \bar{a}(\alpha_1) &\geq \bar{a}(\alpha_2) \end{aligned}$$

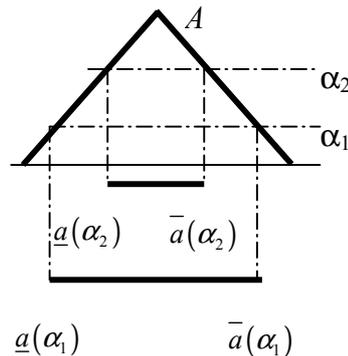


Figura 4.3: Relación entre la variación de  $\alpha$  y los límites del  $\alpha$ -corte correspondiente

Es decir,  $\underline{a}(\alpha)$ ,  $\underline{b}(\alpha)$  son no decrecientes en  $\alpha$  y  $\bar{a}(\alpha)$ ,  $\bar{b}(\alpha)$  son no crecientes en  $\alpha$ .

Esto permite probar que todos aquellos valores de  $\omega = f(a,b)$  que se pueden obtener a partir de los límites inferiores (respectivamente superiores) de los  $\alpha$ -cortes de  $A$  y  $B$  son los límites inferiores (respectivamente superiores) de los  $\alpha$ -cortes de  $C$ . Es decir:

$$(4.5) \quad \forall \omega \mid \omega = f(\underline{a}(\alpha), \underline{b}(\alpha)) \quad \text{con } \alpha \in (0,1] \text{ se cumple que:}$$

$$(4.6) \quad \mu_C(\omega) = \alpha$$

$$(4.7) \quad \omega = \underline{c}(\alpha)$$

La ecuación (4.6) significa que al aplicar el principio de extensión para cada valor de  $\omega$  obtenido según (4.5), el resultado es:

$$(4.8) \quad \mu_C(\omega) = \sup_{\omega=f(a,b)} \min\{\mu_A(a), \mu_B(b)\} = \sup_{\omega=f(a,b)} \min\{\underline{a}(\alpha), \underline{b}(\alpha)\} = \alpha$$

La ecuación (4.7) significa que el valor de  $\omega$  obtenido según (4.5) es el límite inferior del  $\alpha$ -corte de  $C$ , con el mismo valor de  $\alpha$  que en (4.5):

$$(4.9) \quad \omega = \underline{c}(\alpha) = \inf_{\mu_C(x) \geq \alpha} x$$

Para los límites superiores se puede expresar como:

$$(4.10) \quad \forall \omega \mid \omega = f(\bar{a}(\alpha), \bar{b}(\alpha)) \quad \text{con } \alpha \in (0,1] \quad \text{se cumple que:}$$

$$(4.11) \quad \mu_C(\omega) = \sup_{\omega=f(a,b)} \min\{\mu_A(a), \mu_B(b)\} = \sup_{\omega=f(a,b)} \min\{\bar{a}(\alpha), \bar{b}(\alpha)\} = \alpha$$

$$(4.12) \quad \omega = \bar{c}(\alpha) = \sup_{\mu_C(x) \geq \alpha} x$$

El hecho de que se cumplan estas propiedades permite obtener los límites de todos los  $\alpha$ -cortes de  $C$  de forma exacta mediante aritmética de intervalos, pero hay valores de  $C$  que no se pueden obtener operando los límites de los  $\alpha$ -cortes de  $A$  y de  $B$ . En la Figura 4.4 se muestra en línea discontinua la parte de  $\mu_C$  que no es posible obtener mediante aritmética de intervalos.

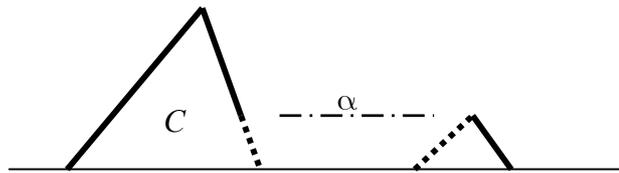


Figura 4.4: En línea discontinua, puntos de  $\mu_C$  que no se obtienen mediante aritmética de intervalos

Comprobemos en primer lugar el resultado propuesto para los *límites inferiores*.

Partimos para ello del valor  $\omega$  que se obtiene de operar los límites inferiores de los  $\alpha$ -cortes de  $A$  y de  $B$ , con el mismo valor de  $\alpha$ :  $\omega = f(\underline{a}(\alpha), \underline{b}(\alpha))$ . Supongamos que  $\omega$  permanece constante. Si el primer parámetro de la función crece, el segundo debe decrecer o permanecer igual, ya que  $f$  es no decreciente en sus parámetros (ver Figura 4.5)

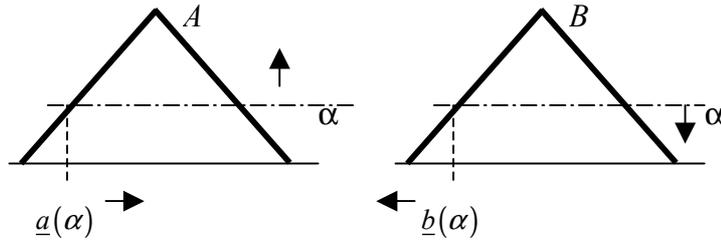


Figura 4.5: Variación de los límites inferiores para que  $\omega$  permanezca constante.

Sean  $\alpha, \alpha_1 \in [0,1]$  Aplicando (4.4) se tiene  $\underline{a}(\alpha_1) > \underline{a}(\alpha) \Rightarrow \alpha_1 \geq \alpha$ , y para que  $\omega = \text{cte}$ , el nuevo valor de  $b$ , si existe, cumple  $\underline{b}(\alpha_2) \leq \underline{b}(\alpha) \Rightarrow \alpha_2 \leq \alpha$  (ver Figura 4.5). Por tanto:

$$\begin{aligned} \min\{\mu_A(\underline{a}(\alpha)), \mu_B(\underline{b}(\alpha))\} &= \min\{\alpha, \alpha\} = \alpha \\ &\geq \min\{\mu_A(\underline{a}(\alpha_1)), \mu_B(\underline{b}(\alpha_2))\} = \min\{\alpha_1, \alpha_2\} = \alpha_2 \end{aligned}$$

Con lo que al aplicar (4.8), el supremo se alcanza para la pareja correspondiente a  $\alpha$ .

Tal y como se ve en la Figura 4.6 si el primer parámetro  $a$  de la función decrece, el supremo también se alcanza para la pareja correspondiente a  $\alpha$ .

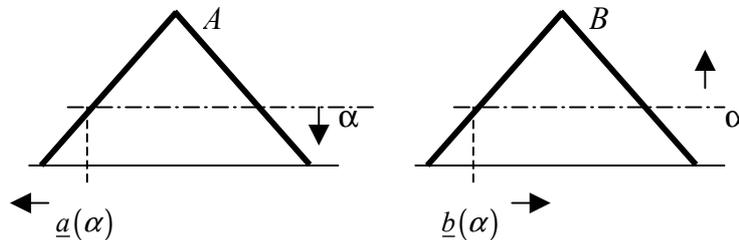


Figura 4.6: Variación de los límites inferiores para que  $\omega$  permanezca constante.

En efecto, aplicando (4.4) se tiene  $\underline{a}(\alpha_1) < \underline{a}(\alpha) \Rightarrow \alpha_1 \leq \alpha$  y para que  $\omega = \text{cte}$ , el nuevo valor de  $b$ , si existe, verifica  $\underline{b}(\alpha_2) \geq \underline{b}(\alpha) \Rightarrow \alpha_2 \geq \alpha$  (ver Figura 4.6). Por tanto:

$$\begin{aligned} \min\{\mu_A(\underline{a}(\alpha)), \mu_B(\underline{b}(\alpha))\} &= \min\{\alpha, \alpha\} = \alpha \\ &\geq \min\{\mu_A(\underline{a}(\alpha_1)), \mu_B(\underline{b}(\alpha_2))\} = \min\{\alpha_1, \alpha_2\} = \alpha_1 \end{aligned}$$

Con lo que al aplicar (4.8), el supremo para un  $\omega$  fijo se alcanza para la pareja correspondiente a  $\alpha$ .

Extendiendo estos resultados a todos los valores  $\alpha_1, \alpha_2$  tal que

$$\omega = f(\underline{a}(\alpha), \underline{b}(\alpha)) = f(\underline{a}(\alpha_1), \underline{b}(\alpha_2))$$

se tiene:

$$\mu_C(\omega) = \sup_{\omega=f(a,b)} \min\{\mu_A(a), \mu_B(b)\} = \sup_{\omega=f(a,b)} \min\{\underline{a}(\alpha), \underline{b}(\alpha)\} = \alpha$$

Queda por demostrar, que el valor  $\omega = f(\underline{a}(\alpha), \underline{b}(\alpha))$  es el límite inferior del  $\alpha$ -corte de  $C$  (4.9). Por reducción al absurdo, supongamos que no se cumple:

$$\omega = \underline{c}(\alpha) = \inf_{\mu_C(x) \geq \alpha} x$$

Eso quiere decir que existe un  $\omega_1 < \omega$  tal que  $\mu_C(\omega_1) = \mu_C(\omega) = \alpha$ . Como  $f$  es no decreciente en sus parámetros, si  $\omega$  disminuye al menos uno de sus parámetros disminuye. Esto es,

Si  $\omega_1 < \omega$  entonces:

- o bien  $f(\underline{a}(\alpha_1), \underline{b}(\alpha)) < f(\underline{a}(\alpha), \underline{b}(\alpha))$  con  $\underline{a}(\alpha_1) < \underline{a}(\alpha)$  y teniendo en cuenta (4.1)  $\alpha_1 \neq \alpha$
- o bien  $f(\underline{a}(\alpha), \underline{b}(\alpha_2)) < f(\underline{a}(\alpha), \underline{b}(\alpha))$  con  $\underline{b}(\alpha_2) < \underline{b}(\alpha)$  y teniendo en cuenta (4.1)  $\alpha_2 \neq \alpha$

Aplicando (4.4) se tiene que  $\alpha_1 < \alpha$  y  $\alpha_2 < \alpha$ .

Por tanto, aplicando el principio de extensión, como al menos uno de los dos valores de  $\alpha$  disminuye, el grado de pertenencia de  $\omega_1$  disminuye:

$$\mu_C(\omega_1) = \sup_{\omega_1=f(a,b)} \min\{\mu_A(a), \mu_B(b)\} < \alpha$$

Con lo que no existe un  $\omega_1 < \omega$  tal que  $\mu_C(\omega_1) = \mu_C(\omega) = \alpha$  y queda probado que  $\omega$  es el límite inferior del  $\alpha$ -corte de  $C$ :

$$\omega = \underline{c}(\alpha) = \inf_{\mu_C(x) \geq \alpha} x$$

De forma similar, se pueden comprobar los resultados propuestos para los *límites superiores*.

Dado un valor  $\omega$  que se obtiene de operar los límites superiores de los  $\alpha$ -cortes de  $A$  y de  $B$ , con el mismo valor de  $\alpha$ ,  $\omega = f(\bar{a}(\alpha), \bar{b}(\alpha))$ , supongamos que  $\omega$  permanece constante.

Si el primer parámetro de la función crece, el segundo debe decrecer o permanecer igual, ya que  $f$  es no decreciente en sus parámetros. En efecto, aplicando (4.4)  $\bar{a}(\alpha_1) > \bar{a}(\alpha) \Rightarrow \alpha_1 \leq \alpha$  y para que  $\omega = \text{cte}$ , el nuevo valor de  $b$ , si existe, es  $\bar{b}(\alpha_2) \leq \bar{b}(\alpha) \Rightarrow \alpha_2 \geq \alpha$ . Con esto se tiene,  $\min\{\mu_A(\bar{a}(\alpha)), \mu_B(\bar{b}(\alpha))\} \geq \min\{\mu_A(\bar{a}(\alpha_1)), \mu_B(\bar{b}(\alpha_2))\}$ . Por tanto el supremo en (4.11) se alcanza para la pareja correspondiente a  $\alpha$ .

Por otro lado, si el primer parámetro de la función decrece, aplicando (4.4) se verifica  $\bar{a}(\alpha_1) < \bar{a}(\alpha) \Rightarrow \alpha_1 \geq \alpha$  y para que  $\omega = \text{cte}$ , el nuevo valor de  $b$ , si existe, cumple  $\bar{b}(\alpha_2) \geq \bar{b}(\alpha) \Rightarrow \alpha_2 \leq \alpha$ . Por tanto se tiene que,  $\min\{\mu_A(\bar{a}(\alpha)), \mu_B(\bar{b}(\alpha))\} \geq \min\{\mu_A(\bar{a}(\alpha_1)), \mu_B(\bar{b}(\alpha_2))\} = \alpha_2$ . Al aplicar (4.11) el supremo se alcanza de nuevo para la pareja correspondiente a  $\alpha$ .

Extendiendo estos resultados a todos los valores  $\alpha_1, \alpha_2$  tal que  $\omega = f(\bar{a}(\alpha), \bar{b}(\alpha)) = f(\bar{a}(\alpha_1), \bar{b}(\alpha_2))$  se tiene que se cumple (4.11).

Queda por demostrar (4.12). Por reducción al absurdo, supongamos que no se cumple que  $\omega = \bar{c}(\alpha) = \sup_{\mu_C(x) \geq \alpha} x$ . En tal caso existe un  $\omega_1 > \omega$  tal que  $\mu_C(\omega_1) = \mu_C(\omega) = \alpha$ . Como  $f$  es no decreciente en sus parámetros, si  $\omega$  crece al menos uno de sus parámetros crece. Esto es, si  $\omega_1 > \omega$  esto quiere decir que:

- bien  $f(\bar{a}(\alpha_1), \bar{b}(\alpha)) > f(\bar{a}(\alpha), \bar{b}(\alpha))$  con  $\bar{a}(\alpha_1) > \bar{a}(\alpha)$  y teniendo en cuenta (4.2)  $\alpha_1 \neq \alpha$
- bien  $f(\bar{a}(\alpha), \bar{b}(\alpha_2)) > f(\bar{a}(\alpha), \bar{b}(\alpha))$  con  $\bar{b}(\alpha_2) > \bar{b}(\alpha)$  y teniendo en cuenta (4.2)  $\alpha_2 \neq \alpha$

Aplicando (4.4) se tiene que  $\alpha_1 < \alpha$  y  $\alpha_2 < \alpha$ . Por tanto, aplicando el principio de extensión, como al menos uno de los dos valores de  $\alpha$  disminuye, el grado de pertenencia de  $\omega_1$  disminuye,  $\mu_C(\omega_1) = \sup_{\omega_1 = f(a,b)} \min\{\mu_A(a), \mu_B(b)\} < \alpha$ . Con lo que no existe un  $\omega_1 > \omega$  tal que  $\mu_C(\omega_1) = \mu_C(\omega) = \alpha$  y queda probado que  $\omega$  es el límite superior del  $\alpha$ -corte de  $C$ .

## Resumen

Los resultados anteriores permiten concluir que si  $A$  y  $B$  son números borrosos y  $f$  es no decreciente ni en  $a$  ni en  $b$ , entonces se pueden calcular los límites inferiores y superiores exactos de los  $\alpha$ -cortes de  $C$  aplicando :

$$(4.13) \quad \underline{c}(\alpha) = f(\underline{a}(\alpha), \underline{b}(\alpha)) \\ \bar{c}(\alpha) = f(\bar{a}(\alpha), \bar{b}(\alpha))$$

De la misma manera, se podría plantear la prueba anterior para el caso en el que  $f$  es no creciente ni en  $a$  ni en  $b$ . El resultado que se obtiene es:

$$(4.14) \quad \bar{c}(\alpha) = f(\underline{a}(\alpha), \underline{b}(\alpha)) \\ \underline{c}(\alpha) = f(\bar{a}(\alpha), \bar{b}(\alpha))$$

De forma similar, para  $f$  no decreciente en  $a$  y no creciente en  $b$ , se tiene:

$$(4.15) \quad \underline{c}(\alpha) = f(\underline{a}(\alpha), \bar{b}(\alpha))$$

$$\bar{c}(\alpha) = f(\bar{a}(\alpha), \underline{b}(\alpha))$$

Por tanto, el conjunto  $C^*$  definido según las ecuaciones anteriores operando los  $\alpha$ -cortes de  $A$  y  $B$  y suponiendo que  $[\underline{c}(\alpha), \bar{c}(\alpha)]$  es el  $\alpha$ -corte resultante, es únicamente una aproximación de  $C$  con los mismos límites inferiores y superiores de los  $\alpha$ -cortes, convexo y menos específico que  $C$ , como se ilustra en el ejemplo de la Figura 4.7.

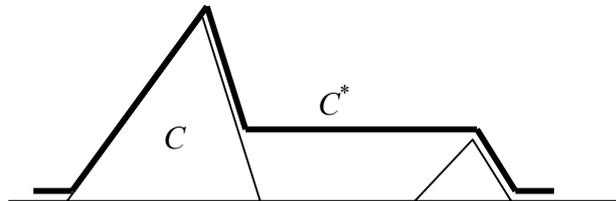


Figura 4.7:  $C$  exacto y  $C^*$  obtenido por aritmética de intervalos

Como  $f$  es discontinua, no se puede concluir que todo el intervalo  $[\underline{c}(\alpha), \bar{c}(\alpha)]$  sea el  $\alpha$ -corte del conjunto borroso  $C$ , a diferencia de lo señalado para funciones continuas. Tal y como ya se ha mencionado, si  $C$  es no convexo, para cada  $\alpha$  hay varios subintervalos asociados y en ese caso, dentro del intervalo  $[\underline{c}(\alpha), \bar{c}(\alpha)]$  hay puntos no pertenecientes al  $\alpha$ -corte de  $C$  (ver la Figura 4.8).

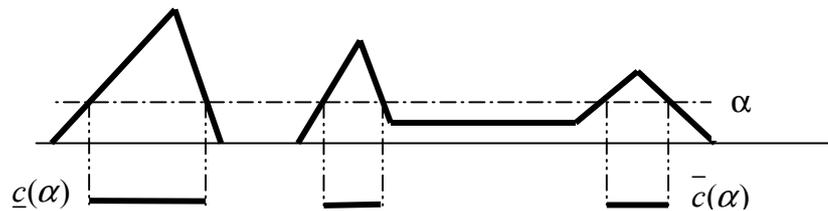


Figura 4.8: Representación del  $\alpha$ -corte de un conjunto borroso no convexo  $C$

### 4.2.3. Extensión de funciones monótonas con parámetros modelados como conjuntos borrosos no convexos

Aunque hasta ahora se ha supuesto que la información de partida se modela mediante números borrosos, los resultados obtenidos son fácilmente generalizables al caso de que los parámetros sean conjuntos borrosos no convexos.

En este caso, si  $A$  es un conjunto borroso no convexo su  $\alpha$ -corte (como se muestra en la Figura 4.9) es un subconjunto del intervalo definido por los límites inferiores y superiores de dicho  $\alpha$ -corte  $[\underline{a}(\alpha), \bar{a}(\alpha)]$ .

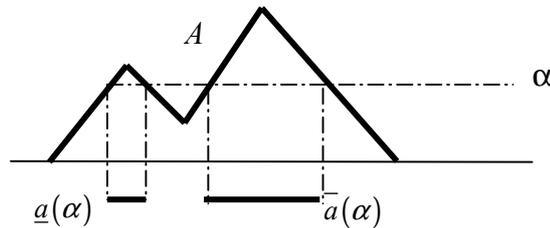


Figura 4.9: Parámetro modelado mediante un conjunto borroso no convexo  $A$ , y  $\alpha$ -corte asociado

El límite inferior del  $\alpha$ -corte de  $A$  se expresa también como  $\underline{a}(\alpha) = \inf_{x/\mu_A(x) \geq \alpha} x$ , y su límite superior se expresa como  $\bar{a}(\alpha) = \sup_{x/\mu_A(x) \geq \alpha} x$

Además cuando crece  $\alpha$ , el límite inferior del  $\alpha$ -corte correspondiente crece (o permanece igual), y el límite superior decrece (o permanece igual), tal y como se observa en la Figura 4.10. Por tanto se tiene que dados  $\alpha_1, \alpha_2 \in [0,1]$ :

$$0 \leq \alpha_1 \leq \alpha_2 \leq 1, \text{ si y sólo si}$$

$$\begin{aligned} \underline{a}(\alpha_1) &\leq \underline{a}(\alpha_2) & \underline{b}(\alpha_1) &\leq \underline{b}(\alpha_2) \\ \bar{a}(\alpha_1) &\geq \bar{a}(\alpha_2) & \bar{b}(\alpha_1) &\geq \bar{b}(\alpha_2) \end{aligned}$$

Se deduce por tanto que  $\underline{a}(\alpha)$ ,  $\underline{b}(\alpha)$  son no decrecientes en  $\alpha$  y  $\bar{a}(\alpha)$ ,  $\bar{b}(\alpha)$  son no crecientes en  $\alpha$ , por lo que los resultados obtenidos en el apartado anterior para argumentos modelados con números borrosos son igualmente aplicables cuando los argumentos se modelan con conjuntos borrosos cualesquiera.

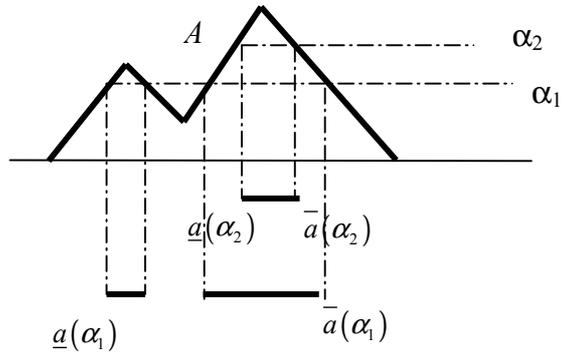


Figura 4.10:  $\alpha$ -cortes de  $A$  para  $\alpha_1$  y  $\alpha_2$

En resumen, si  $A$  y  $B$  son conjuntos borrosos cualesquiera (convexos o no convexos), y  $f$  es no decreciente ni en  $a$  ni en  $b$ , entonces se puede aplicar (4.13):

$$\underline{c}(\alpha) = f(\underline{a}(\alpha), \underline{b}(\alpha))$$

$$\bar{c}(\alpha) = f(\bar{a}(\alpha), \bar{b}(\alpha))$$

De la misma manera, si  $f$  es no creciente ni en  $a$  ni en  $b$ , entonces se puede aplicar (4.14):

$$\bar{c}(\alpha) = f(\underline{a}(\alpha), \underline{b}(\alpha))$$

$$\underline{c}(\alpha) = f(\bar{a}(\alpha), \bar{b}(\alpha))$$

Si  $f$  es no decreciente en  $a$  y no creciente en  $b$ , entonces se puede aplicar (4.15):

$$\underline{c}(\alpha) = f(\underline{a}(\alpha), \bar{b}(\alpha))$$

$$\bar{c}(\alpha) = f(\bar{a}(\alpha), \underline{b}(\alpha))$$

### 4.3. Aproximación mediante aritmética borrosa

En el apartado anterior se ha visto que es posible obtener una aproximación válida convexa de la imagen borrosa de una función discontinua cuando ésta es monótona en todos sus argumentos, aplicando aritmética de intervalos. En este apartado se va a mostrar en qué condiciones se puede obtener una aproximación válida mediante la aplicación de aritmética borrosa. Este estudio amplía los encontrados en la literatura que se centran en operaciones continuas.

Para ilustrar los resultados presentados se analiza la extensión de tres funciones que incluyen la operación redondeo hacia arriba, ya que este tipo de funciones discontinuas son las que aparecen en los algoritmos de planificabilidad de los sistemas de tiempo real.

#### 4.3.1. Extensión de funciones continuas mediante aritmética borrosa

La extensión de una función mediante aritmética borrosa se realiza agrupando la expresión en subexpresiones simples y componiendo posteriormente las subexpresiones. Así, una posible extensión de  $f(a,b,c)=a*(b\oplus c)$ , siendo  $*$  y  $\oplus$  dos operaciones cualquiera, a parámetros borrosos es aplicar aritmética borrosa operando en primer lugar  $U=B\oplus C$  y el resultado operarlo con  $A$  obteniéndose un resultado  $W^*=A*U$ . Cada subexpresión u operación parcial se puede calcular aplicando el principio de extensión; por ejemplo, para  $W^*=A*U$  se tiene  $\mu_{A*U}(\omega) = \sup_{\omega=a*u} \min\{\mu_A(a), \mu_U(u)\}$

Por otro lado, la extensión de una función  $f(a,b,c)=a*(b\oplus c)$  se puede obtener mediante la aplicación estricta del principio de extensión a  $f$  como:

$$\mu_W(\omega) = \sup_{\omega=a*(b\oplus c)} \min\{\mu_A(a), \mu_B(b), \mu_C(c)\}$$

El resultado  $W^*$  de extender una función a parámetros borrosos mediante la aplicación de aritmética borrosa a las operaciones parciales puede coincidir o no con el resultado  $W$  de aplicar el principio de extensión a la función en conjunto ([Klir, 1997]).

En efecto y a modo de ejemplo, la propiedad de subdistributividad ([Klir, Yuan, 1995], [Dubois, Prade, 1980]), establece que:

$$A \cdot (B - C) \subseteq A \cdot B - A \cdot C$$

Por tanto, si la función a extender es  $f(a,b) = a \cdot b - a \cdot c = a \cdot (b - c)$ , la extensión exacta que coincide con la aplicación del principio de extensión es  $F(A,B) = A \cdot (B - C)$ . Por otro lado, si se extiende la función aplicando aritmética borrosa como  $F(A,B) = A \cdot B - A \cdot C$ , el resultado es en general un conjunto borroso menos específico (o igual) que el que proporciona la aplicación del principio de extensión.

Concretamente la discrepancia entre ambos procedimientos aparece cuando la función  $f$  es, por ejemplo, creciente en  $a$  pero algunas de las operaciones de las que está compuesta  $f$  son decrecientes en  $a$ .

Se pueden mostrar estas discrepancias mediante un ejemplo sencillo. Supongamos  $a$  y  $b$  mayores que cero, y la función  $f(a,b) = a / (a+b)$

En este caso,  $f$  es creciente en  $a$  y decreciente en  $b$ , por lo que el cálculo por intervalos visto en el apartado 4.2 establece  $\underline{c}(\alpha) = f(\underline{a}(\alpha), \bar{b}(\alpha))$  y  $\bar{c}(\alpha) = f(\bar{a}(\alpha), \underline{b}(\alpha))$ . Además al ser  $f$  continua en  $a$  y en  $b$ , el intervalo completo  $[\underline{c}(\alpha), \bar{c}(\alpha)]$  es igual al  $\alpha$ -corte del número borroso resultante.

Este resultado difiere del que se obtiene aplicando directamente las operaciones aritméticas a los números borrosos mediante la expresión  $A / (A+B)$ . En efecto, sean  $A$  y  $B$  dos números borrosos triangulares definidos por sus vértices  $A=(0.9,1,1.1)$ ,  $B=(1.9,2,2.1)$ . Para  $\alpha=0$  los límites del  $\alpha$ -corte resultante son:

$$\underline{c}(0) = f(\underline{a}(0), \bar{b}(0)) = \frac{0.9}{0.9 + 2.1} = 0.3$$

$$\bar{c}(0) = f(\bar{a}(0), \underline{b}(0)) = \frac{1.1}{1.1 + 1.9} = 0.366$$

Aplicando directamente aritmética borrosa a las operaciones parciales  $A / (A+B)$  y particularizado para los  $\alpha$ -cortes con  $\alpha=0$ :

$$\frac{[0.9, 1.1]}{[0.9, 1.1] + [1.9, 2.1]} = \frac{[0.9, 1.1]}{[0.9 + 1.9, 1.1 + 2.1]} = \frac{[0.9, 1.1]}{[2.8, 3.2]} = \left[ \frac{0.9}{3.2}, \frac{1.1}{2.8} \right] = [0.28, 0.39]$$

que no coincide con el principio de extensión de la función  $f$ , porque se ha operado el límite inferior de  $A$  (0.9) en el numerador, con el límite superior de  $A$  (1.1) en el denominador. En cambio, aplicando el principio de extensión se utiliza el mismo valor de  $a$  en el numerador y el denominador (ya que se trata de una única variable).

Por otro lado, se puede analizar la extensión de una expresión equivalente de la función  $f$ ,  $f(a,b) = 1 / (1+b/a)$ . La extensión borrosa de esta expresión mediante el principio de extensión es la misma ya que se trata de la misma función. En este caso, la extensión de esta expresión mediante aritmética borrosa proporciona el mismo resultado que el principio de extensión, ya que cada variable aparece una sola vez.

En efecto, aplicando directamente  $1 / (1 + B/A)$  particularizado para los  $\alpha$ -cortes con  $\alpha=0$ :

$$\frac{1}{1 + \frac{[1.9, 2.1]}{[0.9, 1.1]}} = \frac{1}{1 + \left[ \frac{1.9}{1.1}, \frac{2.1}{0.9} \right]} = \frac{1}{1 + [1.72, 2.33]} = \left[ \frac{1}{3.33}, \frac{1}{2.72} \right] = [0.3, 0.366]$$

Estas diferencias aparecen siempre que existen operaciones crecientes y decrecientes en la misma variable. En una función del tipo:  $f(a,b) = a*(a+b)$ , la imagen borrosa de la función obtenida por ambos caminos coincide.

En el caso de la resta, sabemos que la resta de números borrosos no es la operación inversa de la suma, y que, por tanto,  $(\tilde{3} - \tilde{2}) + \tilde{2} \neq \tilde{3}$ , siendo  $\tilde{3}$  el número borroso “aproximadamente 3”.

Sin embargo, es diferente realizar dicha resta, teniendo en cuenta que las variables que modelan son iguales  $f(a,b) = a-b$  tal que  $a=b$ , con lo que  $f(a,a)=a-a$ , y la extensión correcta de esta función es:

$$\mu_C(c) = \sup_{c=f(a)=a-a} \min\{\mu_A(a), \mu_A(a)\}$$

obteniéndose como resultado que el número clásico 0.

Los resultados anteriores se pueden explicar en términos de los  $\alpha$ -corte. Así, si  $a$  y  $b$  son variables independientes, al modelarlas como números borrosos y operar sus  $\alpha$ -cortes, se operan intervalos y cada intervalo resultante describe todo el rango de variación posible de la variable resultante. Sin embargo, si  $a$  y  $b$  son dependientes, al operar intervalos de las mismas el dominio se reduce y el intervalo resultante es menor o igual que el que se obtiene si fueran independientes. Si cada intervalo resultante o  $\alpha$ -corte es menor o igual que el que se tendría en caso de independencia,

esto quiere decir que el conjunto borroso resultante es más específico (o igual) que el que se tendría si las variables fueran independientes.

Por tanto, el resultado de aplicar aritmética borrosa (sin tener en cuenta la dependencia entre variables) es menos específico o igual que aplicar el principio de extensión a la función.

Por tanto podemos enunciar las siguientes conclusiones:

- Cuando una función  $f$  es continua y no todas las operaciones parciales son siempre crecientes (o decrecientes) en la misma variable, el principio de extensión proporciona distinto resultado que la extensión mediante aritmética borrosa  $F(A,B)$  y la aproximación  $F(A,B)$  es menos específica.
- Cuando una función  $f$  es continua y todas las operaciones parciales son siempre crecientes (o decrecientes) en la misma variable, el principio de extensión proporciona el mismo resultado que la extensión mediante aritmética borrosa.

Esta conclusión, presentada en [Klir, 1997] para el caso de funciones continuas se extiende, en el siguiente apartado, para el caso en que la función  $f$  no sea continua.

### 4.3.2. Extensión de funciones discontinuas mediante aritmética borrosa

Analizando ahora la extensión de una función monótona, continua o discontinua, donde todas sus variables son independientes se observa que es posible obtener la imagen borrosa exacta de la función mediante aritmética borrosa. En efecto, sea  $f$  una función compuesta  $f(a,b,c)=h(a,g(b,c))=a \otimes (b \oplus c)$  de variables independientes  $a$ ,  $b$  y  $c$ , siendo  $\otimes$  y  $\oplus$  dos operadores cualesquiera, continuos o discontinuos y donde las variables aparecen una sola vez (esto es, no se considera el caso como por ejemplo  $f(a,b,c)=(a \oplus b) \otimes (b \oplus c)$ )

La extensión de  $f$  a números borrosos, aplicando el principio de extensión viene dada por:

$$\begin{aligned}\mu_W(\omega) &= \sup_{\substack{\omega=f(a,b,c) \\ =h(a,g(b,c))}} \min\{\mu_A(a), \mu_B(b), \mu_C(c)\} = \sup_{\substack{\omega=h(a,u)=a\otimes u \\ u=g(b,c)=b\oplus c}} \min\{\mu_A(a), \min\{\mu_B(b), \mu_C(c)\}\} \\ &= \sup_{\omega=h(a,u)=a\otimes u} \min\left\{\mu_A(a), \sup_{u=g(b,c)=b\oplus c} \min\{\mu_B(b), \mu_C(c)\}\right\} = \sup_{\omega=h(a,u)=a\otimes u} \min\{\mu_A(a), \mu_U(u)\}\end{aligned}$$

Lo que quiere decir que es equivalente aplicar el principio de extensión a la función  $f$  a partir de sus parámetros, que aplicar el principio de extensión a las operaciones parciales o subexpresiones por separado para luego componerlas aplicando el principio de extensión a  $h$ . Por tanto siendo  $\omega=h(a,u)=a\otimes u$  y  $u=g(b,c)=b\oplus c$ , la función de pertenencia de  $W$  es igual que la función de pertenencia  $\mu_W=\mu_{A\otimes U}$  de un conjunto borroso  $W'=A\otimes U$  tal que  $U=B\oplus C$ , con función de pertenencia:

$$\mu_U(u) = \mu_{B\oplus C}(u) = \sup_{u=g(b,c)=b\oplus c} \min\{\mu_B(b), \mu_C(c)\}$$

Aun en el caso de que no exista independencia entre las variables, si una función es discontinua y todas las operaciones parciales son siempre crecientes (o decrecientes) en la misma variable, es posible obtener una aproximación válida de la imagen borrosa de la función aplicando aritmética borrosa.

En efecto, sea  $f(a,b)$  una función monótona discontinua no decreciente en  $a$  y en  $b$ . En  $f$  alguna de las variables aparece varias veces, y  $f$  se puede expresar como función compuesta de otras funciones, por ejemplo  $\omega=f(a,b)=h(g(a,b), m(a,b))$ , donde las funciones parciales  $h$ ,  $g$  y  $m$  son también monótonas no decrecientes en sus parámetros.

El resultado exacto de extender la función  $f$  a parámetros borrosos da como resultado un conjunto borroso  $W$  tal que:

$$\mu_W(\omega) = \sup_{\omega=f(a,b)} \min\{\mu_A(a), \mu_B(b)\}$$

Como  $f$  es no decreciente en sus parámetros, aplicando (4.13) se pueden calcular los límites inferiores y superiores de los  $\alpha$ -cortes de  $W$ :

$$\underline{\omega}(\alpha) = f(\underline{a}(\alpha), \underline{b}(\alpha)) \quad \bar{\omega}(\alpha) = f(\bar{a}(\alpha), \bar{b}(\alpha))$$

Si en lugar de aplicar el principio de extensión se aplica aritmética borrosa se obtiene un conjunto  $W'$  en general distinto de  $W$  pero con los límites exactos de los  $\alpha$ -cortes. El conjunto  $W'$  se puede calcular como  $W' = H(G(A,B), M(A,B)) = H(X,Y)$ . Mediante la función  $g$ , se operan por aritmética borrosa  $A$  y  $B$  y se obtiene  $X$ , y mediante  $m$  se obtiene  $Y$ . Después  $X$  e  $Y$  se operan mediante  $h$  y se obtiene como resultado  $W'$ .

En efecto, si  $x=g(a,b)$  e  $y=m(a,b)$ , y teniendo en cuenta que  $g$  y  $m$  son no decrecientes en sus parámetros, se puede aplicar (4.13) en la extensión de estas funciones a parámetros borrosos con lo que los límites de los  $\alpha$ -cortes de  $X$  e  $Y$  son:

$$\begin{aligned} \underline{x}(\alpha) &= g(\underline{a}(\alpha), \underline{b}(\alpha)) & \bar{x}(\alpha) &= g(\bar{a}(\alpha), \bar{b}(\alpha)) \\ \underline{y}(\alpha) &= m(\underline{a}(\alpha), \underline{b}(\alpha)) & \bar{y}(\alpha) &= m(\bar{a}(\alpha), \bar{b}(\alpha)) \end{aligned}$$

Aplicado de forma similar a la función  $h$ , se obtienen los límites de los  $\alpha$ -cortes de  $W'$  resultado de aplicar aritmética borrosa:

$$\begin{aligned} \underline{\omega}'(\alpha) &= h(\underline{x}(\alpha), \underline{y}(\alpha)) = h(g(\underline{a}(\alpha), \underline{b}(\alpha)), m(\underline{a}(\alpha), \underline{b}(\alpha))) = f(\underline{a}(\alpha), \underline{b}(\alpha)) = \underline{\omega}(\alpha) \\ \bar{\omega}'(\alpha) &= h(\bar{x}(\alpha), \bar{y}(\alpha)) = h(g(\bar{a}(\alpha), \bar{b}(\alpha)), m(\bar{a}(\alpha), \bar{b}(\alpha))) = f(\bar{a}(\alpha), \bar{b}(\alpha)) = \bar{\omega}(\alpha) \end{aligned}$$

Por tanto los límites de los  $\alpha$ -cortes de la extensión borrosa exacta  $W$  coinciden con los de  $W'$  resultado de aplicar aritmética borrosa a la operaciones parciales.

Igual que ocurría con funciones continuas, si no se tiene en cuenta la dependencia entre variables, como ocurre al extender una función mediante aritmética borrosa, en general la aproximación que se obtiene es menos específica que el resultado borroso de aplicar el principio de extensión (que sí tiene en cuenta dicha información), incluso en el caso de que se cumpla que todas las funciones parciales son monótonas en la misma variable. Aunque los límites de los  $\alpha$ -cortes sean exactos, en general la aproximación de la extensión borrosa de una función discontinua obtenida mediante aritmética borrosa es menos específica que la exacta.

Un caso interesante de función discontinua es el redondeo hacia arriba, ya que esta función aparece en los algoritmos de análisis de planificabilidad de los sistemas de tiempo real presentados en el capítulo 2, y que se desea extender a parámetros borrosos en los capítulos 5 y 6. La operación redondeo hacia arriba de  $a$  se expresa como  $\lceil a \rceil$ . En una operación del tipo  $a \lceil a+b \rceil$ , las funciones parciales son siempre no decrecientes en  $a$  y en  $b$  con lo cual, si se aplican las operaciones borrosas a los conjuntos borrosos, los límites inferiores y superiores obtenidos son correctos, aunque el conjunto resultante no es más que una aproximación. En el siguiente apartado se analiza la extensión de este tipo de funciones en detalle.

### 4.3.3. Redondeo borroso hacia arriba

En este apartado se muestran las extensiones borrosas que se obtienen al aplicar el principio de extensión y aritmética borrosa a funciones que contienen la operación redondeo hacia arriba.

La función redondeo por arriba de un número real  $b$  se define como:

$$f(b) = \lceil b \rceil = i \mid i-1 < b \leq i, \quad i \in \mathbb{Z}$$

O también como:  $f(b) = \lceil b \rceil = [b+1] - \left\lfloor \frac{[b]}{b} \right\rfloor$ , donde  $[b]$  es la parte entera de  $b$ .

Su extensión a conjuntos borrosos se puede definir aplicando el principio de extensión como:

$$\mu_{\lceil B \rceil}(i) = \sup_{\substack{i-1 < b \leq i \\ i \in \mathbb{Z}}} \mu_B(b)$$

Con lo que  $\lceil B \rceil$  es un conjunto borroso discreto.

Por ejemplo, sea un número borroso triangular definido por sus vértices  $B=(1.7, 1.93, 2.1)$ . De acuerdo con la definición previa de redondeo borroso hacia arriba, se tiene que  $\lceil B \rceil = (1/2, 0.6/3)$  donde el segundo valor  $b$  de cada par  $\alpha/b$  es un elemento del dominio  $\mathbb{Z}$  y el primer valor  $\alpha$  de cada par  $\alpha/b$  es el grado de pertenencia de  $b$  al conjunto borroso  $\lceil B \rceil$ . El conjunto borroso resultante  $\lceil B \rceil$  aparece representado en la Figura 4.11.

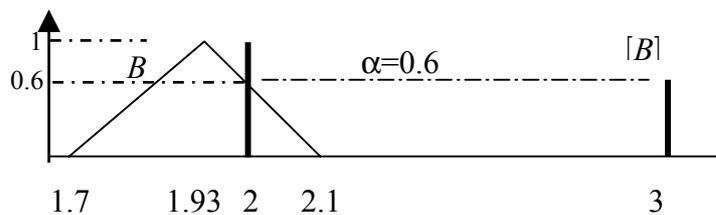


Figura 4.11: Redondeo borroso hacia arriba de  $B$

Se presentan en este apartado tres funciones que contienen redondeo borroso hacia

arriba  $a \cdot \lceil b \rceil$ ,  $a \cdot \lceil a+b \rceil$  y  $a \cdot \left\lceil \frac{a+b}{t_1} \right\rceil + b \cdot \left\lceil \frac{a+b}{t_2} \right\rceil$

**Ejemplo 1:  $a \cdot \lceil b \rceil$**

Aplicando el principio de extensión a la función  $a \cdot \lceil b \rceil$  se tiene:

$$\begin{aligned} \mu_W(\omega) &= \sup_{\omega = a \cdot \lceil b \rceil} \min \{ \mu_A(a), \mu_B(b) \} = \sup_{\substack{\omega = a \cdot i \\ i-1 < b \leq i \\ i \in \mathbb{Z}}} \min \{ \mu_A(a), \mu_B(b) \} = \\ &= \sup_{\substack{\omega = a \cdot i \\ i \in \mathbb{Z}}} \min \left\{ \mu_A(a), \sup_{\substack{i-1 < b \leq i \\ i \in \mathbb{Z}}} \mu_B(b) \right\} = \sup_{\substack{\omega = a \cdot i \\ i \in \mathbb{Z}}} \min \{ \mu_A(a), \mu_{\lceil B \rceil}(i) \} \end{aligned}$$

Es decir, si  $C = \lceil B \rceil$ , entonces  $\mu_W = \mu_{A \cdot \lceil B \rceil} = \mu_{A \cdot C}$  con lo que coinciden los resultados obtenidos al aplicar por un lado el principio de extensión y por otro lado aritmética borrosa.

**Ejemplo 2:  $a \cdot \lceil a+b \rceil$**

Aplicando el principio de extensión a la función  $a \cdot \lceil a+b \rceil$  se tiene:

$$\mu_W(\omega) = \sup_{\omega = a \cdot \lceil a+b \rceil} \min \{ \mu_A(a), \mu_B(b) \}$$

Este resultado no coincide con el obtenido calculando por separado el redondeo  $C = \lceil A+B \rceil$  y luego multiplicando por  $A$ , aplicando aritmética borrosa, ya que  $c$  depende de  $a$ . En efecto, se tiene:

$$\begin{aligned} \mu_{C = \lceil A+B \rceil}(c) &= \sup_{c = \lceil a+b \rceil} \min \{ \mu_A(a), \mu_B(b) \} \\ \mu_{A \cdot C}(\omega') &= \sup_{\omega' = a \cdot c} \min \{ \mu_A(a), \mu_C(c) \} \end{aligned}$$

Sin embargo, sí se pueden obtener algunas conclusiones interesantes acerca de los límites de los  $\alpha$ -cortes resultantes. En efecto, tal y como ya se ha dicho, la función de pertenencia de  $W$  aplicando el principio de extensión es:

$$\mu_W(\omega) = \sup_{\substack{\omega = f(a,b) \\ = a \cdot \lceil a+b \rceil}} \min \{ \mu_A(a), \mu_B(b) \}$$

donde la función  $f$  es discontinua (número infinito de discontinuidades, numerables, de salto finito) y no decreciente en  $a$  y en  $b$ . Por tanto, con las conclusiones obtenidas previamente (4.13), se pueden calcular los límites de los  $\alpha$ -cortes de  $W$  como:

$$\begin{aligned}\underline{\omega}(\alpha) &= \underline{a}(\alpha) \cdot \lceil \underline{a}(\alpha) + \underline{b}(\alpha) \rceil \\ \overline{\omega}(\alpha) &= \overline{a}(\alpha) \cdot \lceil \overline{a}(\alpha) + \overline{b}(\alpha) \rceil\end{aligned}$$

teniendo en cuenta que puede haber puntos del intervalo  $[\underline{\omega}(\alpha), \overline{\omega}(\alpha)]$  que no pertenezcan al  $\alpha$ -corte de  $W$ , ya que  $f$  es discontinua.

En la aproximación sugerida  $W' = A \cdot C$  tal que  $C = \lceil A + B \rceil$ , la función de pertenencia de  $C$  viene dada por:

$$\mu_C(c) = \sup_{\substack{c=g(a,b) \\ =\lceil a+b \rceil}} \min\{\mu_A(a), \mu_B(b)\}$$

Como la función  $g$  es no decreciente en  $a$  y en  $b$ , es posible aplicar (4.13) para obtener los  $\alpha$ -cortes de  $C$ :

$$\begin{aligned}\underline{c}(\alpha) &= \lceil \underline{a}(\alpha) + \underline{b}(\alpha) \rceil \\ \overline{c}(\alpha) &= \lceil \overline{a}(\alpha) + \overline{b}(\alpha) \rceil\end{aligned}$$

A continuación, el conjunto borroso resultante  $C$  se multiplica por  $A$  para obtener el conjunto borroso aproximado  $W'$  ( $W' = A \cdot C$ ), y su función de pertenencia, calculada aplicando el principio de extensión a la función  $h(a,c) = a \cdot c$ , es:

$$\mu_{W'}(\omega') = \sup_{\substack{\omega'=h(a,c) \\ =a \cdot c}} \min\{\mu_A(a), \mu_C(c)\}$$

Como la función  $h$  es no decreciente en  $a$  y en  $c$ , se pueden calcular los límites de los  $\alpha$ -cortes de  $W'$  a partir de los de  $A$  y los de  $C$  como:

$$\begin{aligned}\underline{\omega}'(\alpha) &= \underline{a}(\alpha) \cdot \underline{c}(\alpha) \\ \overline{\omega}'(\alpha) &= \overline{a}(\alpha) \cdot \overline{c}(\alpha)\end{aligned}$$

y sustituyendo por los límites de  $C$ , previamente definidos, se tiene:

$$\underline{\omega}'(\alpha) = \underline{a}(\alpha) \cdot \underline{c}(\alpha) = \underline{a}(\alpha) \cdot \lceil \underline{a}(\alpha) + \underline{b}(\alpha) \rceil = \underline{\omega}(\alpha)$$

$$\overline{\omega}'(\alpha) = \overline{a}(\alpha) \cdot \overline{c}(\alpha) = \overline{a}(\alpha) \cdot \lfloor \overline{a}(\alpha) + \overline{b}(\alpha) \rfloor = \overline{\omega}(\alpha)$$

Por tanto el cálculo aproximado resultante de realizar las operaciones borrosas en distintos pasos, conserva los límites inferiores y superiores de los  $\alpha$ -cortes, al ser  $f$  no decreciente en  $a$  para todas las operaciones parciales, y también en  $b$ . Esta conclusión permite obtener aproximaciones válidas de la imagen borrosa de esta función aplicando aritmética borrosa.

Sean por ejemplo  $A$  y  $B$  dos números borrosos triangulares,  $A = (0.9, 1.03, 1.1)$ ,  $B = (0.8, 0.9, 1)$  y sea la función  $f(a,b) = a \lceil a+b \rceil$ . La Figura 4.12 muestra la función de pertenencia que se obtiene al extender la función mediante el principio de extensión cuando los valores de  $a$  y  $b$  son respectivamente  $A$  y  $B$ .

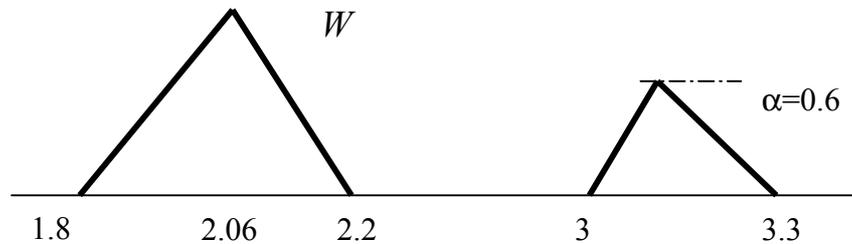


Figura 4.12: Resultado de la extensión de  $f$  a parámetros borrosos

La metodología gráfica propuesta en el anexo A permite obtener este mismo resultado, tal y como se muestra en la Figura 4.13. En este caso la t-norma utilizada es el mínimo y los puntos de  $\alpha_A = \alpha_B$  aparecen en la figura como líneas punteadas en la diagonal principal. Se puede ver que, para todas las parejas de valores de  $a$  y  $b$  que se encuentran por debajo de la línea  $a+b=2$  (discontinua gruesa), el valor de  $\omega = f(a,b) = a \lceil a+b \rceil$  es  $2 \cdot a$ . Las líneas verticales indican líneas de valores constantes de  $\omega$  que en la zona de  $\omega = 2 \cdot a$  va desde 1.8, hasta 2.2. En la línea  $a+b=2$  se produce una discontinuidad en  $f$  debido a la operación redondeo hacia arriba. Por encima de  $a+b=2$   $\omega = f(a,b) = a \lceil a+b \rceil$  vale  $3 \cdot a$ . De nuevo, las líneas verticales indican valores constantes de  $\omega$  y van desde 3 hasta 3.3 (a la derecha). Así queda especificado el soporte de  $W$ , tal y como se ve en la Figura 4.12.

El punto  $(a,b) = (1.03, 0.9)$  se corresponde a los valores de  $a$  y  $b$  para los que  $\alpha_A = \alpha_B = 1$ . En ese punto  $\omega = f(a,b) = 2 \cdot a = 2.06$  y  $\mu_W(\omega) = 1$ .

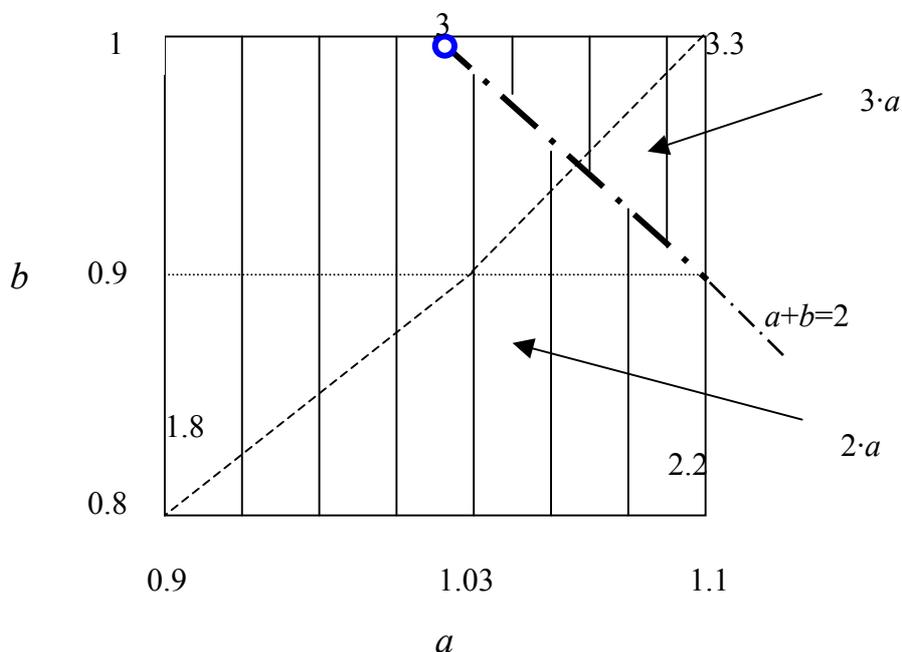


Figura 4.13: Representación gráfica de la extensión de  $f$  a parámetros borrosos mediante el principio de extensión

En la Figura 4.13, el grado de pertenencia de  $\omega=1.8$  es 0 ya que sólo el mayor elemento focal del cuerpo de evidencia conjunto contiene algún elemento de  $\omega=1.8$ , con lo que la integral correspondiente de la asignación básica es cero. El mayor elemento focal del cuerpo de evidencia conjunto es el correspondiente a  $[0.9, 1.1] \times [0.8, 1]$ . Lo que significa que, para la  $t$ -norma *min*, el grado de pertenencia de  $\omega=1.8$  es el  $\alpha$  correspondiente al corte de la línea vertical  $\omega=1.8$  con la diagonal principal  $\alpha_A = \alpha_B$ , esto es,  $\alpha=0$ .

El grado de pertenencia de  $\omega=2.2$  es el mismo que el de  $\omega=1.8$  ya que los elementos focales conjuntos que contienen al menos un elemento de  $\omega=2.2$  son los mismo que los de  $\omega=1.8$ , por tanto  $\mu_W(2.2)=0$ . Lo mismo se puede decir de  $\omega=3$ , con lo que  $\mu_W(3)=0$ .

En el caso de que interese calcular el mayor grado de pertenencia en la zona de  $\omega=3 \cdot a$ , éste se encuentra en el punto de corte entre la línea discontinua  $a+b=2$  y la diagonal  $\alpha_A = \alpha_B$ . El resultado es  $\alpha=0.588 \cong 0.6$  y  $a=1.058$ ,  $\omega=3.17$ . Ese es el punto de la zona  $\omega=3 \cdot a$  contenido en un mayor número de elementos focales conjuntos. A partir de ese punto el grado de pertenencia vuelve a decrecer hasta 0 en  $\omega=3.3$

Haciendo el cálculo aproximado mediante aritmética borrosa  $C=[A+B]$  y a continuación  $W'=A \cdot C$ :

$$A+B=(1.7, 1.93, 2.1)$$

$$C=[A+B]=1/2 \cup 0.6/3$$

La función de pertenencia de  $W'=A \cdot C$  es por tanto:

$$\begin{aligned} \mu_{A \cdot C}(\omega') &= \sup_{\omega'=a \cdot c} \min\{\mu_A(a), \mu_C(c)\} = \sup_{\omega'=a \cdot 2 \cup a \cdot 3} \min\{\mu_A(a), \mu_C(2) \cup \mu_C(3)\} = \\ &= \max_{\omega'} \left\{ \sup_{\omega'=a \cdot 2} \min\{\mu_A(a), 1\}, \sup_{\omega'=a \cdot 3} \min\{\mu_A(a), 0.6\} \right\} = \max_{\omega'} \left\{ \mu_A\left(\frac{\omega'}{2}\right), \min\left\{\mu_A\left(\frac{\omega'}{3}\right), 0.6\right\} \right\} \end{aligned}$$

En la Figura 4.14 se comparan  $W'$  en línea continua y el conjunto borroso exacto  $W$  en línea discontinua. Como se aprecia en la figura, la aproximación conserva, de cada  $\alpha$ -corte, el límite inferior y el límite superior. La aproximación  $W'$  es menos específica que  $W$ , ya que al realizar las operaciones parciales mediante aritmética borrosa se olvida la dependencia existente entre  $a$  y  $c$ .

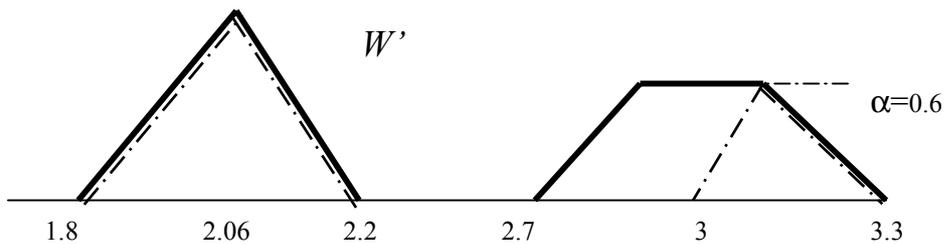


Figura 4.14:  $W'$  obtenido mediante aritmética borrosa

**Ejemplo 3:**  $a \cdot \left[ \frac{a+b}{t_1} \right] + b \cdot \left[ \frac{a+b}{t_2} \right]$

Supongamos que  $a$  y  $b$  se modelan mediante números borrosos  $A$  y  $B$  y que  $t_1$  y  $t_2$  son numéricos y mayores que cero.

Aplicando el principio de extensión a la función  $a \cdot \left[ \frac{a+b}{t_1} \right] + b \cdot \left[ \frac{a+b}{t_2} \right]$  se tiene:

$$\mu_W(\omega) = \sup_{\omega = a \cdot \left[ \frac{a+b}{t_1} \right] + b \cdot \left[ \frac{a+b}{t_2} \right]} \min\{\mu_A(a), \mu_B(b)\}$$

Una posible aplicación de la aritmética borrosa sería:

$$C = \left[ \frac{A+B}{t_1} \right], \quad D = \left[ \frac{A+B}{t_2} \right]$$

y calculando  $W'$  como  $W' = A \cdot C + B \cdot D$

El resultado no es exacto ya que de esta forma no se tiene en cuenta que tanto  $c$  como  $d$  dependen de  $a$  y de  $b$ . Sin embargo, igual que en el caso anterior, se conservan los límites inferiores y superiores del conjunto borroso resultante.

En efecto, la función  $f(a,b) = a \cdot \left[ \frac{a+b}{t_1} \right] + b \cdot \left[ \frac{a+b}{t_2} \right]$  es no decreciente en  $a$  y en  $b$ ,

luego se pueden calcular los límites de los  $\alpha$ -cortes de  $W$  como:

$$\begin{aligned} \underline{\omega}(\alpha) &= \underline{a}(\alpha) \cdot \left[ \frac{\underline{a}(\alpha) + \underline{b}(\alpha)}{t_1} \right] + \underline{b}(\alpha) \cdot \left[ \frac{\underline{a}(\alpha) + \underline{b}(\alpha)}{t_2} \right] \\ \overline{\omega}(\alpha) &= \overline{a}(\alpha) \cdot \left[ \frac{\overline{a}(\alpha) + \overline{b}(\alpha)}{t_1} \right] + \overline{b}(\alpha) \cdot \left[ \frac{\overline{a}(\alpha) + \overline{b}(\alpha)}{t_2} \right] \end{aligned}$$

Como las funciones intermedias utilizadas para calcular  $c = \left[ \frac{a+b}{t_1} \right]$ ,  $d = \left[ \frac{a+b}{t_2} \right]$  y finalmente  $\omega' = a \cdot c + b \cdot d$  son también no decrecientes en  $a$  y en  $b$ , se tiene que

$$\begin{aligned} \underline{\omega}(\alpha) &= \underline{\omega}'(\alpha) \\ \overline{\omega}(\alpha) &= \overline{\omega}'(\alpha) \end{aligned}$$

Por ejemplo, sean  $A$  y  $B$  dos números borrosos triangulares  $A = (0.9, 1.03, 1.1)$ ,  $B = (0.8, 0.9, 1)$ , y sean  $t_1=1$ ,  $t_2=2$ , entonces el conjunto borroso  $W$  que se obtiene al extender mediante el principio de extensión la función  $f(a,b) = a \cdot \left[ \frac{a+b}{t_1} \right] + b \cdot \left[ \frac{a+b}{t_2} \right]$  es el que se muestra en la Figura 4.15.

El resultado  $W$  de extender la función  $f$  mediante el principio de extensión, que aparece en la Figura 4.15, también se puede obtener a partir de la representación gráfica descrita en el anexo A, que se muestra en la Figura 4.16.

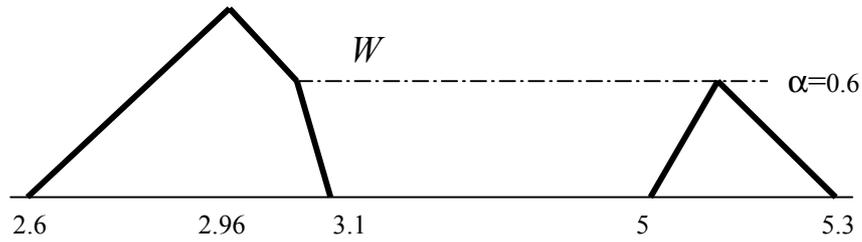


Figura 4.15: Resultado de la extensión de  $f$  a parámetros borrosos

En este caso se distinguen en la Figura 4.16 dos zonas, entre las cuales hay una discontinuidad situada en la línea  $a+b=2$ , que corresponde a la discontinuidad en el primer redondeo de  $f, \lceil (a+b)/t_1 \rceil$  con  $t_1=1$  y también coincide con la discontinuidad del segundo redondeo de  $f, \lceil (a+b)/t_2 \rceil$  con  $t_2=2$ , que se produce cuando  $(a+b)/t_2 = 1$ .

Para las parejas de puntos  $(a,b)$  que se encuentran por debajo de  $a+b=2$ , se tiene que  $\omega=f(a,b)=2 \cdot a+b$ . En dicha zona aparecen representadas las rectas  $\omega=2 \cdot a+b=cte$  como rectas oblicuas de pendiente negativa que van desde  $\omega=2.6$  en el límite inferior izquierdo, hasta  $\omega=3.1$  para  $a=1.1$  y  $b=0.9$ .

Para las parejas  $(a,b)$  que se encuentran por encima de  $a+b=2$ , se tiene  $\omega=f(a,b)=3 \cdot a+2 \cdot b$ . En dicha zona aparecen representadas las rectas  $\omega=3 \cdot a+2 \cdot b=cte$  como rectas oblicuas de pendiente negativa que van desde  $\omega=5$  para  $a=1$  y  $b=1$ , hasta  $\omega=5.3$  para  $a=1.1$  y  $b=1$ .

De esta forma se determina el soporte de  $W$  tal y como aparece en la Figura 4.15.

El grado de pertenencia de  $\omega=2.6$  es cero ya que sólo el elemento focal conjunto mayor de todos contiene a ese punto. A partir de ahí, el grado de pertenencia va creciendo y se calcula como el  $\alpha$  correspondiente al corte entre la recta  $\omega=constante$  y  $\alpha_A=\alpha_B$ . El grado de pertenencia es 1 para  $\omega=2.96$ , que es la recta que pasa por  $\alpha_A=\alpha_B=1$ , ya que todos los elementos focales conjuntos contienen al menos un punto de  $\omega=2.96$ . A partir de ese valor de  $\omega$  el grado de pertenencia va decreciendo hasta  $\omega=3.1$  donde vale cero. De nuevo, el grado de pertenencia de  $\omega=5$  es cero, porque sólo el mayor elemento focal conjunto contiene a ese punto. El grado de pertenencia va creciendo hasta el máximo en esta zona, que se encuentra en el punto de corte entre  $\alpha_A=\alpha_B$  y la línea de discontinuidad  $a+b=2$ . El resultado es  $\alpha=0.588 \cong 0.6$  y  $\omega=5.06$ . Ese es el punto de la zona  $\omega=3 \cdot a+2 \cdot b$  contenido en un mayor número de elementos focales conjuntos. A partir de ese punto el grado de pertenencia vuelve a decrecer hasta 0 en  $\omega=5.3$

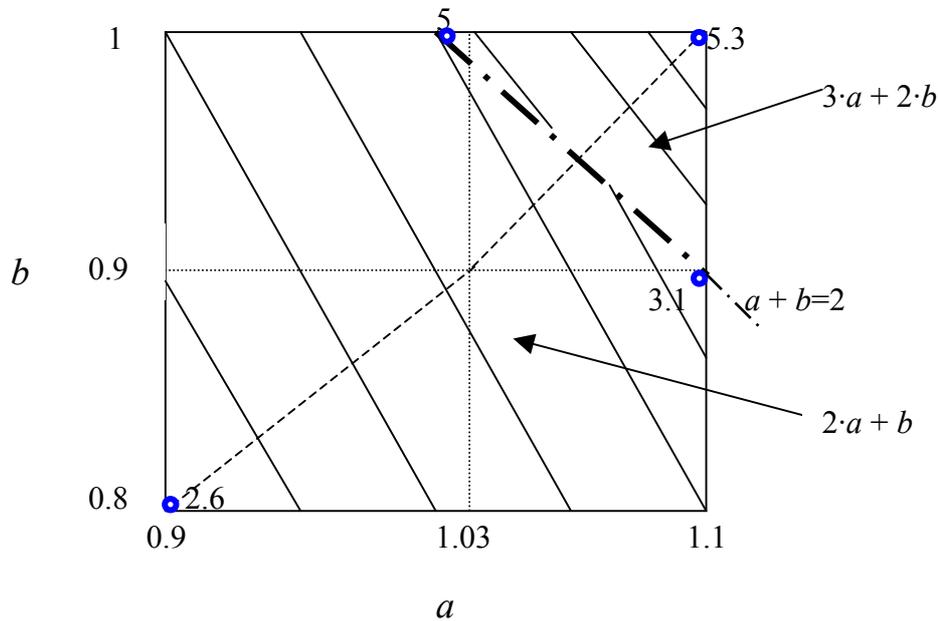


Figura 4.16: Representación gráfica de la extensión de  $f$  a parámetros borrosos mediante el principio de extensión

Haciendo el cálculo aproximado mediante aritmética borrosa se tiene:

$$A+B=(1.7, 1.93, 2.1), (A+B)/t_1=(1.7, 1.93, 2.1) \text{ y } (A+B)/t_2=(0.85, 0.965, 1.05)$$

Aplicando el redondeo se obtiene:

$$C=\lceil (A+B)/t_1 \rceil = 1/2 \cup 0.6/3$$

$$D=\lceil (A+B)/t_2 \rceil = 1/1 \cup 0.6/2$$

$$W'=A \cdot C + B \cdot D$$

En realidad, sólo habría que operar  $A \cdot C + B \cdot D = (A \cdot (1/2) + B \cdot (1/1)) \cup (A \cdot (0.6/3) + B \cdot (0.6/2))$  debido a la relación entre  $C$  y  $D$ , pero si no lo tenemos en cuenta, operaríamos todas las combinaciones:

$$W' = \{ A \cdot (1/2) + B \cdot (1/1) \} \cup \{ A \cdot (1/2) + B \cdot (0.6/2) \} \cup \{ A \cdot (0.6/3) + B \cdot (1/1) \} \cup \\ \{ A \cdot (0.6/3) + B \cdot (0.6/2) \}$$

En la Figura 4.17 se muestra el resultado de cada combinación:

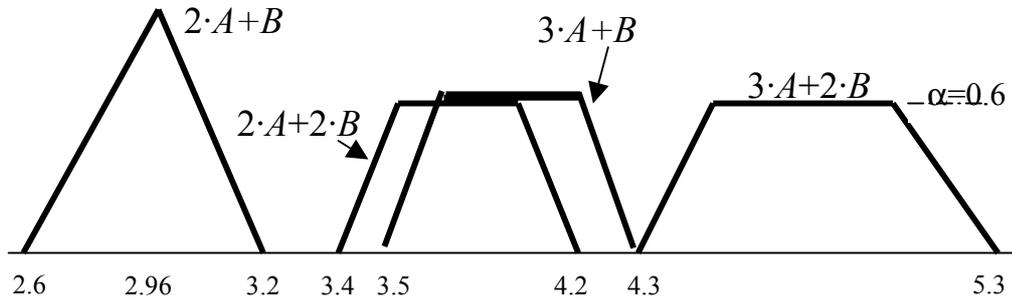


Figura 4.17:  $W'$  como unión de conjuntos borrosos

La unión, obtenida con el operador  $\max$ , es el conjunto  $W'$ , que se muestra en la Figura 4.18 en línea continua y se compara con el conjunto borroso  $W$  resultado de aplicar el principio de extensión en línea discontinua.

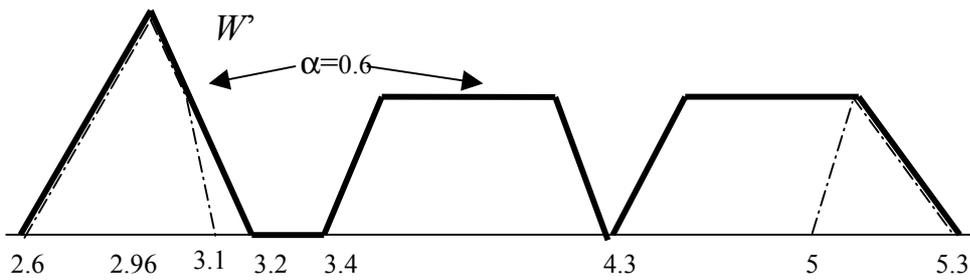


Figura 4.18:  $W'$  obtenido mediante aritmética borrosa

Como se puede apreciar, la aproximación  $W'$  obtenida mediante aritmética borrosa conserva el límite inferior y el límite superior de cada  $\alpha$ -corte.

En el ejemplo que se acaba de presentar, los valores enteros con posibilidad distinta de cero en  $C$  y  $D$  tienen posibilidad 1 y 0.6 en ambos casos. El hecho de que coincidan los valores de posibilidad en ambos conjuntos se debe a los valores escogidos de  $t_1$  y  $t_2$ .

Si en lugar de los anteriores tomamos  $t_1=1$  y  $t_2=1.9$ , existe una combinación adicional que es  $(2\cdot a+2\cdot b)$ , como se aprecia en la Figura 4.20, aunque la conclusión sigue siendo válida. En la Figura 4.19 aparece  $W$  que es el resultado de la extensión de  $f$  a parámetros borrosos mediante el principio de extensión, con los nuevos valores de  $t_1$  y  $t_2$ .

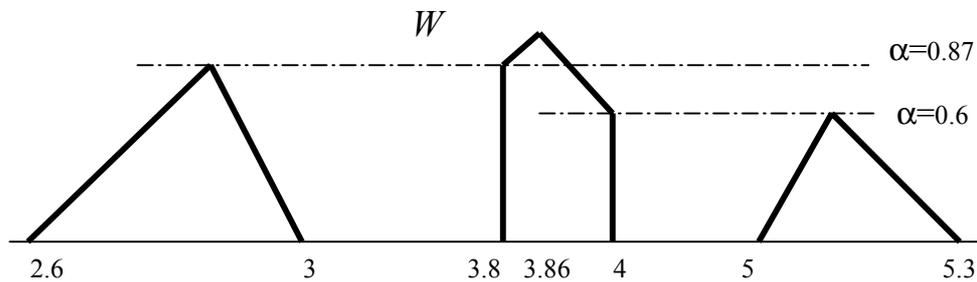


Figura 4.19: Resultado de la extensión de  $f$  a parámetros borrosos

$W$  también se puede obtener mediante la representación gráfica propuesta en esta tesis, tal y como indica la Figura 4.20. En este caso se pueden distinguir tres zonas, con dos líneas de discontinuidad, la primera  $a+b=2$ , que corresponde al primer redondeo de  $f$  ( $\lceil(a+b)/t_1$ ) y la segunda  $a+b=1.9$ , que corresponde al segundo redondeo de  $f$  ( $\lceil(a+b)/t_2$ ).

Para los puntos situados por debajo de  $a+b=1.9$  se tiene  $\omega=f(a,b)=2\cdot a+b$  y las rectas de pendiente negativa en esta zona son las rectas que cumplen  $\omega=2\cdot a+b=cte$ . En esta zona  $\omega$  varía desde  $\omega=2.6$  para  $a=0.9$  y  $b=0.8$  hasta  $\omega=3$  para  $a=1.1$  y  $b=0.8$ . En ambos puntos el grado de pertenencia vale cero ya que el único elemento focal conjunto que contiene a esos puntos es el mayor de todos. Desde  $\omega=2.6$  el grado de pertenencia va creciendo hasta el máximo en esta zona que se encuentra en el punto de corte entre  $\alpha_A=\alpha_B$  y la recta  $a+b=1.9$ . El resultado es  $\alpha=0.87$ . A partir de ese valor, el grado de pertenencia decrece hasta cero en  $\omega=3$ .

Para los puntos situados por encima de  $a+b=1.9$  y por debajo de  $a+b=2$  se tiene  $\omega=f(a,b)=2\cdot a+2\cdot b$  y las rectas de pendiente negativa en esta zona son las rectas que cumplen  $\omega=2\cdot a+2\cdot b=cte$ . En esta zona  $\omega$  varía desde  $\omega=3.8$  para  $a+b=1.9$  hasta  $\omega=4$  para  $a+b=2$ . El grado de pertenencia de  $\omega=3.8$  es el máximo de la zona anterior y vale  $\alpha=0.87$  (ya que están involucrados los mismos elementos focales conjuntos en uno y otro caso). Desde  $\alpha=0.87$  el grado de pertenencia crece hasta llegar a  $\alpha=1$  en  $\omega=3.86$  (para  $a=1.03$  y  $b=0.9$ ). Desde ese valor, decrece hasta que en  $\omega=4$  el grado de pertenencia se hace 0.6.

Para los puntos situados por encima de  $a+b=2$  se tiene que  $\omega=f(a,b)=3\cdot a+2\cdot b$  y las rectas de pendiente negativa en esta zona son las rectas que cumplen  $\omega=3\cdot a+2\cdot b=cte$ . En esta zona  $\omega$  varía desde  $\omega=5$  para  $a=1$  y  $b=1$ , hasta  $\omega=5.3$  para  $a=1.1$  y  $b=1$ . El grado de pertenencia de  $\omega=5$  es cero y va creciendo con  $\omega$  hasta llegar a  $\alpha=0.6$ , el máximo en esta zona, que coincide con el grado de pertenencia de  $\omega=4$  en la zona anterior y se produce en el punto de corte entre  $\alpha_A=\alpha_B$  y  $a+b=2$ . A

partir de ese valor, al crecer  $\omega$  el grado de pertenencia va decreciendo hasta llegar a cero en  $\omega=5.3$ .

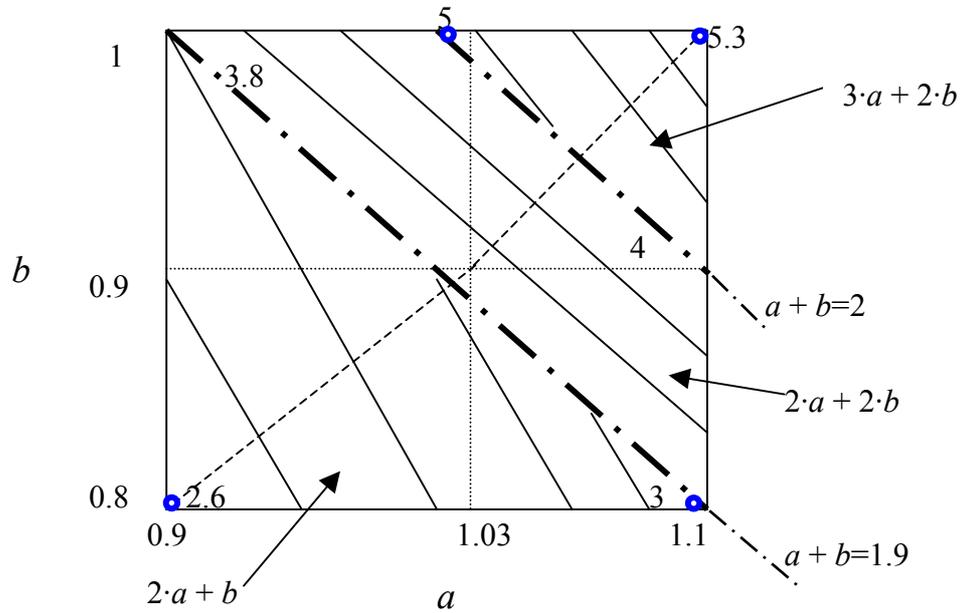


Figura 4.20: Representación gráfica de la extensión de  $f$  a parámetros borrosos mediante el principio de extensión

Aplicando aritmética borrosa el resultado se podría obtener como  $A+B=(1.7, 1.93, 2.1)$ ,  $(A+B)/t_1=(1.7, 1.93, 2.1)$  y  $(A+B)/t_2=(0.89, 1.016, 1.105)$

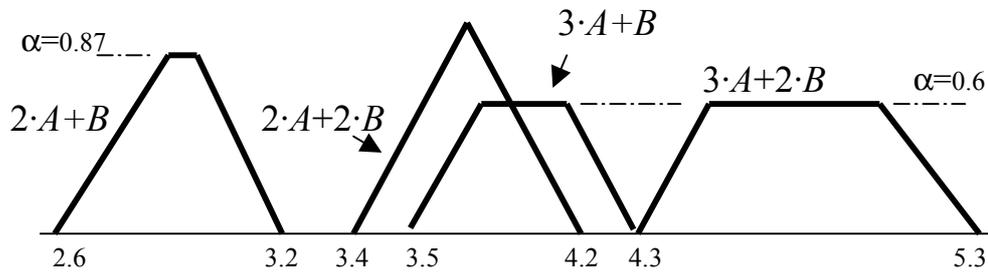
Aplicando el redondeo se tiene:

$$C = \lceil (A+B)/t_1 \rceil = 1/2 \cup 0.6/3$$

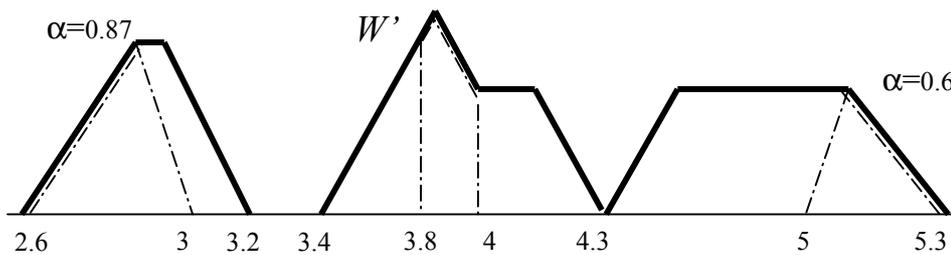
$$D = \lceil (A+B)/t_2 \rceil = 0.87/1 \cup 1/2$$

Finalmente  $W'$  se obtiene como  $W' = A \cdot C + B \cdot D$  (ver Figura 4.21).

$$W' = \{ A \cdot (1/2) + B \cdot (0.87/1) \} \cup \{ A \cdot (1/2) + B \cdot (1/2) \} \cup \{ A \cdot (0.6/3) + B \cdot (0.87/1) \} \cup \{ A \cdot (0.6/3) + B \cdot (1/2) \}$$

Figura 4.21:  $W'$  como unión de conjuntos

La unión de los conjuntos se calcula como el máximo, con lo que el conjunto resultante  $W'$  se muestra en la Figura 4.22 con línea continua, y al compararlo con  $W$  que aparece con línea discontinua, vemos que ambos tienen los mismos límites inferiores y superiores para cada  $\alpha$ -corte

Figura 4.22:  $W'$  obtenido mediante aritmética borrosa

En este capítulo se ha demostrado que cuando una función es monótona en sus variables es posible extender la función a argumentos borrosos aplicando aritmética de intervalos de forma que el resultado obtenido es una aproximación válida del conjunto borroso exacto que proporciona el principio de extensión.

Además, si la función es monótona siempre no creciente o siempre no decreciente en la misma variable en todas las operaciones parciales que involucran a dicha variable, entonces es posible obtener otra aproximación válida del conjunto borroso exacto mediante la aplicación de aritmética borrosa.

Las operaciones analizadas en este capítulo, que incluyen redondeos hacia arriba, aparecerán en el siguiente capítulo dentro de los algoritmos de cálculo de los tiempos de finalización de las tareas en sistemas de tiempo real, por lo que resulta de gran interés la extensión borrosa estas operaciones y sus aproximaciones válidas, con el fin de obtener de forma rápida los valores exactos de planificabilidad borrosa.

## **5. Análisis borroso de planificabilidad en sistemas con prioridades fijas**

En capítulos anteriores se ha propuesto una definición de planificabilidad borrosa y se ha enunciado la condición que debe cumplir toda función para que se pueda extender a argumentos borrosos mediante aritmética borrosa o mediante aritmética de intervalos, con los mismos resultados que la aplicación del principio de extensión. En este capítulo se proponen extensiones borrosas del análisis de planificabilidad para sistemas con prioridades fijas, generalizando los tests necesarios y suficientes para parámetros numéricos (“crisp”) ya existentes en la literatura. Las extensiones se basan en aritmética borrosa y en aritmética de intervalos presentadas en el capítulo anterior.

En primer lugar, en el apartado 5.1 se describe el algoritmo de análisis de planificabilidad con parámetros numéricos, cuando los plazos son menores que los periodos. Este algoritmo calcula los peores tiempos de respuesta de las tareas supuesto que existe un instante crítico.

En el apartado 5.2 se extiende este algoritmo a plazos y tiempos de ejecución borrosos, mediante el principio de extensión, aritmética borrosa y aritmética de intervalos, y se resuelve también la extensión a periodos borrosos, y a tiempos de bloqueo y “release jitter” o retrasos borrosos de las activaciones.

En el apartado 5.3 se define el concepto de óptimo en la asignación de prioridades fijas cuando los parámetros temporales de las tareas se modelan como conjuntos borrosos, concepto que se basa en la maximización de la posibilidad y necesidad de planificabilidad. También se enuncia la propiedad que deben cumplir los plazos para que la asignación obtenida de acuerdo con el orden de los plazos borrosos sea óptima.

Las extensiones propuestas previamente se comparan, en el apartado 5.4, con la extensión propuesta por Litoiu y Tadei para el análisis de sistemas con prioridades fijas y plazos menores que los periodos basada en aritmética borrosa. Se muestra que los resultados obtenidos por los autores pueden llegar a ser muy pesimistas debido a la simplificación de algunas operaciones borrosas.

Una vez propuestas las extensiones borrosas del análisis de planificabilidad, en el apartado 5.5 se hace un estudio comparativo entre los resultados que se obtienen modelando los parámetros mediante números borrosos y los que se obtienen modelándolos mediante distribuciones de probabilidad. Para ello se utilizan dos tipos de funciones de transformación entre distribuciones de posibilidad y de probabilidad descritas en el anexo C y se observa que los resultados que se obtienen con el modelo probabilista son iguales o muy parecidos a los que se obtienen con el modelo posibilista.

En el apartado 5.6 se describe el algoritmo de análisis temporal con plazos mayores que los periodos y se extiende a parámetros borrosos, teniendo en cuenta la

existencia de tiempos de bloqueo por compartir recursos y retardos modelados como “release jitter”.

En el apartado 5.7 se extiende a modo de ejemplo el algoritmo de cálculo de los tiempos de bloqueo cuando el algoritmo de sincronización es el PIP (descrito en el capítulo 2).

Por último, en el apartado 5.8 se extiende el análisis de planificabilidad de sistemas con tareas aperiódicas cuando éstas son atendidas por dos tipos de servidor aperiódico: el Servidor Esporádico y el Servidor Aplazable, y las tareas periódicas tienen sus plazos menores o iguales a sus periodos.

## 5.1. Algoritmo de análisis de planificabilidad con plazos menores que los periodos

Joseph y Padya [Joseph, Padya, 1986] y Audsley [Audsley et al., 1991] desarrollaron de forma independiente el algoritmo (5.1) para calcular el peor tiempo de respuesta de las tareas en un sistema de tiempo real con tareas periódicas, plazos menores que los periodos, expulsables y con prioridades fijas, supuesto que existe un instante crítico (ver el capítulo 2). Ese algoritmo, permite obtener una condición necesaria y suficiente de planificabilidad.

Para ello se calcula de forma iterativa el tiempo de finalización  $wc_i$  de cada tarea en su primera ejecución. Solamente se calcula la primera ejecución de la tarea, ya que cuando las prioridades son fijas, el peor tiempo de respuesta de una tarea se da en su primera ejecución a partir del instante crítico.

En (5.1) las tareas se encuentran ordenadas según su prioridad de forma que la tarea 1 tiene la prioridad más alta, su tiempo de ejecución es  $p_1$ , su plazo es  $d_1$  y su periodo  $t_1$ . Para cada tarea  $i$ , su tiempo de finalización en la primera ejecución, o lo que es lo mismo, su peor tiempo de respuesta  $wc_i$  viene dado por el valor final de  $s_{i,k}$  cuando termina su cálculo iterativo. El valor de  $s_{i,k}$  va aumentando en las sucesivas iteraciones hasta que se estabiliza cuando  $s_{i,k} = s_{i,k-1}$  obteniéndose el peor tiempo de respuesta  $wc_i$  de la tarea  $i$ . Cuanto más cerca se encuentre el valor inicial de  $s$ ,  $s_{in_i}$ , del tiempo de finalización, más rápido encuentra el algoritmo la solución. En cada iteración se compara  $s_{i,k}$  con el plazo de la tarea correspondiente. Como se cumple que  $s_{i,k} \leq wc_i$ , si  $s_{i,k}$  es mayor que el plazo, entonces el tiempo de finalización también es mayor que el plazo, se interrumpe el cálculo de  $wc_i$  y se concluye que la tarea  $i$  no es planificable.

```

CRISP ALG
wc0 = 0
for i=1 to n_tareas do
  ALG BASICO:
  (5.1) {
    si,0 = s_ini
    while (si,k-1 < si,k(si,k-1))
      { if si,k > di tarea i no planificable; exit
        k = k + 1 }
    wci = si,k
    dado, si,k(si,k-1) = ∑j=1i pj · ⌈  $\frac{s_{i,k-1}}{t_j}$  ⌉
  }
end

```

En la ecuación principal de (5.1):

$$s_{i,k}(s_{i,k-1}) = \sum_{j=1}^i p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

donde  $\lceil \cdot \rceil$  representa la operación redondeo hacia arriba, los términos del sumatorio que van de  $j=1$  hasta  $j=i-1$  son la interferencia (o tiempo de CPU ocupada) de las tareas más prioritarias que  $i$  en la ejecución de  $i$ . Es decir, en el tiempo  $s_{i,k}$  se contabiliza cuántas veces se ejecutan las tareas más prioritarias, y que retrasan por tanto la ejecución de la tarea  $i$ .

Si se modifica el algoritmo (5.1) para que continúe iterando hasta que se obtiene el tiempo de finalización de la tarea, se obtiene (5.2) y de esta forma es posible calcular  $wc_i$  aunque éste sobrepase su plazo. Como se verá más adelante, esta formulación resultará especialmente útil para su extensión a parámetros borrosos. Además cuando se detecta que se ha sobrepasado el periodo se termina la iteración (esto es, finaliza el bucle *while*), porque este algoritmo no es válido para calcular los tiempos de finalización de las tareas cuando éstos sobrepasan los periodos.

Si no se terminara la iteración al sobrepasar el periodo ( $s_{i,k}$  vuelve a ser mayor que  $s_{i,k-1}$  y se sigue ejecutando el bucle *while*), lo que se obtendría es el tiempo de finalización de la tarea  $i$  en su ejecución  $m$ , siendo  $m$  la primera ejecución en la que la tarea consigue finalizar dentro de su periodo (suponiendo que antes de la ejecución  $m$  se finalizaron las ejecuciones anteriores). Si no existieran plazos, siendo el único requisito que todas las tareas se ejecuten, este análisis permitiría comprobar si es posible cumplir el requisito, aunque una ejecución termine en algún periodo posterior. La condición para ello es que la primera ejecución que ha podido finalizar dentro de su periodo termine dentro del hiperperiodo.

```

CRISP ALG
 $wc_0 = 0$ 
for  $i=1$  to  $n\_tareas$  do
  ALG BASICO:
  {
     $s_{i,0} = s\_in_i$ 
    while  $(s_{i,k-1} < s_{i,k}(s_{i,k-1})) \text{ AND } (s_{i,k} \leq t_i) \quad k = k+1$ 
     $wc_i = s_{i,k}$ 
    dado,  $s_{i,k}(s_{i,k-1}) = \sum_{j=1}^i p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$ 
  }
  if  $(wc_i > d_i)$  tarea  $i$  no planificable
end
    
```

(5.2)

En la literatura se han encontrado distintas inicializaciones posibles del algoritmo. Así, en [Shin, Ramanathan, 1994] el valor inicial de  $s$ ,  $s\_in_i$  es  $\sum_{j=1}^i p_j$ . Este algoritmo se aceleró en [Litoiu, Tadei, 1997] refinando el valor inicial de  $s$  con  $s\_in_i = wc_{i-1} + p_i$  ya que está asegurado que lo más pronto que puede terminar una tarea  $i$  es el instante en que finaliza la tarea anterior  $i-1$  más su propio tiempo de ejecución  $p_i$ .

A título de ejemplo, para ilustrar el algoritmo (5.2), supongamos que se tiene un sistema de tiempo real formado por tres tareas periódicas cuyos periodos son  $t_1=3$ ,  $t_2=5$  y  $t_3=15$ , con tiempos de ejecución  $p_1= 1$ ,  $p_2= 1.9$  y  $p_3= 1$  y con plazos iguales a los periodos.

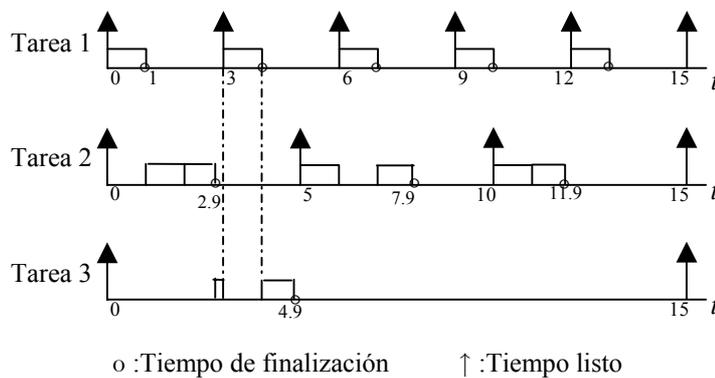


Figura 5.1: Gráfico de planificación de un sistema con prioridades fijas

En la Figura 5.1 se muestra el gráfico de planificación de este sistema en un hiperperiodo. Cuando las prioridades se asignan de acuerdo con el algoritmo RMS, la tarea 1 es la más prioritaria ya que tiene el menor periodo y la tarea 3 la menos

prioritaria. Al terminar la tarea 1 su ejecución, se ejecuta la tarea 2 hasta que finaliza en el instante 2.9. A continuación la tarea 3 comienza a ejecutarse, pero después de 0.1 unidades de tiempo, en el instante 3, la tarea 1 vuelve a estar lista e interrumpe a la tarea 3. Cuando la tarea 1 termina su segunda ejecución, la tarea 3 continúa hasta finalizar en 4.9. Por tanto, los peores tiempos de respuesta de las tareas valen 1, 2.9 y 4.9 respectivamente, y el sistema es planificable.

Aplicando el algoritmo (5.2), con la inicialización  $s_{in_i} = wc_{i-1} + p_i$ , se tiene:

*Tarea 1:*

$p_1 = 1, d_1 = 3, t_1 = 3$
$s_{1,0} = s_{in_1} = wc_0 + p_1 = p_1 = 1$
$s_{1,1} = p_1 \lceil s_{1,0} / t_1 \rceil = 1 \cdot 1 = 1 = s_{1,0}$
$wc_1 = s_{1,1} = 1$

Por tanto  $wc_1 = 1$ , y como se cumple que  $wc_1 < d_1$ , la tarea 1 es planificable.

*Tarea 2:*

$p_1 = 1, p_2 = 1.9, d_1 = 3, d_2 = 5, t_1 = 3, t_2 = 5$
$s_{2,0} = s_{in_2} = wc_1 + p_2 = 1 + 1.9 = 2.9$
$s_{2,1} = p_1 \lceil s_{2,0} / t_1 \rceil + p_2 \lceil s_{2,0} / t_2 \rceil = 1 \cdot \lceil 2.9/3 \rceil + 1.9 \cdot \lceil 2.9/5 \rceil = 1 + 1.9 = 2.9 = s_{2,0}$
$wc_2 = s_{2,1} = 2.9$

Por tanto  $wc_2 = 2.9$ , y como se cumple que  $wc_2 < d_2$ , la tarea 2 también es planificable.

Tarea 3:

$p_1=1, p_2=1.9, p_3=1, d_1=3, d_2=5, d_3=15, t_1=3, t_2=5, t_3=15$
$s_{3,0} = wc_2 + p_3 = 2.9 + 1 = 3.9$
$s_{3,1} = p_1 \lceil s_{3,0} / t_1 \rceil + p_2 \lceil s_{3,0} / t_2 \rceil + p_3 \lceil s_{3,0} / t_3 \rceil = 1 \cdot \lceil 3.9/3 \rceil + 1.9 \cdot \lceil 3.9/5 \rceil + 1 \cdot \lceil 3.9/15 \rceil = 1 \cdot 2 + 1.9 + 1 = 4.9$
$s_{3,2} = p_1 \lceil s_{3,1} / t_1 \rceil + p_2 \lceil s_{3,1} / t_2 \rceil + p_3 \lceil s_{3,1} / t_3 \rceil = 1 \cdot \lceil 4.9/3 \rceil + 1.9 \cdot \lceil 4.9/5 \rceil + 1 \cdot \lceil 4.9/15 \rceil = 1 \cdot 2 + 1.9 + 1 = 4.9 = s_{3,1}$
$wc_3 = s_{3,2} = 4.9$

El peor tiempo de respuesta de la tarea 3 es  $wc_3 = 4.9$ , y como se cumple que  $wc_3 < d_3$ , la tarea 3 también es planificable. Por tanto, tal y como ya se dijo antes, el sistema es planificable.

Este análisis se puede ampliar para tener en cuenta la existencia de tiempos de bloqueo en las tareas, causados por el acceso de una tarea menos prioritaria a un recurso compartido (ver capítulo 2). En efecto, si denominamos  $b_i$  al tiempo de bloqueo de una tarea  $i$ , el tiempo de finalización de cada tarea se obtiene calculando  $s$  de forma iterativa según (5.3) hasta que se verifica  $s_{i,k} = s_{i,k-1}$ .

En este algoritmo el tiempo de bloqueo se trata como un tiempo extra de ejecución al comienzo del periodo de ocupación, aumentando los tiempos de respuesta de las ejecuciones en dicho periodo.

$$(5.3) \quad s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

siendo la inicialización de  $s$ :

$$(5.4) \quad s_{i,0} = b_i + \sum_{j=1}^i p_j$$

Además se puede tener en cuenta la existencia de “release jitter”  $j_i$ , de forma que la ecuación a resolver iterativamente es:

$$(5.5) \quad s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1} + j_j}{t_j} \right\rceil$$

Cuando  $s_{i,k} = s_{i,k-1}$ , el tiempo de finalización de la tarea  $i$  es  $wc_i = s_{i,k} + j_i$

## 5.2. Extensión del análisis de planificabilidad con plazos menores que los periodos

En primer lugar se va a analizar la extensión del análisis anterior a plazos y tiempos de ejecución borrosos aplicando tres métodos diferentes: el principio de extensión (apartado 5.2.1), aritmética borrosa (apartado 5.2.2) y aritmética de intervalos (apartado 5.2.3). Las extensiones propuestas basadas en aritmética borrosa y en aritmética de intervalos proporcionan los mismos resultados de posibilidad y necesidad de que el sistema sea planificable que la aplicación del principio de extensión y de forma significativamente más rápida que éste, a pesar de que los tiempos de finalización borrosos obtenidos son sólo aproximaciones de los que se obtienen mediante el principio de extensión.

Por último, en el apartado 5.2.4 se extiende el análisis a periodos borrosos.

### 5.2.1. Principio de extensión

Tal y como se vio en el capítulo 3 cuando algunos de los parámetros que definen las tareas se modelan mediante números borrosos, la planificabilidad del sistema se puede medir mediante un grado de posibilidad  $\Pi_{planif}$  y un grado de necesidad  $N_{planif}$  que se definen como los valores mínimos de posibilidad y necesidad de que cada tarea  $i$  en cada ejecución  $j$  finalice antes del correspondiente plazo absoluto  $d_{ij}$ , donde  $d_{ij}=(j-1) \cdot t_i + d_i$ . Si por ejemplo los tiempos de ejecución se modelan como números borrosos, los tiempos de finalización son también borrosos y se tiene:

$$\Pi_{planif} = \min(\Pi(WC_{ij} < d_{ij}))$$

$$N_{planif} = \min(N(WC_{ij} < d_{ij}))$$

Cuando existe un instante crítico y las prioridades de las tareas son fijas, el peor tiempo de respuesta de cada tarea se produce en la primera ejecución, con lo que es suficiente comparar los tiempos de finalización en la primera ejecución  $wc_{i,1} = wc_i$  con los plazos  $d_i$ . Por tanto la posibilidad y necesidad de que el sistema sea planificable son respectivamente:

$$(5.6) \Pi_{planif} = \min_{i=1..n\_tareas} (\Pi(WC_i < d_i))$$

$$N_{planif} = \min_{i=1..n\_tareas} (N(WC_i < d_i))$$

Si los plazos son también números borrosos, entonces la comparación entre el tiempo de finalización y el plazo en (5.6) es una comparación entre conjuntos borrosos (ver anexo B) y se debe sustituir el plazo numérico  $d_i$  por el plazo borroso  $D_i$ .

Para calcular los tiempos de finalización borrosos  $WC_i$  se extiende el algoritmo previamente presentado (5.2) a parámetros borrosos. La forma más general de extender una función a parámetros borrosos es mediante la aplicación del principio de extensión. En el caso de que la t-norma utilizada sea el mínimo (ver capítulo 3), la extensión de la función  $wc_i=f(p_h)$  a tiempos de ejecución borrosos  $P_h$ , con  $h \in \mathbb{N} \mid 1 \leq h \leq i$ , da como resultado un conjunto borroso  $WC_i$  cuya función de pertenencia se calcula como:

$$\mu_{WC_i}(wc_i) = \sup_{wc_i=f(p_h)} \min\{\mu_{P_h}(p_h)\} \quad \forall h \in \mathbb{N} \mid 1 \leq h \leq i$$

donde  $f$  es la función definida por el algoritmo básico que aparece en (5.2) y que se describe a continuación:

**ALG BASICO:**

$$(5.7) \quad \left\{ \begin{array}{l} s_{i,0} = s\_ini \\ \text{while } (s_{i,k-1} < s_{i,k}(s_{i,k-1})) \text{ AND } (s_{i,k} \leq t_i) \quad k = k + 1 \\ wc_i = s_{i,k} \\ \text{dado, } s_{i,k}(s_{i,k-1}) = \sum_{h=1}^i p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil \end{array} \right.$$

if ( $wc_i > d_i$ ) tarea  $i$  no planificable

Nótese que  $f$  es no decreciente en  $p_h$ , discontinua en  $p_h$  (debido a la operación redondeo), y que  $p_h > 0$  por lo que el resultado de la extensión de  $f$  a parámetros borrosos es, en general, un tiempo de finalización borroso no convexo.

Si retomamos el ejemplo anterior formado por tres tareas periódicas de periodos  $t_1=3$ ,  $t_2=5$  y  $t_3=15$ , y se definen los tiempos de ejecución como números borrosos triangulares  $P_1=(0.9, 1, 1.05)$ ,  $P_2=(1.8, 1.9, 2)$  y  $P_3=(0.9, 1, 1.1)$ , la aplicación del principio de extensión permite calcular los tiempos de finalización borrosos de las tareas.

En la Figura 5.2 se muestra el tiempo borroso de finalización de la tarea 2 en su primera ejecución, obtenido mediante el principio de extensión.

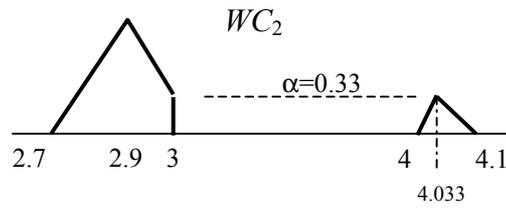


Figura 5.2: Tiempo borroso de respuesta de la tarea 2, mediante el principio de extensión

En este caso, como el tiempo de respuesta de la tarea 2 depende de los tiempos de ejecución de las tareas 1 y 2, es decir, es función de dos argumentos, la distribución de posibilidad de  $WC_2$  se puede obtener mediante la representación gráfica propuesta en el anexo A, tal y como se muestra a continuación en la Figura 5.3.

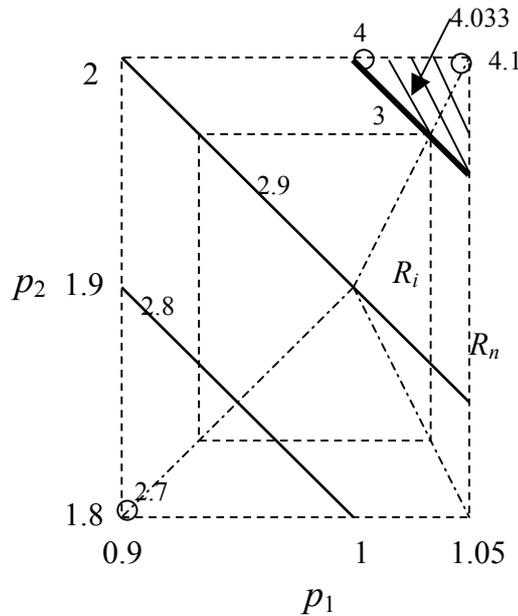


Figura 5.3: Representación gráfica del tiempo de finalización de la tarea 2 en función de los tiempos de ejecución de las tareas 1 y 2

Se pueden distinguir dos zonas en la Figura 5.3. Cuando la suma de los tiempos de ejecución de las tareas 1 y 2 es menor o igual que 3 (zona 1), el tiempo de finalización de la tarea 2 es  $p_1+p_2$ , ya que la tarea 2 se ejecuta inmediatamente después de que la tarea 1 termine su primera ejecución. Cuando la suma de los tiempos de ejecución de las tareas 1 y 2 es mayor que 3 (zona 2), la ejecución de la tarea 2 se ve interrumpida por la segunda ejecución de la tarea 1, que se produce en

el instante 3. La tarea 2 no finaliza hasta después de la segunda ejecución de la tarea 1, por tanto el tiempo de finalización de la tarea 2 es  $2 \cdot p_1 + p_2$ .

En la primera zona,  $w_{c_2}$  varía desde 2.7 hasta 3. La posibilidad de  $w_{c_2}=2.7$  es 0 ya que sólo el elemento focal conjunto externo  $R_n$  contiene al punto correspondiente. A medida que crece  $w_{c_2}$ , la posibilidad va aumentando hasta que para  $w_{c_2}=2.9$  la posibilidad es 1. Desde 2.9 hasta 3 la posibilidad va decreciendo y en  $w_{c_2}=3$  la posibilidad es 0.33 (1/3 de la diagonal correspondiente a los límites superiores), que es la integral correspondiente a la asignación básica conjunta de los elementos focales desde  $R_n$  hasta  $R_i$ . Esos son los elementos focales conjuntos que contienen al menos un punto de la línea  $w_{c_2}=3$ . Después de  $w_{c_2}=3$  hay una discontinuidad.

En la zona 2 de la Figura 5.3 la función toma valores desde 4 hasta 4.1. Para  $w_{c_2}=4$  la posibilidad es 0 y al crecer  $w_{c_2}$  va creciendo la posibilidad hasta  $w_{c_2}=4.033$  cuya posibilidad es 0.33, la misma que la de  $w_{c_2}=3$ , ya que son los mismos elementos focales los que intervienen en el cálculo. A partir de  $w_{c_2}=4.033$  la posibilidad decrece hasta que en  $w_{c_2}=4.1$  vale 0.

Tal y como ya se dijo en el capítulo 3, el cálculo de  $WC_i$  mediante la aplicación del principio de extensión puede ser muy lento. Si se aplica muestreando los soportes  $]P_i[$  de los tiempos de ejecución borrosos  $P_i$  para una resolución  $r$  en  $\alpha$  (ver Figura 5.4), el número de muestras de cada soporte es  $n_m = (2/r) - 1$  (en el caso de que el núcleo del número borroso lo forme un solo punto). En el cálculo de la distribución de posibilidad del tiempo de finalización de la tarea menos prioritaria  $WC_{n\_tar}$  mediante el principio de extensión hay que ejecutar el algoritmo básico con parámetros numéricos (5.7) para todas las combinaciones de los tiempos de ejecución de las tareas, es decir se ejecuta  $n_m^{n\_tar}$  número de veces. En general, para una tarea  $i$ , el algoritmo  $f$  se ejecuta  $n_m^i$  número de veces. Por tanto, para el cálculo de los tiempos de finalización borrosos de todas las tareas el número de veces que se ejecuta  $f$  es  $\sum_{i=1..n\_tar} (n_m^i)$ . Según sea el número de tareas el cálculo se puede hacer inviable.

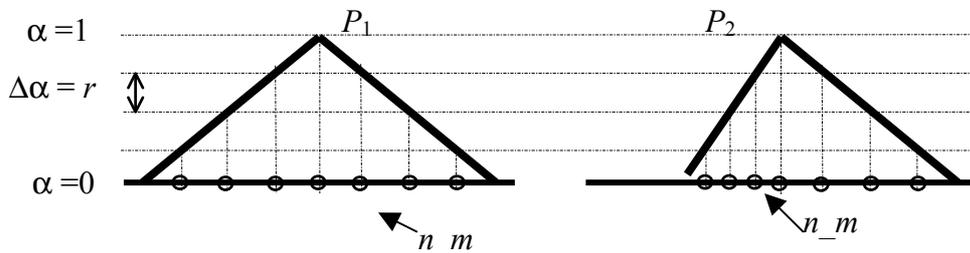


Figura 5.4: Muestreo de  $]P_1[$  y  $]P_2[$  para una resolución  $r$  en  $\alpha$

### 5.2.2. Aritmética borrosa

La segunda alternativa que se contempla para la extensión del análisis a parámetros borrosos es la basada en aritmética borrosa. En (5.8) se muestra la extensión del algoritmo de cálculo de la planificabilidad basada en aritmética borrosa, suponiendo que los tiempos de finalización borrosos son fuertemente menores que los periodos. Aunque los tiempos de finalización borrosos que se obtienen son aproximaciones, los valores de planificabilidad borrosa son los mismos que se obtienen con el principio de extensión.

	<p><b>Algoritmo de Aritmética Borrosa</b></p> <p><math>WC_0 = 0</math> (crisp)</p> <p>for <math>i = 1</math> to <math>n\_tarefas</math> do</p> <p style="padding-left: 20px;"><math>S_{i,0} = WC_{i-1} + P_i</math></p> <p style="padding-left: 20px;">while <math>(S_{i,k-1} \neq S_{i,k}(S_{i,k-1}))</math> <math>k=k+1</math></p> <p style="padding-left: 40px;"><math>WC_i = S_{i,k}</math></p> <p style="padding-left: 40px;"><math>\Pi / N(WC_i \leq d_i)</math></p> <p>end</p> <p><math>\Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq d_i) \} \quad \forall i</math></p> <p>dado <math>S_{i,k}(S_{i,k-1}) = \sum_{j=1}^i P_j \cdot \left\lceil \frac{S_{i,k-1}}{t_j} \right\rceil</math></p>
--	---

Recordemos que el redondeo clásico hacia arriba que se define como  $h(b) = \lceil b \rceil = i \mid i-1 < b \leq i, i \in \mathbb{Z}$ , se puede extender a argumento borroso mediante el principio de extensión como:

$$\mu_{\lceil B \rceil}(i) = \sup_{\substack{i-1 < b \leq i \\ b \in \mathbb{Z}}} \mu_B(b)$$

donde el resultado  $\lceil B \rceil$  es un conjunto borroso discreto.

Veamos como efectivamente, las aproximaciones de los tiempos de finalización borrosos obtenidos mediante aritmética borrosa en este algoritmo son efectivamente aproximaciones válidas.

Para la tarea 1, se obtiene  $WC_1 = P_1$ , resultado que coincide con la aplicación del principio de extensión a  $wc_1 = 0 + p_1$ .

Para la tarea 2,  $S_{2,0}$  vale  $S_{2,0} = P_1 + P_2$  tanto mediante la aplicación del algoritmo (5.8) como mediante la aplicación del principio de extensión a la función  $s_{2,0} = wc_1 + p_2 =$

$p_1 + p_2$ . En la siguiente iteración, con  $k=1$ , la aproximación  $S_{2,1}$  que se obtiene mediante aritmética borrosa se calcula como:

$$S_{2,1}(S_{2,0}) = P_1 \cdot \left[ \frac{S_{2,0}}{t_1} \right] + P_2 \cdot \left[ \frac{S_{2,0}}{t_2} \right] = P_1 \cdot \left[ \frac{P_1 + P_2}{t_1} \right] + P_2 \cdot \left[ \frac{P_1 + P_2}{t_2} \right]$$

y es uno de los casos ejemplo analizados en detalle en el capítulo 4 (concretamente la extensión de la función  $a \cdot [(a+b)/t_1] + b \cdot [(a+b)/t_2]$ ).

Realizando las operaciones de aritmética borrosa de forma independiente, como se indica en (5.9), no se tiene en cuenta la relación funcional existente entre  $c = [(p_1 + p_2)/t_1]$  y  $d = [(p_1 + p_2)/t_2]$ :

$$(5.9) \quad C = \left[ \frac{P_1 + P_2}{t_1} \right], \quad D = \left[ \frac{P_1 + P_2}{t_2} \right], \quad \text{y} \quad S_{2,1} = P_1 \cdot C + P_2 \cdot D$$

Como los términos  $c = [(p_1 + p_2)/t_1]$  y  $d = [(p_1 + p_2)/t_2]$  son no decrecientes en  $p_1$  y en  $p_2$ , se tiene que los límites inferiores y superiores de los  $\alpha$ -cortes de  $C$  y  $D$  se pueden calcular a partir de los límites inferiores y superiores de los  $\alpha$ -cortes de  $P_1$  y  $P_2$  como:

$$\begin{aligned} \underline{c}(\alpha) &= \left[ \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_1} \right] & \bar{c}(\alpha) &= \left[ \frac{\bar{p}_1(\alpha) + \bar{p}_2(\alpha)}{t_1} \right] \\ \underline{d}(\alpha) &= \left[ \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_2} \right] & \bar{d}(\alpha) &= \left[ \frac{\bar{p}_1(\alpha) + \bar{p}_2(\alpha)}{t_2} \right] \end{aligned}$$

Como  $s_{2,1} = p_1 \cdot c + p_2 \cdot d$  es creciente en  $p_1$ , en  $p_2$ , en  $c$  y en  $d$ , los límites de los  $\alpha$ -cortes de  $S_{2,1}$  se pueden calcular como:

$$\begin{aligned} \underline{s}_{2,1}(\alpha) &= \underline{p}_1(\alpha) \cdot \underline{c}(\alpha) + \underline{p}_2(\alpha) \cdot \underline{d}(\alpha) = \underline{p}_1(\alpha) \cdot \left[ \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_1} \right] + \underline{p}_2(\alpha) \cdot \left[ \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_2} \right] \\ \bar{s}_{2,1}(\alpha) &= \bar{p}_1(\alpha) \cdot \bar{c}(\alpha) + \bar{p}_2(\alpha) \cdot \bar{d}(\alpha) = \bar{p}_1(\alpha) \cdot \left[ \frac{\bar{p}_1(\alpha) + \bar{p}_2(\alpha)}{t_1} \right] + \bar{p}_2(\alpha) \cdot \left[ \frac{\bar{p}_1(\alpha) + \bar{p}_2(\alpha)}{t_2} \right] \end{aligned}$$

Por otro lado, aplicando el principio de extensión a la función

$$s_{2,1}(s_{2,0}) = p_1 \cdot \left[ \frac{p_1 + p_2}{t_1} \right] + p_2 \cdot \left[ \frac{p_1 + p_2}{t_2} \right]$$

se obtiene el conjunto borroso exacto  $S_{2,1}^{pe}$  cuya función de pertenencia viene dada por:

$$\mu_{S_{2,1}^{pe}}(s_{2,1}^{pe}) = \sup_{s_{2,1}^{pe}=f(p_1,p_2)=p_1 \cdot \left\lfloor \frac{p_1+p_2}{t_1} \right\rfloor + p_2 \cdot \left\lfloor \frac{p_1+p_2}{t_2} \right\rfloor} \min\{\mu_{p_1}(p_1), \mu_{p_2}(p_2)\}$$

Como  $f$  es no decreciente en  $p_1$  y  $p_2$ , se pueden calcular los límites inferiores y superiores de los  $\alpha$ -cortes de la siguiente forma:

$$\begin{aligned} \underline{s}_{2,1}^{pe}(\alpha) &= \underline{p}_1(\alpha) \cdot \left\lfloor \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_1} \right\rfloor + \underline{p}_2(\alpha) \cdot \left\lfloor \frac{\underline{p}_1(\alpha) + \underline{p}_2(\alpha)}{t_2} \right\rfloor = \underline{s}_{2,1}(\alpha) \\ \overline{s}_{2,1}^{pe}(\alpha) &= \overline{p}_1(\alpha) \cdot \left\lfloor \frac{\overline{p}_1(\alpha) + \overline{p}_2(\alpha)}{t_1} \right\rfloor + \overline{p}_2(\alpha) \cdot \left\lfloor \frac{\overline{p}_1(\alpha) + \overline{p}_2(\alpha)}{t_2} \right\rfloor = \overline{s}_{2,1}(\alpha) \end{aligned}$$

Como estos límites coinciden con los límites de los  $\alpha$ -cortes de  $S_{2,1}$  se deduce que  $S_{2,1}$  es una aproximación válida de  $S_{2,1}^{pe}$ .

De forma similar, para las siguientes iteraciones,  $s$  va creciendo hasta que se estabiliza al encontrar el resultado del tiempo de finalización. El valor  $s_{2,k}$  es no decreciente con  $s_{2,k-1}$ , además de ser creciente con  $p_1$  y  $p_2$ . Por tanto, el cálculo iterativo de  $wc_2$  mediante el algoritmo básico es monótono creciente con los tiempos de ejecución y su extensión mediante aritmética borrosa proporciona una aproximación válida del tiempo borroso de finalización exacto, con lo que  $\Pi(WC_i \leq d_i) = \Pi(WC_i^{pe} \leq d_i)$  y  $N(WC_i \leq d_i) = N(WC_i^{pe} \leq d_i)$ .

Nótese que en el algoritmo (5.8) la comparación entre iteraciones sucesivas de  $S$  se ha expresado como “*while*( $S_{i,k-1} \neq S_{i,k}$ )”, de forma diferente a como aparecía en el algoritmo básico con parámetros numéricos, donde la condición era “*while*( $s_{i,k-1} < s_{i,k}$ )”. Se considera que la forma correcta de extender la comparación es la propuesta en (5.8), tal y como se describe en el siguiente estudio.

#### *Extensión de la comparación $s_{i,k-1} < s_{i,k}$ del algoritmo básico con parámetros numéricos*

A modo de ejemplo se analiza primero la extensión de la función real de parámetros numéricos  $b=g(a)=2 \cdot a$ . Si  $a$  se modela mediante un número borroso  $A$ , el principio de extensión permite extender  $g$  de tal forma que el resultado es un número borroso  $B = 2 \cdot A$ . Si  $A$  es un número borroso triangular definido por sus vértices  $A = (0.1, 0.2,$

0.3), entonces el número borroso  $B$  obtenido es  $B = (0.2, 0.4, 0.6)$ , tal y como se muestra en la Figura 5.5.

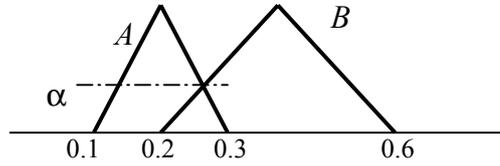


Figura 5.5: Comparación entre dos números borrosos

Comparando  $A$  y  $B$ , la posibilidad de que  $B$  sea mayor que  $A$  es  $\Pi(B > A) = 1$  y la de  $A$  mayor que  $B$  es  $\Pi(A > B) = \alpha$ . Pero en este resultado se ha olvidado la relación funcional existente entre ambas variables  $a$  y  $b$ . En la Figura 5.6 se muestra la distribución de posibilidad conjunta de  $a$  y  $b$ , donde se ha resaltado la zona del dominio conjunto que cumple  $a > b$ . Para cualquier pareja  $(a, b)$  situada en esa zona, la distribución de posibilidad conjunta vale 0. Afirmar que el valor de posibilidad de que  $A$  sea mayor que  $B$  es distinto de cero significa que existen parejas  $(a, b)$  en el soporte conjunto de  $A$  y  $B$ , tal que  $a > b$  con posibilidad distinta de cero. Pero esa condición no se cumple si tenemos en cuenta la relación funcional  $b = 2 \cdot a$ .

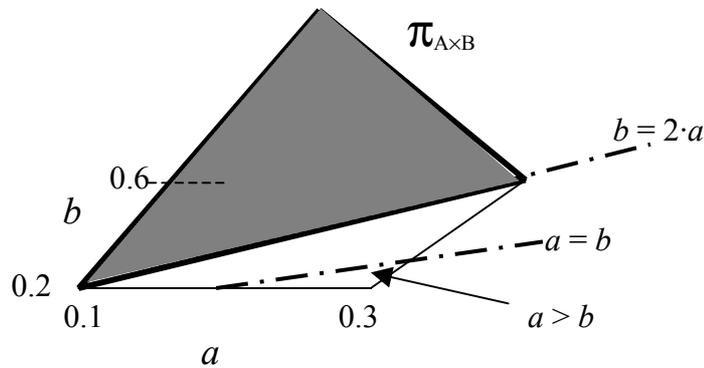


Figura 5.6: Distribución de posibilidad conjunta de  $A$  y  $B$  tal que  $b = 2 \cdot a$

Si se analiza nuevamente el algoritmo (5.7), se puede observar que entre  $s_{i,k-1}$  y  $s_{i,k}$ , se cumple la relación funcional:

$$s_{i,k} = g(s_{i,k-1}) = \sum_{h=1}^i p_h \cdot \left[ \frac{s_{i,k-1}}{t_h} \right]$$

Teniendo en cuenta, además, la forma iterativa en que se va obteniendo cada valor de  $s$ , se puede observar que la función es siempre no decreciente en  $s$ :  $s_{i,k}$  es siempre mayor que  $s_{i,k-1}$  y en el momento que se iguala el mayor tiempo de respuesta de la tarea  $i$  ha sido encontrado, y se pasa a buscar el de la siguiente tarea.

Por tanto, al extender la función a tiempos de ejecución borrosos,  $s_{i,k-1}$  y  $s_{i,k}$  son borrosos, pero siempre se tendrá que  $\Pi(S_{i,k-1} > S_{i,k})=0$  debido a la relación funcional existente entre ambos, y la condición de fin del bucle *while* en (5.8) (tiempo de respuesta de la tarea encontrado), se puede expresar en términos borrosos como  $S_{i,k-1} = S_{i,k}$ . Recordemos que  $S_{i,k-1} = S_{i,k}$  ssi  $\forall x \in \mathbb{R}$ ,  $\mu_{S_{i,k}}(x) = \mu_{S_{i,k-1}}(x)$

### Ejemplo:

Como se ha visto antes, se pueden realizar las operaciones del algoritmo basado en aritmética borrosa (5.8) de forma independiente sin tener en cuenta las relaciones funcionales entre  $c$  y  $d$ , siendo la aproximación del tiempo borrosa de finalización una aproximación válida. Se presenta a modo de ejemplo el mismo sistema de tiempo real anterior, formado por tres tareas periódicas de periodos  $t_1=3$ ,  $t_2=5$  y  $t_3=15$ , con tiempos de ejecución borrosos triangulares definidos por sus vértices  $P_1=(0.9, 1, 1.05)$ ,  $P_2=(1.8, 1.9, 2)$  y  $P_3=(0.9, 1, 1.1)$ .

Para la tarea 1, inicialmente  $WC_0 = 0$  con lo que  $S_{1,0} = 0+ P_1 = P_1$ . En la iteración  $k=1$ , se tiene  $S_{1,1} = P_1 \cdot \lceil S_{1,0} / t_1 \rceil = P_1$ . Como  $S_{1,1} = S_{1,0}$ , el tiempo de respuesta borroso de la tarea 1 es  $S_{1,1}$ , esto es,  $WC_1 = S_{1,1} = P_1$ .

Para la tarea 2, con  $k=0$  se tiene que  $S_{2,0} = P_1 + P_2=(2.7, 2.9, 3.05)$  y se muestra en la Figura 5.7. En la siguiente iteración  $k=1$ , se calcula  $S_{2,1}$  como:

$$\begin{aligned} S_{2,1} &= P_1 \cdot \lceil S_{2,0} / t_1 \rceil + P_2 \cdot \lceil S_{2,0} / t_2 \rceil \\ &= (0.9, 1, 1.05) \cdot \lceil (0.9, 0.966, 1.0166) \rceil + (1.8, 1.9, 2) \cdot \lceil (0.54, 0.58, 0.61) \rceil \\ &= (0.9, 1, 1.05) \cdot \{1/1 \cup 0.33/2\} + (1.8, 1.9, 2) \cdot \{1/1\} \\ &= \{(0.9, 1, 1.05) + (1.8, 1.9, 2)\} \cup \{\min((0.9, 1, 1.05) \cdot 2, 0.33) + (1.8, 1.9, 2)\} \end{aligned}$$

donde el segundo valor  $b$  de cada par  $\alpha/b$  es un elemento del dominio  $\mathbb{Z}$  y el primer valor  $\alpha$  de cada par  $\alpha/b$  es el valor de pertenencia de  $b$  al conjunto borroso  $[B]$ , es decir,  $0.33/2$  significa que el número entero 2 tiene valor de posibilidad 0.33 en el conjunto borroso  $\lceil S_{2,0} / t_1 \rceil$ , o dicho de otra forma, tiene un grado de pertenencia al conjunto borroso  $\lceil S_{2,0} / t_1 \rceil$  igual a 0.33.

El primer término de la unión, suma de dos números borrosos triangulares, es otro número triangular dado por  $(2.7, 2.9, 3.05)$ . Se puede ver esta parte de  $S_{2,1}$  en la Figura 5.7. El segundo término de la unión es suma de  $\min((0.9, 1, 1.05) \cdot 2, 0.33)$

más un número triangular (1.8,1.9,2). La función de pertenencia del primer sumando  $\min((0.9,1,1.05) \cdot 2, 0.33) = \min\{\mu_{2 \cdot P_1}(x), 0.33\}$ , es el número borroso triangular  $2 \cdot (0.9,1,1.05) = (1.8,2,2.1)$  truncado a la altura  $\alpha=0.33$ . Por tanto, la función de pertenencia de la suma queda como  $\min\{\mu_{2 \cdot P_1 + P_2}(x), 0.33\}$ . Éste es el número borroso triangular  $(1.8,2,2.1) + (1.8,1.9,2) = (3.6, 3.9, 4.1)$  truncado a la altura  $\alpha=0.33$ , como aparece en la Figura 5.7.

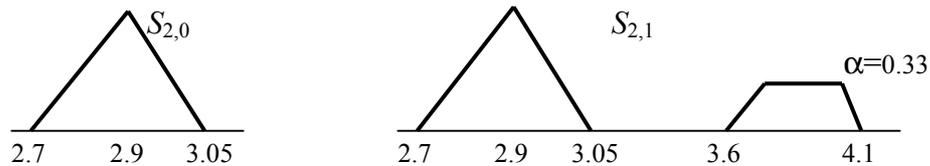


Figura 5.7: Representación de  $S_{2,0}$  y  $S_{2,1}$  del ejemplo

Comparando  $S_{2,1}$  con  $S_{2,0}$  se puede ver que  $S_{2,1} \neq S_{2,0}$ , con lo que se continuaría, dentro del bucle *while*, en la búsqueda de  $WC_2$ , calculando la siguiente iteración  $S_{2,2}$ . Sin embargo vemos que una parte del conjunto borroso  $S_{2,1}$  sí que coincide con  $S_{2,0}$ . Expresado con valores numéricos, si para cada punto del soporte de esa zona de  $S_{2,1}$  se vuelve a repetir el bucle haciendo  $k=2$ , volverá a salir el mismo punto, puesto que el algoritmo ha encontrado el resultado en dicha zona. En cambio, para los puntos situados en la zona que va de 3.6 a 4.1, se tiene que  $s_{2,1} \neq s_{2,0}$  y hay que volver a iterar. Por tanto, es suficiente recalculer la parte de la derecha del conjunto borroso  $S_{2,1}$  continuando el algoritmo con  $S_{2,1} = \{ \min((0.9, 1, 1.05) \cdot 2, 0.33) + (1.8,1.9,2) \} = \min((3.6,3.9,4.1), 0.33)$ .

Así, con  $k=2$  se tiene:

$$\begin{aligned} S_{2,2} &= P_1 \cdot \lceil S_{2,1} / t_1 \rceil + P_2 \cdot \lceil S_{2,1} / t_2 \rceil \\ &= (0.9, 1, 1.05) \cdot \lceil \min((1.2, 1.3, 1.366), 0.33) \rceil + (1.8, 1.9, 2) \cdot \lceil \min((0.72, 0.78, 0.82), 0.33) \rceil \\ &= (0.9, 1, 1.05) \cdot \{ 0.33/2 \} + (1.8, 1.9, 2) \cdot \{ 0.33/1 \} = \min((3.6, 3.9, 4.1), 0.33) \end{aligned}$$

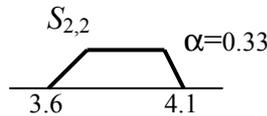


Figura 5.8: Representación gráfica de  $S_{2,2}$

Como  $S_{2,2} = S_{2,1}$  el tiempo de respuesta borroso de la tarea 2 es la unión del conjunto triangular  $(2.7, 2.9, 3.05)$  y de  $\min((3.6, 3.9, 4.1), 0.33)$ . El resultado  $WC_2 = \{(2.7, 2.9, 3.05) \cup \min((3.6, 3.9, 4.1), 0.33)\}$  obtenido mediante este algoritmo basado en aritmética borrosa se puede ver en la Figura 5.9.

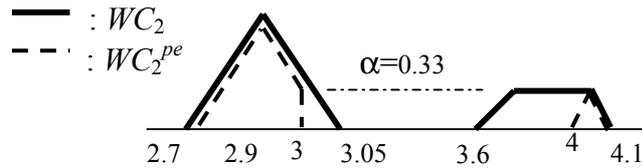


Figura 5.9:  $WC_2$  obtenido mediante aritmética borrosa, comparado con el resultado exacto

En la misma Figura 5.9 se muestra también el resultado exacto  $WC_2^{pe}$  obtenido mediante la aplicación del principio de extensión (calculado gráficamente en la Figura 5.3 y mostrado en la Figura 5.2). Se puede comprobar que, efectivamente, tienen los mismos límites de los  $\alpha$ -cortes.

Para la tarea 3, con  $k=0$ , se tiene:

$$S_{3,0} = WC_2 + P_3 = \{(2.7, 2.9, 3.05) \cup \min((3.6, 3.9, 4.1), 0.33)\} + (0.9, 1, 1.1)$$

$$= (3.6, 3.9, 4.15) \cup \{\min((4.5, 4.9, 5.2), 0.33)\}$$

La representación gráfica de  $S_{3,0}$  se muestra en la Figura 5.10. En la iteración correspondiente a  $k=1$  el conjunto borroso  $S_{3,1}$  se puede calcular como:

$$S_{3,1} = P_1 \cdot [S_{3,0} / t_1] + P_2 \cdot [S_{3,0} / t_2] + P_3 \cdot [S_{3,0} / t_3]$$

$$= (0.9, 1, 1.05) \cdot [(1.2, 1.3, 1.38) \cup \min((1.5, 1.63, 1.73), 0.33)]$$

$$+ (1.8, 1.9, 2) \cdot [(0.72, 0.78, 0.83) \cup \min((0.9, 0.98, 1.04), 0.33)]$$

$$+ (0.9, 1, 1.1) \cdot [(0.24, 0.26, 0.276) \cup \min((0.3, 0.33, 0.35), 0.33)]$$

$$= (0.9, 1, 1.05) \cdot \{1/2\} + (1.8, 1.9, 2) \cdot \{0.33/1 \cup 0.33/2\} + (0.9, 1, 1.1) \cdot \{1/1\}$$

$$= (4.5, 4.9, 5.2) \cup \{\min((6.3, 6.8, 7.2), 0.33)\}$$

Este conjunto borroso no convexo  $S_{3,1}$  aparece también en la Figura 5.10.

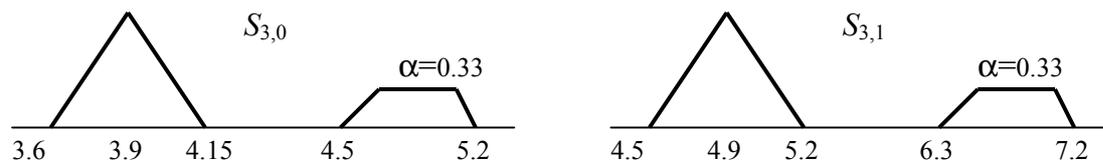


Figura 5.10: Representación de  $S_{3,0}$  y  $S_{3,1}$  del ejemplo

Como  $S_{3,1} \neq S_{3,0}$ , se vuelve a iterar con  $k=2$ :

$$\begin{aligned}
 S_{3,2} &= P_1 \cdot \lceil S_{3,1} / t_1 \rceil + P_2 \cdot \lceil S_{3,1} / t_2 \rceil + P_3 \cdot \lceil S_{3,1} / t_3 \rceil \\
 &= (0.9, 1, 1.05) \cdot \lceil (1.5, 1.63, 1.73) \cup \min((2.1, 2.26, 2.4), 0.33) \rceil \\
 &\quad + (1.8, 1.9, 2) \cdot \lceil (0.9, 0.98, 1.04) \cup \min((1.26, 1.36, 1.44), 0.33) \rceil \\
 &\quad + (0.9, 1, 1.1) \cdot \lceil (0.3, 0.326, 0.34) \cup \min((0.42, 0.45, 0.48), 0.33) \rceil \\
 &= (0.9, 1, 1.05) \cdot \{ 1/2 \cup 0.33/3 \} \\
 &\quad + (1.8, 1.9, 2) \cdot \{ 0.33/1 \cup 0.33/2 \} \\
 &\quad + (0.9, 1, 1.1) \cdot \{ 1/1 \} \\
 &= (4.5, 4.9, 5.2) \cup \{ \min((7.2, 7.8, 8.25), 0.33) \}
 \end{aligned}$$

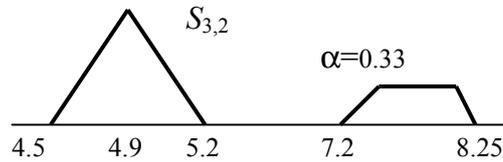


Figura 5.11: Representación de  $S_{3,2}$

Vemos que  $S_{3,2} \neq S_{3,1}$ , aunque la parte correspondiente a (4.5, 4.9, 5.2) coincide. Por tanto (4.5, 4.9, 5.2) es parte de la solución del tiempo borroso de respuesta de la tarea 3. Para calcular el resto, continuamos el algoritmo igualando  $S_{3,2}$  a la parte que es diferente, esto es,  $S_{3,2} = \{ \min((7.2, 7.8, 8.25), 0.33) \}$

Así, con  $k=3$ ,

$$\begin{aligned}
 S_{3,3} &= P_1 \cdot \lceil S_{3,2} / t_1 \rceil + P_2 \cdot \lceil S_{3,2} / t_2 \rceil + P_3 \cdot \lceil S_{3,2} / t_3 \rceil \\
 &= (0.9, 1, 1.05) \cdot \lceil \min((2.4, 2.6, 2.75), 0.33) \rceil \\
 &\quad + (1.8, 1.9, 2) \cdot \lceil \min((1.44, 1.56, 1.65), 0.33) \rceil \\
 &\quad + (0.9, 1, 1.1) \cdot \lceil \min((0.48, 0.52, 0.55), 0.33) \rceil \\
 &= (0.9, 1, 1.05) \cdot \{ 0.33/3 \} \\
 &\quad + (1.8, 1.9, 2) \cdot \{ 0.33/2 \} \\
 &\quad + (0.9, 1, 1.1) \cdot \{ 0.33/1 \} \\
 &= \{ \min((7.2, 7.8, 8.25), 0.33) \}
 \end{aligned}$$

Como  $S_{3,3} = S_{3,2}$ , el tiempo de respuesta de la tarea 3 es la unión de (4.5, 4.9, 5.2) obtenido antes, con  $\{ \min((7.2, 7.8, 8.25), 0.33) \}$ , es decir,  $WC_3 = (4.5, 4.9, 5.2) \cup \{ \min((7.2, 7.8, 8.25), 0.33) \}$ , que aparece representado en la Figura 5.12.

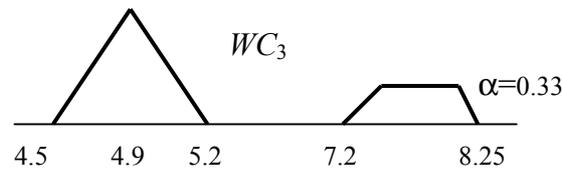


Figura 5.12: Representación de  $WC_3$  obtenido mediante aritmética borrosa

Una vez calculadas las aproximaciones de los peores tiempos de finalización de las tareas, se comparan con los correspondientes plazos. En la Figura 5.13 se han dibujado unos plazos como ejemplo.

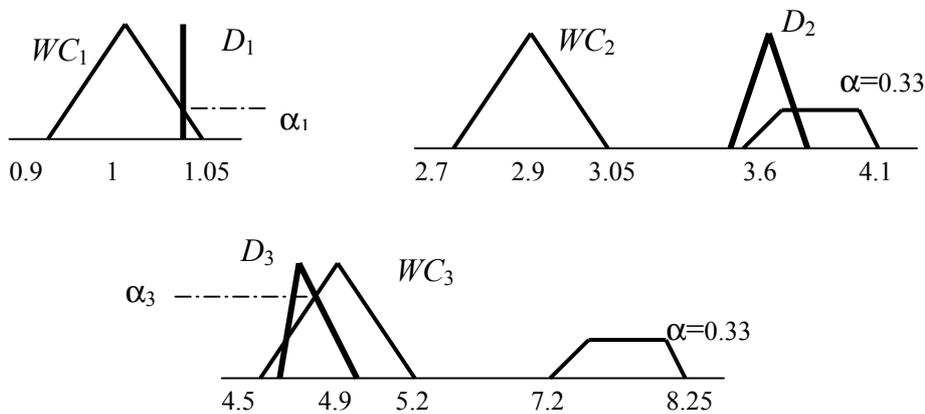


Figura 5.13: Comparación de los tiempos de respuesta borrosos con los plazos

Para los plazos que aparecen en la Figura 5.13, los valores de posibilidad y necesidad de que las tareas finalicen antes de sus correspondientes plazos son:

$$\Pi(WC_1 \leq D_1) = 1 \text{ y } N(WC_1 \leq D_1) = 1 - \Pi(WC_1 > D_1) = 1 - \alpha_1$$

$$\Pi(WC_2 \leq D_2) = 1 \text{ y } N(WC_2 \leq D_2) = 1 - \Pi(WC_2 > D_2) = 1 - 0.33$$

$$\Pi(WC_3 \leq D_3) = \alpha_3 \text{ y } N(WC_3 \leq D_3) = 1 - \Pi(WC_3 > D_3) = 0$$

Y los valores de planificabilidad borrosa del sistema son:

$$\Pi_{planif} = \min\{ \Pi(WC_1 \leq D_1), \Pi(WC_2 \leq D_2), \Pi(WC_3 \leq D_3) \} = \alpha_3$$

$$N_{planif} = \min\{ N(WC_1 \leq D_1), N(WC_2 \leq D_2), N(WC_3 \leq D_3) \} = 0$$

En este apartado se ha mostrado que la aplicación del algoritmo de planificabilidad (5.8) basado en aritmética borrosa permita calcular el valor de planificabilidad exacta a partir de tiempos de finalización borrosos que son aproximaciones válidas de los exactos.

### 5.2.3. Aritmética de intervalos

Tal y como ya se ha dicho, la función  $f$  (el algoritmo básico) que calcula el peor tiempo de respuesta  $wc_i$  de cada tarea es una función no decreciente en los tiempos de ejecución  $p_j$  de las tareas más prioritarias, con lo que se cumple:

$$\underline{wc}_i(\alpha) = f\left(\underline{p}_j(\alpha)\right)$$

$$\overline{wc}_i(\alpha) = f\left(\overline{p}_j(\alpha)\right)$$

Por tanto, es posible obtener aproximaciones válidas convexas mediante el cálculo de intervalos, siendo éstos los  $\alpha$ -cortes de los tiempos de ejecución borrosos. La aproximación conserva los límites inferiores y superiores de cada  $\alpha$ -corte resultante, pero puede haber puntos del intervalo que no pertenezcan al  $\alpha$ -corte exacto tal y como se aprecia en la Figura 5.14, donde se muestra la aproximación convexa  $WC_2$  del tiempo de finalización de la tarea 2 en comparación con el resultado exacto  $WC_2^{pe}$ .

Para acelerar el cálculo de los valores de posibilidad y necesidad de que la tarea finalice antes de su plazo, es suficiente con calcular el punto de corte entre el plazo y la aproximación convexa del tiempo de finalización, resaltado en la Figura 5.14.

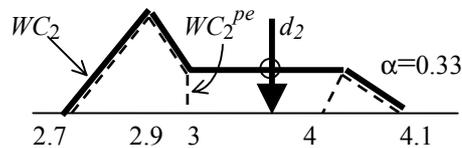


Figura 5.14: Tiempo borroso de finalización de la tarea 2 obtenido mediante el principio de extensión  $WC_2^{pe}$  y mediante aritmética de intervalos  $WC_2$ .

En este apartado se propone un algoritmo basado en aritmética de intervalos, suponiendo primero que los plazos son numéricos, y después se extiende a plazos borrosos.

En (5.10), el algoritmo basado en aritmética de intervalos permite calcular por bipartición el punto de corte entre el tiempo de finalización borroso y el plazo para cada tarea, y con ese valor de  $\alpha$  del punto de corte calcula la posibilidad y necesidad de que la tarea sea planificable. El mínimo de todas ellas es la planificabilidad borrosa del sistema.

**ALG Aritmética de Intervalos**

$$\underline{wc}_0(0) = 0$$

for  $i = 1$  to  $n\_tarefas$  do

$$\underline{wc}_i(0) = \text{ALG BASICO} \left( p_j = \underline{p}_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \right)$$

$$\underline{wc}_i(1) = \text{ALG BASICO} \left( p_j = \underline{p}_j(1); s\_in_i = \underline{wc}_{i-1}(0) + p_i \right)$$

$$\overline{wc}_i(1) = \text{ALG BASICO} \left( p_j = \overline{p}_j(1); s\_in_i = \underline{wc}_{i-1}(0) + p_i \right)$$

$$\overline{wc}_i(0) = \text{ALG BASICO} \left( p_j = \overline{p}_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \right)$$

ALG ZONA

end

(5.10)  $\Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq d_i) \} \quad \forall i$

En (5.10) y en general en todos los demás algoritmos que se van a proponer en este capítulo de la tesis basados en aritmética de intervalos, los argumentos de la llamada al algoritmo básico ALG BASICO son aquellos que en el algoritmo están sin definir, como el valor inicial de  $s$  o bien aquellos que por ser borrosos hay que precisar en qué valor concreto del conjunto borroso hay que aplicar el algoritmo básico. En este caso, como el tiempo de ejecución de las tareas es borroso, se indica como argumento de la llamada al algoritmo el valor numérico del tiempo de ejecución para el que hay que calcular el tiempo de finalización de la tarea, que es siempre el límite inferior o superior de un  $\alpha$ -corte de los tiempos borrosos de ejecución.

El algoritmo ALG BASICO en (5.7) se rescribe en (5.11), de momento por simplificar se supone que se verifica  $s_{i,k} \leq t_i$ . El valor inicial  $s\_in_i$  con el que se llama al algoritmo básico desde (5.7) es  $\underline{wc}_{i-1}(0) + p_i$  porque es seguro que el tiempo de finalización de la tarea  $i$  estará después de ese instante de tiempo.

De hecho, para cada valor  $\alpha$  se podría inicializar el algoritmo básico con  $\underline{wc}_{i-1}(\alpha) + p_i$ , inicialización que está más cerca de la solución que  $\underline{wc}_{i-1}(0) + p_i$ . Sin embargo, consideramos que realizar el cálculo de  $\underline{wc}_{i-1}(\alpha)$  únicamente para la inicialización, alargaría los tiempos de ejecución del algoritmo.

$$(5.11) \quad \boxed{\begin{array}{l} \text{ALG BASICO:} \\ \left\{ \begin{array}{l} s_{i,0} = s_{in_i} \\ \text{while } (s_{i,k-1} < s_{i,k}(s_{i,k-1})) \quad k = k + 1 \\ wc_i = s_{i,k} \end{array} \right. \\ \text{dado, } s_{i,k}(s_{i,k-1}) = \sum_{h=1}^i p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil \end{array}}$$

El Algoritmo ZONA (5.12) determina la zona en la que hay que realizar la búsqueda por bipartición: en los límites superiores o en los límites inferiores de los  $\alpha$ -cortes de  $WC_i$ .

$$(5.12) \quad \boxed{\begin{array}{l} \text{ALG ZONA} \\ \text{if } (d_i \leq \underline{wc}_i(0)) \quad \Pi(WC_i \leq d_i) = 0, \quad N(WC_i \leq d_i) = 0 \\ \text{if } (d_i \geq \overline{wc}_i(0)) \quad \Pi(WC_i \leq d_i) = 1, \quad N(WC_i \leq d_i) = 1 \\ \text{if } (\underline{wc}_i(1) \leq d_i \leq \overline{wc}_i(1)) \quad \Pi(WC_i \leq d_i) = 1, \quad N(WC_i \leq d_i) = 0 \\ \text{if } (\underline{wc}_i(0) \leq d_i \leq \underline{wc}_i(1)) \quad \text{Bipartición inf} \\ \text{if } (\overline{wc}_i(1) \leq d_i \leq \overline{wc}_i(0)) \quad \text{Bipartición sup} \end{array}}$$

Cuando el punto de corte entre el tiempo de finalización borroso  $WC_i$  y el plazo se encuentra en la zona de los límites inferiores de los  $\alpha$ -cortes de  $WC_i$ , la necesidad de que la tarea finalice antes de su plazo es cero. En este caso se llama al algoritmo *Bipartición inf* (5.13) para realizar la búsqueda por bipartición del punto de corte. *Bipartición inf* obtiene el punto de corte y con él el valor de posibilidad de que la tarea finalice antes de su plazo.

$$\begin{array}{l}
 \text{Bipartición inf} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0 \\
 \text{while } \left( \left| \underline{wc}_i(\alpha) - d_i \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(\alpha); s\_in_i = \underline{wc}_{i-1}(0) + p_i \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 \text{if } \left( \underline{wc}_i(\alpha) < d_i \right) \alpha_0 = \alpha \\
 \text{if } \left( \underline{wc}_i(\alpha) > d_i \right) \alpha_1 = \alpha \\
 \text{end} \\
 \Pi(WC_i \leq d_i) = \alpha; \quad N(WC_i \leq d_i) = 0
 \end{array}
 \tag{5.13}$$

En cambio, si el punto de corte entre el tiempo de finalización borroso  $WC_i$  y el plazo se encuentra en la zona de los límites superiores de los  $\alpha$ -cortes de  $WC_i$ , la posibilidad de que la tarea finalice antes de su plazo es 1. En este caso se llama al algoritmo *Bipartición sup* (5.14) para realizar la búsqueda del punto de corte por bipartición y obtener la necesidad de que la tarea finalice antes de su plazo.

$$\begin{array}{l}
 \text{Bipartición sup} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0 \\
 \text{while } \left( \left| \overline{wc}_i(\alpha) - d_i \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \overline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); s\_in_i = \underline{wc}_{i-1}(0) + p_i \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 \text{if } \left( \overline{wc}_i(\alpha) < d_i \right) \alpha_1 = \alpha \\
 \text{if } \left( \overline{wc}_i(\alpha) > d_i \right) \alpha_0 = \alpha \\
 \text{end} \\
 \Pi(WC_i \leq d_i) = 1; \quad N(WC_i \leq d_i) = 1 - \alpha
 \end{array}
 \tag{5.14}$$

*Mejora de la inicialización del Algoritmo Básico:*

Para aumentar la rapidez del algoritmo básico, en lugar de inicializar  $s$  siempre a  $\underline{wc}_{i-1}(0) + p_i$  en los diversos cálculos de  $wc_i$  con distintos  $\alpha$ , se utilizará este valor sólo para inicializar el cálculo de  $\underline{wc}_i(0)$ .

Para los demás cálculos de  $wc_i$  con valor de  $\alpha$  distinto de cero, o para calcular  $\overline{wc}_i(0)$ ,  $s$  se inicializa a  $\underline{wc}_i(0)$ , ya que está asegurado que cualquier tiempo de finalización de la tarea  $i$  será mayor que su propio límite inferior  $\underline{wc}_i(0)$ .

Aún se puede mejorar esta inicialización de  $s$  en la búsqueda por bipartición, si cada vez que se localiza el punto de corte en el subintervalo derecho de la zona de búsqueda de  $wc_i$ , se inicializa  $s$  con el límite inferior de ese subintervalo derecho. Por otro lado, cuando se localiza el punto de corte en el subintervalo izquierdo de la zona de búsqueda de  $wc_i$ ,  $s$  se inicializa con el mismo valor que en la partición anterior, que es el límite inferior del subintervalo izquierdo.

Veamos en primer lugar los dos ejemplos de la Figura 5.15 donde el punto de corte entre el tiempo de finalización borroso  $WC_i$  y el plazo  $d_i$  está en la zona de los límites inferiores de los  $\alpha$ -cortes de  $WC_i$ .

En los dos ejemplos de la Figura 5.15, para calcular  $\underline{wc}_i(0)$ , se inicializa  $s$  a  $\underline{wc}_{i-1}(0)+p_i$ . Para calcular  $\underline{wc}_i(1)$ , se inicializa  $s$  a  $\underline{wc}_i(0)$ . Como se verifica que  $\underline{wc}_i(0) < d_i < \underline{wc}_i(1)$ , la zona de búsqueda por bipartición se encuentra en los límites inferiores. Así pues, para calcular  $\underline{wc}_i(0.5)$ ,  $s$  se inicializa a  $\underline{wc}_i(0)$ . Con ello, el intervalo  $[\underline{wc}_i(0), \underline{wc}_i(1)]$  queda dividido en dos subintervalos  $[\underline{wc}_i(0), \underline{wc}_i(0.5)]$  y  $[\underline{wc}_i(0.5), \underline{wc}_i(1)]$ .

En el ejemplo de la izquierda de la Figura 5.15, se puede ver, una vez calculada  $\underline{wc}_i(0.5)$ , que  $\underline{wc}_i(0) < d_i < \underline{wc}_i(0.5)$ , es decir, el punto de corte se encuentra en el subintervalo izquierdo  $[\underline{wc}_i(0), \underline{wc}_i(0.5)]$ . Por tanto, para calcular  $\underline{wc}_i(0.25)$   $s$  se inicializa al límite inferior de dicho subintervalo izquierdo:  $\underline{wc}_i(0)$ .

En el ejemplo de la derecha de la Figura 5.15, se verifica que  $\underline{wc}_i(0.5) < d_i < \underline{wc}_i(1)$ , es decir, el punto de corte se encuentra en el subintervalo derecho, que es  $[\underline{wc}_i(0.5), \underline{wc}_i(1)]$ . Para calcular  $\underline{wc}_i(0.75)$ ,  $s$  se inicializa a  $\underline{wc}_i(0.5)$  que es el límite inferior del subintervalo derecho.

A continuación se presentan dos ejemplos en la Figura 5.16 donde el punto de corte entre el tiempo de finalización borroso  $WC_i$  y el plazo  $d_i$  está en la zona de los límites superiores de los  $\alpha$ -cortes de  $WC_i$ . En los dos ejemplos de esta figura, para calcular  $\overline{wc}_i(1)$  se inicializa  $s$  a  $\underline{wc}_i(1)$ , y para calcular  $\overline{wc}_i(0)$   $s$  se inicializa a  $\overline{wc}_i(1)$ . Como se cumple que  $\overline{wc}_i(1) < d_i < \overline{wc}_i(0)$ , la zona de búsqueda se encuentra en los límites

superiores. Para calcular  $\overline{wc}_i(0.5)$ ,  $s$  se inicializa a  $\overline{wc}_i(1)$ . De esta forma, el intervalo  $[\overline{wc}_i(1), \overline{wc}_i(0)]$  queda dividido en dos subintervalos  $[\overline{wc}_i(1), \overline{wc}_i(0.5)]$  y  $[\overline{wc}_i(0.5), \overline{wc}_i(0)]$ .

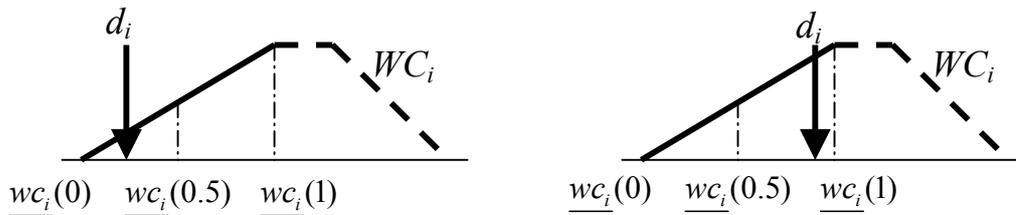


Figura 5.15: Dos ejemplos de búsqueda por bipartición en la zona de los límites inferiores

En el ejemplo de la izquierda de la Figura 5.16, se verifica  $\overline{wc}_i(1) < d_i < \overline{wc}_i(0.5)$ , con lo que el punto de corte se encuentra en el subintervalo izquierdo  $[\overline{wc}_i(1), \overline{wc}_i(0.5)]$ . Así, para calcular  $\overline{wc}_i(0.75)$ ,  $s$  se inicializa a  $\overline{wc}_i(1)$ , que es el límite inferior de dicho subintervalo.

Por otro lado, en el ejemplo de la derecha de la Figura 5.16 se tiene que  $\overline{wc}_i(0.5) < d_i < \overline{wc}_i(0)$ , lo que significa que la zona de búsqueda se encuentra en el subintervalo derecho  $[\overline{wc}_i(0.5), \overline{wc}_i(0)]$ . Para calcular  $\overline{wc}_i(0.25)$ ,  $s$  se inicializa a  $\overline{wc}_i(0.5)$ , que es el límite inferior de dicho subintervalo.

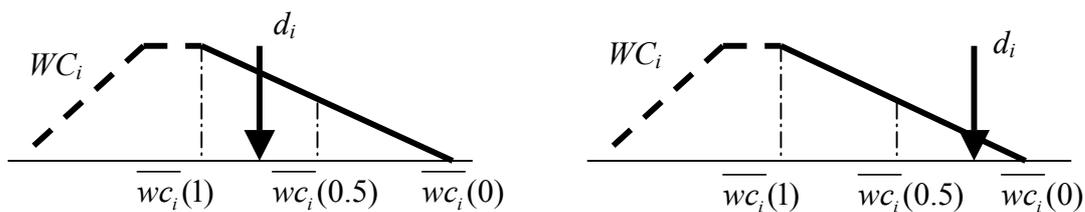


Figura 5.16: Dos ejemplos de búsqueda por bipartición en la zona de los límites superiores

Si la forma del conjunto borroso  $WC_i$  fuera como la representada en las figuras anteriores, podría pensarse que el avance logrado con la nueva inicialización de  $s$  es pequeño. Sin embargo, no hay que olvidar que la función que calcula el peor tiempo de una tarea es discontinua, debido a la ejecución de tareas más prioritarias. Con

esto, en el caso de los límites inferiores, la distancia que puede haber de  $\underline{wc}_i(0)$  a  $\underline{wc}_i(1)$  puede ser considerable debido a la interferencia de tareas más prioritarias (ver Figura 5.17), y actualizar la inicialización de  $s$ , en lugar de utilizar siempre  $\underline{wc}_i(0)$ , puede llegar a ser muy beneficioso.

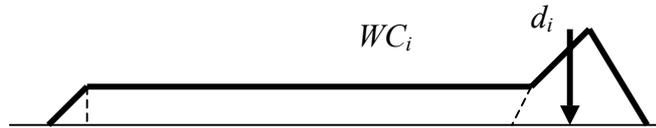


Figura 5.17: Ejemplo de  $WC_i$  cuando interfieren tareas más prioritarias

Con esta inicialización, el algoritmo basado en aritmética de intervalos queda modificado con los nuevos valores de  $s\_in_i$ .

**ALG Aritmética de Intervalos**

$$\underline{wc}_0(0) = 0$$

for  $i=1$  to  $n\_tareas$  do

$$\underline{wc}_i(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{matrix} \right)$$

$$\underline{wc}_i(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(1); s\_in_i = \underline{wc}_i(0) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_i(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(1); s\_in_i = \underline{wc}_i(1) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_i(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(0); s\_in_i = \overline{wc}_i(1) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{matrix} \right)$$

ALG ZONA

end

(5.15)  $\Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq d_i) \} \quad \forall i$

El algoritmo ZONA es el mismo que aparece en (5.12), mientras que el algoritmo *Bipartición inf* que va calculando la cota inferior del intervalo de búsqueda en cada paso y con ella inicializa el algoritmo básico queda como sigue:

**Bipartición inf**  
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$ ,  $cota\_inf = \underline{wc}_i(0)$ ,  $cota\_sup = \underline{wc}_i(1)$   
 while  $(|\underline{wc}_i(\alpha) - d_i| > \varepsilon)$   

$$\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$$

$$\underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(\alpha); \\ s\_in_i = cota\_inf \end{array} \right)_{\substack{1 \leq j \leq i \\ 1 \leq j \leq i}}$$
 if  $(\underline{wc}_i(\alpha) < d_i)$   $\alpha_0 = \alpha$  ,  $cota\_inf = \underline{wc}_i(\alpha)$   
 if  $(\underline{wc}_i(\alpha) > d_i)$   $\alpha_1 = \alpha$  ,  $cota\_sup = \underline{wc}_i(\alpha)$   
 end  
 $\Pi(WC_i \leq d_i) = \alpha$  ;  $N(WC_i \leq d_i) = 0$

De forma similar, el algoritmo *Bipartición sup* inicializa el algoritmo básico con la cota inferior del intervalo de búsqueda:

**Bipartición sup**  
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$ ,  $cota\_inf = \overline{wc}_i(1)$ ,  $cota\_sup = \overline{wc}_i(0)$   
 while  $(|\overline{wc}_i(\alpha) - d_i| > \varepsilon)$   

$$\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$$

$$\overline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); \\ s\_in_i = cota\_inf \end{array} \right)_{\substack{1 \leq j \leq i \\ 1 \leq j \leq i}}$$
 if  $(\overline{wc}_i(\alpha) < d_i)$   $\alpha_1 = \alpha$  ,  $cota\_inf = \overline{wc}_i(\alpha)$   
 if  $(\overline{wc}_i(\alpha) > d_i)$   $\alpha_0 = \alpha$  ,  $cota\_sup = \overline{wc}_i(\alpha)$   
 end  
 $\Pi(WC_i \leq d_i) = 1$  ;  $N(WC_i \leq d_i) = 1 - \alpha$

### Plazo borroso

Cuando los plazos son borrosos se pueden calcular los valores de posibilidad y necesidad de que el sistema sea planificable aplicando de nuevo aritmética de intervalos.

Como se puede ver en la Figura 5.18, la posibilidad y necesidad de que la tarea finalice antes de su plazo borroso se puede calcular a partir del punto de corte entre

ambos conjuntos borrosos. Concretamente, en la Figura 5.18 a) dicho punto es el punto de corte entre los límites superiores de los  $\alpha$ -cortes de  $WC_i$  y los límites inferiores de  $D_i$ . En la Figura 5.18 b) es el punto de corte entre los límites inferiores de  $WC_i$  y los límites superiores de  $D_i$ .

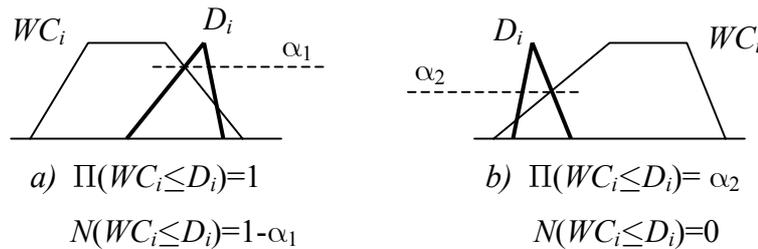


Figura 5.18: Dos ejemplos de comparación entre tiempo de finalización y plazo borrosos

El algoritmo (5.15) basado en aritmética de intervalos permanece igual, sustituyendo el valor numérico  $d_i$  por el valor borroso  $D_i$ , mientras que el algoritmo ZONA (5.16) que determina la zona de búsqueda, compara los límites de los  $\alpha$ -cortes de  $WC_i$  con los límites de los  $\alpha$ -cortes de  $D_i$ .

**ALG ZONA**

if ( $\underline{d}_i(0) \geq \overline{wc}_i(0)$ )  $\Pi(WC_i \leq D_i) = 1$ ,  $N(WC_i \leq D_i) = 1$   
 if ( $\overline{d}_i(0) \leq \underline{wc}_i(0)$ )  $\Pi(WC_i \leq D_i) = 0$ ,  $N(WC_i \leq D_i) = 0$   
 if ( $\overline{wc}_i(1) \geq \underline{d}_i(1)$ )  $N(WC_i \leq D_i) = 0$   
     else if ( $\overline{wc}_i(1) < \underline{d}_i(1)$ ) Bipartición sup  
 if ( $\underline{wc}_i(1) \leq \overline{d}_i(1)$ )  $\Pi(WC_i \leq D_i) = 1$   
     else if ( $\underline{wc}_i(1) > \overline{d}_i(1)$ ) Bipartición inf

(5.16)

El algoritmo de bipartición de los límites inferiores de  $WC_i$  (5.17), hace las comparaciones con los límites superiores del plazo  $D_i$  correspondiente. Así pues, cuando para un valor de  $\alpha$ ,  $\overline{d}_i(\alpha)$  está a la derecha de  $\underline{wc}_i(\alpha)$ , el punto de corte se encuentra en el subintervalo derecho, con lo que para el siguiente valor de  $\alpha$ ,  $s$  se inicializa a  $\underline{wc}_i(\alpha)$ . En la Figura 5.19 a), la zona de búsqueda es  $[\underline{wc}_i(0), \underline{wc}_i(1)]$ . Para  $\alpha=0.5$  se verifica  $\underline{wc}_i(\alpha) < \overline{d}_i(\alpha)$ , con lo que el punto de corte se encuentra en

el subintervalo derecho  $[\underline{wc}_i(0.5), \underline{wc}_i(1)]$ . Para calcular el tiempo de finalización correspondiente a  $\alpha=0.75$ , se inicializa  $s$  a  $\underline{wc}_i(0.5)$ . Por otro lado, en la Figura 5.19 b), el punto de corte se encuentra en el subintervalo izquierdo  $[\underline{wc}_i(0), \underline{wc}_i(0.5)]$ , ya que se verifica  $\bar{d}_i(\alpha) < \underline{wc}_i(\alpha)$ , con  $\alpha=0.5$ . Por tanto, para calcular el tiempo de finalización correspondiente a  $\alpha=0.25$ , se inicializa  $s$  a  $\underline{wc}_i(0)$ .

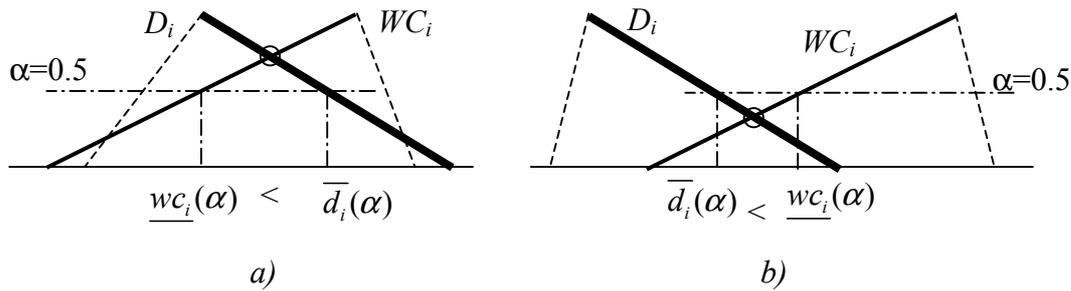


Figura 5.19: Búsqueda por bipartición en los límites inferiores de  $WC_i$

```

Bipartición inf
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$ ,  $cota\_inf = \underline{wc}_i(0)$ ,  $cota\_sup = \underline{wc}_i(1)$ 
while  $(|\underline{wc}_i(\alpha) - \bar{d}_i(\alpha)| > \epsilon)$ 
     $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ 
     $\underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(\alpha); s\_in_i = cota\_inf \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
    if  $(\underline{wc}_i(\alpha) < \bar{d}_i(\alpha))$   $\alpha_0 = \alpha$  ,  $cota\_inf = \underline{wc}_i(\alpha)$ 
    if  $(\underline{wc}_i(\alpha) > \bar{d}_i(\alpha))$   $\alpha_1 = \alpha$  ,  $cota\_sup = \underline{wc}_i(\alpha)$ 
end
 $\Pi(WC_i \leq D_i) = \alpha$  ;  $N(WC_i \leq D_i) = 0$ 
    
```

El algoritmo de bipartición de los límites superiores de  $WC_i$  (5.18) hace las comparaciones con los límites inferiores del plazo  $D_i$  correspondiente. Así pues, cuando para un valor de  $\alpha$ ,  $\bar{d}_i(\alpha)$  está a la derecha de  $\underline{wc}_i(\alpha)$ , el punto de corte se

encuentra en el subintervalo derecho, por lo que el siguiente valor de  $s$  se inicializa a  $\overline{wc}_i(\alpha)$ .

En la Figura 5.20 a), la zona de búsqueda es  $[\overline{wc}_i(1), \overline{wc}_i(0)]$ . Para  $\alpha=0.5$  se puede ver que  $\overline{wc}_i(\alpha) < \underline{d}_i(\alpha)$ , con lo que el punto de corte se encuentra en el subintervalo derecho  $[\overline{wc}_i(0.5), \overline{wc}_i(0)]$ . Para calcular el tiempo de finalización correspondiente a  $\alpha=0.25$ , se inicializa  $s$  a  $\overline{wc}_i(0.5)$ . Por otro lado, en la Figura 5.20 b), el punto de corte se encuentra en el subintervalo izquierdo  $[\overline{wc}_i(1), \overline{wc}_i(0.5)]$ , ya que se cumple que  $\underline{d}_i(\alpha) < \overline{wc}_i(\alpha)$ , con  $\alpha=0.5$ . Por tanto, para calcular el tiempo de finalización correspondiente a  $\alpha=0.75$ , se inicializa  $s$  a  $\overline{wc}_i(1)$ .

```

Bipartición sup
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$  ,  $cota\_inf = \overline{wc}_i(1)$  ,  $cota\_sup = \overline{wc}_i(0)$ 
while  $(|\overline{wc}_i(\alpha) - \underline{d}_i(\alpha)| > \epsilon)$ 
     $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ 
     $\overline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(\alpha); s\_in_i = cota\_inf \\ 1 \leq j \leq i \\ 1 \leq j \leq i \end{matrix} \right)$ 
    if  $(\overline{wc}_i(\alpha) < \underline{d}_i(\alpha))$   $\alpha_1 = \alpha$  ,  $cota\_inf = \overline{wc}_i(\alpha)$ 
    if  $(\overline{wc}_i(\alpha) > \underline{d}_i(\alpha))$   $\alpha_0 = \alpha$  ,  $cota\_sup = \overline{wc}_i(\alpha)$ 
end
 $\Pi(WC_i \leq D_i) = 1$  ;  $N(WC_i \leq D_i) = 1 - \alpha$ 
    
```

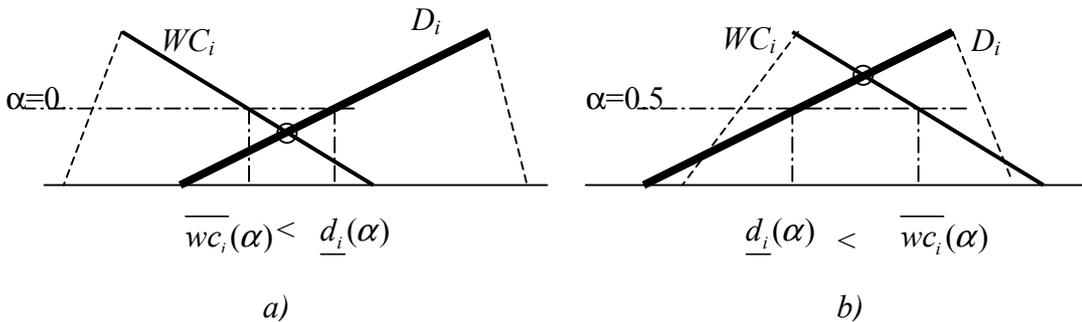


Figura 5.20: Búsqueda por bipartición en los límites superiores de  $WC_i$

En los algoritmos anteriores propuestos en este apartado se supone que los tiempos de finalización se encuentran siempre antes de la finalización del periodo. Resulta sencillo adaptar el algoritmo de aritmética borrosa para contemplar el caso en que el tiempo de finalización sobrepase el periodo. Modificando el algoritmo básico como se indica en (5.19), cuando se detecta que  $s_{i,k}$  ha sobrepasado el periodo se interrumpe la iteración y se iguala  $wc_i$  a  $t_i + \varepsilon_t$ , donde  $\varepsilon_t$  es un número real positivo cualquiera. Eso supone truncar el tiempo borroso de finalización como se indica en la Figura 5.21. La comparación con el plazo sigue dando el valor exacto de posibilidad y necesidad de que la tarea sea planificable.

(5.19) **ALG BASICO:**

$$\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k} \text{ and } s_{i,k} \leq t_i) \quad k = k + 1 \\ \text{if } s_{i,k} = s_{i,k-1} \text{ then } wc_i = s_{i,k} \\ \text{else } wc_i = t_i + \varepsilon_t \\ \text{dado, } s_{i,k}(s_{i,k-1}) = \sum_{h=1}^i p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil \end{array} \right.$$

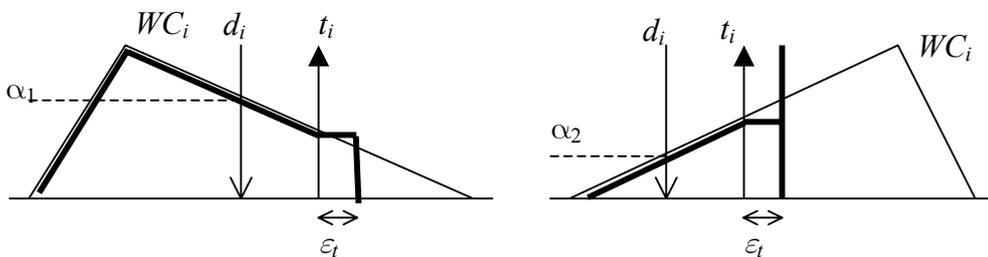


Figura 5.21: Aproximación al tiempo borroso de finalización cuando existe sobrepaso del periodo.

En el caso del algoritmo basado en aritmética de intervalos, todas las operaciones son operaciones con números reales y la búsqueda por bipartición determina el número de llamadas al algoritmo básico. El número de veces que se ejecuta el algoritmo básico es  $n \cdot n\_tarefas$ , donde  $n$  se obtiene como se indica a continuación.

Inicialmente, se calcula  $wc_i$  para  $\alpha=0$  y  $\alpha=1$ , con lo que  $n = 2$  y la longitud del intervalo o resolución en  $\alpha$  es  $r=1$ . Se calcula otro valor de  $wc_i$  para  $\alpha=0.5$ , con lo que  $n = 3$  y la longitud del intervalo es  $1/2$ . En general, al calcular  $n$  valores de  $wc_i$  por bipartición, la longitud del intervalo es  $1/2^{n-2}=r$ . Así pues, despejando  $n$  en función de la resolución  $r$  que se desea obtener, se tiene:

$$n \geq 2 - \frac{\log(r)}{\log(2)}$$

El número de veces que se ejecuta el algoritmo básico modificado es ( $n \cdot n\_tareas$ ), independientemente de la forma de los tiempos de ejecución borrosos.

Además, hay que tener en cuenta que gracias a la mejora introducida en la inicialización el tiempo que tarda el algoritmo básico para una tarea no es  $n$  veces el tiempo  $tt_1$  que tarda en la primera llamada ( $n \cdot tt_1$ ), sino ( $tt_1 + tt_2 + \dots + tt_n$ ), tal que  $tt_1 \geq tt_2 \geq \dots \geq tt_n$

#### 5.2.4. Extensión del análisis para periodos, tiempos de bloqueo y retraso de la activación borrosos

Supongamos que existe un instante crítico y que los periodos de las tareas son números borrosos, lo que quiere decir que a más ejecuciones de las tareas, el instante listo es un número borroso menos específico y con mayor soporte.

Tal y como se dijo anteriormente, la función  $f$  que representa el algoritmo básico es no decreciente en los tiempos de ejecución. Además, se puede observar fácilmente que  $wc_i$  es no creciente respecto a los periodos, es decir, si los periodos aumentan,  $wc_i$  disminuye o permanece igual.

Según lo estudiado en el capítulo 4, si  $f$  es no decreciente en  $a$  y no creciente en  $b$ , los límites inferiores y superiores del los  $\alpha$ -cortes del resultado se pueden obtener como  $\underline{c}(\alpha) = f(\underline{a}(\alpha), \overline{b}(\alpha))$  y  $\overline{c}(\alpha) = f(\overline{a}(\alpha), \underline{b}(\alpha))$

Por tanto, los límites de los  $\alpha$ -cortes de los tiempos de finalización se pueden calcular como:

$$\underline{wc}_i(\alpha) = f\left(\underline{p}_j(\alpha), \overline{t}_j(\alpha)\right)$$

$$\overline{wc}_i(\alpha) = f\left(\overline{p}_j(\alpha), \underline{t}_j(\alpha)\right)$$

Se puede aplicar tanto aritmética borrosa como aritmética de intervalos para la obtención de aproximaciones válidas de los tiempos de finalización. Para extender el

algoritmo (5.8) basado en aritmética borrosa basta con sustituir el periodo numérico  $t_i$  por periodo borroso  $T_i$ .

Además de considerar los tiempos de ejecución y los periodos borrosos, también se puede extender el algoritmo de análisis de planificabilidad a tiempos de bloqueo y “release jitter” borrosos extendiendo la función iterativa clásica:

$$s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1} + j_j}{t_j} \right\rceil$$

$$wc_i = s_{i,k} + j_i$$

Se puede ver fácilmente que el algoritmo básico que tiene en cuenta estos dos parámetros  $b_i$  y  $j_i$  es no decreciente en ambos, con lo que se puede aplicar tanto aritmética de intervalos como aritmética borrosa para extender el análisis a parámetros borrosos, ya que se verifica:

$$\underline{wc}_i(\alpha) = f \left( \underbrace{p_j(\alpha)}_{1 \leq j \leq i}, \underbrace{b_i(\alpha)}_{1 \leq j \leq i}, \underbrace{j_j(\alpha)}_{1 \leq j < i}, \underbrace{t_j(\alpha)}_{1 \leq j < i} \right)$$

$$\overline{wc}_i(\alpha) = f \left( \overbrace{p_j(\alpha)}_{1 \leq j \leq i}, \overbrace{b_i(\alpha)}_{1 \leq j \leq i}, \overbrace{j_j(\alpha)}_{1 \leq j < i}, \overbrace{t_j(\alpha)}_{1 \leq j < i} \right)$$

El algoritmo de análisis de planificabilidad basado en aritmética borrosa con tiempos de ejecución, plazos, periodos, tiempos de bloqueo y “release jitter” borrosos queda como sigue:

<p><b>Algoritmo de Aritmética Borrosa</b></p> <p><math>WC_0 = 0</math> (crisp)</p> <p>for <math>i = 1</math> to <math>n\_tarefas</math> do</p> <p style="padding-left: 20px;"><math>S_{i,0} = WC_{i-1} + P_i</math></p> <p style="padding-left: 20px;">while <math>(S_{i,k-1} \neq S_{i,k}(S_{i,k-1}))</math> <math>k=k+1</math></p> <p style="padding-left: 20px;"><math>WC_i = S_{i,k} + J_i</math></p> <p style="padding-left: 20px;"><math>\Pi / N(WC_i \leq D_i)</math></p> <p>end</p> <p><math>\Pi_s / N_s = \min \{ \Pi / N(WC_i \leq D_i) \} \quad \forall i</math></p> <p>dado <math>S_{i,k}(S_{i,k-1}) = B_i + P_i + \sum_{j=1}^{i-1} P_j \cdot \left\lceil \frac{S_{i,k-1} + J_j}{T_j} \right\rceil</math></p>
---

También se puede calcular la planificabilidad mediante aritmética de intervalos para los mismos parámetros borrosos. Cuando los tiempos de ejecución, los tiempos de bloqueo, los “release jitter” y los periodos son borrosos, los argumentos de la llamada al algoritmo básico son los valores numéricos de  $p_i$ ,  $b_i$ ,  $j_i$  y  $t_i$  con los que se debe calcular el tiempo de finalización, así como la inicialización  $s\_in_i$ .

**ALG Aritmética de Intervalos**

$\underline{wc}_0(0) = 0$

for  $i = 1$  to  $n\_tarefas$  do

$$\underline{wc}_i(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(0); b_i = \underline{b}_i(0); j_j = j_j(0); t_j = \overline{t}_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \\ \hline 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\underline{wc}_i(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(1); b_i = \underline{b}_i(1); j_j = j_j(1); t_j = \overline{t}_j(1); s\_in_i = \underline{wc}_i(0) \\ \hline 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_i(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); t_j = t_j(1); s\_in_i = \underline{wc}_i(1) \\ \hline 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_i(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); t_j = t_j(0); s\_in_i = \overline{wc}_i(1) \\ \hline 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

ALG ZONA

end

$\Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq D_i) \} \quad \forall i$

El algoritmo básico que tiene en cuenta los tiempos de bloqueo y los “release jitter” es:

**ALG BASICO:**

$$\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ \quad wc_i = s_{i,k} + j_i \\ \text{dado, } s_{i,k} = b_i + p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1} + j_h}{t_h} \right\rceil \end{array} \right.$$

El algoritmo ZONA que aparece en (5.16) permanece igual y los algoritmos Bipartición inferior y Bipartición superior que realizan la llamada al algoritmo básico con los límites de los  $\alpha$ -cortes de los parámetros borrosos son ahora:

**Bipartición inf**
 $\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_i(0), \text{cota\_sup} = \underline{wc}_i(1)$ 

 while  $\left( \left| \underline{wc}_i(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right)$ 

$$\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$$

$$\underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{c} p_j = \underline{p}_j(\alpha); b_i = \underline{b}_i(\alpha); j_j = \underline{j}_j(\alpha); t_j = \underline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$$

 if  $\left( \underline{wc}_i(\alpha) < \underline{d}_i(\alpha) \right) \alpha_0 = \alpha, \text{cota\_inf} = \underline{wc}_i(\alpha)$ 

 if  $\left( \underline{wc}_i(\alpha) > \underline{d}_i(\alpha) \right) \alpha_1 = \alpha, \text{cota\_sup} = \underline{wc}_i(\alpha)$ 

end

$$\Pi(WC_i \leq D_i) = \alpha; \quad N(WC_i \leq D_i) = 0$$

**Bipartición sup**
 $\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{wc}_i(1), \text{cota\_sup} = \overline{wc}_i(0)$ 

 while  $\left( \left| \overline{wc}_i(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right)$ 

$$\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$$

$$\overline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{c} p_j = \overline{p}_j(\alpha); b_i = \overline{b}_i(\alpha); j_j = \overline{j}_j(\alpha); t_j = \underline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$$

 if  $\left( \overline{wc}_i(\alpha) < \underline{d}_i(\alpha) \right) \alpha_1 = \alpha, \text{cota\_inf} = \overline{wc}_i(\alpha)$ 

 if  $\left( \overline{wc}_i(\alpha) > \underline{d}_i(\alpha) \right) \alpha_0 = \alpha, \text{cota\_sup} = \overline{wc}_i(\alpha)$ 

end

$$\Pi(WC_i \leq D_i) = 1; \quad N(WC_i \leq D_i) = 1 - \alpha$$

Ambos algoritmos (Bipartición superior y Bipartición inferior) hacen las llamadas al algoritmo básico con los valores de los límites de los  $\alpha$ -cortes de  $P_i$ ,  $B_i$ ,  $J_i$  y  $T_i$  que correspondan.

### 5.3. Asignación óptima de prioridades fijas con parámetros borrosos

En los apartados anteriores se ha propuesto un conjunto de algoritmos para el análisis de planificabilidad para un sistema con parámetros temporales modelados mediante conjuntos borrosos, dada una asignación de prioridades fijas. Sin embargo no se ha abordado cómo elegir la mejor asignación de prioridades.

Recordemos que cuando los parámetros son numéricos, se dice que el algoritmo de planificación DMS es óptimo entre los algoritmos de prioridades fijas, cuando los plazos son menores o iguales que los periodos, porque si existe una asignación de prioridades fijas que sea planificable, la asignación dada por DMS también lo es.

Más aún, DMS es óptimo en el sentido de la regla de Jackson que dice que “dado un conjunto de  $n$  tareas independientes, cualquier algoritmo que ejecute las tareas en orden no decreciente de los plazos es óptimo en el sentido de minimizar el retraso máximo” [Buttazzo, 1997]. O lo que es lo mismo, si todos los tiempos de finalización están antes de los correspondientes plazos, dadas dos asignaciones planificables y una de ellas basada en DMS, la distancia mínima de los tiempos de finalización a los plazos correspondientes es máxima para la asignación DMS.

Hay que tener en cuenta que cuando el plazo es mayor que el periodo, ni RMS ni DMS son óptimos [Lehoczky, 1990].

Sin embargo, cuando los parámetros que definen las tareas son números borrosos y la planificabilidad del sistema se expresa con un grado de posibilidad y de necesidad, se hace necesario redefinir el concepto de optimalidad.

La definición que se propone en esta tesis es la siguiente:

*Diremos que un algoritmo de planificación es óptimo si maximiza la posibilidad y la necesidad de que el sistema sea planificable.*

No se plantea el problema como una búsqueda multicriterio ya que ambas medidas se maximizan a la vez. Con esta definición y basándose en la definición de óptimo clásico, se puede probar que el algoritmo DMS es óptimo también con parámetros borroso. La asignación de prioridades dada por DMS cuando los plazos son numéricos y los tiempos de ejecución son números borrosos es óptima. Esta afirmación es un caso particular de la siguiente: si los plazos son números borrosos y existe un orden fuerte entre ellos, la asignación de prioridades obtenida por dicho orden es óptima.

En [Burns, Wellings, 1989] se prueba, con parámetros numéricos, que el algoritmo de planificación DMS es óptimo, en el sentido de que si una asignación de prioridades fijas es planificable, la asignación obtenida con DMS es también planificable. En lo que sigue se amplía dicha prueba para mostrar que DMS también es óptimo en el sentido de que minimiza el retraso numérico máximo. A partir de la prueba se desarrolla la prueba extendida a parámetros borrosos.

### Prueba con parámetros numéricos

DMS es óptimo si para cualquier conjunto de tareas, que es planificable por una asignación de prioridades  $\sigma$ , es también planificable por DMS. La prueba de optimalidad consiste en transformar las prioridades asignadas por  $\sigma$ , hasta llegar a la ordenación DMS. Cada paso de la transformación conserva la planificabilidad.

Se considera que todas las tareas se encuentran ordenadas según sus prioridades. Sean  $i$  y  $j$  dos tareas con prioridades  $pr_i$  y  $pr_j$  adyacentes, tal que en  $\sigma$  se verifica  $pr_i > pr_j$  y  $d_i > d_j$ .

Se define una nueva asignación  $\sigma'$  a partir de  $\sigma$ , intercambiando las prioridades de  $i$  y  $j$ . Veamos la planificabilidad del sistema con  $\sigma'$ :

- Ninguna tarea con prioridad mayor que  $pr_i$  se ve afectada por el cambio de prioridades más bajas que él.
- Ninguna tarea con prioridad menor que  $pr_j$  se ve afectada, ya que sufren la misma interferencia en su ejecución de  $i$  y de  $j$ .
- La tarea  $j$ , que era planificable en  $\sigma$ , tiene ahora una prioridad mayor sufriendo así menos interferencia, con lo que sigue siendo planificable en  $\sigma'$ .

Falta por probar que ahora la tarea  $i$ , que ha visto reducida su prioridad, sigue siendo planificable.

En  $\sigma$ ,  $wc_j \leq d_j$ ,  $d_j < d_i$  y  $d_i \leq t_i$  y por tanto el proceso  $i$  interfiere solamente una vez durante la ejecución de  $j$ .

Al intercambiar las prioridades, el nuevo tiempo de respuesta de  $i$ ,  $wc_i'$  es igual al antiguo tiempo de respuesta de  $j$ ,  $wc_j$ . Esto es cierto ya que con ambas asignaciones el tiempo de computación  $p_i + p_j$  se ha completado con el mismo nivel de interferencia de procesos más prioritarios. El proceso  $j$  ha sido disparado una sola vez durante  $wc_j$  y por tanto, interfiere una vez durante la ejecución de  $i$  en  $\sigma'$ . Por ello:

$$wc_i' = wc_j \leq d_j < d_i$$

Por tanto, la tarea  $i$  es planificable después del cambio de prioridades.

Continuando este proceso de cambio de prioridad de dos tareas según el criterio anterior, se conserva cada vez la planificabilidad, hasta obtener  $\sigma_{DMS}$ , que a su vez es también planificable.

Esta prueba, propuesta en [Burns, Wellings, 1989], no analiza sin embargo la distancia de los tiempos de finalización a los plazos correspondientes. En lo que sigue se amplía la prueba para mostrar que DMS maximiza la mínima distancia de los tiempos de finalización a los plazos.

Por un lado se tiene que  $d_i - wc_i > d_j - wc_j$  porque en  $\sigma$   $i$  es más prioritaria, terminando antes su ejecución y porque  $d_i > d_j$ . Por tanto la distancia mínima de las dos es la de  $j$  en la asignación  $\sigma$ .

De la última relación  $wc_i' = wc_j \leq d_j < d_i$ , se puede observar que la distancia del nuevo tiempo de respuesta de  $i$ ,  $wc_i'$  hasta su plazo  $d_i$  es mayor que la distancia del antiguo tiempo de respuesta de  $j$   $wc_j$  a su plazo  $d_j$ :

$$d_i - wc_i' > d_j - wc_j$$

Además, como en  $\sigma'$  la tarea  $j$  se ejecuta antes por tener mayor prioridad, su distancia al plazo aumenta:

$$d_j - wc_j' > d_j - wc_j$$

En consecuencia, las dos tareas que intercambian sus prioridades al pasar de  $\sigma$  a  $\sigma'$  tienen en  $\sigma'$  una distancia mayor a sus plazos que el mínimo de ambos en  $\sigma$  (que era la distancia de  $j$ ). Podemos afirmar con esto que en cada paso, además de mantener la planificabilidad, se aumenta la mínima distancia de los tiempos de respuesta de esas tareas a sus plazos.

### Prueba con parámetros borrosos

A partir de esta prueba se puede extender el análisis al caso de tener parámetros borrosos, y existir orden fuerte entre los plazos. Recordemos (ver el capítulo 3) que un número borroso  $A$  es fuertemente mayor que un número borroso  $B$  ( $A >_s B$ ) si  $A \neq B$  y los lados verifican:

$$f_A(x) \leq f_B(x)$$

$$g_A(x) \geq g_B(x)$$

siendo  $f_A$  la función que define el lado izquierdo de un número borroso  $A$  y  $g_A$  la función que define el lado derecho de  $A$ .

En la Figura 5.22 se muestran dos ejemplos, uno en el que  $A$  es fuertemente mayor que  $B$  ( $A >_s B$ ) y otro en el que  $A'$  no es fuertemente mayor que  $B'$  ( $A' \not>_s B'$ )

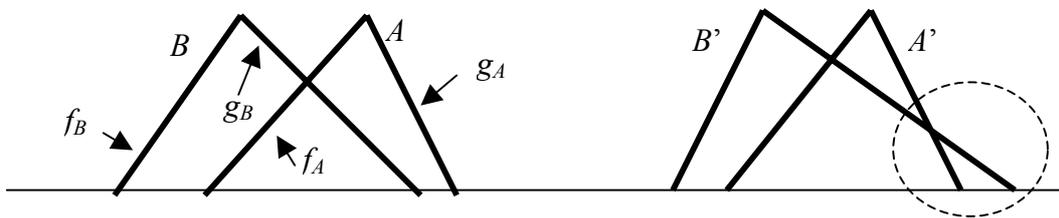


Figura 5.22: Ejemplo de ( $A >_s B$ ) y ( $A' \not>_s B'$ )

Supongamos que tanto los tiempos de ejecución como los plazos son números borrosos ( $P_i$  y  $D_i$ ) y que existe orden fuerte entre los plazos. Como los tiempos de ejecución son borrosos, los tiempos de finalización también lo son ( $WC_i$ ). Supongamos además que los plazos son fuertemente menores que los periodos y que el sistema es planificable según la asignación  $\sigma$ , con un determinado grado de posibilidad y de necesidad.

Sean  $i$  y  $j$  dos tareas con prioridades adyacentes, tal que en  $\sigma$   $pr_i > pr_j$  y  $D_i >_s D_j$ .

Supongamos que la de menor prioridad,  $j$  es planificable con un cierto valor de posibilidad y necesidad.

Se define la asignación  $\sigma'$  a partir de  $\sigma$ , intercambiando las prioridades de  $i$  y  $j$ . Veamos la planificabilidad en  $\sigma'$ :

- Ninguna tarea con prioridad mayor que  $pr_i$  se ve afectada por el cambio de prioridades más bajas que él.
- Ninguna tarea con prioridad menor que  $pr_j$  se ve afectada, ya que sufren la misma interferencia en su ejecución de  $i$  y de  $j$ .

- La tarea  $j$ , que era planificable en  $\sigma$  con un valor de posibilidad y necesidad, tiene ahora una prioridad mayor sufriendo así menos interferencia. Al intercambiar las prioridades, el nuevo tiempo de respuesta de  $i$ ,  $WC_i'$  es igual al antiguo tiempo de respuesta de  $j$ ,  $WC_j$ . Esto es cierto ya que con ambas asignaciones el tiempo de ejecución  $P_i + P_j$  (suma de dos números borrosos) se ha completado con el mismo nivel de interferencia de tareas más prioritarias.  $WC_i' = WC_j = WC_j' + P_i$ . Como ambos sumandos son fuertemente mayores que cero,  $WC_j'$  es fuertemente menor que  $WC_j$ . Además  $D_j - WC_j'$  es fuertemente mayor que  $D_j - WC_j$  con lo que la posibilidad y la necesidad de que  $j$  termine antes de su plazo en  $\sigma'$  es mayor o igual que el mínimo de ambas tareas en  $\sigma$ . Es decir, la tarea  $j$  se puede considerar más planificable con las asignaciones de  $\sigma'$  en el sentido de que la posibilidad y necesidad de que sea planificable es mayor o igual que las de  $\sigma$ .
- La tarea  $i$  en  $\sigma$  tiene más prioridad que  $j$ :  $WC_j = WC_i + P_j$  con lo que  $WC_i$  es fuertemente menor que  $WC_j$ . Además  $D_j <_s D_i$  con lo que la posibilidad y necesidad de  $i$  de terminar antes que  $D_i$  es mayor o igual que la de  $j$  de terminar antes de  $D_j$ . O lo que es lo mismo, la distancia mínima de las dos es la de  $j$  en  $\sigma$ . Se define la distancia como diferencia entre el plazo y el tiempo de respuesta borrosos.
- Falta por probar que la tarea  $i$  en  $\sigma'$ , que ha visto reducida su prioridad, sigue siendo planificable con un valor de posibilidad y necesidad mayor o igual que el mínimo de ambas tareas encontrado en  $\sigma$ .

En  $\sigma$ ,  $WC_j$  es menor que  $D_j$  con un cierto valor de posibilidad y necesidad,  $D_j <_s D_i$  y  $D_i <_s t_i$  y por tanto el proceso  $i$  interfiere solamente una vez durante la ejecución de  $j$ .

Al intercambiar las prioridades, el nuevo tiempo de respuesta de  $i$ ,  $WC_i'$  es igual al antiguo tiempo de respuesta de  $j$ ,  $WC_j$ . El proceso  $j$  ha sido disparado una sola vez durante  $WC_j$  y por tanto, interfiere una vez durante la ejecución de  $i$  en  $\sigma'$ . Por ello:

$WC_i' = WC_j$  es fuertemente menor que  $D_j$  y, a su vez,  $D_j <_s D_i$

$D_i - WC_i'$  es fuertemente mayor que  $D_j - WC_j$  con lo que la posibilidad y necesidad de que  $i$  termine antes de su plazo en  $\sigma'$  es mayor que el mínimo de ambas tareas en  $\sigma$ . Cuando la necesidad de que  $i$  termine antes de su plazo en  $\sigma'$  sea menor que 1, la diferencia entre el plazo y el tiempo de respuesta correspondiente no es fuertemente mayor que cero.

En consecuencia, las dos tareas que intercambian sus prioridades al pasar de  $\sigma$  a  $\sigma'$  tienen en  $\sigma'$  una distancia fuertemente mayor a sus plazos que el mínimo de ambas

en  $\sigma$  (que era la distancia de  $j$ ). Podemos afirmar con esto que en cada paso se aumenta la mínima distancia de los tiempos de respuesta afectados a sus plazos y la posibilidad y necesidad de que el sistema sea planificable aumenta o permanece igual.

Por lo tanto la asignación de prioridades que establece DMS cuando existe orden fuerte entre los plazos borrosos es óptima ya que maximiza la posibilidad y necesidad de que el sistema sea planificable.

Nótese que también es posible presentar extender la prueba de la regla de Jackson [Buttazzo, 1997], a parámetros borrosos.

En efecto, sea  $\sigma$  una asignación de prioridades diferente de la dada por DMS, esto es, existen dos tareas  $a$  y  $b$  con prioridades adyacentes con  $D_a <_s D_b$ , tal que la tarea  $b$  precede a la tarea  $a$  en  $\sigma$ . Sea  $\sigma'$  una asignación obtenida a partir de  $\sigma$  intercambiando las prioridades de ambas tareas, de forma que tarea  $a$  precede a la tarea  $b$  en  $\sigma'$ .

En  $\sigma$  la tarea  $a$  termina  $P_a$  unidades de tiempo después que la tarea  $b$ , con lo que  $WC_a >_s WC_b$ . Como  $D_a <_s D_b$  entonces:

$\Pi(WC_a \leq D_a) \leq \Pi(WC_b \leq D_b)$  y  $N(WC_a \leq D_a) \leq N(WC_b \leq D_b)$  esto es, el valor mínimo de posibilidad y necesidad se obtiene para el tiempo de finalización de la tarea  $a$ .

En  $\sigma'$  se analizan dos casos:

- Tarea  $a$ : el mínimo valor de posibilidad y necesidad entre  $a$  y  $b$  es el correspondiente a  $a$ :  $\Pi(WC'_a \leq D_a)$  y  $N(WC'_a \leq D_a)$ . Como en  $\sigma'$  la tarea  $a$  se ejecuta antes que la tarea  $b$ , el tiempo de finalización de la tarea  $a$  en  $\sigma'$  cumple lo siguiente:  $WC'_a <_s WC_a$  con lo que,  $\Pi(WC_a \leq D_a) \leq \Pi(WC'_a \leq D_a)$  y  $N(WC_a \leq D_a) \leq N(WC'_a \leq D_a)$
- Tarea  $b$ : el mínimo valor de posibilidad y necesidad entre  $a$  y  $b$  es el correspondiente a  $b$ :  $\Pi(WC'_b \leq D_b)$  y  $N(WC'_b \leq D_b)$ . Como en  $\sigma'$  la tarea  $a$  se ejecuta antes que la tarea  $b$ ,  $WC'_b = WC_a$  y como  $D_a <_s D_b$  se tiene que  $\Pi(WC_a \leq D_a) \leq \Pi(WC'_b \leq D_b)$  y  $N(WC_a \leq D_a) \leq N(WC'_b \leq D_b)$ .

En ambos casos los valores de posibilidad y necesidad aumentan o permanecen igual. Realizando un número finito de transposiciones de tareas se puede pasar de  $\sigma$  a la asignación de prioridades dada por DMS:  $\sigma_{DMS}$ . Y como en cada paso la

posibilidad y necesidad de que el sistema sea planificable no decrece,  $\sigma_{DMS}$  es óptimo.

Se puede extender de forma similar esta conclusión a la asignación de prioridades dinámicas mediante el algoritmo EDF, si existe orden fuerte entre los plazos absolutos de las tareas.

### 5.4. Comparación con la extensión propuesta por Litoiu y Tadei

El único trabajo encontrado en la literatura de extensión del algoritmo de planificabilidad de tareas con prioridades fijas a parámetros borrosos es el de Litoiu y Tadei. En [Litoiu, Tadei, 1997] se propone un algoritmo de búsqueda de asignación de prioridades óptima cuando los plazos son borrosos y los tiempos de ejecución son numéricos. En [Litoiu, Tadei, 2001b], [Litoiu, Tadei, 2001a] extienden el algoritmo a tiempos de ejecución y plazos borrosos y la búsqueda de la asignación óptima a tiempos de ejecución borrosos. Además, en [Litoiu, Tadei, 2001b] se hace un estudio comparativo de distintos tipos de ordenación de números borrosos y en [Litoiu, Tadei, 2001a] se presenta un caso de estudio de diseño de un sistema de tiempo real.

En el modelo presentado por dichos autores se introduce una función de coste denominada satisfacción de planificabilidad para medir el cumplimiento de los plazos.

Como ya se ha visto en el capítulo 2, en general hay que estudiar el cumplimiento de todos los plazos situados dentro de un hiperperiodo, es decir, dar una medida del grado en el que los tiempos de finalización  $wc_{ij}$  de la tarea  $i$  respetan los correspondientes plazos  $d_{ij}$ .

Si los plazos son borrosos y los tiempos de ejecución son numéricos, la función de satisfacción de planificabilidad  $S_{d_i}(wc_{ij})$  de una tarea  $i$  en su ejecución  $j$  propuesta por los autores depende del tiempo de finalización de la tarea  $i$  en la ejecución  $j$  que es un valor numérico, y se calcula como ([Litoiu, Tadei, 2001a; Litoiu, Tadei, 2001b]):

$$S_{d_i}(wc_{ij}) = \begin{cases} 1 & \text{if } wc_{ij} < a_{ij} \\ 1 - \frac{\int_{a_{ij}}^{wc_{ij}} \mu(D_{ij}) d(D_{ij})}{\int_{a_{ij}}^{b_{ij}} \mu(D_{ij}) d(D_{ij})} & \text{if } a_{ij} \leq wc_{ij} \leq b_{ij} \\ 0 & \text{if } wc_{ij} > b_{ij} \end{cases}$$

Donde  $\mu(D_{ij})$  es la función de pertenencia del plazo borroso  $D_{ij}$ . La función  $S_{di}(wc_{ij})$  es continua y estrictamente decreciente en el soporte  $[a_{ij}, b_{ij}]$  de  $D_{ij}$ , ya que el plazo se ha modelado mediante un número borroso, y por tanto convexo (ver Figura 5.23).

Para una asignación de prioridades determinada, la satisfacción de planificabilidad es el mínimo de las satisfacciones de todas las tareas en todas las ejecuciones dentro de un hiperperiodo. Si se desea encontrar la asignación de prioridades óptima, el problema consiste en maximizar dicha satisfacción mínima.

Por otro lado, si los plazos son numéricos y los tiempos de ejecución son borrosos se define la satisfacción de planificabilidad  $S_{wci}(d_{ij})$  para la tarea  $i$  en su ejecución  $j$  como ([Litoiu, Tadei, 2001a; Litoiu, Tadei, 2001b]):

$$S_{wci}(d_{ij}) = \begin{cases} 0 & \text{if } d_{ij} < e_{ij} \\ \frac{\int_{e_{ij}}^{d_{ij}} \mu(WC_{ij}) d(WC_{ij})}{\int_{e_{ij}}^{f_{ij}} \mu(WC_{ij}) d(WC_{ij})} & \text{if } e_{ij} \leq d_{ij} \leq f_{ij} \\ 1 & \text{if } d_{ij} > f_{ij} \end{cases}$$

donde  $\mu(WC_{ij})$  es la función de pertenencia del tiempo de finalización borroso  $WC_{ij}$ .  $S_{wci}(d_{ij})$  es estrictamente creciente en el soporte  $[e_{ij}, f_{ij}]$  de  $WC_{ij}$  (ver Figura 5.23). Los tiempos de finalización borrosos obtenidos por los autores son conjuntos borrosos convexos debido a la simplificación que utilizan del cociente borroso, que transforman en cociente clásico, y por eso  $S_{wci}$  es estrictamente creciente. Sin embargo, tal y como se ha visto en esta tesis esos tiempos no tienen por qué ser convexos, en cuyo caso la función de satisfacción será una función monótona creciente, no estrictamente creciente.

De nuevo, para una asignación de prioridades determinada, la satisfacción de planificabilidad es el mínimo de las satisfacciones de todas las tareas en todas las ejecuciones dentro de un hiperperiodo. Si se desea encontrar la asignación de prioridades óptima, el problema consiste en maximizar dicha satisfacción mínima.

Cuando tanto los tiempos de ejecución como los plazos son borrosos la satisfacción de planificabilidad  $S_{ij}$  de una tarea  $i$  en su ejecución  $j$  se obtiene como el punto de corte de  $S_{di}$  y  $S_{wci}$  (ver Figura 5.23). Con lo que el problema de planificación consiste en maximizar el mínimo  $S_{ij}$ .

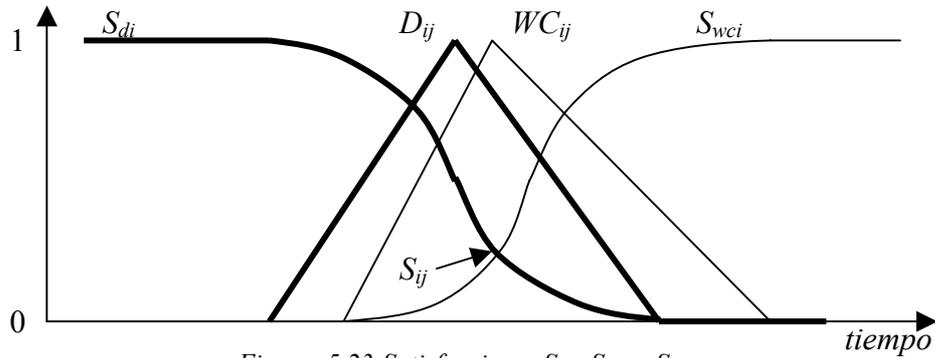


Figura 5.23: Satisfacciones  $S_{di}$ ,  $S_{wei}$  y  $S_{ij}$ .

Cuando existe un instante crítico el algoritmo (5.2) calcula el peor tiempo de ejecución de cada tarea. La extensión propuesta por Litoiu y Tadei para tiempos de ejecución borrosos (5.20) se basa en la aplicación de la aritmética borrosa y permite obtener los tiempos de finalización borrosos de las tareas, a partir de los cuales se calcularán las satisfacciones de planificabilidad. Suponen que los tiempos de finalización están antes de los periodos correspondientes.

$$\begin{aligned}
 & WC_0 = 0 \text{ (crisp)} \\
 & \text{for } i = 1 \text{ to } n\_tareas \text{ do} \\
 & \quad S_{i,0} = WC_{i-1} + P_i \\
 & \quad \text{while } (S_{i,k-1} ' < ' S_{i,k}) \quad k = k + 1 \\
 & \quad \quad WC_i = S_{i,k} \\
 & \text{end} \\
 & \text{dado } S_{i,k}(S_{i,k-1}) = B_i + \sum_{j=1}^i P_j \cdot \lceil S_{i,k-1} ' / ' t_j \rceil
 \end{aligned}
 \tag{5.20}$$

Los autores consideran además los tiempos de bloqueo que aparecen cuando se retrasa la ejecución de una tarea porque otra menos prioritaria está utilizando un recurso compartido.

Las operaciones del algoritmo (5.20) son operaciones de aritmética borrosa, salvo el cociente  $' / ' t_j$  y el redondeo hacia arriba  $\lceil \cdot \rceil$ . En (5.20)  $S_{i,k-1} ' / ' t_j$  se simplifica para el valor numérico más desfavorable  $f_i/t_j$  siendo  $f_i$  el límite superior del soporte de  $S_{i,k-1}$ . Por tanto, la operación redondeo se realiza con números clásicos. Los autores lo justifican por el hecho de que así se simplifican las operaciones y porque se contemplan mejor los requisitos de tiempo real donde se debe considerar el peor caso. Con  $' < '$  expresan una ordenación de números borrosos.

Consideramos que la extensión propuesta por Litoiu y Tadei presenta algunos problemas, relacionados con la simplificación de la división borrosa, el empleo de la función de satisfacción definida por los autores y la ordenación de los números borrosos.

### Simplificación de la división borrosa

Para hacer un análisis para el peor caso, basta con tener el mayor valor de los tiempos de ejecución y ejecutar el algoritmo de planificabilidad con parámetros numéricos. Pero si lo que se desea es aprovechar al máximo la información disponible, que está modelada en forma de conjuntos borrosos, hay que procurar no perder información en el proceso de extensión del análisis de planificabilidad. De poco sirve realizar algunas operaciones mediante aritmética borrosa si por otro lado en otras se introducen simplificaciones pesimistas que pueden hacer que el resultado sea parecido o incluso igual que el que se obtendría al realizar el análisis clásico del peor caso.

Para ver el efecto que puede tener la simplificación en la división borrosa, de forma aislada al cálculo de satisfacción, vamos a calcular las medidas de posibilidad y necesidad de planificabilidad si se aplica el cociente simplificado para el peor caso y a compararlas con las que se obtienen mediante el principio de extensión.

En el ejemplo planteado anteriormente, un sistema formado por tres tareas periódicas de periodos  $t_1=3$ ,  $t_2=5$  y  $t_3=15$ , con tiempos de ejecución borrosos triangulares definidos por sus vértices  $P_1 = (0.9, 1, 1.05)$ ,  $P_2 = (1.8, 1.9, 2)$  y  $P_3 = (0.9, 1, 1.1)$  y plazos  $d_1=2$ ,  $d_2=3.1$  y  $d_3=15$ , para la tarea 2 inicialmente se tiene  $S_{2,0} = WC_1 + P_2 = P_1 + P_2 = (2.7, 2.9, 3.05)$ .

Según la simplificación introducida por los autores, en lugar de calcular  $S_{2,1} = \lceil S_{2,0}/t_j \rceil$  se aproxima como  $\lceil 3.05/t_j \rceil$ , con lo que  $S_{2,1} = (3.6, 3.9, 4.1)$ . En la siguiente iteración se tiene  $S_{2,2} = S_{2,1} = WC_2$ . Por tanto, tal y como se puede ver en la Figura 5.24, se tiene que  $\Pi(WC_2 < d_2) = 0$  y  $N(WC_2 < d_2) = 0$  y los valores de posibilidad y necesidad de que el sistema sea planificable son  $\Pi_{planif} = 0$  y  $N_{planif} = 0$ .

Por otro lado, mediante la aplicación del principio de extensión se tiene que  $\Pi(WC_2^{pe} < d_2) = 1$  y  $N(WC_2^{pe} < d_2) = 0.66$ . Como  $WC_1^{pe}$  y  $WC_3^{pe}$  son fuertemente menores que sus correspondientes plazos, los valores de planificabilidad del sistema son  $\Pi_{planif} = 1$  y  $N_{planif} = 0.66$ .

Se puede observar que el resultado obtenido con la simplificación propuesta por Litoiu y Tadei es muy pesimista en comparación con la aplicación del principio de

extensión. Se podría haber escogido un ejemplo con posibilidad 1 y necesidad menor que 1 tan próxima como queramos a 1, que sin embargo con la simplificación que proponen hubiera dado lugar a valores de posibilidad y de necesidad nulos.

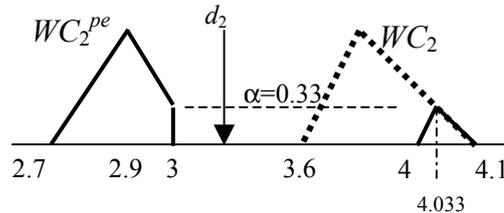


Figura 5.24: Tiempo de finalización exacto  $WC_2^{pe}$  y aproximación  $WC_2$  obtenida simplificando la división borrosa

### Satisfacción

Los autores expresan la planificabilidad del sistema en términos de grado de satisfacción, cuya función definen como se ha mostrado previamente. Esto supone realizar una transformación más o menos justificada de la información de partida. En esta tesis se propone trabajar directamente con la información disponible en forma de conjuntos borrosos y expresar la planificabilidad del sistema mediante las medidas de posibilidad y planificabilidad existentes ya en la teoría de la Posibilidad. Evitar transformaciones de la información evita pérdidas de información, así como problemas de interpretación adicionales.

Además los autores no justifican la aplicación de la aritmética borrosa. Tal y como se ha demostrado en esta tesis, los resultados de los tiempos de finalización borrosos obtenidos mediante aritmética borrosa son menos específicos que los obtenidos mediante el principio de extensión. A pesar de ser aproximaciones, la medida de posibilidad y de necesidad que se obtiene de la comparación con los plazos es la misma. Aun en el caso de que Litoiu y Tadei no hubieran realizado la simplificación de la división borrosa, no parece que el resultado expresado en términos de satisfacción de planificabilidad coincida con el que se habría obtenido mediante el principio de extensión, ya que las integrales del tiempo de finalización borroso exacto y del aproximado por aritmética borrosa son diferentes en general, como en el ejemplo de la Figura 5.25.

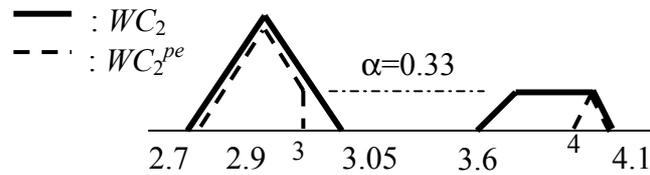


Figura 5.25:  $WC_2$  obtenido mediante aritmética borrosa y tiempo de finalización exacto  $WC_2^{pe}$

### Ordenación de números borrosos

Otro problema detectado es que los autores proponen utilizar métodos de ordenación de números borrosos para la comparación entre  $S_{i,k}$  y  $S_{i,k-1}$  ([Baas, Kwakernaak, 1977], [Watson et al., 1979], [Liou, Wang, 1992], [Buckley, 1985], [Bortolan, Degani, 1985]), lo que puede llevar a confusión ya que se olvida la relación funcional existente entre ambos, tal y como se ha analizado anteriormente en este capítulo.

Por otro lado, cuando los plazos son borrosos, los autores proponen un algoritmo de búsqueda de la asignación de prioridades que maximiza la satisfacción de planificabilidad. En [Litoiu, Tadei, 2001b] los autores comparan los resultados obtenidos según diferentes métodos de ordenación de números borrosos, aplicados a la ordenación de los plazos [Baas, Kwakernaak, 1977], [Watson et al., 1979], [Liou, Wang, 1992], [Buckley, 1985], [Bortolan, Degani, 1985].

Tal y como afirman los autores, los resultados de planificabilidad varían según el método de ordenación empleado, de forma que es cuestión de elegir el método más adecuado. A pesar del estudio comparativo de distintos métodos de ordenación de números borrosos que aparece en el artículo, consideramos que no se puede llegar a la conclusión de que un método es mejor que otro, ya que la satisfacción (o la posibilidad y necesidad de planificabilidad) depende no sólo de los plazos, sino también de los tiempos de ejecución.

Así, para los mismos plazos, dependiendo de los tiempos de ejecución de las tareas, será mejor una asignación de prioridades u otra (asociada a un orden de los plazos borrosos) y la asignación de prioridades propuesta por el método de ordenación puede ser o no la de máxima planificabilidad.

Esto se ilustra mediante el siguiente ejemplo. Dadas dos tareas tarea  $a$ , y tarea  $b$ , con plazos borrosos  $D_a$  y  $D_b$ , se trata de determinar qué tarea debe ser la más prioritaria. En la Figura 5.26 se muestran dos situaciones diferentes: que el tiempo de respuesta de la tarea menos prioritaria sea  $WC_2$  o  $WC_2'$ . En ambos casos se supone que el tiempo de finalización de la tarea más prioritaria está muy por debajo de ambos plazos con lo que los valores de posibilidad y necesidad de que la tarea más

prioritaria sea planificable son 1. Así pues, la planificabilidad del sistema vendrá dada por la planificabilidad de la tarea menos prioritaria.

a) Si el tiempo borroso de respuesta de la tarea menos prioritaria es  $WC_2$  en la Figura 5.26, los valores de planificabilidad que se obtienen con las dos asignaciones de prioridades posibles son los siguientes:

- Si el orden escogido es primero ejecutar la tarea  $a$  y luego la tarea  $b$ , el tiempo de finalización de la tarea  $b$  es  $WC_2$ , ya que es la menos prioritaria. Por tanto los valores de planificabilidad de esta tarea son  $\Pi(WC_2 \leq D_b) = 1$  y  $N(WC_2 \leq D_b) = 0.75$
- Si el orden escogido es primero la tarea  $b$  y luego la tarea  $a$ , entonces hay que comparar  $WC_2$  con  $D_a$ , con lo que  $\Pi(WC_2 \leq D_a) = 0.8$  y  $N(WC_2 \leq D_a) = 0$ .

Por tanto en este caso es mejor que la tarea  $a$  sea más prioritaria.

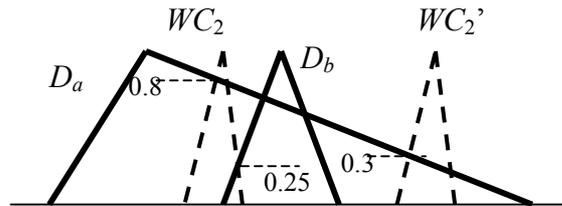


Figura 5.26: Dos casos de tiempo de finalización de la tarea menos prioritaria:  $WC_2$  y  $WC_2'$

b) Si el tiempo borroso de respuesta de la tarea menos prioritaria es  $WC_2'$  en la Figura 5.26:

- Si el orden escogido es primero la tarea  $a$  y luego la tarea  $b$ , hay que comparar  $WC_2'$  con  $D_b$  dando  $\Pi(WC_2' \leq D_b) = 0$  y  $N(WC_2' \leq D_b) = 0$
- Si el orden escogido es primero la tarea  $b$  y luego la tarea  $a$ , entonces hay que comparar  $WC_2'$  con  $D_a$  dando  $\Pi(WC_2' \leq D_a) = 0.3$  y  $N(WC_2' \leq D_a) = 0$

Por tanto en este caso es mejor que la tarea  $b$  sea más prioritaria.

Es decir, para los mismos plazos borrosos, dependiendo de los tiempos de finalización (y por tanto de los tiempos de ejecución) borrosos, la asignación óptima varía.

## 5.5. Comparación entre el análisis de planificabilidad posibilista y el probabilista

Tal y como ya se vio en el capítulo 3, y más en detalle en el anexo C, la probabilidad y la posibilidad modelan diferentes tipos de incertidumbre ( $N(E) \leq P(E) \leq \Pi(E)$ ). A pesar de ello, existen diversas propuestas de transformación entre distribuciones de probabilidad y de posibilidad. Tras haber analizado la planificabilidad de sistemas de tiempo real modelando la incertidumbre mediante números borrosos, cabe preguntarse si los resultados son comparables a los que se hubieran obtenido de haber modelado la incertidumbre mediante distribuciones de probabilidad.

Para comparar los resultados obtenidos por ambos caminos se propone el siguiente procedimiento. Se estudiarán todas las combinaciones posibles de asignaciones de prioridades fijas y se buscará la mejor asignación en términos de planificabilidad. Para ello se va a analizar un ejemplo de un sistema formado por tres tareas periódicas, con periodos numéricos. Los tiempos de ejecución y los plazos son inciertos y se modelan mediante conjuntos borrosos en un caso o mediante distribuciones de probabilidad en el otro. Cada asignación de prioridades tiene por tanto unos valores de posibilidad y necesidad de que sea planificable (que se calcularán mediante aritmética de intervalos) en el modelo posibilista, y un valor de probabilidad de que sea planificable en el modelo probabilista.

La comparación se realizará en los siguientes términos:

- La mejor asignación para cada modelo
- Grado de planificabilidad de la mejor asignación en relación con las otras asignaciones
- Rapidez del método cuantificada en número de llamadas al algoritmo básico, que es idéntico en ambos casos
- La comparación se realiza exigiendo una precisión similar ( $\epsilon$ ) en el valor de posibilidad y necesidad y en el valor de probabilidad.
- Se van a utilizar dos tipos de transformación entre distribuciones de posibilidad y de probabilidad, elegidas porque son fácilmente interpretables, de aplicación sencilla, y porque en ambos casos la aplicación de la transformación inversa permite obtener la distribución de posibilidad de partida.

Estas transformaciones son la basada en normalización (que en los ejemplos se denominará Prob<sub>1</sub>) y la basada en los principios de máxima especificidad y razón insuficiente o “insufficient reason” (que en los ejemplos se denominará Prob<sub>2</sub>), descritas ambas en el anexo C.

Dada una distribución de posibilidad, la probabilidad que se obtiene aplicando la transformación basada en normalización es:

$$p_i = \pi_i / \sum_j \pi_j \quad j=1,n$$

En el caso continuo, el denominador pasa a ser el área de la distribución de posibilidad.

La transformación inversa es  $\pi_i = p_i / p_1$ , con  $p_1 \geq p_2 \geq \dots \geq p_n$ , que en el caso continuo se expresa como  $\pi(x) = p(x) / p_m$ , con  $p_m = \sup_x p(x)$

Por tanto, si se modela un tiempo de ejecución mediante un número borroso triangular, la función de densidad de probabilidad asociada mediante esta transformación es también triangular y escalada para que su área sea 1. Ver Figura 5.27

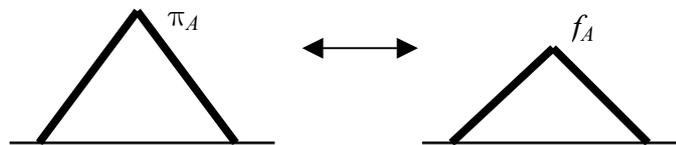


Figura 5.27: Transformación normalizada probabilidad-posibilidad

La transformación de posibilidad a probabilidad basada en máxima especificidad y razón insuficiente se aplica de la siguiente forma: dada una distribución de posibilidad  $\pi$ ,

- En el intervalo unidad: se selecciona un valor de  $\alpha$  de forma aleatoria (distribución uniforme) en  $(0,1]$  y se considera  $A_\alpha = \{x | \pi(x) \geq \alpha\}$
- En el  $\alpha$ -corte seleccionado  $A_\alpha$ : se obtiene un valor de  $x$  de forma aleatoria (distribución uniforme) en el intervalo  $A_\alpha$ .

Esta transformación aparece ilustrada en la Figura 5.28 a). La transformación inversa a esta se puede expresar de la siguiente forma:

$$\pi(x_i) = \sum_{j=1}^n \min(p(x_i), p(x_j))$$

La solución continua es  $\pi(x) = \int_{-\infty}^{\infty} \min(p(x), p(t)) dt$ . Esta transformación aparece ilustrada en la Figura 5.28 b).

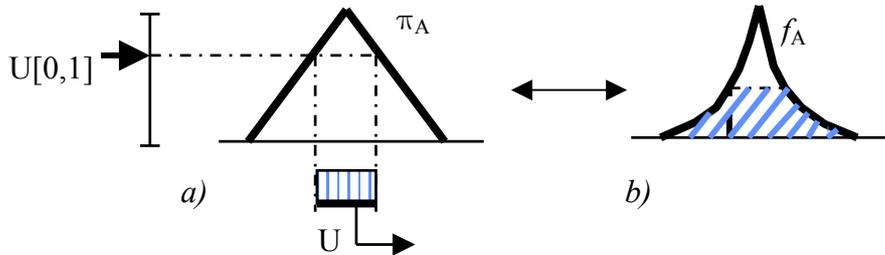


Figura 5.28: Transformación de posibilidad a probabilidad basada en máxima especificidad y razón insuficiente. Transformación de probabilidad a posibilidad inversa de la anterior

Ejemplo 1:

Los periodos son  $t_1 = 4$ ,  $t_2 = 6$  y  $t_3 = 16$ . Los tiempos de ejecución son números borrosos triangulares:  $P_1=(0.4, 0.5, 0.7)$ ,  $P_2=(0.9, 1, 1.1)$  y  $P_3=(0.7, 0.8, 0.9)$ . Los plazos son también números borrosos triangulares:  $D_1=(1.5, 2, 4)$ ,  $D_2=(2.5, 2.75, 3)$  y  $D_3=(2, 3, 3.5)$ .

$\epsilon=0.1$	Nº llamadas	Asignación de prioridades					
		3,2,1	2,3,1	3,1,2	1,3,2	2,1,3	1,2,3
$\pi$	90	$\Pi=0.875$ $N=0$	$\Pi=0.875$ $N=0$	$\Pi=1$ $N=0.75$	$\Pi=1$ $N=0.75$	$\Pi=1$ $N=0.5$	$\Pi=1$ $N=0.5$
Prob <sub>1</sub>	1500	0.57	0.5	1	1	0.92	0.93
Prob <sub>2</sub>	1500	0.42	0.44	1	1	0.96	0.96

Tabla 5-1: Ejemplo 1. Comparación Posibilidad-Probabilidad

Los resultados obtenidos por el modelado posibilista se corresponden a la fila indicada con  $\pi$ . Los resultados obtenidos por el modelado probabilista basado en normalización se corresponden a la fila indicada con Prob<sub>1</sub> y los basados en máxima

especificidad y razón insuficiente se corresponden a la fila indicada con Prob<sub>2</sub>. En cada columna aparece una asignación de prioridades de manera que el primer número es la tarea más prioritaria y el último es la menos prioritaria. Por ejemplo 3,1,2 indica que la tarea 3 es la más prioritaria y la tarea 2 la menos prioritaria.

Se puede ver en este ejemplo, ver Tabla 5-1, que las asignaciones que maximizan la posibilidad y necesidad también maximizan la probabilidad. Para una precisión  $\epsilon=0.1$ , se observa que hay gran diferencia en el número de llamadas al algoritmo básico por ambos caminos.

Ejemplo 2:

En este segundo ejemplo, los periodos son  $t_1 = 4$ ;  $t_2 = 6$ ;  $t_3 = 16$ . Los tiempos de ejecución son números borrosos triangulares definidos por sus vértices:  $P_1=(0.5, 0.7, 0.9)$ ,  $P_2=(0.8, 0.9, 1)$  y  $P_3=(0.7, 0.8, 1.2)$ . Los plazos son también números borrosos triangulares:  $D_1=(1.5, 2, 4)$ ,  $D_2=(2.5, 2.75, 3)$  y  $D_3=(2, 3, 3.5)$ . Se puede ver en este ejemplo, ver Tabla 5-2, que las asignaciones que maximizan la posibilidad y necesidad también maximizan la probabilidad. Para una precisión de 0.02, se observa que hay gran diferencia en el número de llamadas al algoritmo básico por ambos caminos.

$\epsilon=0.02$	Nº llamadas	Asignación de prioridades					
		3,2,1	2,3,1	3,1,2	1,3,2	2,1,3	1,2,3
$\pi$	112	$\Pi=0.83$ $N=0$	$\Pi=0.83$ $N=0$	$\Pi=1$ $N=0.37$	$\Pi=1$ $N=0.37$	$\Pi=1$ $N=0.34$	$\Pi=1$ $N=0.34$
Prob <sub>1</sub>	15000	0.47	0.45	<b>0.90</b>	<b>0.93</b>	0.80	0.82
Prob <sub>2</sub>	15000	0.32	0.33	<b>0.96</b>	<b>0.95</b>	0.88	0.91

Tabla 5-2: Ejemplo 2. Comparación Posibilidad-Probabilidad

Además, se ha observado que para un número de llamadas comparable al de la posibilidad, esto es, con 126 llamadas al algoritmo básico, con el modelo de probabilidad no es posible distinguir correctamente la mejor asignación de entre las cuatro últimas.

## Ejemplo 3:

En este ejemplo la solución obtenida por teoría de la posibilidad es única, y coincide con la de probabilidad, con las dos transformaciones analizadas. Es la correspondiente a la asignación de prioridades 1,3,2 como se puede ver en la Tabla 5-3.

Los periodos son  $t_1 = 4$ ,  $t_2 = 6$  y  $t_3 = 16$ . Los tiempos de ejecución son números borrosos triangulares:  $P_1=(0.5, 0.7, 0.9)$ ,  $P_2=(0.8, 0.9, 1)$  y  $P_3=(0.7, 0.8, 1.2)$ . Los plazos son también números borrosos triangulares:  $D_1=(1.2, 2, 4)$ ,  $D_2=(2.5, 2.75, 3)$  y  $D_3=(2, 3, 3.5)$ . Se puede ver en la Tabla 5-3, que la asignación que maximiza la posibilidad y necesidad también maximiza la probabilidad. Para una precisión de 0.02, se observa que hay gran diferencia en el número de llamadas al algoritmo básico por ambos caminos.

$\epsilon=0.02$	Nº llamadas	Asignación de prioridades					
		3,2,1	2,3,1	3,1,2	1,3,2	2,1,3	1,2,3
$\pi$	115	$\Pi=0.83$ $N=0$	$\Pi=0.83$ $N=0$	$\Pi=1$ $N=0.34$	<b><math>\Pi=1</math></b> <b><math>N=0.37</math></b>	$\Pi=1$ $N=0.34$	$\Pi=1$ $N=0.34$
Prob <sub>1</sub>	12000	0.41	0.4	0.83	<b>0.91</b>	0.74	0.8
Prob <sub>2</sub>	12000	0.31	0.28	0.92	<b>0.97</b>	0.89	0.9

Tabla 5-3: Ejemplo 3. Comparación Posibilidad-Probabilidad

## Ejemplo 4:

En este ejemplo las soluciones obtenidas por teoría de la posibilidad 2,1,3 y 1,2,3, no coinciden con la de probabilidad 1,3,2 (prácticamente igual que 3,1,2), con las dos transformaciones analizadas. Ver la Tabla 5-4.

Los periodos son  $t_1 = 4$ ,  $t_2 = 6$  y  $t_3 = 16$ . Los tiempos de ejecución son números borrosos triangulares:  $P_1=(0.375, 0.4, 0.5)$ ,  $P_2=(0.875, 0.9, 1)$  y  $P_3=(0.85, 0.875, 1)$ . Los plazos son también números borrosos triangulares:  $D_1=(1.75, 2, 2.25)$ ,  $D_2=(2, 2.5, 3.5)$  y  $D_3=(1.5, 3, 3.75)$ . Se puede ver en la Tabla 5-4, que en este ejemplo no se ha exigido la misma precisión  $\epsilon$  en el modelo posibilista y en el probabilista. De

hecho, en probabilidad, para un número mucho mayor de llamadas al algoritmo básico (3600 frente a 111), la precisión conseguida es bastante menor (0.08 aproximadamente frente a 0.01).

	Nº llamadas	Asignación de prioridades					
		3,2,1	2,3,1	3,1,2	1,3,2	2,1,3	1,2,3
$\varepsilon=0.01$ $\pi$	111	$\Pi=0.437$ $N=0$	$\Pi=0.437$ $N=0$	$\Pi=1$ $N=0.39$	$\Pi=1$ $N=0.39$	$\Pi=1$ $N=0.453$	$\Pi=1$ $N=0.453$
$\varepsilon=0.09$ Prob <sub>1</sub>	3600	0	0	<b>0.91</b>	<b>0.92</b>	0.82	0.82
$\varepsilon=0.08$ Prob <sub>2</sub>	3600	0	0	<b>0.95</b>	<b>0.96</b>	0.91	0.9

Tabla 5-4: Ejemplo 4. Comparación Posibilidad-Probabilidad

Por el camino de la posibilidad, se optaría por dar a la tarea 3 la prioridad más baja, mientras que por el camino de la probabilidad se daría la prioridad más baja a la tarea 2. Se puede ver que aunque la probabilidad de estas asignaciones (3,1,2 y 1,3,2) es mayor que la de las asignaciones elegidas por la vía de la posibilidad (2,1,3 y 1,2,3), sus valores de probabilidad están cercanos, sobre todo en comparación con las otras asignaciones, cuya probabilidad es 0. Se observa que también los valores de posibilidad y necesidad están cercanos y claramente más alejados de las otras asignaciones.

Es interesante destacar, que en todos los ejemplos que se fueron analizando las asignaciones óptimas proporcionadas por el camino de la posibilidad y de la probabilidad eran las mismas y que fue preciso diseñar muy cuidadosamente el ejemplo 4 para lograr un caso en el que no coincidieran.

Ejemplo 5:

Veamos un ejemplo en el que en la ejecución de una tarea interfieren varias ejecuciones de otras tareas. Se puede considerar que el tiempo de ejecución aleatorio obtenido de la tarea más prioritaria es el mismo en las distintas ejecuciones que interfieren en la menos prioritaria, o bien se pueden generar diferentes valores del

tiempo de ejecución, independientes para cada ejecución. En la Tabla 5-5, se ha añadido el caso Prob<sub>2</sub> con la modificación de generar números aleatorios independientes para las distintas ejecuciones. Este caso se denomina P<sub>2-ind</sub>

Los periodos son  $t_1 = 3$ ,  $t_2 = 5$  y  $t_3 = 15$ . Los tiempos de ejecución son números borrosos triangulares:  $P_1=(0.9, 1, 1.1)$ ,  $P_2=(1.9, 2, 2.1)$  y  $P_3=(1.9, 2, 2.1)$ . Los plazos son también números borrosos triangulares:  $D_1=(1.9, 2, 2.1)$ ,  $D_2=(3.5, 4, 4.5)$  y  $D_3=(13.4, 13.7, 14.7)$

$\epsilon=0.1$	Nº llamadas	Asignación de prioridades					
		3,2,1	2,3,1	3,1,2	1,3,2	2,1,3	1,2,3
$\pi$	91	$\Pi = 0$ $N = 0$	$\Pi = 0$ $N = 0$	$\Pi = 0$ $N = 0$	$\Pi = 0$ $N = 0$	$\Pi = 0$ $N_{pl} = 0$	$\Pi = 1$ $N = 0.125$
Prob <sub>1</sub>	345	0	0	0	0	0	<b>0.66</b>
Prob <sub>2</sub>	345	0	0	0	0	0	<b>0.66</b>
Prob <sub>2-ind</sub>	645	0	0	0	0	0	<b>0.7</b>

Tabla 5-5: Ejemplo 5. Comparación Posibilidad-Probabilidad

## 5.6. Extensión del análisis de planificabilidad con plazos mayores que los periodos

Los algoritmos de análisis de planificabilidad presentados en los apartados anteriores sólo son válidos cuando los plazos son menores o iguales que los periodos. Para realizar el análisis de planificabilidad con parámetros numéricos cuando los plazos pueden ser menores, iguales o mayores que los periodos, se puede aplicar (5.21).

Este algoritmo fue desarrollado por Lehoczky ([Lehoczky, 1990]) y fue obtenido también, de forma independiente, por Tindell ([Tindell, 1992], [Tindell et al., 1994]). Es aplicable tanto a tareas periódicas como esporádicas, es decir tareas no periódicas con tiempos entre llegadas acotados.

Al realizar el análisis de planificabilidad del sistema, se puede comprobar en primer lugar el límite de utilización, que debe estar por debajo del 100%. Si no fuera así, los tiempos de respuesta crecerían indefinidamente y el sistema no sería planificable.

Si se cumple dicha condición necesaria, se calculan los peores tiempos de respuesta de las tareas y se comparan con los plazos correspondientes. El cálculo del peor tiempo de cada tarea se puede hacer mediante la aplicación del algoritmo (5.21), en el que las tareas están ordenadas de acuerdo a su prioridad, siendo la tarea 1 la más prioritaria.

En (5.21), para cada tarea se inicializa  $s$  y el contador  $n$ . Para cada tarea  $i$ , el contador  $n$  varía de 1 hasta el número de ejecuciones de la tarea  $i$  que hay dentro de un periodo de ocupación.

El periodo de ocupación se prolonga mientras los tiempos de respuesta  $e_{i,n}$  son mayores que el periodo  $t_i$ .

Cuando  $s_{i,k}=s_{i,k-1}$  el tiempo de finalización de la tarea  $i$  en su ejecución  $n$  es  $s_{i,k}$ . Si el tiempo de respuesta correspondiente a esta ejecución es menor o igual que el periodo  $t_i$ , el periodo de ocupación ha terminado y se puede obtener el peor tiempo de la tarea  $i$ , como el máximo de todos sus tiempos de respuesta.

$$\begin{array}{l}
 \text{for } i=1 \text{ to } n\_tareas \text{ do} \\
 s_{i,0} = \sum_{j=1}^i p_j, \quad n=1 \\
 \text{loop} \\
 \quad \text{while } (s_{i,k} \neq s_{i,k-1}) \{ \text{if } (s_{i,k} > t_i(n-1) + d_i) \text{ tarea } i \text{ no planificable} \\
 \quad \quad \quad k = k + 1 \} \\
 \quad e_{i,n} = s_{i,k} - t_i(n-1) \\
 \quad \text{if } (e_{i,n} > t_i) \{ n = n + 1; \quad s_{i,k} = s_{i,k} + p_i \} \\
 \quad \text{else } \{ wc_i = \max(e_{i,x}) \quad \forall x \in \mathbb{N}, 1 \leq x \leq n; \text{ Siguiente tarea} \} \\
 \text{endloop} \\
 \text{end} \\
 (5.21) \quad s_{i,k}(s_{i,k-1}) = n \cdot p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil
 \end{array}$$

*Ejemplo A:*

En este ejemplo el peor tiempo de respuesta se produce en la segunda ejecución de una tarea.

Sean dos tareas con periodos  $t_1=9$  y  $t_2=15$ , los tiempos de ejecución son  $p_1=2$  y  $p_2=11.5$  y los plazos son  $d_1=7$  y  $d_2=15.1$

Aplicando el algoritmo clásico (5.21) tal y como se detalla a continuación, se tiene que  $w_{c_{1,1}}=2$ ,  $w_{c_{2,1}}=15.5$ ,  $w_{c_{2,2}}=31$ ,  $w_{c_{2,3}}=44.5$ .

Para la tarea 1, inicialmente  $s_{1,0}=p_1=2$  y  $n=1$ . Para  $k=1$ ,  $s_{1,1}=1 \cdot p_1 = 2 = s_{1,0}$ . Por tanto,  $w_{c_{1,1}}=2$ . Como  $w_{c_{1,1}}=2 < t_1$ , se pasa a analizar la siguiente tarea.

Para la tarea 2, inicialmente se tiene que  $s_{2,0}=p_1+p_2=13.5$  y  $n=1$ .

Para  $k=1$ ,  $s_{2,1}=1 \cdot p_2 + p_1 \cdot \lceil s_{2,0} / t_1 \rceil = 11.5 + 2 \cdot \lceil 13.5 / 9 \rceil = 15.5$ . Con  $k=2$ ,  $s_{2,2}=11.5 + 2 \cdot \lceil 15.5 / 9 \rceil = 15.5 = s_{2,1}$ . Por tanto el tiempo de finalización de la tarea 2 en su primera ejecución es  $w_{c_{2,1}}=15.5$ . Como  $w_{c_{2,1}} > t_2$ , hay que calcular el tiempo de finalización de esta tarea en la siguiente ejecución  $n=2$ .

Por tanto, continuando con  $n=2$ ,  $s_{2,2}=15.5+11.5=27$ . Para  $k=3$ ,  $s_{2,3}=2 \cdot 11.5 + 2 \cdot \lceil 27 / 9 \rceil = 23 + 6 = 29$ . Para  $k=4$ ,  $s_{2,4}=2 \cdot 11.5 + 2 \cdot \lceil 29 / 9 \rceil = 23 + 8 = 31$ . Para  $k=5$ ,  $s_{2,5}=2 \cdot 11.5 + 2 \cdot \lceil 31 / 9 \rceil = 23 + 8 = 31 = s_{2,4}$ . Por tanto el tiempo de finalización de la tarea 2 en su segunda ejecución es  $w_{c_{2,2}}=31$ . El periodo de ocupación no ha terminado porque  $w_{c_{2,2}} > 2 \cdot t_2$ , por tanto hay que calcular el tiempo de finalización de esta tarea en su tercera ejecución.

Así pues, para  $n=3$ ,  $s_{2,5}=31+11.5=42.5$ . Con  $k=6$ ,  $s_{2,6}=3 \cdot 11.5 + 2 \cdot \lceil 42.5 / 9 \rceil = 44.5$ . Con  $k=7$ ,  $s_{2,7}=3 \cdot 11.5 + 2 \cdot \lceil 44.5 / 9 \rceil = 44.5 = s_{2,6}$ . Por tanto el tiempo de finalización de la tarea 2 en su tercera ejecución es  $w_{c_{2,3}}=44.5$ . Como se cumple que  $w_{c_{2,3}} < 3 \cdot t_2$ , el periodo de ocupación ha terminado y no es necesario calcular siguientes ejecuciones de la tarea 2.

Por tanto, el peor tiempo de respuesta de la tarea 2 se produce en su segunda ejecución y vale 16. Como el peor tiempo de respuesta no respeta su plazo  $d_2$  la tarea y el sistema son no planificables.

Si por el contrario el tiempo de ejecución de la tarea 2 fuera menor que en el caso anterior y de valor  $p_2=10.6$ , el tiempo de respuesta de esta tarea también sería menor, de forma que en la primera ejecución se tiene  $w_{c_{2,1}}=14.6$ . Como es menor que el periodo, este sería el peor tiempo de respuesta y no haría falta seguir calculando otras ejecuciones de dicha tarea.

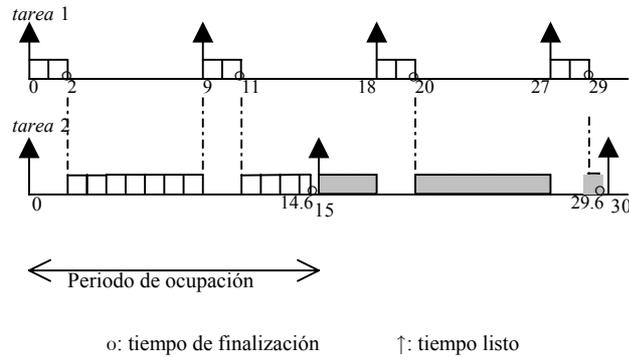


Figura 5.29: Ejemplo A: tiempo de finalización menor que el periodo

De hecho, si se aplicara el algoritmo para la segunda ejecución, el resultado obtenido de tiempo de finalización  $wc_{2,2}=29.2$  sería incorrecto ya que el periodo de ocupación habría finalizado antes. El resultado correcto es  $wc_{2,2}=t_2+p_2+2\cdot p_1 = 29.6$ , 0.4 unidades mayor que el resultado anterior, que es precisamente el tiempo de espera desde que finaliza la primera ejecución en 14.6 hasta que se produce el siguiente tiempo listo de la tarea 2 en el instante 15 (ver Figura 5.29).

Para la posterior extensión del algoritmo a parámetros borrosos, se detalla en lo que sigue el resultado del algoritmo en el caso de que se ignore la comprobación de final de periodo de ocupación, es decir, si a pesar de que un tiempo de finalización esté antes del siguiente tiempo listo, se sigue calculando la siguiente ejecución de la tarea.

En efecto, para una tarea  $i$  cualquiera, si  $wc_{i,1} < t_i$ , los demás tiempos de finalización de esa tarea  $wc_{i,k}$  también son menores que el siguiente tiempo listo  $wc_{i,k} < k \cdot t_i$ , y además, como el peor tiempo de respuesta se da en la primera ejecución, se cumple que  $t_i - wc_{i,1} < k \cdot t_i - wc_{i,k}$  y también,  $d_i - wc_{i,1} < d_{i,k} - wc_{i,k} \quad \forall k$ . Como en este caso el resultado  $wc_{i,k}$  obtenido al aplicar el algoritmo ignorando la comprobación de fin de periodo de ocupación es incorrecto, se tiene que  $wc_{i,k} \neq wc_{i,k}^e$ . Al eliminar dicha comprobación, el algoritmo considera que el periodo de ocupación no ha terminado, supone que no existen huecos o tiempos de espera en la ejecución y por tanto se tiene que  $wc_{i,k} \leq wc_{i,k}^e$  con lo que  $d_i - wc_{i,1} < d_{i,k} - wc_{i,k} \leq d_{i,k} - wc_{i,k}$ . Es decir, aunque se calculen más ejecuciones de la tarea, el peor caso se sigue dando en la primera ejecución, y el resultado de peor tiempo de respuesta proporcionado por el algoritmo es correcto.

Cuando en la primera ejecución hay sobrepaso del periodo, empieza un periodo de ocupación. Entonces,  $\forall k$   $wc_{i,k}$  es el resultado correcto si los tiempos de finalización anteriores  $\forall j=1..k-1$  sobrepasan el siguiente tiempo listo, es decir,  $wc_{i,j} \geq j \cdot t_i$ .

Si existe una ejecución  $j$  anterior a  $k$  ( $j < k$ ) que no sobrepasa el periodo  $wc_{i,j} < j \cdot t_i$ , entonces ese tiempo de finalización  $wc_{i,k}$  no es correcto y puede sobrepasar o no el tiempo listo correspondiente. En el cálculo exacto se cumple que el peor tiempo de respuesta de una tarea se produce dentro del periodo de ocupación, en la ejecución  $p$  tal que  $p \leq j$ . Por tanto, como la ejecución  $k$  está fuera del periodo de ocupación se tiene que  $wc_{i,k} - k \cdot t_i \leq wc_{i,p} - p \cdot t_i$  y también,  $wc_{i,k} - d_{i,k} \leq wc_{i,p} - d_{i,p}$ . El resultado incorrecto  $wc_{i,k}$  obtenido mediante el algoritmo cumple que  $wc_{i,k} \leq wc_{i,k}^e$ . Con lo que  $wc_{i,k} - d_{i,k} \leq wc_{i,k}^e - d_{i,k} \leq wc_{i,p} - d_{i,p}$ . Es decir, el peor caso se sigue dando en  $p$ , con lo que el peor tiempo de respuesta que proporciona el algoritmo es exacto.

El algoritmo (5.21) se puede ampliar para tener en cuenta la existencia de tiempos de bloqueo en las tareas, causadas por el acceso de una tarea menos prioritaria a un recurso compartido. El tiempo de bloqueo de una tarea  $i$  se denomina  $b_i$ .

La diferencia con el algoritmo (5.21) está en la inicialización de  $s$  (5.22) y en la ecuación principal de cálculo iterativo (5.23). En este algoritmo, el tiempo de bloqueo se trata como un tiempo extra de ejecución al comienzo del periodo de ocupación, aumentando los tiempos de respuesta de las iteraciones en dicho periodo.

$$(5.22) \quad s_{i,0} = b_i + \sum_{j=1}^i p_j$$

$$(5.23) \quad s_{i,k}(s_{i,k-1}) = b_i + n \cdot p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil$$

En [Tindell et al., 1994], [Burns, Wellings, 1996] además se extiende el algoritmo al caso de que exista "release jitter":

$$(5.24) \quad s_{i,k}(s_{i,k-1}) = b_i + n \cdot p_i + \sum_{j=1}^{i-1} p_j \cdot \left\lceil \frac{s_{i,k-1} + j_j}{t_j} \right\rceil$$

$$e_{i,n} = s_{i,k} - t_i(n-1) + j_i$$

### Ejemplo B:

El sistema de este ejemplo consta ahora de tres tareas, con periodos  $t_1=10$ ,  $t_2=15$  y  $t_3=35$ , tiempos de ejecución  $p_1=4$ ,  $p_2=6$  y  $p_3=6$ , tiempos de bloqueo  $b_1=0$ ,  $b_2=2$  y

$b_3=0$ , y plazos  $d_1=10$ ,  $d_2=16$ ,  $d_3=35$ . La tarea 1 es la más prioritaria y la 3 la menos prioritaria. Como se puede ver, la tarea 2 es la única con tiempo de bloqueo y que además tiene su plazo mayor que el periodo.

En primer lugar la utilización total del procesador es  $4/10 + 6/15 + 6/35 = 0.97$ , que está por debajo del 100%, por lo que se debe seguir con el análisis de planificabilidad.

En la Figura 5.30 se representan las ejecuciones de las tareas 1 y 2 durante el periodo de ocupación. Se dibujan las ejecuciones de la tarea 1 que al ser más prioritaria expulsa a la tarea 2. Además, se tiene en cuenta el bloqueo que sufre la tarea 2 debido a tareas menos prioritarias añadiendo al comienzo de su ejecución el tiempo de bloqueo.

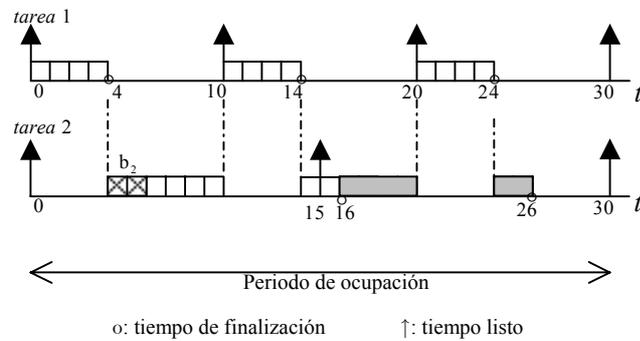


Figura 5.30: Ejemplo B: Tarea con bloqueo y plazo mayor que el periodo

El tiempo de respuesta de la tarea 2 en su primera ejecución es 16 y en la segunda es  $26-15=11$ . Por tanto el peor tiempo de respuesta de la tarea 2 es 16, que es menor que el plazo.

Veamos ahora el cálculo del peor tiempo de la tarea 2 mediante la aplicación del algoritmo (5.23). Inicialmente  $s_{i,0} = b_i + \sum_{j=1}^i p_j$ , con lo que  $s_{2,0} = 2+4+6 = 12$

Calculamos en primer lugar el tiempo de respuesta de la tarea 2 en su primera ejecución  $n=1$ .

Para  $k=1$ , aplicando (5.23) se tiene  $s_{2,1} = 2+1 \cdot (6)+4 \cdot \lceil 12/10 \rceil = 16$

Como  $s_{2,1} \neq s_{2,0}$  se vuelve a calcular para  $k=2$ ,  $s_{2,2} = 2+1 \cdot (6)+4 \cdot \lceil 16/10 \rceil = 16$

Como  $s_{2,2} = s_{2,1}$  este es el tiempo de finalización de la tarea 2 en su ejecución  $n=1$

El tiempo de respuesta de la tarea 2 en su ejecución  $n=1$  es, según la ecuación  $e_{i,n}=s_{i,k}-t_i \cdot (n-1)$ ,  $e_{2,1}=s_{2,2}-t_2 \cdot (1-1)=16-0=16$

El periodo de ocupación no ha terminado, ya que el tiempo de respuesta  $e_{2,1}$  es mayor que el periodo:  $e_{2,1} > 15$ . Por tanto se calcula el tiempo de respuesta en su segunda ejecución  $n=2$ ,  $s_{2,2}=s_{2,2} + p_2 = 16+6=22$

Para  $k=3$ ,  $s_{2,3} = 2+2 \cdot (6)+4 \cdot \lceil 22/10 \rceil = 26$ . Como  $s_{2,3} \neq s_{2,2}$  se continua con  $k=4$ ,  $s_{2,4} = 2+2 \cdot (6)+4 \cdot \lceil 22/10 \rceil = 26$

Como  $s_{2,4} = s_{2,3}$  este es el tiempo de finalización de la tarea 2 en su ejecución  $n=2$ . El tiempo de respuesta de la tarea 2 en su ejecución  $n=2$  es, según la ecuación  $e_{i,n}=s_{i,k} - t_i \cdot (n-1)$ ,  $e_{2,2}=s_{2,4}-t_2 \cdot (2-1)=26-15=11$ . El periodo de ocupación ha terminado, ya que el tiempo de respuesta  $e_{2,2}$  es menor que el periodo:  $e_{2,2} < 15$ .

Hay dos ejecuciones en el periodo de ocupación. El máximo de los dos tiempos de respuesta es  $e_{2,1} = 16$  y éste es el peor tiempo de la tarea 2  $wc_2$ . Como  $wc_2 \leq d_2$ , esta tarea es planificable.

### 5.6.1. Extensión del algoritmo a tiempos de ejecución y plazos borrosos.

Si el plazo y el periodo son numéricos, se puede distinguir claramente qué algoritmo hay que extender para modelar el sistema con tiempos de ejecución borrosos: o el algoritmo para plazos menores que los periodos o el que admite plazos mayores. Pero si, por ejemplo, el plazo es también borroso, entonces es posible que para ciertos valores del soporte sea menor que el periodo y para otros sea mayor (ver Figura 5.31). El objetivo que se plantea en este apartado es extender el algoritmo (5.21) de forma que se pueda obtener la posibilidad y necesidad de planificabilidad también cuando no se verifica que los plazos son menores o iguales que los periodos.



Figura 5.31: Ejemplo en el que no existe orden fuerte entre el plazo y el periodo

La extensión de este algoritmo a argumentos borrosos es muy similar a la planteada en el apartado 5.2 para el algoritmo de cálculo de los peores tiempos cuando los plazos son menores o iguales que las tareas.

**Algoritmo basado en aritmética borrosa**

El algoritmo propuesto basado en aritmética borrosa es el que se muestra en (5.25).

*Cálculo de  $wc^c_{i,n}$*

$wc^c_{i,n}$  es el instante del punto de corte entre  $WC_{i,n}$  y  $D_{i,n}=(n-1) \cdot t_i + D_i$ . Se entiende por punto de corte el punto necesario para calcular los valores de posibilidad  $\Pi(WC_{i,n} \leq D_{i,n})$  y necesidad  $N(WC_{i,n} \leq D_{i,n})$ , como se indica en la Figura 5.32.

Si el tiempo de finalización borroso se calcula mediante aritmética borrosa (5.25), este tiempo puede ser un conjunto borroso no convexo. En tal caso, el punto de corte sería el de su aproximación convexa con el plazo, tal y como se muestra en la Figura 5.33. Si no hay punto de corte y  $WC_{i,n} >_s D_{i,n}$  entonces  $wc^c_{i,n} = 0$ , y se termina porque la posibilidad y necesidad de que el tiempo de finalización de la tarea  $i$  en su ejecución  $n$  esté antes del plazo son 0), y la tarea y el sistema no son planificables, por lo que no es necesario seguir el análisis.

```

for  $i=1$  to  $n\_tarefas$  do
 $S_{i,0} = \sum_{j=1}^i P_j$ ,  $n = 1$ 
loop
while ( $S_{i,k} \neq S_{i,k-1}$ )  $k = k + 1$ 
 $WC_{i,n} = S_{i,k}$ 
if ( $wc^c_{i,n} > n \cdot t_i$ )
 $n = n + 1$ ;  $S_{i,k} = S_{i,k} + P_i$ 
else
 $\Pi_i / N_i = \min(\Pi / N(WC_{i,x} \leq D_{i,x})) \forall x \in \mathbb{N} | 1 \leq x \leq n$ 
Siguiente tarea
end
endloop
end

```

(5.25)  $S_{i,k}(S_{i,k-1}) = n \cdot P_i + \sum_{j=1}^{i-1} P_j \cdot \left\lceil \frac{S_{i,k-1}}{t_j} \right\rceil$

$\Pi_{planif} / N_{planif} = \min(\Pi_i / N_i) \forall i \in \mathbb{N} | 1 \leq i \leq n\_tarefas$

Si no hay punto de corte y  $WC_{i,n} <_s D_{i,n}$  entonces:

Si  $\Pi(WC_{i,n} > n \cdot t_i) \neq 0$ ,  $wc^c_{i,n} = n \cdot t_i + \epsilon$  y se continúa calculando la siguiente ejecución de  $i$ .

Si  $\Pi(WC_{i,n} > n \cdot t_i) = 0$ ,  $wc_{i,n}^c = 0$  (y se termina con esa tarea).

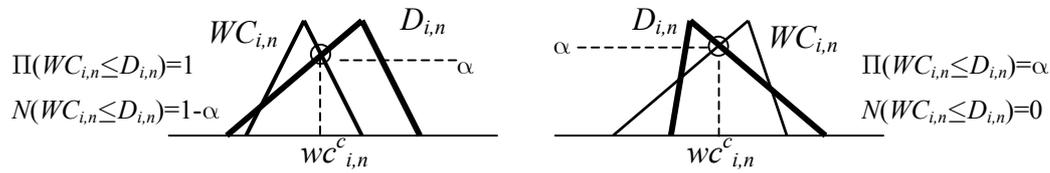


Figura 5.32: Punto de corte  $wc_{i,n}^c$  que se obtiene de la comparación entre  $WC_{i,n}$  y  $D_{i,n}$ .

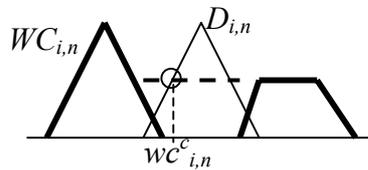


Figura 5.33: Punto de corte cuando el tiempo de finalización es un conjunto borroso no convexo

### Descripción del algoritmo basado en aritmética borrosa:

Si el tiempo de finalización de la tarea  $i$  en su ejecución  $n$  es fuertemente menor que el siguiente instante listo, el periodo de ocupación ha terminado y no hace falta analizar la siguiente ejecución  $n+1$ . El peor caso en el valor de planificabilidad borrosa hay que buscarlo dentro del periodo de ocupación. Concretamente, si  $n=1$ , entonces el peor caso se da en la primera ejecución.

Se pueden distinguir dos casos básicos: que el punto de corte se encuentre antes del periodo o después del periodo.

- 1) El punto de corte entre el tiempo de finalización y el plazo es menor que el siguiente tiempo listo:  $wc_{i,n}^c < n \cdot t_i$ , o lo que es lo mismo, el punto de corte entre el tiempo de respuesta y el plazo está antes que el periodo  $e_{i,n}^c < t_i$ .

En ese caso el peor valor de planificabilidad borrosa de la tarea se da en una ejecución  $p \leq n$ . Por tanto no es necesario analizar iteraciones posteriores y se continua con la siguiente tarea.

*Prueba:*

- a)  $\Pi(E_{i,n} \leq D_i) \neq 0$  y  $N(E_{i,n} \leq D_i) = 0$

De acuerdo con la Figura 5.34, se define  $\alpha_n$  como:

$$(5.26) \alpha_n = \max \alpha \mid \underline{e}_{i,n}(\alpha) \leq \overline{d}_i(\alpha)$$

de forma que  $\Pi(E_{i,n} \leq D_i) = \alpha_n$

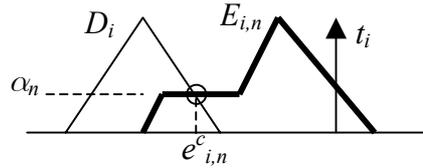


Figura 5.34: Punto de corte antes del periodo y  $N(E_{i,n} \leq D_i) = 0$

Como todos los puntos  $\underline{e}_{i,n}(\alpha) \mid \alpha \leq \alpha_n$  están antes de  $t_i$ , para dichos valores de tiempo de respuesta el periodo de ocupación ha terminado y, por tanto, el peor tiempo de respuesta de la tarea  $i$  se da en una ejecución  $p$  dentro del periodo de ocupación. Es decir:

$$(5.27) \forall \alpha \leq \alpha_n, \exists p \leq n \mid \underline{e}_{i,p}(\alpha) \geq \underline{e}_{i,n}(\alpha)$$

$$p \text{ es tal que } \underline{e}_{i,p}(\alpha) = \max \underline{e}_{i,j}(\alpha) \quad \forall j \leq n$$

En la ejecución  $p$  se define  $\alpha_p$ :

$$(5.28) \alpha_p = \max \alpha \mid \underline{e}_{i,p}(\alpha) \leq \overline{d}_i(\alpha)$$

Teniendo en cuenta (5.26), (5.27) y (5.28), se cumple que  $\alpha_p \leq \alpha_n$ . Y por tanto  $\Pi(E_{i,p} \leq D_i) \leq \Pi(E_{i,n} \leq D_i)$

Falta probar que para ejecuciones posteriores el valor de planificabilidad borrosa es mayor que en  $p$ :

$\forall \alpha \leq \alpha_n$ , el periodo de ocupación ha terminado. Como  $\alpha_p \leq \alpha_n$ , se cumple que  $\underline{e}_{i,n+j}(\alpha) \leq \underline{e}_{i,p}(\alpha)$

Si se define  $\alpha_{n+j} = \max \leq \alpha \mid \overline{e_{i,n+j}}(\alpha) \leq \overline{d_i}(\alpha)$ , teniendo en cuenta lo anterior, se cumple que  $\alpha_{n+j} \geq \alpha_p$ , y por tanto  $\Pi(E_{i,n+j} \leq D_i) \geq \Pi(E_{i,p} \leq D_i)$ . Esto significa que el peor caso se da en una ejecución menor o igual a  $n$ , con lo que no es necesario calcular siguientes ejecuciones de la tarea  $i$ .

b)  $\Pi(E_{i,n} \leq D_i) = 1$  y  $N(E_{i,n} \leq D_i) \neq 0$

De acuerdo con la Figura 5.35, ahora se define  $\alpha_n$  como  $\alpha_n = \min \alpha \mid \overline{e_{i,n}}(\alpha) \leq \overline{d_i}(\alpha)$ , de forma que  $N(E_{i,n} \leq D_i) = 1 - \alpha_n$

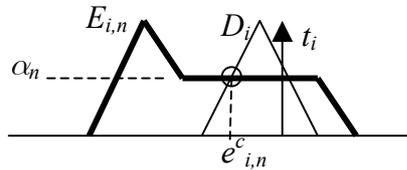


Figura 5.35: Punto de corte antes del periodo y  $N(E_{i,n} \leq D_i) \neq 0$

Como todos los puntos  $\overline{e_{i,n}}(\alpha) \mid \alpha \geq \alpha_n$  están antes de  $t_i$ , para dichos valores el periodo de ocupación ha terminado y por tanto, el peor tiempo de respuesta de la tarea  $i$  se da en una ejecución  $p$  dentro del periodo de ocupación. Es decir,

$$\forall \alpha \geq \alpha_n, \exists p \leq n \mid \overline{e_{i,p}}(\alpha) \geq \overline{e_{i,n}}(\alpha) \text{ y } p \text{ es tal que } \overline{e_{i,p}}(\alpha) = \max \overline{e_{i,j}}(\alpha) \quad \forall j \leq n$$

En la ejecución  $p$  se define ahora  $\alpha_p$  como:

$$\alpha_p = \min \alpha \mid \overline{e_{i,p}}(\alpha) \leq \overline{d_i}(\alpha)$$

Teniendo en cuenta lo anterior, se cumple que  $\alpha_p \geq \alpha_n$ , y por tanto

$$N(E_{i,p} \leq D_i) = 1 - \alpha_p \leq N(E_{i,n} \leq D_i) = 1 - \alpha_n$$

Falta probar que para ejecuciones posteriores el valor de planificabilidad borrosa es mayor que en  $p$ :

$\forall \alpha \geq \alpha_n$ , el periodo de ocupación ha terminado. Como  $\alpha_p \geq \alpha_n$ , se cumple que  $\overline{e_{i,n+j}}(\alpha) \leq \overline{e_{i,p}}(\alpha)$

Se define  $\alpha_{n+j} = \min \alpha \mid \overline{e_{i,n+j}}(\alpha) \leq \underline{d_i}(\alpha)$ . Teniendo en cuenta lo anterior se cumple que  $\alpha_{n+j} \leq \alpha_p$ , con lo que  $N(E_{i,n+j} \leq D_i) = 1 - \alpha_{n+j} \geq N(E_{i,p} \leq D_i) = 1 - \alpha_p$

Es decir, el peor caso se da en una ejecución menor o igual a  $n$ , con lo que no es necesario calcular siguientes ejecuciones de la tarea  $i$ .  $\square$

- 2) Veamos ahora el caso en el que el punto de corte entre el tiempo de finalización y el plazo está a la derecha del siguiente tiempo listo,  $wc_{i,n}^c > n \cdot t_i$  o, lo que es lo mismo, el punto de corte entre el tiempo de respuesta y el plazo está después que el periodo  $e_{i,n}^c > t_i$ . En tal caso, el periodo de ocupación no ha terminado y hay que seguir calculando la ejecución  $n+1$ , por lo que el resultado que se obtiene de planificabilidad borrosa de la tarea es exacto incluso si el punto de corte  $e_{i,n}^c$  se da en una zona incorrecta de  $E_{i,n}$ .

*Prueba:*

Tal y como ya se describió en el algoritmo clásico si, una vez terminado el periodo de ocupación, en lugar de finalizar se sigue calculando la siguiente ejecución de la misma tarea, el resultado de dicho tiempo de respuesta es incorrecto, y concretamente es menor que el tiempo de respuesta exacto. Partiendo de dicha afirmación, supongamos que el tiempo de respuesta de la ejecución anterior  $n-1$  tiene parte del soporte menor que el periodo, tal y como se indica en la Figura 5.36 a). Entonces se cumple que,

$$\forall \alpha \leq \alpha_0, \underline{e_{i,n-1}}(\alpha) \leq t_i \Rightarrow \underline{e_{i,n}}(\alpha) \leq \underline{e_{i,n}^{pe}}(\alpha)$$

En la Figura 5.36 b) se muestra en trazo discontinuo la zona del tiempo de respuesta correspondiente a  $\underline{e_{i,n}^{pe}}(\alpha)$ .

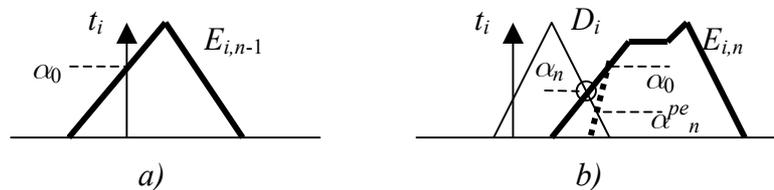


Figura 5.36: Tiempos de respuesta de la tarea  $i$  en a) ejecución  $n-1$ , b) ejecución  $n$

Se va a probar que, incluso si el punto de corte entre el plazo y el tiempo de respuesta está en la zona incorrecta, el valor de planificabilidad de la tarea es exacto.

Es decir, si  $\alpha_n = \max \alpha \mid \bar{d}_i(\alpha) \geq \underline{e}_{i,n}(\alpha)$ , suponiendo que el punto de corte está en la zona incorrecta, se tiene que  $\alpha_n < \alpha_0$ , como en la Figura 5.36.

Por otro lado, el valor de  $\alpha$  del punto de corte exacto se define como  $\alpha^{pe}_n = \max \alpha \mid \bar{d}_i(\alpha) \geq \underline{e}_{i,n}^{pe}(\alpha)$ . Como  $\alpha_n < \alpha_0$ ,  $\underline{e}_{i,n}(\alpha_n) \leq \underline{e}_{i,n}^{pe}(\alpha_n)$  y se tiene que  $\alpha^{pe}_n \leq \alpha_n$ . O lo que es lo mismo,

$$(5.29) \quad \Pi(E_{i,n}^{pe} \leq D_i) \leq \Pi(E_{i,n} \leq D_i).$$

Para los valores de  $\alpha$  menores que  $\alpha_0$ , el periodo de ocupación ha terminado, y como el peor tiempo de respuesta se da dentro del periodo de ocupación se cumple:

$\exists p \leq n-1 \mid \underline{e}_{i,p}(\alpha) \geq \underline{e}_{i,n}^{pe}(\alpha)$ ,  $\forall \alpha \leq \alpha_0$ , donde  $p$  es tal que  $\underline{e}_{i,p}(\alpha)$  es el máximo tiempo de respuesta.

En la ejecución  $p$  se define  $\alpha_p = \max \alpha \mid \bar{d}_i(\alpha) \geq \underline{e}_{i,p}(\alpha)$  y como  $p$  es el peor caso,  $\alpha_p \leq \alpha^{pe}_n$ . O lo que es lo mismo,  $\Pi(E_{i,p} \leq D_i) \leq \Pi(E_{i,n}^{pe} \leq D_i)$  y aplicando (5.29) se tiene que  $\Pi(E_{i,p} \leq D_i) \leq \Pi(E_{i,n} \leq D_i)$ , es decir el valor de planificabilidad borrosa de la tarea  $i$  sigue siendo  $\Pi(E_{i,p} \leq D_i)$ . Por tanto, al comparar los tiempos de finalización borrosos obtenidos mediante la extensión borrosa con los plazos, la peor ejecución coincidirá con la que se tendría si los tiempos de finalización fueran exactos y los valores de posibilidad y necesidad de que la tarea sea planificable también coinciden.

La demostración en el caso de tener  $\Pi(E_{i,n} \leq D_i) = 1$  y  $N(E_{i,n} \leq D_i) \neq 0$ , no se muestra porque es similar a la ya presentada.  $\square$

### Algoritmo basado en aritmética de intervalos

A continuación se muestra el algoritmo basado en aritmética de intervalos, en el que se busca el punto de corte mediante el método de bipartición. De nuevo, según la

posición de dicho punto de corte, se calculan siguientes ejecuciones de la misma tarea o se pasa a la siguiente tarea.

En primer lugar se presenta el algoritmo general.

```

ALG Aritmética de Intervalos
 $\underline{wc}_{0,1}(0) = 0$ 
for  $i = 1$  to  $n\_tareas$  do
   $\underline{wc}_{i,0}(0) = \underline{wc}_{i-1,1}(0)$  ,  $n = 1$ 
  loop
     $\underline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(0); s\_in_i = \underline{wc}_{i,n-1}(0) + p_i \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
     $\underline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(1); s\_in_i = \underline{wc}_{i,n}(0) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
     $\overline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p_j}(1); s\_in_i = \underline{wc}_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
     $\overline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p_j}(0); s\_in_i = \overline{wc}_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
    ALG ZONA
    if  $wc_{i,n}^c > n \cdot t_i$ 
       $n = n + 1$ 
    else
       $\Pi_i / N_i = \min \{ \Pi / N (WC_{i,x} \leq D_{i,x}) \} \quad \forall x \in \mathbb{N} | 1 \leq x \leq n$ 
      siguiente tarea
    end if
  end loop
end
 $\Pi_{planif} / N_{planif} = \min \{ \Pi_i / N_i \} \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas$ 

```

donde  $D_{i,n} = (n-1) \cdot t_i + D_i$  y el algoritmo básico que se muestra a continuación ya aparecía en el algoritmo de parámetros numéricos (5.21).

```

ALG BASICO:

$$\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_{i,n} = s_{i,k} \\ \text{dado, } s_{i,k} = n \cdot p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil \end{array} \right.$$


```

El algoritmo ZONA establece la zona de búsqueda por bipartición.

$$\begin{aligned}
 & \text{ALG ZONA} \\
 (5.30) \quad & \text{if } (\underline{d}_{i,n}(0) \geq \overline{wc}_{i,n}(0)) \quad \Pi(WC_{i,n} \leq D_{i,n}) = 1, \quad N(WC_{i,n} \leq D_{i,n}) = 1, \\
 & \quad \quad \quad \text{if } (\overline{wc}_{i,n}(0) \geq nt_i) \quad wc_{i,n}^c = nt_i + \delta \\
 & \quad \quad \quad \text{else } wc_{i,n}^c = 0 \\
 & \text{if } (\overline{d}_{i,n}(0) \leq \underline{wc}_{i,n}(0)) \quad \Pi(WC_{i,n} \leq D_{i,n}) = 0, \quad N(WC_{i,n} \leq D_{i,n}) = 0, \quad wc_{i,n}^c = 0 \\
 & \text{if } (\overline{wc}_{i,n}(1) \geq \underline{d}_{i,n}(1)) \quad N(WC_{i,n} \leq D_{i,n}) = 0 \\
 & \quad \quad \text{else if } (\overline{wc}_{i,n}(1) < \underline{d}_{i,n}(1)) \quad \text{Bipartición sup} \\
 & \text{if } (\underline{wc}_{i,n}(1) \leq \overline{d}_{i,n}(1)) \quad \Pi(WC_{i,n} \leq D_{i,n}) = 1 \\
 & \quad \quad \text{else if } (\underline{wc}_{i,n}(1) > \overline{d}_{i,n}(1)) \quad \text{Bipartición inf}
 \end{aligned}$$

Y los algoritmos de búsqueda por bipartición en la zona de límites inferiores o de límites superiores de los  $\alpha$ -cortes de los tiempos de finalización son los siguientes.

$$\begin{aligned}
 & \text{Bipartición inf} \\
 & \alpha_0 = 0, \quad \alpha_1 = 1, \quad \alpha = 0, \quad \text{cota\_inf} = \underline{wc}_{i,n}(0), \quad \text{cota\_sup} = \underline{wc}_{i,n}(1) \\
 & \text{while } \left( \left| \underline{wc}_{i,n}(\alpha) - \overline{d}_{i,n}(\alpha) \right| > \varepsilon \right) \\
 & \quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 & \quad \underline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(\alpha); \quad s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 & \quad \text{if } \left( \underline{wc}_{i,n}(\alpha) < \overline{d}_{i,n}(\alpha) \right) \quad \alpha_0 = \alpha, \quad \text{cota\_inf} = \underline{wc}_{i,n}(\alpha) \\
 & \quad \text{if } \left( \underline{wc}_{i,n}(\alpha) > \overline{d}_{i,n}(\alpha) \right) \quad \alpha_1 = \alpha, \quad \text{cota\_sup} = \underline{wc}_{i,n}(\alpha) \\
 & \text{end} \\
 & \Pi(WC_{i,n} \leq D_{i,n}) = \alpha; \quad N(WC_{i,n} \leq D_{i,n}) = 0; \quad wc_{i,n}^c = \underline{wc}_{i,n}(\alpha)
 \end{aligned}$$

```

Bipartición sup
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$ ,  $cota\_inf = \overline{wc}_{i,n}(1)$ ,  $cota\_sup = \overline{wc}_{i,n}(0)$ 
while  $\left( \left| \overline{wc}_{i,n}(\alpha) - \underline{d}_{i,n}(\alpha) \right| > \varepsilon \right)$ 
     $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ 
     $\overline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(\alpha); s\_in_i = cota\_inf \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
    if  $\left( \overline{wc}_{i,n}(\alpha) < \underline{d}_{i,n}(\alpha) \right)$   $\alpha_1 = \alpha$  ,  $cota\_inf = \overline{wc}_{i,n}(\alpha)$ 
    if  $\left( \overline{wc}_{i,n}(\alpha) > \underline{d}_{i,n}(\alpha) \right)$   $\alpha_0 = \alpha$  ,  $cota\_sup = \overline{wc}_{i,n}(\alpha)$ 
end
 $\Pi(WC_{i,n} \leq D_{i,n}) = 1$  ;  $N(WC_{i,n} \leq D_{i,n}) = 1 - \alpha$  ;  $wc_{i,n}^c = \overline{wc}_{i,n}(\alpha)$ 
    
```

*Extensión del ejemplo A aplicando el algoritmo basado en aritmética borrosa:*

Veamos ahora la extensión a tiempos de ejecución borrosos  $P_1=(2,2,2)$  y  $P_2=(10.6, 11.1, 11.5)$  del ejemplo 2 planteado anteriormente, donde los periodos son  $t_1=9$  y  $t_2=15$ , y los plazos son  $d_1=7$  y  $d_2=15.1$ .

Aplicando el algoritmo extendido se obtiene que para la tarea 2 el tiempo de finalización en su primera ejecución es  $WC_{2,1}=(14.6, 15.1, 15.5)$ . Como el punto de corte entre el plazo y el tiempo de finalización se encuentra después del siguiente tiempo listo, se calcula la siguiente ejecución de la tarea 2. El tiempo de finalización en la segunda ejecución de la tarea 2 es  $WC_{2,2}=(29.2, 30.2, 31)$  (ver Figura 5.37). De nuevo, el punto de corte está después del siguiente tiempo listo por tanto se calcula el tiempo de finalización de la tercera ejecución  $WC_{2,3}=(41.8, 43.3, 44.5)$ . Como este tiempo es menor que el tiempo listo correspondiente 45, el periodo de ocupación ha terminado. El peor caso en el cálculo de planificabilidad borrosa se dará en la primera o en la segunda ejecución.

Los tiempos de finalización  $WC_{2,2}$  y  $WC_{2,3}$  son aproximaciones que no conservan los límites de los  $\alpha$ -cortes. Esto es debido a que, tal y como se vio en el ejemplo con parámetros numéricos, para aquellos valores de  $wc_{2,1}$  menores que el periodo, el cálculo de  $wc_{2,2}$  mediante este algoritmo, que considera que se ha superado el periodo, es incorrecto y para aquellos valores de  $wc_{2,2}$  menores que  $(2 \cdot \text{periodo})$ , el cálculo de  $wc_{2,3}$  es incorrecto. Cuando el tiempo  $wc_{2,1}$  es menor que el periodo, el tiempo  $wc_{2,2}$  también es menor que  $2 \cdot \text{periodo}$ , ya que en ese caso el peor tiempo de respuesta se da en la primera ejecución. Por tanto, la zona incorrecta de  $wc_{2,2}$  está siempre antes del instante listo siguiente. Como el plazo es mayor que el periodo, el punto de corte entre el tiempo de finalización y el plazo se encuentra siempre en la

zona correcta del tiempo de finalización que es la de aquellos valores que superan el periodo, y el resultado de planificabilidad borrosa es exacto.

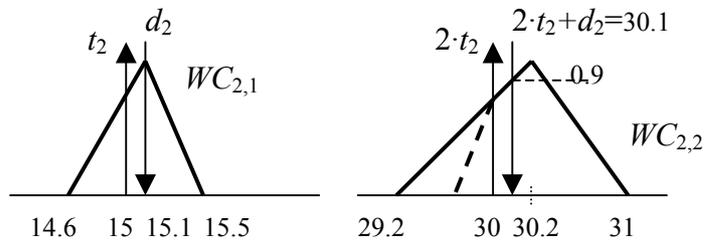


Figura 5.37: Tiempos de finalización de la tarea 2 en su primera y segunda ejecución

En la Figura 5.37 se marca con línea discontinua la forma que tendría el tiempo borroso exacto en la segunda ejecución, cuando en la primera ejecución no se supera el periodo.

Se puede ver que en la primera ejecución  $\Pi(WC_{2,1} \leq d_2) = 1$  y  $N(WC_{2,1} \leq d_2) = 0$ , y en la segunda  $\Pi(WC_{2,2} \leq 2 \cdot d_2) = 0.9$  y  $N(WC_{2,2} \leq 2 \cdot d_2) = 0$ . El peor caso se da en la segunda ejecución, con lo que la planificabilidad borrosa del sistema es  $\Pi_S = 0.9$  y  $N_S = 0$ .

### 5.6.2. Extensión del algoritmo con tiempos de bloqueo y retraso de la activación

El algoritmo de planificación considerando tiempos de bloqueo borrosos y “release jitter” (5.24) es similar a la extensión borrosa propuesta antes, añadiendo los tiempos de bloqueo y “jitter” borrosos. En (5.31) se muestra el algoritmo basado en aritmética borrosa, para tiempos de bloqueo, “release jitter”, tiempos de ejecución y plazos borrosos.

El algoritmo propuesto basado en aritmética borrosa proporciona una aproximación del tiempo de finalización de cada tarea en cada ejecución dentro del periodo de ocupación. En general dicha aproximación no tiene los límites exactos de los  $\alpha$ -cortes. A pesar de ello se verá que el resultado de planificabilidad borrosa es exacto.

$$\begin{aligned}
 & \text{for } i=1 \text{ to } n\_tareas \text{ do} \\
 & \quad S_{i,0} = B_i + \sum_{j=1}^i P_j ; \quad n=1 \\
 & \quad \text{loop} \\
 & \quad \quad \text{while } (S_{i,k} \neq S_{i,k-1}) \quad k = k+1 \\
 & \quad \quad \quad WC_{i,n} = S_{i,k} + J_i \\
 & \quad \quad \quad \text{if } (wc_{i,n}^c > n \cdot t_i) \\
 & \quad \quad \quad \quad n = n+1 ; \quad S_{i,k} = S_{i,k} + P_i \\
 & \quad \quad \quad \text{else} \\
 & \quad \quad \quad \quad \Pi_i / N_i = \min(\Pi / N(WC_{i,x} \leq D_{i,x})) \quad \forall x \in \mathbb{N} | 1 \leq x \leq n \\
 & \quad \quad \quad \quad \text{Siguiete tarea} \\
 & \quad \quad \quad \text{end} \\
 & \quad \text{endloop} \\
 & \text{end} \\
 (5.31) \quad & S_{i,k}(S_{i,k-1}) = B_i + n \cdot P_i + \sum_{j=1}^{i-1} P_j \cdot \left\lceil \frac{S_{i,k-1} + J_j}{t_j} \right\rceil \\
 & \Pi_{planif} / N_{planif} = \min(\Pi_i / N_i) \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas
 \end{aligned}$$

Y el algoritmo basado en aritmética de intervalos, con búsqueda por bipartición es:

$$\begin{aligned}
 & \text{ALG Aritmética de Intervalos} \\
 & \underline{wc}_{0,1}(0) = 0 \\
 & \text{for } i=1 \text{ to } n\_tareas \text{ do} \\
 & \quad \underline{wc}_{i,0}(0) = \underline{wc}_{i-1,1}(0) ; \quad n=1 \\
 & \quad \text{loop} \\
 & \quad \quad \underline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(0); b_i = \underline{b}_i(0); j_j = j_j(0); s\_in_i = \underline{wc}_{i,n-1}(0) + p_i \\ \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \end{array} \right) \\
 & \quad \quad \underline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(1); b_i = \underline{b}_i(1); j_j = j_j(1); s\_in_i = \underline{wc}_{i,n}(0) \\ \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \end{array} \right) \\
 & \quad \quad \overline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); s\_in_i = \overline{wc}_{i,n}(1) \\ \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \end{array} \right) \\
 & \quad \quad \overline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); s\_in_i = \overline{wc}_{i,n}(1) \\ \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \end{array} \right) \\
 & \quad \text{ALG ZONA} \\
 & \quad \text{if } wc_{i,n}^c > n \cdot t_i \\
 & \quad \quad n = n+1 \\
 & \quad \quad \text{else} \\
 & \quad \quad \quad \Pi_i / N_i = \min\{\Pi / N(WC_{i,x} \leq D_{i,x})\} \quad \forall x \in \mathbb{N} | 1 \leq x \leq n \\
 & \quad \quad \quad \text{siguiete tarea} \\
 & \quad \quad \quad \text{end if} \\
 & \quad \text{end loop} \\
 & \text{end} \\
 & \Pi_{planif} / N_{planif} = \min\{\Pi_i / N_i\} \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas
 \end{aligned}$$

Donde  $D_{i,n} = (n-1) \cdot t_i + D_i$  y el algoritmo básico es

<p><b>ALG BASICO:</b></p> $\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_{i,n} = s_{i,k} + j_i \\ \text{dado, } s_{i,k} = b_i + n \cdot p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1} + j_h}{t_h} \right\rceil \end{array} \right.$
--

El algoritmo ZONA es el mismo que aparece en (5.30). Los algoritmos Bipartición inferior y Bipartición superior son:

<p><b>Bipartición inf</b></p> $\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_{i,n}(0), \text{cota\_sup} = \underline{wc}_{i,n}(1)$ <p>while <math>\left( \left  \underline{wc}_{i,n}(\alpha) - \underline{d}_{i,n}(\alpha) \right  &gt; \varepsilon \right)</math></p> $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ $\underline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \underline{p}_j(\alpha); b_i = \underline{b}_i(\alpha); j_j = \underline{j}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \quad \underline{1 \leq j \leq i} \end{array} \right)$ <p>if <math>\left( \underline{wc}_{i,n}(\alpha) &lt; \underline{d}_{i,n}(\alpha) \right)</math> <math>\alpha_0 = \alpha, \text{cota\_inf} = \underline{wc}_{i,n}(\alpha)</math></p> <p>if <math>\left( \underline{wc}_{i,n}(\alpha) &gt; \underline{d}_{i,n}(\alpha) \right)</math> <math>\alpha_1 = \alpha, \text{cota\_sup} = \underline{wc}_{i,n}(\alpha)</math></p> <p>end</p> $\Pi(WC_{i,n} \leq D_{i,n}) = \alpha; \quad N(WC_{i,n} \leq D_{i,n}) = 0; \quad wc_{i,n}^c = \underline{wc}_{i,n}(\alpha)$
--

<p><b>Bipartición sup</b></p> $\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{wc}_{i,n}(1), \text{cota\_sup} = \overline{wc}_{i,n}(0)$ <p>while <math>\left( \left  \overline{wc}_{i,n}(\alpha) - \overline{d}_{i,n}(\alpha) \right  &gt; \varepsilon \right)</math></p> $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ $\overline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); b_i = \overline{b}_i(\alpha); j_j = \overline{j}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \overline{1 \leq j \leq i} \quad \overline{1 \leq j \leq i} \quad \overline{1 \leq j \leq i} \quad \overline{1 \leq j \leq i} \end{array} \right)$ <p>if <math>\left( \overline{wc}_{i,n}(\alpha) &lt; \overline{d}_{i,n}(\alpha) \right)</math> <math>\alpha_1 = \alpha, \text{cota\_inf} = \overline{wc}_{i,n}(\alpha)</math></p> <p>if <math>\left( \overline{wc}_{i,n}(\alpha) &gt; \overline{d}_{i,n}(\alpha) \right)</math> <math>\alpha_0 = \alpha, \text{cota\_sup} = \overline{wc}_{i,n}(\alpha)</math></p> <p>end</p> $\Pi(WC_{i,n} \leq D_{i,n}) = 1; \quad N(WC_{i,n} \leq D_{i,n}) = 1 - \alpha; \quad wc_{i,n}^c = \overline{wc}_{i,n}(\alpha)$
---

*Extensión del ejemplo B:*

El sistema consta de tres tareas, de periodos  $t_1=10$ ,  $t_2=15$  y  $t_3=35$ . Los tiempos de ejecución son:  $P_1=(3.9, 4, 4.1)$ ,  $P_2=(4.8, 5.9, 6.1)$  y  $P_3=6$ . Los tiempos de bloqueo son:  $B_1=0$ ,  $B_2=(1.9, 2, 2.1)$ ,  $B_3=0$ . Los plazos son:  $D_1=10$ ,  $D_2=(15.9, 16, 16.1)$ ,  $D_3=35$ .

La tarea 1 es la más prioritaria y la 3 la menos prioritaria. Como se puede ver, la tarea 2 es la única con tiempo de bloqueo y que además tiene su plazo mayor que el periodo.

Veamos los cálculos de tiempos para la tarea 2.

$$S_{2,0} = B_2 + P_1 + P_2 = (10.6, 11.9, 12.3)$$

$$n=1$$

$$S_{2,1} = B_2 + 1 \cdot P_2 + P_1 \cdot \lceil S_{2,0}/t_1 \rceil = B_2 + P_2 + \{P_1 \cdot 1/2\} = (14.5, 15.9, 16.4)$$

$$S_{2,2} = B_2 + 1 \cdot P_2 + P_1 \cdot \lceil S_{2,1}/t_1 \rceil = B_2 + P_2 + \{P_1 \cdot 1/2\} = S_{2,1}$$

Como se cumple que  $S_{2,2} = S_{2,1}$  éste es el tiempo de finalización de la tarea 2 en su primera ejecución  $WC_{2,1} = S_{2,2}$  (ver Figura 5.38).

Como  $\Pi(S_{2,2} > 1 \cdot t_2) \neq 0$ , el periodo de ocupación no ha terminado y que hay que calcular el tiempo de finalización en la segunda ejecución. Sin embargo, para determinados valores del tiempo de finalización no se cumple que sea mayor que el periodo: para los límites inferiores de los  $\alpha$ -cortes con  $\alpha$  menor que 0.357. Es decir, sólo para tiempos de ejecución mayores que 5.19 (que es el límite inferior del  $\alpha$ -corte con  $\alpha=0.357$ ) se cumple que el tiempo de finalización en la primera ejecución supera el periodo.

En la segunda ejecución  $n=2$  de la tarea,  $S_{2,2} = S_{2,2} + P_2 = (19.3, 21.8, 22.5)$ . Ver Figura 5.38.

Para  $k=3$ ,  $S_{2,3} = B_2 + 2 \cdot P_2 + P_1 \cdot \lceil S_{2,2}/t_1 \rceil = (11.5, 13.8, 14.3) + \{P_1 \cdot 1/3\} = (23.2, 25.8, 26.6)$ . Ver Figura 5.38.

Como  $S_{2,3} \neq S_{2,2}$  se continúa iterando,  $k=4$

$$S_{2,4} = B_2 + 2 \cdot P_2 + P_1 \cdot \lceil S_{2,3}/t_1 \rceil = (11.5, 13.8, 14.3) + \{P_1 \cdot 1/3\} = S_{2,3}$$

Como  $S_{2,4} = S_{2,3}$  se cumple que  $WC_{2,2} = S_{2,4}$

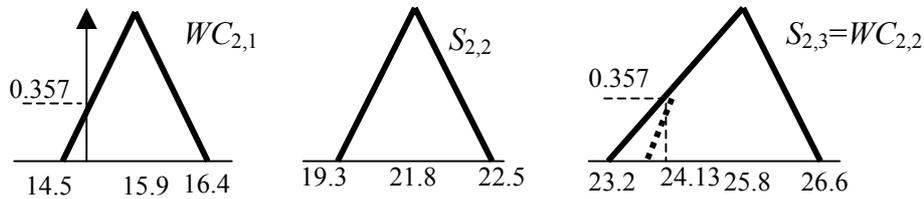


Figura 5.38: Cálculo iterativo del tiempo de finalización de la tarea 2 en su primera y segunda ejecución

Como el tiempo de finalización en la primera ejecución supera el periodo sólo para tiempos de ejecución mayores que 5.19 (que es el límite inferior del  $\alpha$ -corte con  $\alpha=0.357$ ), el tiempo de finalización de esta tarea en la segunda ejecución calculado mediante este algoritmo es válido sólo a partir de  $w_{C_{2,2}}=24.13$ , que es el límite inferior del  $\alpha$ -corte con  $\alpha=0.357$  (ver Figura 5.38). Para valores menores, los valores exactos de la función de pertenencia están por debajo. En la Figura 5.38 aparece marcado con línea discontinua. Si el plazo es mayor que el periodo, el punto de corte del plazo con el tiempo de finalización en el caso más desfavorable estará siempre a la derecha de dicho punto, en este caso, se trata de la primera ejecución.

Como  $\Pi(S_{2,4} > 2 \cdot t_2) = 0$  quiere decir que el periodo de ocupación ha terminado.

Hay que comparar cada tiempo de finalización con el plazo correspondiente (ver Figura 5.39).

$$\begin{aligned} \Pi(WC_{2,1} < D_2) &= 1 & N(WC_{2,1} < D_2) &= 1 - 0.833 = 0.166 \\ \Pi(WC_{2,2} < 1 \cdot t_2 + D_2) &= 1 & N(WC_{2,2} < 1 \cdot t_2 + D_2) &= 1 \end{aligned}$$

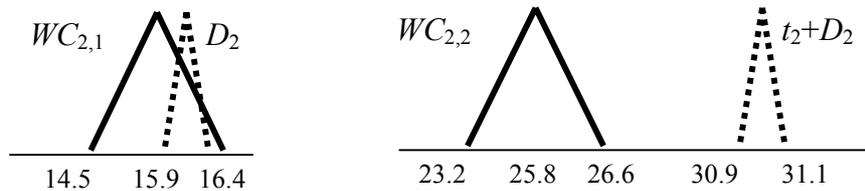


Figura 5.39: Comparación de los tiempos de finalización de la tarea 2 con los correspondientes plazos

Habría que coger el mínimo de estos valores junto con los resultantes de cálculo de peores tiempos de la tarea 1 y de la tarea 3, para obtener la posibilidad y necesidad de que el sistema sea planificable.

### 5.6.3. Extensión del algoritmo con periodos borrosos

Cuando los periodos son borrosos y los demás parámetros son numéricos, los tiempos de respuesta de las tareas son conjuntos borrosos. Existe una dependencia funcional entre el tiempo de respuesta de la tarea  $i$  y los periodos de las tareas más prioritarias.

Los tiempos de respuesta de las tareas son no crecientes con los periodos. Por esa razón, el algoritmo con parámetros numéricos se puede extender a periodos borrosos mediante aritmética borrosa o mediante aritmética de intervalos:

$$(5.32) \quad \underline{e}_{i,n} = f(\underline{t}_j) \quad \overline{e}_{i,n} = f(\overline{t}_j) \quad \forall j \in \mathbb{N} / 1 \leq j < i$$

Por otro lado, el tiempo de respuesta de  $i$  es independiente de su propio periodo  $t_i$ . Como  $wc_{i,n} = e_{i,n} + (n-1) \cdot t_i$  comparar  $wc_{i,n}$  con  $n \cdot t_i$  (que es el siguiente instante listo) es lo mismo que, restando a ambos términos  $(n-1) \cdot t_i$ , comparar  $e_{i,n}$  con  $t_i$ . Por tanto, al extender a números borrosos no hay relación funcional a tener en cuenta en la comparación de conjuntos borrosos  $E_{i,n}$  y  $T_i$ . Por el contrario, no se deben comparar los conjuntos borrosos  $WC_{i,n}$  y  $n \cdot T_i$  como si fuesen independientes, porque  $wc_{i,n}$  depende de  $t_i$ .

Por la misma razón se puede comparar directamente  $E_{i,n}$  con su plazo  $D_i$  ya que son independientes. No se debe comparar, por el contrario  $WC_{i,n}$  con  $D_{i,n}$  como si fueran independientes ya que  $wc_{i,n} = e_{i,n} + (n-1) \cdot t_i$  y  $d_{i,n} = d_i + (n-1) \cdot t_i$

Por tanto, el algoritmo extendido a periodos, tiempos de ejecución, tiempos de bloqueo y plazos borrosos en términos de aritmética borrosa queda:

```

for  $i = 1$  to  $n\_tareas$  do
 $S_{i,0} = B_i + \sum_{j=1}^i P_j$  ;  $n = 1$ 
loop
while ( $S_{i,k} \neq S_{i,k-1}$ )  $k = k + 1$ 
 $E_{i,n} = S_{i,k} + J_i - (n-1) \cdot T_i$ 
if ( $\Pi(E_{i,n} > T_i) \neq 0$ )
 $n = n + 1$  ;  $S_{i,k} = S_{i,k} + P_i$ 
else
 $\Pi_i / N_i = \min(\Pi / N(E_{i,x} \leq D_i)) \quad \forall x \in \mathbb{N} | 1 \leq x \leq n$ 
Siguiete tarea
end
endloop
end
 $S_{i,k}(S_{i,k-1}) = B_i + n \cdot P_i + \sum_{j=1}^{i-1} P_j \cdot \left\lceil \frac{S_{i,k-1} + J_j}{T_j} \right\rceil$ 
 $\Pi_{planif} / N_{planif} = \min(\Pi_i / N_i) \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas$ 

```

Y el algoritmo basado en aritmética de intervalos con búsqueda por bipartición:

```

ALG Aritmética de Intervalos
 $wc_{0,1}(0) = 0$ 
for  $i = 1$  to  $n\_tareas$  do
 $wc_{i,0}(0) = wc_{i-1,1}(0)$  ;  $n = 1$ 
loop
 $wc_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(0); b_i = \underline{b}_i(0); j_j = j_j(0); t_j = \overline{t}_j(0); s\_in_i = wc_{i,n-1}(0) + p_i \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
 $wc_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(1); b_i = \underline{b}_i(1); j_j = j_j(1); t_j = \overline{t}_j(1); s\_in_i = wc_{i,n}(0) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
 $\overline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); t_j = t_j(1); s\_in_i = wc_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
 $\overline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); t_j = t_j(0); s\_in_i = \overline{wc}_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$ 
ALG ZONA
if [ $e_{i,n}^c > t_i(0)$ ]  $n = n + 1$ 
else  $\Pi_i / N_i = \min\{\Pi / N(E_{i,x} \leq D_i)\} \quad \forall x \in \mathbb{N} | 1 \leq x \leq n$  ; siguiente tarea
end if
end loop
end
 $\Pi_{planif} / N_{planif} = \min\{\Pi_i / N_i\} \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas$ 

```

El algoritmo ZONA que tiene en cuenta la existencia de periodos borrosos es:

**ALG ZONA**

$$\underline{e}_{i,n}(0) = \underline{wc}_{i,n}(0) - (n-1)\underline{t}_i(0)$$

$$\underline{e}_{i,n}(1) = \underline{wc}_{i,n}(1) - (n-1)\underline{t}_i(1)$$

$$\overline{e}_{i,n}(0) = \overline{wc}_{i,n}(0) - (n-1)\overline{t}_i(0)$$

$$\overline{e}_{i,n}(1) = \overline{wc}_{i,n}(1) - (n-1)\overline{t}_i(1)$$

if ( $\underline{d}_i(0) \geq \underline{e}_{i,n}(0)$ )  $\Pi(E_{i,n} \leq D_i) = 1$ ,  $N(E_{i,n} \leq D_i) = 1$ ,

if [ $\overline{e}_{i,n}(0) > \underline{t}_i(0)$ ]  $e_{i,n}^c = \overline{t}_i(0) + \delta$

else  $e_{i,n}^c = 0$

if ( $\overline{d}_i(0) \leq \overline{e}_{i,n}(0)$ )  $\Pi(E_{i,n} \leq D_i) = 0$ ,  $N(E_{i,n} \leq D_i) = 0$ ,  $e_{i,n}^c = 0$

if ( $\underline{e}_{i,n}(1) \geq \underline{d}_i(1)$ )  $N(E_{i,n} \leq D_i) = 0$

else if ( $\overline{e}_{i,n}(1) < \underline{d}_i(1)$ ) Bipartición sup

if ( $\underline{e}_{i,n}(1) \leq \overline{d}_i(1)$ )  $\Pi(E_{i,n} \leq D_i) = 1$

else if ( $\overline{e}_{i,n}(1) > \overline{d}_i(1)$ ) Bipartición inf

donde los algoritmos de búsqueda por bipartición en los límites superiores y en los límites inferiores de los  $\alpha$ -cortes son:

**Bipartición inf**

$$\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_{i,n}(0), \text{cota\_sup} = \underline{wc}_{i,n}(1)$$

while ( $|e_{i,n}(\alpha) - \overline{d}_i(\alpha)| > \varepsilon$ )

$$\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$$

$$\underline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(\alpha); b_i = \underline{b}_i(\alpha); j_j = j_j(\alpha); t_j = \overline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

if ( $e_{i,n}(\alpha) < \overline{d}_i(\alpha)$ )  $\alpha_0 = \alpha$ ,  $\text{cota\_inf} = \underline{wc}_{i,n}(\alpha)$

if ( $e_{i,n}(\alpha) > \overline{d}_i(\alpha)$ )  $\alpha_1 = \alpha$ ,  $\text{cota\_sup} = \underline{wc}_{i,n}(\alpha)$

end

$$\Pi(E_{i,n} \leq D_i) = \alpha; \quad N(E_{i,n} \leq D_i) = 0; \quad e_{i,n}^c = \underline{e}_{i,n}(\alpha)$$

```

Bipartición sup
 $\alpha_0 = 0$  ,  $\alpha_1 = 1$  ,  $\alpha = 0$ ,  $cota\_inf = \overline{wc}_{i,n}(1)$ ,  $cota\_sup = \overline{wc}_{i,n}(0)$ 
while  $(|\overline{e}_{i,n}(\alpha) - d_i(\alpha)| > \varepsilon)$ 
     $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$ 
     $\overline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); b_i = \overline{b}_i(\alpha); j_j = \overline{j}_j(\alpha); t_j = \overline{t}_j(\alpha); s\_ini = cota\_inf \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$ 
    if  $(\overline{e}_{i,n}(\alpha) < d_i(\alpha))$   $\alpha_1 = \alpha$  ,  $cota\_inf = \overline{wc}_{i,n}(\alpha)$ 
    if  $(\overline{e}_{i,n}(\alpha) > d_i(\alpha))$   $\alpha_0 = \alpha$  ,  $cota\_sup = \overline{wc}_{i,n}(\alpha)$ 
end
 $\Pi(E_{i,n} \leq D_i) = 1$  ;  $N(E_{i,n} \leq D_i) = 1 - \alpha$  ;  $e_{i,n}^e = \overline{e}_{i,n}(\alpha)$ 

```

## 5.7. Extensión del algoritmo de cálculo del tiempo de bloqueo para el protocolo PIP

Uno de los protocolos de sincronización, descritos en el capítulo 2, para implementar la sección crítica de las tareas cuando éstas comparten recursos es el protocolo PIP o protocolo básico de herencia de prioridades.

Recordemos que en este protocolo se utiliza un semáforo para implementar la sección crítica. Cuando el recurso está ocupado, la petición se encola de acuerdo con su prioridad. Al encolar una petición, la prioridad de la tarea que está utilizando el recurso se eleva temporalmente al valor de la prioridad encolada y así se evita que haya expulsión en la sección crítica por parte de otra tarea, lo que produciría una inversión de prioridades no acotada. Al liberar el semáforo la tarea que está al comienzo de la cola accede al recurso.

El tiempo de bloqueo que puede sufrir cada tarea cuando hay recursos compartidos en el sistema depende del tipo de protocolo utilizado, de la asignación de prioridades y de los tiempos de utilización de los recursos. En el capítulo 2 se presentó el algoritmo de cálculo de los tiempos de bloqueo cuando el protocolo es PIP.

En la extensión de dicho algoritmo a argumentos borrosos se considera que las prioridades de las tareas son fijas y que el tiempo de utilización de un recurso  $r$  por parte de las tareas es borroso  $RU_i$ . A modo de ejemplo se muestra la extensión del algoritmo de cálculo de los tiempos de bloqueo para este tipo de protocolo:

1. Se ordenan las tareas de acuerdo con su prioridad. La tarea 1 es la más prioritaria.
2. Inicialización de  $B$ :  $B = 0$  (clásico)
3. Comienzo con la tarea menos prioritaria:  $i = n$
4.  $B_i = B$
5.  $B = \widetilde{\max}(RU_i, B)$
6. Si  $i = 1$ , termina el algoritmo. Si  $pr_{i-1} > pc_r$ , entonces  $B = 0$  (clásico)
7. Iterar:  $i = i-1$  y continúa en el paso 4.

Donde la operación borrosa  $\widetilde{\max}$  indica el máximo de una serie de números borrosos [Dubois, Prade, 1980]. El máximo de dos números borrosos es otro número borroso obtenido tal y como se indica en la Figura 5.40 con línea discontinua:

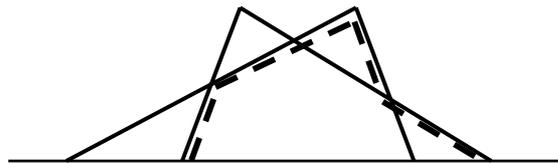


Figura 5.40: Máximo de dos números borrosos

Se utiliza la función máximo extendida, ya que el resultado no tiene porqué ser igual a ninguno de los operandos. Si se hubiera extendido directamente la expresión “Si  $ru_i > b$ , entonces  $b = ru_i$ ” a números borrosos, el resultado podría ser o  $B$  o  $RU_i$

## 5.8. Planificabilidad de sistemas con servidores aperiódicos

En este apartado se va a estudiar la extensión del análisis de planificabilidad de sistemas formados por tareas periódicas planificadas según un criterio de prioridades fijas y por tareas aperiódicas que son atendidas mediante un servidor. Se analizan dos tipos de servidor, que ya fueron descritos en el estado del arte: Servidor Esporádico (Sporadic Server: SS) [Sprunt et al., 1989] y Servidor Aplazable (Deferrable Server: DS) [Lehoczky et al., 1987],[Strosnider et al., 1995].

### 5.8.1. Servidor Esporádico

Cuando se utiliza un Servidor Esporádico para atender a las tareas aperiódicas, el peor tiempo de respuesta de la tarea periódica  $i$  se obtiene cuando  $s_{i,k}=s_{i,k-1}$  mediante la ecuación iterativa

$$s_{i,k} = p_i + \sum_{j=1}^{i-1} \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil \cdot p_j + \left\lceil \frac{s_{i,k-1}}{t_s} \right\rceil \cdot c_s$$

Se puede ver que existe interferencia del servidor de tareas aperiódicas en la ejecución de las tareas periódicas, retrasando su respuesta. Esta interferencia depende de la capacidad del servidor  $c_s$  y del periodo del mismo  $t_s$ .

Tal y como se puede ver en la ecuación, desde el punto de vista de la planificabilidad, el Servidor Esporádico (SS) se puede sustituir por una tarea periódica con el mismo periodo y tiempo de ejecución [Sprunt et al., 1989].

Por tanto, la extensión de la fórmula anterior a valores borrosos de la capacidad y periodo del servidor sigue el mismo razonamiento que se presentó para extender el análisis de planificabilidad de tareas periódicas con tiempos de ejecución y periodos borrosos. Al igual que ocurre con los tiempos de ejecución, el algoritmo iterativo es no decreciente con la capacidad del servidor. Al igual que pasa con los periodos de las tareas periódicas, el algoritmo es no creciente con el periodo del servidor SS. Por tanto se cumple que los límites inferiores y superiores de los  $\alpha$ -cortes de los tiempos de finalización borrosos se pueden obtener a partir de los límites de la capacidad y del periodo del servidor de la siguiente forma:

$$\begin{aligned} \underline{wc}(\alpha) &= f\left(\underline{c}_s(\alpha), \overline{t}_s(\alpha)\right) \\ \overline{wc}(\alpha) &= f\left(\overline{c}_s(\alpha), \underline{t}_s(\alpha)\right) \end{aligned}$$

donde  $f$  es el algoritmo iterativo.

Así pues, el algoritmo de planificabilidad basado en aritmética borrosa cuando los tiempos de ejecución, los plazos, los periodos, la capacidad del servidor y el periodo del servidor son borrosos es:

**Algoritmo de Aritmética Borrosa**  
 $WC_0 = 0$  (crisp)  
 for  $i=1$  to  $n\_tarefas$  do  
      $S_{i,0} = WC_{i-1} + P_i$   
     while  $(S_{i,k-1} \neq S_{i,k}(S_{i,k-1}))$   $k=k+1$   
          $WC_i = S_{i,k}$   
          $\Pi / N(WC_i \leq D_i)$   
 end  
 $\Pi_s / N_s = \min \{ \Pi / N(WC_i \leq D_i) \} \quad \forall i$   
 dado  $S_{i,k}(S_{i,k-1}) = P_i + \sum_{j=1}^{i-1} P_j \cdot \left\lceil \frac{S_{i,k-1}}{T_j} \right\rceil + C_s \cdot \left\lceil \frac{S_{i,k-1}}{T_s} \right\rceil$

Donde  $C_s$  es la capacidad borrosa del servidor y  $T_s$  es el periodo borroso del servidor.

El algoritmo basado en aritmética de intervalos, para los mismos parámetros borrosos y con la inicialización propuesta en el apartado 5.2.3 es:

**ALG Aritmética de Intervalos**  
 $wc_0(0) = 0$   
 for  $i=1$  to  $n\_tasks$  do  
      $wc_i(0) = \text{ALG BASICO} \left( p_j = p_j(0); c_s = c_s(0); t_s = \bar{t}_s(0); t_j = \bar{t}_j(0); s\_in_i = wc_{i-1}(0) + p_i \right)$   
      $wc_i(1) = \text{ALG BASICO} \left( p_j = p_j(1); c_s = c_s(1); t_s = \bar{t}_s(1); t_j = \bar{t}_j(1); s\_in_i = wc_i(0) \right)$   
      $\bar{wc}_i(1) = \text{ALG BASICO} \left( p_j = \bar{p}_j(1); c_s = \bar{c}_s(1); t_s = \underline{t}_s(1); t_j = \underline{t}_j(1); s\_in_i = wc_i(1) \right)$   
      $\bar{wc}_i(0) = \text{ALG BASICO} \left( p_j = \bar{p}_j(0); c_s = \bar{c}_s(0); t_s = \underline{t}_s(0); t_j = \underline{t}_j(0); s\_in_i = \bar{wc}_i(1) \right)$   
     ALG ZONA  
 end  
 (5.33)  $\Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq D_i) \} \quad \forall i$

Donde el algoritmo básico es:

**ALG BASICO:**

$$\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ \quad wc_i = s_{i,k} \\ \text{dado, } s_{i,k} = p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil + c_s \cdot \left\lceil \frac{s_{i,k-1}}{t_s} \right\rceil \end{array} \right.$$

El algoritmo ZONA (5.16), es el mismo que se mostró en el apartado de prioridades fijas y plazos menores o iguales que los periodos. Los algoritmos Bipartición Superior y Bipartición Inferior son:

$$\begin{aligned}
 & \textbf{Bipartición inf} \\
 & \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_i(0), \text{cota\_sup} = \underline{wc}_i(1) \\
 & \text{while } \left( \left| \underline{wc}_i(\alpha) - \bar{d}_i(\alpha) \right| > \varepsilon \right) \\
 & \quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 & \quad \underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \underline{p}_j(\alpha); c_s = \underline{c}_s(\alpha); t_s = \underline{t}_s(\alpha); t_j = \underline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 & \quad \text{if } \left( \underline{wc}_i(\alpha) < \bar{d}_i(\alpha) \right) \quad \alpha_0 = \alpha, \quad \text{cota\_inf} = \underline{wc}_i(\alpha) \\
 & \quad \text{if } \left( \underline{wc}_i(\alpha) > \bar{d}_i(\alpha) \right) \quad \alpha_1 = \alpha, \quad \text{cota\_sup} = \underline{wc}_i(\alpha) \\
 & \text{end} \\
 & \Pi(WC_i \leq D_i) = \alpha; \quad N(WC_i \leq D_i) = 0
 \end{aligned}
 \tag{5.34}$$

$$\begin{aligned}
 & \textbf{Bipartición sup} \\
 & \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{wc}_i(1), \text{cota\_sup} = \overline{wc}_i(0) \\
 & \text{while } \left( \left| \overline{wc}_i(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right) \\
 & \quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 & \quad \overline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); c_s = \overline{c}_s(\alpha); t_s = \overline{t}_s(\alpha); t_j = \overline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 & \quad \text{if } \left( \overline{wc}_i(\alpha) < \underline{d}_i(\alpha) \right) \quad \alpha_1 = \alpha, \quad \text{cota\_inf} = \overline{wc}_i(\alpha) \\
 & \quad \text{if } \left( \overline{wc}_i(\alpha) > \underline{d}_i(\alpha) \right) \quad \alpha_0 = \alpha, \quad \text{cota\_sup} = \overline{wc}_i(\alpha) \\
 & \text{end} \\
 & \Pi(WC_i \leq D_i) = 1; \quad N(WC_i \leq D_i) = 1 - \alpha
 \end{aligned}
 \tag{5.35}$$

### 5.8.2. Servidor Aplazable

Cuando el servidor utilizado para atender a las tareas aperiódicas es el Servidor Aplazable (DS) no resulta tan sencillo extraer conclusiones sobre la extensión del análisis de planificabilidad como en el caso anterior.

La fórmula utilizada para calcular el peor tiempo de una tarea  $i$ , cuando  $s_{i,k} = s_{i,k-1}$  es:

$$s_{i,k} = p_i + \sum_{j=1}^{i-1} \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil \cdot p_j + \left( 1 + \left\lceil \frac{s_{i,k-1} - c_s}{t_s} \right\rceil \right) \cdot c_s$$

En este caso la monotonicidad de la función aparece enmascarada. Si consideráramos que  $s_{i,k-1}$  es constante, la función:

$$\left( 1 + \left\lceil \frac{s_{i,k-1} - c_s}{t_s} \right\rceil \right) \cdot c_s$$

no es monótona en  $c_s$ . La forma de dicha función sería la indicada en la Figura 5.41.

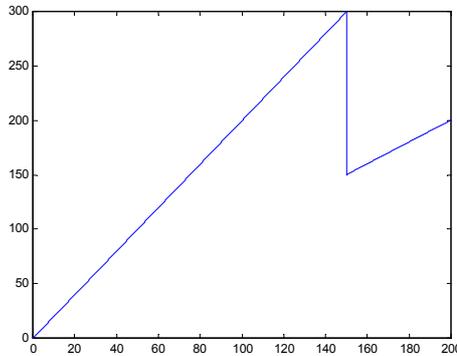


Figura 5.41: Último sumando de  $s_{i,k}$  en función de  $c_s$  suponiendo que  $s_{i,k-1}$  es constante

Sin embargo, hay que tener en cuenta que  $s_{i,k-1}$  no es constante sino que a su vez depende de  $c_s$ . A continuación se presenta la prueba de que el algoritmo iterativo es monótono no decreciente en  $c_s$ .

En primer lugar, suponemos que  $c_s < t_s$ , ya que el servidor debe dejar tiempo para la ejecución de las tareas periódicas. Para simplificar, estudiamos la respuesta de la tarea periódica más prioritaria  $i=1$  y por tanto no hay interferencia de otras tareas periódicas. Este supuesto no afecta al resultado.

Comenzamos el análisis partiendo de  $s_{1,0}=0$ . Como  $c_s < t_s$ , ( $-1 < -c_s/t_s < 0$ ), y por tanto,

$$s_{1,1} = p_1 + \left( 1 + \left\lceil \frac{s_{1,0} - c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + c_s$$

En la siguiente iteración, sustituyendo  $s_{1,1}$  por su valor:

$$s_{1,2} = p_1 + \left( 1 + \left\lceil \frac{p_1 + c_s - c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + (1 + n_2) \cdot c_s \quad \text{donde } n_2 \in \mathbb{N} - \{0\}$$

En la siguiente iteración, podemos ver la relación del nuevo valor  $n_3$  con respecto a  $n_2$ :

$$s_{1,3} = p_1 + \left( 1 + \left\lceil \frac{p_1 + (1 + n_2)c_s - c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + \left( 1 + \left\lceil \frac{p_1 + n_2 \cdot c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + (1 + n_3) \cdot c_s$$

donde  $n_3 \in \mathbb{N} - \{0\}$ . Además como  $\frac{p_1 + n_2 \cdot c_s}{t_s} > \frac{p_1}{t_s}$ , entonces  $\left\lceil \frac{p_1 + n_2 \cdot c_s}{t_s} \right\rceil \geq \left\lceil \frac{p_1}{t_s} \right\rceil$ , o

lo que es lo mismo,  $n_3 \geq n_2$

En la siguiente iteración se tiene:

$$s_{1,4} = p_1 + \left( 1 + \left\lceil \frac{p_1 + (1 + n_3)c_s - c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + \left( 1 + \left\lceil \frac{p_1 + n_3 \cdot c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + (1 + n_4) \cdot c_s$$

donde  $n_4 \in \mathbb{N} - \{0\}$ . Además como  $n_3 \geq n_2$  entonces  $\left\lceil \frac{p_1 + n_3 \cdot c_s}{t_s} \right\rceil \geq \left\lceil \frac{p_1 + n_2 \cdot c_s}{t_s} \right\rceil$ , o lo

que es lo mismo,  $n_4 \geq n_3$ .

Para una iteración  $k$ , se tiene:

$$s_{1,k} = p_1 + \left( 1 + \left\lceil \frac{p_1 + (1 + n_{k-1})c_s - c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + \left( 1 + \left\lceil \frac{p_1 + n_{k-1} \cdot c_s}{t_s} \right\rceil \right) \cdot c_s = p_1 + (1 + n_k) \cdot c_s$$

donde  $n_k \in \mathbb{N} - \{0\}$ .

Además como  $n_{k-1} \geq n_{k-2}$  entonces  $\left\lceil \frac{p_1 + n_{k-1} \cdot c_s}{t_s} \right\rceil \geq \left\lceil \frac{p_1 + n_{k-2} \cdot c_s}{t_s} \right\rceil$ , o lo que es lo

mismo,  $n_k \geq n_{k-1}$ .

De forma similar se continuaría el análisis hasta que  $s_{i,k} = s_{i,k-1}$

Si observamos ahora el efecto que tiene un incremento en  $c_s$  sobre el tiempo de respuesta:

Al aumentar  $c_s$  también aumenta  $s_{1,1} = p_1 + c_s$ . También  $s_{1,2} = p_1 + (1 + n_2) \cdot c_s$  aumenta. El tercer valor de  $s$ ,  $s_{1,3} = p_1 + \left(1 + \left\lceil \frac{p_1 + n_2 \cdot c_s}{t_s} \right\rceil\right) \cdot c_s = p_1 + (1 + n_3) \cdot c_s$  ve aumentado, o permanece igual, tanto el término  $n_2$  como el factor  $c_s$ , con lo cual aumenta.

El cuarto  $s_{1,4} = p_1 + \left(1 + \left\lceil \frac{p_1 + n_3 \cdot c_s}{t_s} \right\rceil\right) \cdot c_s = p_1 + (1 + n_4) \cdot c_s$  ve aumentado, o permanece igual, tanto el término  $n_4$  (porque  $n_3$  aumenta o permanece igual y  $c_s$  aumenta) como el factor  $c_s$ .

Para cualquier iteración, el valor de  $s$  aumenta al aumentar  $c_s$ , con lo que el valor del peor tiempo de la tarea también aumenta.

$$s_{1,k} = p_1 + \left(1 + \left\lceil \frac{p_1 + n_{k-1} \cdot c_s}{t_s} \right\rceil\right) \cdot c_s = p_1 + (1 + n_k) \cdot c_s$$

En consecuencia, tenemos que el tiempo de finalización de la tarea  $i$  en el peor caso es monótona creciente con la capacidad del servidor.

Por otro lado, el tiempo de finalización es monótono no creciente con el periodo de las tareas, como ocurre con los periodos de las tareas periódicas,.

Por tanto, al igual que con el servidor SS, los límites inferiores y superiores de los  $\alpha$ -cortes de los tiempos de finalización borrosos se pueden obtener a partir de los límites de la capacidad y del periodo del servidor como:

$$\underline{wc}(\alpha) = f(\underline{c_s}(\alpha), \bar{t_s}(\alpha))$$

$$\overline{wc}(\alpha) = f(\bar{c_s}(\alpha), \underline{t_s}(\alpha))$$

donde  $f$  es el algoritmo iterativo.

Como hay operaciones parciales crecientes y otras decrecientes en  $c_s$ , no es posible extender el análisis mediante aritmética borrosa ya que daría una aproximación no válida de los tiempos de finalización borrosos.

El algoritmo basado en aritmética de intervalos con búsqueda por bipartición es el mismo que aparece en (5.33), donde el algoritmo básico es:

<p><b>ALG BASICO:</b></p> $\left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ \quad wc_i = s_{i,k} \\ \text{dado, } s_{i,k} = p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil + c_s \cdot \left( 1 + \left\lceil \frac{s_{i,k-1} - c_s}{t_s} \right\rceil \right) \end{array} \right.$
--

El algoritmo ZONA (5.16) es el mismo que para prioridades fijas y plazos menores o iguales que los periodos. Los algoritmos Bipartición Superior (5.35) y Bipartición Inferior (5.34) coinciden con los presentados para el servidor SS.

Concluyendo, en este capítulo se han propuesto extensiones borrosas del análisis de planificabilidad para tiempos de ejecución, plazos, periodos, tiempos de bloqueo, “release jitter”, y capacidad y periodo de los servidores SS y DS borrosos. Se han propuesto algoritmos, tanto para el caso de que los plazos sean fuertemente menores que los periodos, como para cualquier plazo. Las extensiones basadas en aritmética borrosa y la basadas en aritmética de intervalos proporcionan los mismos resultados que la aplicación del principio de extensión. Únicamente no es posible extender el algoritmo mediante aritmética borrosa en el caso de tener un servidor DS si la capacidad del servidor es borrosa.

## **6. Análisis borroso de planificabilidad en sistemas con prioridades dinámicas: EDF**

Tal y como se describió en el capítulo 2, el algoritmo “Earliest Deadline First” (EDF) es un algoritmo de planificación dinámica expulsor que selecciona la tarea a ejecutar de acuerdo con su plazo absoluto, de forma que la tarea con el plazo más cercano se ejecuta con la prioridad más alta. EDF no hace suposición alguna sobre la periodicidad de las tareas, por lo que se puede usar para planificar tanto tareas periódicas como aperiódicas.

Se pueden encontrar en la literatura diversos análisis para determinar si el sistema es planificable cuando las tareas son periódicas. Dichos análisis distinguen dos situaciones diferentes: que el plazo sea igual al periodo o que el plazo sea menor que el periodo. En este capítulo se describen los análisis de planificabilidad y se extienden a tiempos de ejecución, plazos y periodos borrosos.

En el apartado 6.1 se extiende el análisis de planificabilidad basado en la utilización total del procesador a tiempos de ejecución y periodos borrosos cuando los plazos de ejecución de las tareas son iguales a los periodos.

En el apartado 6.2 se extiende el análisis de planificabilidad cuando los plazos de las tareas son menores que los periodos. En ese caso el análisis se basa en la demanda del procesador y se extiende a tiempos de ejecución borrosos, sin embargo no se puede extender este análisis en general a plazos y periodos borrosos aplicando aritmética de intervalos o aritmética borrosa con los mismos resultados de posibilidad y necesidad que la aplicación del principio de extensión. Por último en este apartado cuando los plazos son borrosos se definen la posibilidad y necesidad total como los valores de posibilidad y necesidad de planificabilidad teniendo en cuenta todas las asignaciones de prioridades según las posibles ordenaciones de los plazos, y se muestra que, si existe orden fuerte entre los plazos absolutos borrosos, la posibilidad y necesidad total coincide con la posibilidad y necesidad dadas por la asignación de prioridades asociada al orden fuerte.

## **6.1. Plazos iguales a los periodos**

En capítulos anteriores se expuso que en sistemas con prioridades fijas la forma de determinar si un sistema es planificable es calcular los tiempos de finalización de las tareas en un hiperperiodo y compararlos con los correspondientes plazos. Cuando existe un instante crítico es suficiente con calcular los tiempos de finalización en la primera ejecución ya que ese es el mayor tiempo de respuesta de cada tarea. Por otro lado, la condición del límite de utilización total del procesador no proporciona una

condición necesaria y suficiente en este caso, sino sólo suficiente ([Liu, Layland, 1973]).

En cambio, cuando un sistema tiene los plazos de las tareas iguales a los periodos y se planifica según el algoritmo de planificación EDF, la condición dada por el límite de utilización sí proporciona una condición necesaria y suficiente de planificabilidad del sistema, por lo que no es preciso calcular los tiempos de finalización de las tareas ([Buttazzo, 1997]).

Cuando para todas las tareas periódicas el plazo es igual al periodo, se dice que el sistema es planificable con EDF si y sólo si la utilización total es menor o igual que 1.

$$(6.1) \sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} \leq 1$$

En este caso el límite de utilización es 1, por lo que las tareas pueden utilizar el procesador al 100% y ser planificables. En el ejemplo de la Figura 6.1 se compara la planificación dos tareas A y B obtenida mediante RMS y la obtenida mediante EDF. Los periodos de las tareas son  $t_A = 5$  y  $t_B = 7$ , los plazos son iguales a los periodos y los tiempos de ejecución son  $p_A = 2$  y  $p_B = 4$ .

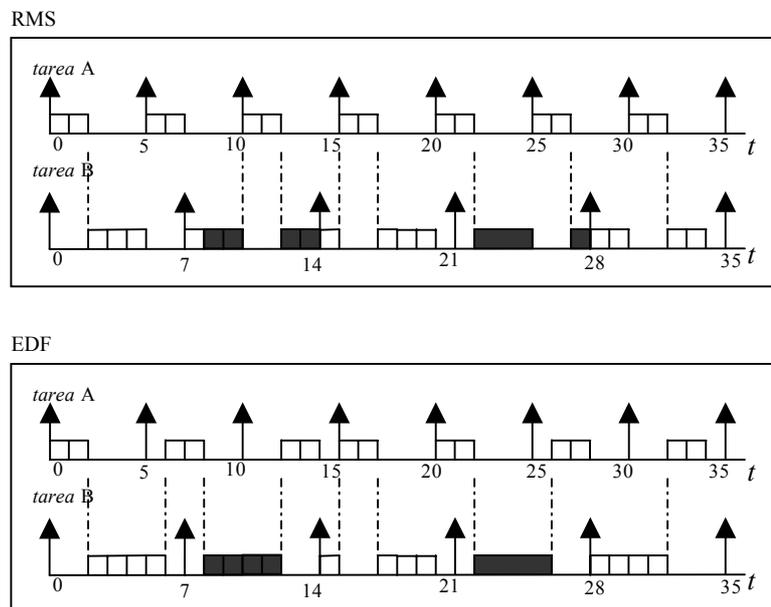


Figura 6.1: Dos tareas planificadas mediante RMS y mediante EDF

La utilización total del procesador en este ejemplo es  $U = 2/5 + 4/7 = 0.97$ , lo que quiere decir que el 97% del tiempo del procesador se utiliza en la ejecución de tareas periódicas quedando disponible el 3% restante. Como  $U > \ln 2$ , no es posible garantizar la planificabilidad de las tareas con RMS, mientras que sí se garantiza con EDF.

Tal y como se ve en la Figura 6.1, con RMS se produce un sobrepaso del plazo de la tarea  $B$  en su primera ejecución, mientras que con EDF todas las tareas terminan antes de su plazo. También se puede apreciar la diferencia en el número de veces que las tareas son expulsadas. En el ejemplo se ve que con RMS la tarea  $B$  es expulsada cinco veces en un hiperperiodo, definido como el mínimo común múltiplo de los periodos (ver capítulo 2), y en este caso vale  $h = t_A \cdot t_B$ . En cambio, con EDF la tarea  $B$  es expulsada una sola vez en  $h$  [Buttazzo, 1997].

A continuación se presenta la condición de planificabilidad expresada en función del hiperperiodo  $h$ . Tanto (6.1), como la nueva condición se van a utilizar para extender el análisis de planificabilidad basado en la utilización del procesador a parámetros borrosos.

### 6.1.1. Ecuación de planificabilidad de EDF en función del hiperperiodo

Dado el hiperperiodo  $h$  de las tareas de un sistema, podemos decir que transcurrido un tiempo  $h$  de ejecución del sistema, éste se encuentra en la misma situación que al principio del hiperperiodo siempre y cuando la utilización del procesador sea menor o igual al 100%. Por tanto, es suficiente con analizar el sistema en un hiperperiodo.

El algoritmo de planificación EDF permite una utilización de la CPU del 100% correspondiente a un total aprovechamiento de la CPU, por lo que el tiempo en que la CPU está ocupada durante un hiperperiodo, sin que se sobrepase ningún plazo, es como máximo  $h$ .

En un hiperperiodo  $h$ , la tarea  $i$  se ejecuta un número  $h/t_i$  de veces y el tiempo de CPU que ocupa la tarea  $i$  en  $h$  es  $p_i \cdot (h/t_i)$ . Así, teniendo en cuenta la ejecución de todas las tareas del sistema, el tiempo total que la CPU está ocupada en un hiperperiodo es:

$$\sum_{i=1}^{n_{tar}} p_i \cdot \frac{h}{t_i}$$

Como ya se ha dicho, el tiempo máximo en que la CPU puede estar ocupada es el propio hiperperiodo, con lo que la condición de planificabilidad queda:

$$(6.2) \sum_{i=1}^{n\_tar} p_i \cdot \frac{h}{t_i} \leq h$$

Dividiendo ambos términos por  $h$ , se obtiene la ecuación de partida (6.1), por tanto, ambas desigualdades son equivalentes.

En el caso límite en que el término de la izquierda de (6.2) es igual a  $h$ , la CPU está utilizada al 100%, y se puede decir que el tiempo de finalización de la última tarea ejecutada (en su última ejecución dentro del hiperperiodo) es  $h$ . De forma similar, si la utilización es mayor del 100%, el primer término de la desigualdad se puede interpretar como el tiempo de finalización de la última tarea ejecutada en ese hiperperiodo, tiempo que es mayor que  $h$ . Esto se debe a que en  $h$  la CPU está ocupada en todo momento.

Cuando la utilización es menor que 100%, entonces el resultado del primer término de la desigualdad es menor o igual que el tiempo de finalización de la última tarea ejecutada en  $h$ , ya que ese término no tiene en cuenta que puede haber “huecos” en la ocupación de la CPU, es decir, que no tienen porqué ejecutarse todas la tareas seguidas sin interrupción en la ocupación de la CPU, sino que por el contrario, normalmente existen huecos en los que la CPU queda libre, desde que una tarea finaliza una ejecución, hasta que la siguiente está lista.

### 6.1.2. Extensión del análisis de planificabilidad a tiempos de ejecución borrosos

En este capítulo se propone extender la ecuación de utilización del procesador a parámetros borrosos para obtener las medidas de posibilidad y de necesidad de que el sistema sea planificable, en lugar de aplicar el cálculo inicialmente propuesto a partir de los tiempos borrosos de finalización de las tareas dentro de un hiperperiodo y que se expresa como:

$$\begin{aligned} \Pi_{planif} &= \min(\Pi(WC_{ij} < d_{ij})) \\ N_{planif} &= \min(N(WC_{ij} < d_{ij})) \quad \forall i \in \mathbb{N} \mid 1 \leq i \leq n\_tarefas \end{aligned}$$

El resultado obtenido de planificabilidad borrosa que se obtiene a partir de la utilización total es el mismo que el que se obtiene si se calculan los valores de posibilidad y de necesidad de planificabilidad según las expresiones anteriores, pero es más rápido de calcular mediante la ecuación de utilización.

Siendo así, se puede extender la ecuación (6.1) que se describe a continuación:

$$\sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} \leq 1,$$

a tiempos de ejecución borrosos  $P_i$ , según se indica en (6.3), donde 1 es un número clásico, según se describe a continuación.

$$(6.3) \quad \sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} \leq 1$$

Extender (6.1) a parámetros borrosos es equivalente a extender (6.2), por lo que en la expresión de (6.1) con parámetros borrosos el 1 de la desigualdad sigue siendo numérico, independientemente de si el hiperperiodo es borroso o numérico. Al aplicar el principio de extensión a (6.2), se ha de escoger el mismo valor de  $h$  a un lado y al otro de la ecuación, por lo que se puede simplificar, obteniéndose el valor numérico 1 en (6.3).

Por tanto, hay que comparar los dos términos de (6.3) calculando la medida de posibilidad y la medida de necesidad (6.4) de que la utilización total borrosa sea menor o igual que 1, para dar el grado de planificabilidad del sistema:

$$(6.4) \quad \Pi_{planif} = \Pi \left( \sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} \leq 1 \right) \quad N_{planif} = N \left( \sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} \leq 1 \right)$$

Nótese que el término de la izquierda de (6.1) es una función continua creciente en  $p_i$ . Por esa razón, se puede obtener mediante aritmética de intervalos y aritmética borrosa el resultado exacto de utilización borrosa del procesador.

*Ejemplo:*

Sea un sistema de tiempo real formado por dos tareas periódicas cuyos periodos son  $t_1=3$  y  $t_2=5$  y cuyos tiempos de ejecución están modelados como números borrosos triangulares  $P_1=(0.9, 1, 1.1)$  y  $P_2=(3.2, 3.3, 3.6)$ .

Se puede calcular el primer término de (6.3) mediante aritmética borrosa como:

$$\sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} = \frac{(0.9, 1, 1.1)}{3} + \frac{(3.2, 3.3, 3.6)}{5} = (0.3, 0.333, 0.366) + (0.64, 0.66, 0.72)$$

Con lo que el número borroso triangular resultante es  $\sum_{i=1}^{n \text{ tar}} \frac{P_i}{t_i} = (0.94, 0.993, 1.086)$

Comparando la utilización total borrosa con el número clásico 1, se obtienen las siguientes medidas borrosas de planificabilidad del sistema, que se ilustran en la Figura 6.2.

$$\Pi_{planif} = \Pi \left( \sum_{i=1}^2 \frac{P_i}{t_i} \leq 1 \right) = 1$$

$$N_{planif} = N \left( \sum_{i=1}^2 \frac{P_i}{t_i} \leq 1 \right) = 1 - \Pi \left( \sum_{i=1}^2 \frac{P_i}{t_i} > 1 \right) = 1 - 0.928 = 0.072$$

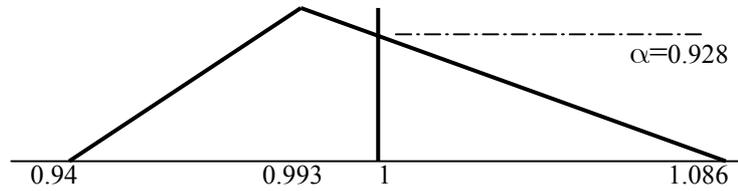


Figura 6.2: Representación gráfica de la condición borrosa de planificabilidad extendiendo (6.1)

Por otro lado, si en lugar de extender (6.1), se extiende (6.2) a tiempos de ejecución borrosos, el hiperperiodo  $h$  vale 15, y se tiene:

$$\sum_{i=1}^2 P_i \cdot \frac{15}{t_i} = (14.1, 14.9, 16.3)$$

Comparando dicho número borroso triangular con el hiperperiodo  $h=15$ , se tienen los siguientes valores de posibilidad y de necesidad, ilustrados en la Figura 6.3.

$$\Pi_{planif} = \Pi \left( \sum_{i=1}^2 P_i \cdot \frac{15}{t_i} \leq 15 \right) = 1$$

$$N_{planif} = N \left( \sum_{i=1}^2 P_i \cdot \frac{15}{t_i} \leq 15 \right) = 1 - 0.928 = 0.072$$

En la Figura 6.3 se muestra el número borroso triangular obtenido del cálculo borroso anterior (14.1, 14.9, 16.3), el plazo de la tarea y un ejemplo, en línea discontinua, de cómo podría ser el tiempo de finalización de la última tarea que se ejecuta en  $h$ , (el tiempo de finalización de la tarea 1 en su quinta ejecución) cuando éste es menor que  $h$ , de forma que el tiempo de finalización es mayor o igual que (14.1, 14.9, 16.3). Cuando el tiempo de finalización es mayor o igual que  $h$ , éste coincide con el tramo correspondiente del número borroso triangular (14.1, 14.9, 16.3).

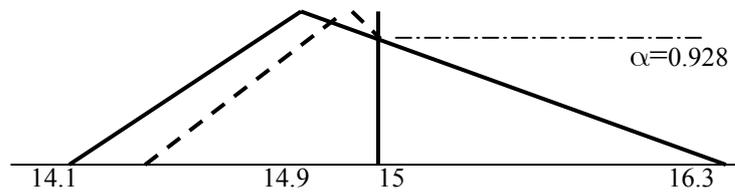


Figura 6.3: Representación gráfica de la condición borrosa de planificabilidad extendiendo (6.2)

Se puede observar que los resultados de posibilidad y de necesidad coinciden al extender las ecuaciones (6.1) y (6.2) a tiempos de ejecución borrosos.

También es posible obtener los tiempos de finalización de las tareas mediante simulación del sistema de tiempo real en un hiperperiodo. Si los tiempos de ejecución son números borrosos, se pueden obtener aproximaciones válidas de los tiempos de finalización borrosos mediante simulación por intervalos de los tiempos de ejecución, tal y como se ha descrito en los capítulos 3 y 5, ya que los tiempos de finalización son funciones no decrecientes con los tiempos de ejecución.

Tomemos como ejemplo un sistema de tiempo real con tres tareas periódicas  $A$ ,  $B$  y  $C$  con periodos iguales a sus plazos y de valores  $t_A=3$ ,  $t_B=5$  y  $t_C=15$ . Sus tiempos de ejecución son números borrosos triangulares  $P_A=(0.9;1;1.1)$ ,  $P_B=(1.9;2;2.1)$  y  $P_C=(1.9;2;2.1)$ . La Figura 6.4 muestra las distribuciones de posibilidad del tiempo de finalización de  $C$  en su primera ejecución obtenidas aplicando el principio de extensión y simulación por intervalos.

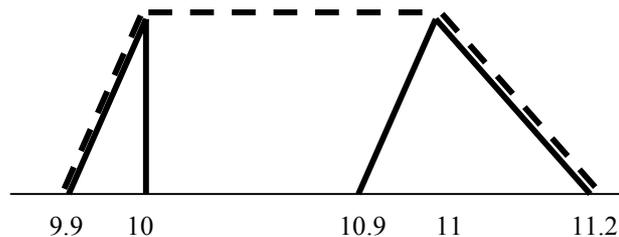


Figura 6.4: Tiempo de finalización de  $C$  en la primera ejecución mediante el principio de extensión (línea continua) y mediante simulación por intervalos (línea discontinua)

### 6.1.3. Extensión del análisis de planificabilidad a periodos borrosos

Además de extender la ecuación (6.1) a tiempos de ejecución borrosos, también se puede analizar la forma de extenderla a periodos borrosos.

Es importante distinguir el modelado de los periodos como números borrosos de otras dos situaciones diferentes en las que se modela la incertidumbre en torno al periodo, ya sea el plazo o ya sea una incertidumbre fija en el tiempo listo, tal y como se describe a continuación.

a) *El periodo es numérico y el plazo es borroso en torno al periodo numérico.*

En este caso al ser el periodo numérico y el plazo borroso, periodo y plazo no coinciden, por lo que no se puede extender la ecuación (6.1) a parámetros borrosos, ya que sólo es válida cuando ambos coinciden. En [Terrier et al., 1995] los autores confunden ambos conceptos, tratando de definir un hiperperiodo a partir de los plazos borrosos en lugar de hacerlo a partir de los periodos numéricos, y no resuelven el análisis de planificabilidad en esa situación.

b) *El periodo  $t$  es numérico, y existe una incertidumbre fija  $R$  en torno al tiempo listo.*

En este caso, el periodo es numérico y existe incertidumbre acerca del instante de activación inicial de las tareas, y por tanto de igual forma en las sucesivas activaciones o instantes listos. La incertidumbre no aumenta al aumentar el número de ejecuciones, como ocurre al modelar los periodos como números borrosos.

Así, el instante listo en la primera ejecución es  $0 \cdot t + R$ , donde  $R$  es un número borroso con núcleo  $=\{0\}$  (ver Figura 6.5)

Y el instante listo en la  $n^a$  ejecución es  $(n-1) \cdot t + R$ , donde el primer sumando es numérico y el segundo es borroso.

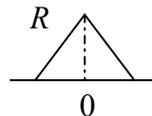


Figura 6.5: Número borroso  $R$  con núcleo  $=\{0\}$

En este caso, tanto los periodos como los plazos son numéricos y coinciden, y la incertidumbre aparece en todos los tiempos listos de las tareas. Recordemos que si no existe un instante crítico, es decir si no hay un instante en el que todas las tareas están listas a la vez, entonces no se puede aplicar la ecuación (6.1).

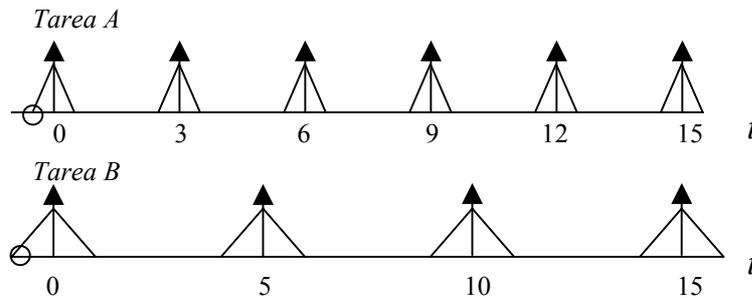


Figura 6.6: Dos tareas con tiempo listo borroso en torno a periodos numéricos

En la Figura 6.6 se muestran los tiempos listos borrosos de dos tareas con periodos numéricos  $t_A=3$  y  $t_B=5$  respectivamente. Los números borrosos  $R_1$  y  $R_2$  que modelan la incertidumbre en torno al periodo son distintos y aparecen dibujados. Si nos fijamos por ejemplo en los instantes en que están listas por primera vez las tareas  $A$  y  $B$  en el límite inferior del  $\alpha$ -corte con  $\alpha=0$  (que en la figura se han marcado con un círculo), se observa que no coinciden.

Por tanto, si se desea calcular la planificabilidad borrosa del sistema cuando los periodos y los plazos son numéricos y hay una incertidumbre  $R$  en torno a los periodos para modelar el tiempo listo, no se puede utilizar la fórmula (6.1). En este caso, se pueden obtener los tiempos de finalización borrosos mediante simulación y posteriormente compararlos con los plazos con el fin de calcular la planificabilidad del sistema. Si se calcula la planificabilidad por este camino, hay que tener en cuenta que el tiempo de finalización de una tarea y el plazo correspondiente (que coincide con el tiempo listo de la siguiente ejecución) no son independientes y por tanto no se pueden comparar ambos números borrosos como si lo fueran.

Siendo el tiempo de ejecución un número clásico, el tiempo de finalización resultante es un conjunto borroso debido a que el tiempo listo es borroso. En el caso más sencillo, para la tarea más prioritaria, en su primera ejecución, el tiempo de finalización es el tiempo listo más el tiempo de ejecución  $WC_1=R+p_1$ . Existe, por tanto una relación funcional entre el tiempo de finalización y el tiempo listo. El plazo con el que se compara ese tiempo de finalización también es borroso ya que este tiempo es el plazo relativo, que es numérico, más el tiempo listo inicial, que es borroso. Como el plazo relativo numérico es igual al periodo se tiene  $D_{1,1}=R+d_1=R+t_1$

Por tanto, para obtener las medidas de posibilidad y de necesidad de que el tiempo de finalización esté antes del plazo correspondiente hay que extender  $r+p_1 < r+t_1$  que, simplificando, es lo mismo que extender  $p_1 < t_1$ . Por lo que:

$$\begin{aligned} \Pi(WC_1 < D_{1,1}) &= \Pi(p_1 < t_1) \\ N(WC_1 < D_{1,1}) &= N(p_1 < t_1) \end{aligned}$$

que se reduce a una comparación entre números clásicos.

Una vez aclarados los dos casos anteriores, a continuación se estudia la extensión del análisis de planificabilidad a periodos borrosos.

Si el periodo  $T_i$  es borroso, al aumentar el número de ejecuciones aumenta la incertidumbre del tiempo listo correspondiente, tal y como se muestra en la Figura 6.7.

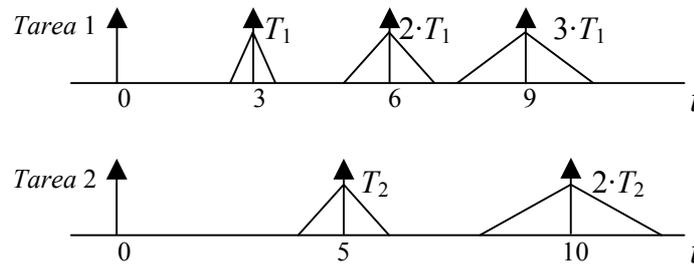


Figura 6.7: Dos tareas con periodos borrosos

Así, si existe un instante crítico se tiene:

Tiempo listo en la primera ejecución =  $0 \cdot T_i$

Tiempo listo en la  $n^a$  ejecución =  $(n-1) \cdot T_i$

Como ya se ha mencionado, la ecuación (6.1):

$$\sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} \leq 1$$

es equivalente a su expresión (6.2) en función del hiperperiodo  $h$ :

$$\sum_{i=1}^{n_{tar}} p_i \cdot \frac{h}{t_i} \leq h$$

Sin embargo, al extender el análisis a periodos borrosos, deja de existir un hiperperiodo definido de forma cierta y precisa. Por tanto es necesario estudiar el problema en detalle para determinar si es posible extender (6.1) a periodos borrosos.

El significado del hiperperiodo, cuando los periodos son borrosos, se puede ilustrar mediante el análisis de intervalos. Para cada límite de cada  $\alpha$ -corte de los periodos borrosos se obtiene una tupla de periodos numéricos, y se puede calcular el hiperperiodo numérico asociado a la tupla de periodos. Aunque los hiperperiodos pueden ser muy diferentes para las distintas tuplas, como se verá en el siguiente ejemplo, se puede aplicar para cada una de ellas la ecuación (6.2) y por tanto, simplificando, se puede aplicar a cada tupla la ecuación (6.1). Por lo que es equivalente a aplicar aritmética de intervalos a la extensión borrosa de (6.1), donde el 1 es un número clásico, a pesar de no existir un hiperperiodo cierto.

Sean dos tareas con periodos borrosos  $T_1=(2.9, 3, 3.1)$  y  $T_2=(4.9, 5, 5.1)$

Primero se analizan los periodos con  $\alpha=1$ , es decir,  $t_1=3$  y  $t_2=5$ . Para estos periodos, el hiperperiodo es  $h=15$ .

Los periodos asociados al límite superior del  $\alpha$ -corte con  $\alpha=0$  son  $t_1=3.1$  y  $t_2=5.1$ , que cambiando de escala con  $k=10$  se tiene  $k \cdot t_1=31$  y  $k \cdot t_2=51$ . El valor del hiperperiodo es  $h=31 \cdot 51=1581$ . En un hiperperiodo, la tarea 1 se ejecuta 51 veces (que es  $1581/31$ ) y la tarea 2 se ejecuta 31 veces (que es  $1581/51$ ).

Aplicando (6.2) y el cambio de escala  $k$ , se tiene:

$$p_1 \cdot (h \cdot k / k \cdot t_1) + p_2 \cdot (h \cdot k / k \cdot t_2) \leq h$$

$$p_1 \cdot 10 \cdot (1581/31) + p_2 \cdot 10 \cdot (1581/51) \leq 1581$$

$$p_1 \cdot 10 \cdot (51) + p_2 \cdot 10 \cdot (31) \leq 1581$$

Para los periodos asociados al límite inferior del  $\alpha$ -corte con  $\alpha=0$ ,  $t_1=2.9$  y  $t_2=4.9$ , el valor del hiperperiodo es  $h=29 \cdot 49=1421$ , aplicando de nuevo un cambio de escala  $k=10$ .

De forma similar, los periodos asociados al límite superior del  $\alpha$ -corte con  $\alpha=0.7$  son  $t_1=3.03$  y  $t_2=5.03$  que, cambiando de escala con  $k=100$ , se tiene  $k \cdot t_1=303$  y  $k \cdot t_2=503$ . El valor del hiperperiodo es  $h=303 \cdot 503=152409$ . La tarea 1 se ejecuta 503 veces en un hiperperiodo y la tarea 2 se ejecuta 303 veces.

Como se ve, para diferentes valores de los periodos, el hiperperiodo  $h$  varía mucho pero la aplicación de la ecuación (6.2) es correcta. Si se divide para cada pareja de periodos, los dos términos de  $p_1 \cdot (h \cdot k / k \cdot t_1) + p_2 \cdot (h \cdot k / k \cdot t_2) \leq h$  por  $h$ , y se simplifica el factor  $k$ , la ecuación resultante es (6.1) aplicada a los límites de los  $\alpha$ -cortes de los periodos borrosos  $T_i$ , es decir, se ha resuelto mediante aritmética de intervalos la ecuación borrosa:

$$(6.5) \quad \sum_{i=1}^{n\_tar} \frac{p_i}{T_i} \leq 1$$

donde el término de la derecha es numérico.

Se ha comprobado que a pesar de no existir un único hiperperiodo para el análisis del sistema (problema planteado y no resuelto por Terrier [Terrier et al., 1995]), se puede extender la condición de planificabilidad (6.1) a periodos borrosos.

En el mismo ejemplo de antes, con periodos borrosos  $T_1=(2.9, 3, 3.1)$  y  $T_2=(4.9, 5, 5.1)$ , si los tiempos de ejecución numéricos son  $p_1=1$  y  $p_2=3.4$ , aplicando (6.5) se tiene:

$$p_1/T_1 + p_2/T_2 = 1/(2.9, 3, 3.1) + 3.4/(4.9, 5, 5.1)$$

Y las medidas de posibilidad y necesidad que se obtienen son:

$$\Pi_{planif} = \Pi ( (p_1/T_1 + p_2/T_2) \leq 1 ) = 0.45 \quad N_{planif} = 0$$

Para ese valor de  $\alpha=0.45$  los periodos asociados al límite superior del  $\alpha$ -corte son  $t_1=3.055$  y  $t_2=5.055$ . Para dichos periodos la utilización numérica asociada del procesador es 1, el valor de  $h$  es  $h=(3055 \cdot 5055)/5=3088605$ , la tarea 1 se ejecuta 1011 veces en  $h$  y la tarea 2 se ejecuta 611 veces.

Si además de periodos borrosos los tiempos de ejecución también son borrosos, las medidas de posibilidad y de necesidad de que el sistema sea planificable son:

$$\Pi_{planif} = \Pi \left( \sum_{i=1}^{n\_tar} \frac{P_i}{T_i} \leq 1 \right) \quad N_{planif} = N \left( \sum_{i=1}^{n\_tar} \frac{P_i}{T_i} \leq 1 \right)$$

El resultado es el mismo que si se aplica el principio de extensión, aritmética de intervalos o aritmética borrosa ya que la función  $\sum_{i=1..n\_tar} (p_i / t_i)$  es continua, creciente en  $p_i$  y decreciente en  $t_i$ .

## 6.2. Plazos menores que los periodos

Cuando los plazos no coinciden con los periodos la ecuación (6.1) es una condición necesaria pero no suficiente de planificabilidad, y en su lugar el análisis se puede realizar usando un criterio de demanda del procesador [Buttazzo, 1997]. Veamos primero este enfoque en el caso de que los plazos sean iguales a los periodos.

La demanda de procesador de una tarea  $i$  en un intervalo  $[tm, tm+l]$  es el tiempo de procesamiento que la tarea  $i$  requiere en dicho intervalo y que debe completarse en el instante  $tm+l$  o antes. Con el algoritmo EDF, este es el tiempo de procesamiento requerido en  $[tm, tm+l]$  que debe ser ejecutado con plazos menores o iguales que  $tm+l$ .

Para un conjunto de tareas periódicas, con plazos iguales a los periodos, invocadas en el instante  $tm=0$ , la demanda de procesador acumulada en un intervalo  $[0,l]$  es la cantidad total de tiempo de procesamiento  $c_P(0,l)$  que se tiene que ejecutar con plazos menores o iguales a  $l$ . Esto es,

$$c_P(0,l) = \sum_{i=1}^n \left\lfloor \frac{l}{t_i} \right\rfloor \cdot p_i$$

donde el redondeo hacia abajo de un número  $x$  se expresa como  $\lfloor x \rfloor$ . El teorema de Jeffay y Stone afirma que la planificabilidad de un conjunto de tareas periódicas planificadas mediante EDF queda garantizada si y sólo si la demanda de procesador acumulada en un intervalo  $[0,l]$  es menor que el tiempo disponible, esto es, que la longitud del intervalo  $l$ . O lo que es lo mismo: *Un conjunto de tareas periódicas es planificable con EDF si y sólo si para todo  $l, l \geq 0$ ,*

$$(6.6) \quad l \geq \sum_{i=1}^n \left\lfloor \frac{l}{t_i} \right\rfloor \cdot p_i$$

Es suficiente con comprobar (6.6) para los valores de  $l$  iguales a los tiempos listos menores que el hiperperiodo  $h$ . Si se cumple la ecuación para los tiempos listos  $l=r_k$ , también se cumple para todo  $l \in [r_k, r_{k+1})$ , ya que se verifica:

$$\forall l \in [r_k, r_{k+1}), \quad \left\lfloor \frac{l}{t_i} \right\rfloor = \left\lfloor \frac{r_k}{t_i} \right\rfloor$$

donde  $r_k$  es el tiempo listo de una tarea en una ejecución cualquiera y  $r_{k+1}$  es el primer tiempo listo que se produce después de  $r_k$  ya sea de la misma tarea o de otra. Además, los valores de  $l$  para los que se debe comprobar la ecuación (6.6) se pueden

reducir a los tiempos listos que se encuentran dentro del periodo de ocupación. El *periodo de ocupación* es el menor intervalo  $[0, l^*]$  en el que el tiempo total de procesamiento  $w_p(l^*)$  requerido en  $[0, l^*]$  se ejecuta por completo, donde:

$$w_p(l^*) = \sum_{i=1}^n \left\lceil \frac{l^*}{t_i} \right\rceil \cdot p_i$$

El periodo de ocupación  $b_p$  se puede definir como:

$$b_p = \min \{ l^* \mid w_p(l^*) = l^* \}$$

El algoritmo para calcular el periodo de ocupación  $b_p$  es:

Periodo ocupado

$$l^* = \sum_{i=1}^n p_i$$

$$l' = w_p(l^*)$$

$$h = \text{mcm}(t_1, t_2, \dots, t_n)$$

while  $(l' \neq l^*)$  and  $(l' \leq h)$

{  $l^* = l'$

$l' = w_p(l^*)$  }

if  $(l' \leq h)$   $b_p = l^*$

else  $b_p = \infty$

(6.7)

Cuando el sistema está sobrecargado el procesador está siempre ocupado y el periodo de ocupación es infinito. Si el sistema no está sobrecargado, el periodo de ocupación termina al comenzar un tiempo de procesador disponible, o bien con el tiempo listo de una tarea periódica.

De acuerdo con lo anterior, para determinar la planificabilidad de un conjunto de tareas periódicas, se debe comprobar la ecuación (6.6) para todo  $l \in \mathcal{R}$ , donde:

$$\mathcal{R} = \{r_{i,j} \mid r_{i,j} \leq \min(b_p, h), 1 \leq i \leq n, j \geq 1\}$$

y donde  $r_{i,j}$  es el tiempo listo de la tarea  $i$  en su ejecución  $j$ .

Veamos a continuación la extensión al caso de tener plazos menores que los periodos. El enfoque de demanda del procesador queda expresado mediante el siguiente teorema ([Buttazzo, 1997]). Si  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ , entonces un conjunto de tareas periódicas con plazos menores que los periodos es planificable si y sólo si

$$(6.8) \forall l \in \mathcal{D} \quad l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

Ejemplo:

Sean tres tareas con periodos  $t_1=4$ ,  $t_2=6$  y  $t_3=7$ , tiempos de ejecución  $p_1=1$ ,  $p_2=2.5$  y  $p_3=2.3$ , y plazos  $d_1=4$ ,  $d_2=6$  y  $d_3=6$ . Nótese que para la tarea 3 el plazo no coincide con el periodo. Por tanto el que la utilización total sea menor o igual a uno es una condición necesaria pero no suficiente para que el sistema sea planificable.

En este caso se tiene:  $\sum_{i=1}^{n_{tar}} \frac{p_i}{t_i} = 0.995 \leq 1$

La Figura 6.8 muestra una parte de la planificación de las tareas, concretamente, hasta que la tarea 1 sobrepasa su plazo en el instante 48 finalizando su ejecución en 48.1. El hiperperiodo de este sistema es  $h=84$ . El periodo de ocupación, aplicando el algoritmo (6.7), es  $b_p=83.6$  y es menor que el hiperperiodo. Tal y como se ve en el ejemplo, el sobrepaso del plazo se puede producir en cualquier punto del periodo de ocupación. No se tiene por qué dar ni en la primera ejecución de las tareas ni al final del periodo de ocupación. El segundo término de la desigualdad (6.8), que corresponde a la demanda de procesador, a menudo coincide con el tiempo de finalización de la tarea en una determinada ejecución, pero no siempre es así.

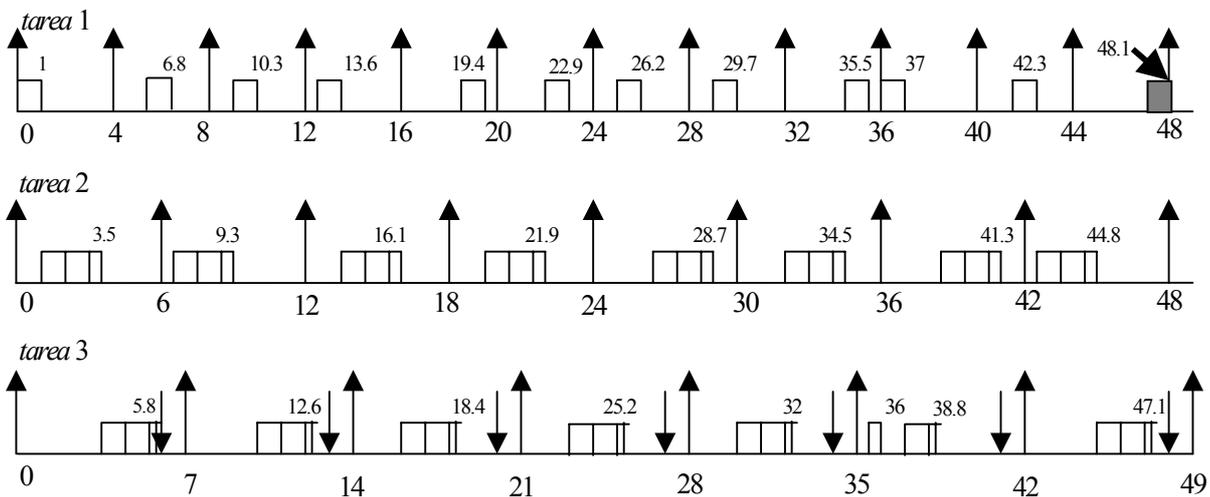


Figura 6.8: Planificación de tres tareas mediante EDF

Por ejemplo, comprobando la ecuación para  $l = d_{3,3} = 27$  se tiene:

$$27 \geq \sum_{i=1}^3 \left( \left\lfloor \frac{27 - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i = 25.2$$

En este caso, el segundo término de la desigualdad coincide con el tiempo de finalización de la tarea 3 en su cuarta ejecución, tal y como se ve en la Figura 6.8.

En cambio, para  $l = d_{1,9} = 40$  se obtiene:

$$40 \geq \sum_{i=1}^3 \left( \left\lfloor \frac{40 - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i = 36.5$$

que es distinto, el tiempo de finalización de la tarea 1 en su décima ejecución  $w_{c_{1,10}} = 37$ . Esta diferencia se debe a que no se contabiliza que la tarea 3 ha ejecutado una parte de su sexta ejecución, concretamente 0.5 unidades de tiempo.

### 6.2.1. Extensión del análisis de planificabilidad a tiempos de ejecución borrosos

El objetivo de este apartado es proporcionar medidas de posibilidad y de necesidad de planificabilidad del sistema extendiendo la desigualdad (6.8) a tiempos de ejecución borrosos mediante la aplicación de aritmética borrosa y aritmética de intervalos. Recordemos la desigualdad citada:

$$\forall l \in \mathcal{D} \quad l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

donde  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ .

Se puede observar que los valores de  $l$  dependen de los periodos y de los plazos, y son independientes de los tiempos de ejecución. Estos valores de  $l$  son los plazos absolutos de las tareas en las sucesivas ejecuciones, dentro del periodo de ocupación o del hiperperiodo y no son borrosos ya que en este caso sólo los tiempos de ejecución lo son.

El segundo término de la desigualdad, que es la demanda de procesador  $c_p$ , es una función creciente en los tiempos de ejecución  $p_i$ . Por tanto, aplicando la propiedad de

funciones monótonas (ver capítulo 4), es posible obtener los límites inferiores y superiores exactos de los  $\alpha$ -cortes de la demanda borrosa de procesador, como:

$$\underline{c}_p(\alpha) = f(\underline{p}_i(\alpha)) \quad \overline{c}_p(\alpha) = f(\overline{p}_i(\alpha))$$

$$\text{siendo } f(p_i) = \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

Una vez obtenida la demanda borrosa aproximada  $C_p$ , se compara con el valor de  $l$ , que es numérico. Como la aproximación borrosa obtenida es válida, los valores de posibilidad y necesidad de que  $l$  sea menor o igual que la demanda borrosa son exactos.

Así, para tiempos de ejecución borrosos, se puede extender el análisis de planificabilidad tanto mediante aritmética borrosa (6.9) como mediante aritmética de intervalos (6.10).

Mediante aritmética borrosa (6.9), se pueden calcular las medidas borrosas de planificabilidad extendiendo la desigualdad a tiempos de ejecución borrosos y tomando el mínimo de las medidas de posibilidad y de necesidad para cada valor que toma  $l$ :

$$(6.9) \quad \Pi_{planif} = \min \left\{ \Pi \left( l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot P_i \right) \right\}$$

$$N_{planif} = \min \left\{ N \left( l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot P_i \right) \right\}$$

$$\forall l \in \mathcal{D}, \forall 1 \leq i \leq n\_tarefas$$

Donde  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ , y  $b_p$  es el periodo de ocupación y  $h$  es el hiperperiodo. El periodo de ocupación se calcula aplicando (6.7).

Como ya se ha dicho, también es posible obtener las medidas borrosas exactas de planificabilidad aplicando aritmética de intervalos, y para acelerar el cálculo se puede aplicar la búsqueda por bipartición del punto de corte entre la demanda borrosa del procesador  $C_p$  y el valor numérico que toma  $l$ . El algoritmo basado en aritmética de intervalos con búsqueda por bipartición es:

	<p><b>ALG Aritmética de Intervalos</b></p> <p><math>\underline{wc}_0(0) = 0</math></p> <p><math>\forall l_{i,k} \in D</math></p> $\underline{c}_p(0) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(0)$ $\underline{c}_p(1) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(1)$ $\overline{c}_p(1) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(1)$ $\overline{c}_p(0) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(0)$ <p>ALG ZONA'</p> <p>end</p>
(6.10)	$\Pi_{planif} / N_{planif} = \min \{ \Pi(C_p(l_{i,k}) \leq l_{i,k}) / N(C_p(l_{i,k}) \leq l_{i,k}) \} \quad \forall i, k$

donde el algoritmo ZONA' (6.11) determina si la zona de búsqueda del punto de corte se encuentra en los límites superiores de los  $\alpha$ -cortes de  $C_p$ , o bien en los límites inferiores.

	<p><b>ALG ZONA'</b></p> <p>if (<math>l_{i,k} \geq \overline{c}_p(0)</math>) <math>\Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1</math>, <math>N(C_p(l_{i,k}) \leq l_{i,k}) = 1</math>,</p> <p>if (<math>l_{i,k} \leq \underline{c}_p(0)</math>) <math>\Pi(C_p(l_{i,k}) \leq l_{i,k}) = 0</math>, <math>N(C_p(l_{i,k}) \leq l_{i,k}) = 0</math>,</p> <p>if (<math>\overline{c}_p(1) \geq l_{i,k}</math>) <math>N(C_p(l_{i,k}) \leq l_{i,k}) = 0</math></p> <p style="padding-left: 20px;">else if (<math>\overline{c}_p(1) &lt; l_{i,k}</math>) Bipartición sup'</p> <p>if (<math>\underline{c}_p(1) \leq l_{i,k}</math>) <math>\Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1</math></p> <p style="padding-left: 20px;">else if (<math>\underline{c}_p(1) &gt; l_{i,k}</math>) Bipartición inf'</p>
(6.11)	

Si la zona de búsqueda se encuentra en los límites inferiores, la necesidad de  $C_p(l_{i,k}) \leq l_{i,k}$  es 0. El algoritmo Bipartición inf' (6.12) realiza la búsqueda por bipartición, con la precisión  $\varepsilon$  deseada, y el valor  $\alpha$  del punto de corte es la posibilidad de  $C_p(l_{i,k}) \leq l_{i,k}$ .

$$\begin{array}{l}
 \text{Bipartición inf'} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{c}_p(0), \text{cota\_sup} = \underline{c}_p(1) \\
 \text{while } \left( \left| \underline{c}_p(\alpha) - l_{i,k} \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \underline{c}_p(\alpha) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(\alpha) \\
 \text{if } \left( \underline{c}_p(\alpha) < l_{i,k} \right) \quad \alpha_0 = \alpha, \quad \text{cota\_inf} = \underline{c}_p(\alpha) \\
 \text{if } \left( \underline{c}_p(\alpha) > l_{i,k} \right) \quad \alpha_1 = \alpha, \quad \text{cota\_sup} = \underline{c}_p(\alpha) \\
 \text{end} \\
 \Pi(C_p(l_{i,k}) \leq l_{i,k}) = \alpha; \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 0
 \end{array}
 \tag{6.12}$$

De forma similar, si la zona de búsqueda se encuentra en los límites superiores la posibilidad de  $C_p(l_{i,k}) \leq l_{i,k}$  es cero, y el algoritmo Bipartición sup' (6.13) realiza la búsqueda del punto de corte por bipartición para calcular la necesidad de  $C_p(l_{i,k}) \leq l_{i,k}$ .

$$\begin{array}{l}
 \text{Bipartición sup'} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{c}_p(1), \text{cota\_sup} = \overline{c}_p(0) \\
 \text{while } \left( \left| \overline{c}_p(\alpha) - l_{i,k} \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \overline{c}_p(\alpha) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(\alpha) \\
 \text{if } \left( \overline{c}_p(\alpha) < l_{i,k} \right) \quad \alpha_1 = \alpha, \quad \text{cota\_inf} = \overline{c}_p(\alpha) \\
 \text{if } \left( \overline{c}_p(\alpha) > l_{i,k} \right) \quad \alpha_0 = \alpha, \quad \text{cota\_sup} = \overline{c}_p(\alpha) \\
 \text{end} \\
 \Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1; \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 1 - \alpha
 \end{array}
 \tag{6.13}$$

*Ejemplo:*

Recordemos el ejemplo ya planteado anteriormente con todos los parámetros numéricos, en el que el sistema está formado por tres tareas con periodos  $t_1=4$ ,  $t_2=6$  y  $t_3=7$ , tiempos de ejecución  $p_1=1$ ,  $p_2=2.5$  y  $p_3=2.3$ , y plazos  $d_1=4$ ,  $d_2=6$  y  $d_3=6$ .

El hiperperiodo es  $h=84$ , el periodo de ocupación es  $b_p=83.6$  y se calcula aplicando el algoritmo (6.7). Si se aplica (6.8) para determinar la planificabilidad del sistema,

se observa que la desigualdad no se cumple para  $l=48$ , ya que para ese valor se tiene  $c_p=48.1$ .

Si se modelan los tiempos de ejecución como números borrosos triangulares  $P_1=(0.8 \ 0.9 \ 1.1)$ ,  $P_2=(2.3 \ 2.4 \ 2.6)$  y  $P_3=(2.1 \ 2.2 \ 2.4)$ , podemos obtener los valores de posibilidad y necesidad de que el sistema sea planificable aplicando el algoritmo basado en aritmética de intervalos (6.10):

1. Se determina la zona de búsqueda por bipartición:

Para  $\overline{p}_i(0)$ ,  $U > 1$ , por tanto no es necesario calcular  $c_p$  ya que el sistema no es planificable para estos valores de los tiempos de ejecución.

Para  $\overline{p}_i(1)$ ,  $U \leq 1$ ,  $b_p=11.9$ , se cumple (6.8) para todo  $l$ . Por tanto el sistema es planificable.

2. De acuerdo con el algoritmo ZONA', el punto de corte entre  $C_p$  y  $l$  se encuentra en los límites superiores de los  $\alpha$ -cortes.

3. Bipartición:

Para  $\alpha=0.5$ ,  $\overline{p}_i(0.5)$ ,  $U \leq 1$ ,  $b_p=83.6$  y  $l=48$ . En este caso no se cumple (6.8) ya que para ese valor de  $l$  se tiene que  $c_p=48.1$ .

Se sigue aumentando sucesivamente el valor de  $\alpha$  hasta que para  $\alpha=0.5626$  se tiene que con  $\overline{p}_i(0.5+0.0625)$ ,  $U \leq 1$ ,  $b_p=47.93$ , se cumple (6.8) para todo  $l$ . Por tanto el sistema es planificable para los límites superiores de los  $\alpha$ -cortes con  $\alpha=0.5625$

Por el contrario, para  $\alpha=0.5+0.0625/2=0.53125$ , los tiempos de ejecución son  $\overline{p}_i(0.53125)$ , y no se cumple (6.8).

Si consideramos que la precisión conseguida en  $\alpha$  es suficiente, las medidas de planificabilidad borrosa son:

$$\Pi_{planif}=1 \qquad N_{planif}=1-0.5625=0.4375$$

### 6.2.2. Extensión del análisis de planificabilidad a tiempos de ejecución, periodos y plazos borrosos

En este apartado se va a analizar si es posible extender la condición de planificabilidad de este tipo de sistemas a periodos y plazos borrosos, además de los tiempos de ejecución, aplicando aritmética borrosa y aplicando aritmética de intervalos. Para ello, se va a comprobar si las aproximaciones de la demanda borrosa

del procesador obtenidas por ambos métodos son aproximaciones válidas y si el resultado de comparar la demanda borrosa con  $L$  es exacto.

Analizaremos en primer lugar si la demanda de procesador es monótona con los plazos y los periodos. Sea  $l = k \cdot t_j + d_j$ . La demanda del procesador para ese valor de  $l$  es:

$$c_p(t_j, d_j) = \sum_{i=1}^n \left( \left\lfloor \frac{k \cdot t_j + d_j - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

Si  $i=j$  entonces  $\left( \left\lfloor \frac{k \cdot t_j + d_j - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i = (k+1) \cdot p_i$

y este sumando de la demanda del procesador es independiente del periodo  $t_j$  y del plazo  $d_j$

Por el contrario, si  $i \neq j$  entonces  $\left( \left\lfloor \frac{k \cdot t_j + d_j - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i = g(t_j^+, d_j^+, t_i^-, d_i^-, p_i^+)$

donde  $g(t_j^+, d_j^+, t_i^-, d_i^-, p_i^+)$  indica que este sumando es una función no decreciente con  $t_j, d_j$  y  $p_i$  y no creciente con  $t_i$  y  $d_i$ .

Por tanto, para todo  $i, j$  se tiene  $c_p = g'(t_j^+, d_j^+, t_i^-, d_i^-, p_i^+)$

Por otro lado, se observa de manera inmediata que  $l = f(t_j^+, d_j^+)$ , esto es,  $l$  es no decreciente con  $t_j$  y  $d_j$ , ya que:

$$l \in \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$$

El valor borroso  $L$  para  $t_j$  y  $d_j$  borrosos se calcula de forma exacta mediante aritmética borrosa o mediante aritmética de intervalos, ya que se trata de una función continua y monótona.

Por otro lado, se pueden calcular aproximaciones borrosas de  $c_p$  con los límites exactos de los  $\alpha$ -cortes mediante aritmética de intervalos, o bien aplicando aritmética borrosa, dado que se verifica la propiedad de las funciones monótonas:

$$\begin{aligned} \underline{c}_p(\alpha) &= g'(\underline{t}_j(\alpha), \underline{d}_j(\alpha), \bar{t}_i(\alpha), \bar{d}_i(\alpha), \underline{p}_i(\alpha)) \\ \bar{c}_p(\alpha) &= g'(\bar{t}_j(\alpha), \bar{d}_j(\alpha), \underline{t}_i(\alpha), \underline{d}_i(\alpha), \bar{p}_i(\alpha)) \end{aligned}$$

Por tanto es posible extender a periodos y plazos borrosos los dos términos de la desigualdad:

$$\forall l \in \mathcal{D} \quad l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

donde  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ .

El paso siguiente es la comparación de los conjuntos borrosos  $L$  y la aproximación válida borrosa  $C_p$ . Los resultados de posibilidad y necesidad que se obtienen de la comparación entre  $L$  y la demanda del procesador borrosa exacta coinciden con los que se obtienen si se sustituye la demanda exacta por una aproximación válida, siempre que no exista relación funcional entre los conjuntos que se están comparando. Sin embargo, en este caso existe una relación funcional entre  $l$  y  $c_p$  ya que ambos dependen de  $t_j$  y  $d_j$ . Es preciso tener en cuenta dicha relación ya que los valores de posibilidad y necesidad que se obtienen al compararlos como si fueran independientes no coinciden con la aplicación del principio de extensión.

En el siguiente ejemplo se muestra que, a pesar de tener aproximaciones válidas de la demanda del procesador, las medidas borrosas resultantes de comparar dichos conjuntos con  $L$ , olvidando la relación funcional existente entre ellos, no coinciden con los resultados que proporciona el principio de extensión.

*Ejemplo:*

Sea un sistema formado por dos tareas y cuyo periodo de ocupación es tal que  $\mathcal{D} = \{d_{1,0} = d_1, d_{2,0} = d_2\}$ , es decir, sólo se analiza la primera ejecución de cada tarea. Además, sólo se produce una ejecución de cada tarea en dicho periodo, con lo que  $\lfloor (l-d_i) / t_i \rfloor < 1$ . En este caso  $c_p$  coincide con el tiempo de finalización de la tarea.

Si  $d_1 < d_2$  entonces para  $l = d_1$ , se tiene  $c_p = p_1 = wc_1$ , y para  $l = d_2$ ,  $c_p = p_1 + p_2 = wc_2$ . Es decir, si  $d_1$  es menor que  $d_2$  entonces la tarea 1 se ejecuta antes y su tiempo de finalización es  $wc_1 = p_1$  que coincide con la demanda de procesador para  $l = d_1$ . En este caso la tarea 2 se ejecuta después de la tarea 1 y su tiempo de finalización es  $wc_2 = p_1 + p_2$  que coincide con la demanda de procesador para  $l = d_2$ .

Si  $d_1 > d_2$  entonces para  $l = d_2$ , se tiene  $c_p = p_2 = wc_2$ , y para  $l = d_1$ ,  $c_p = p_1 + p_2 = wc_1$ . Es decir, si  $d_2$  es menor que  $d_1$  entonces la tarea 2 se ejecuta antes y su tiempo de finalización

es  $wc_2 = p_2$  que coincide con la demanda de procesador para  $l=d_2$ . La tarea 1 se ejecuta después de la tarea 1 y su tiempo de finalización es  $wc_1 = p_1+p_2$  que coincide con la demanda de procesador para  $l=d_1$ .

Para que el sistema sea planificable se tiene que cumplir que  $l \geq c_p$  para  $l=d_1$  y  $l=d_2$ . O lo que es lo mismo se debe verificar:

$$(6.14) \quad \{ (d_1 < d_2) \text{ AND } (p_1 \leq d_1) \text{ AND } (p_1+p_2 \leq d_2) \} \text{ OR} \\ \text{OR } \{ (d_2 < d_1) \text{ AND } (p_2 \leq d_2) \text{ AND } (p_1+p_2 \leq d_1) \}$$

Si se extiende el análisis a parámetros borrosos, la posibilidad de que el sistema sea planificable se puede obtener aplicando el principio de extensión de la siguiente forma:

$$\Pi_{planif} = \sup \left\{ \sup_{\substack{d_1 < d_2 \\ p_1 \leq d_1 \\ p_1+p_2 \leq d_2}} \min \left( \mu_{P_1}(p_1), \mu_{P_1+P_2}(p_1+p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2) \right), \right. \\ \left. \sup_{\substack{d_1 > d_2 \\ p_1+p_2 \leq d_1 \\ p_2 \leq d_2}} \min \left( \mu_{P_1+P_2}(p_1+p_2), \mu_{P_2}(p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2) \right) \right\}$$

O lo que es lo mismo:

$$(6.15) \quad \Pi_{planif} = \sup \left\{ \min_{\substack{d_1 < d_2 \\ p_1 \leq d_1 \\ p_1+p_2 \leq d_2}} \left( \mu_{P_1}(p_1), \mu_{P_1+P_2}(p_1+p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2) \right), \right. \\ \left. \min_{\substack{d_1 > d_2 \\ p_1+p_2 \leq d_1 \\ p_2 \leq d_2}} \left( \mu_{P_1+P_2}(p_1+p_2), \mu_{P_2}(p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2) \right) \right\}$$

La necesidad de que el sistema sea planificable, se puede expresar como uno menos la posibilidad de que no sea planificable, y para que el sistema no sea planificable basta con que no se cumpla  $l \geq c_p$  para alguno de los valores de  $l$ ,  $l=d_1$ , o bien  $l=d_2$ . Es decir, para que el sistema no sea planificable se debe verificar:

$$(6.16) \quad \{ [(p_1 > d_1) \text{ AND } (d_1 < d_2)] \text{ OR } [(p_1+p_2 > d_1) \text{ AND } (d_1 > d_2)] \} \\ \text{OR } \{ [(p_2 > d_2) \text{ AND } (d_2 < d_1)] \text{ OR } [(p_1+p_2 > d_2) \text{ AND } (d_2 > d_1)] \}$$

Extendiendo la expresión anterior a parámetros borrosos, la posibilidad de que el sistema no sea planificable es:

$$\begin{aligned} \Pi_{Nplanif} = \sup \{ & \sup_{\substack{d_1 < d_2 \\ p_1 > d_1}} \min(\mu_{P_1}(p_1), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \sup_{\substack{d_1 > d_2 \\ p_1 + p_2 > d_1}} \min(\mu_{P_1+P_2}(p_1 + p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \sup_{\substack{d_1 > d_2 \\ p_2 > d_2}} \min(\mu_{P_2}(p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \sup_{\substack{d_1 < d_2 \\ p_1 + p_2 > d_2}} \min(\mu_{P_1+P_2}(p_1 + p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)) \} \end{aligned}$$

Con lo que la necesidad de que el sistema sea planificable es uno menos dicha posibilidad, esto es:

$$\begin{aligned} N_{planif} = 1 - \Pi_{Nplanif} = 1 - \sup \{ & \min_{\substack{d_1 < d_2 \\ p_1 > d_1}}(\mu_{P_1}(p_1), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \min_{\substack{d_1 > d_2 \\ p_1 + p_2 > d_1}}(\mu_{P_1+P_2}(p_1 + p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \min_{\substack{d_1 > d_2 \\ p_2 > d_2}}(\mu_{P_2}(p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)), \\ & \min_{\substack{d_1 < d_2 \\ p_1 + p_2 > d_2}}(\mu_{P_1+P_2}(p_1 + p_2), \mu_{D_1}(d_1), \mu_{D_2}(d_2)) \} \end{aligned}$$

En el ejemplo de la Figura 6.9 se puede ver que la posibilidad de que  $d_2$  sea mayor que  $d_1$  es  $\alpha_1$ .

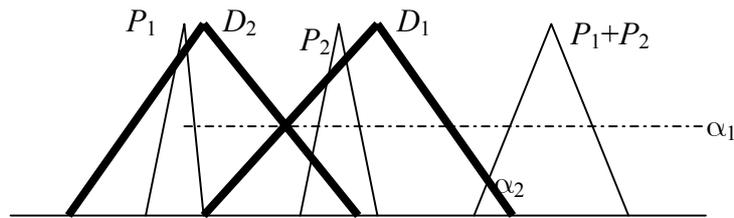


Figura 6.9: Ejemplo de tiempos de finalización de dos tareas y sus plazos

A continuación se muestra por separado la comparación con cada valor de  $l$ , esto es, con cada plazo. Se observa en la Figura 6.10 que la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 \leq D_1)]$  es  $\alpha_1$ , y la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_1 + P_2 \leq D_1)]$  es  $\alpha_2$ .

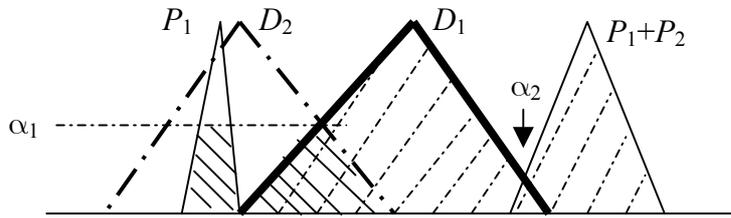


Figura 6.10: Tiempo de finalización de la tarea 1 y comparación con su plazo

Por otro lado, comparando con el otro plazo  $D_2$ , la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 + P_2 \leq D_2)]$  es 0, y la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_2 \leq D_2)]$  es  $\alpha_3$ , tal y como se muestra en la Figura 6.11.

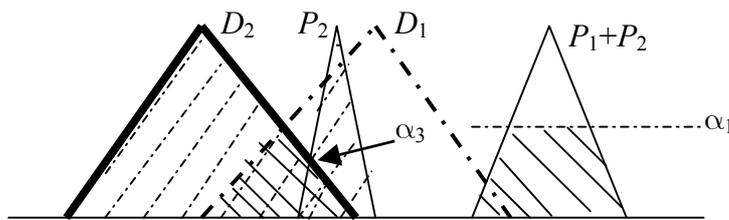


Figura 6.11: Tiempo de finalización de la tarea 2 y comparación con su plazo

Para obtener la posibilidad de que el sistema sea planificable, estudiamos en primer lugar el primer término de (6.14):

$$(d_1 < d_2) \text{ AND } (p_1 \leq d_1) \text{ AND } (p_1 + p_2 \leq d_2)$$

La posibilidad de ese término es el mínimo de la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 \leq D_1)]$  y de la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 + P_2 \leq D_2)]$ . Esto es, el mínimo de  $\alpha_1$  y 0, que es igual a 0:

$$\begin{aligned} & \Pi \{ (D_1 < D_2) \text{ AND } (P_1 \leq D_1) \text{ AND } (P_1 + P_2 \leq D_2) \} \\ &= \min ( \Pi \{ (D_1 < D_2) \text{ AND } (P_1 \leq D_1) \}, \Pi \{ (D_1 < D_2) \text{ AND } (P_1 + P_2 \leq D_2) \} ) \\ &= \min (\alpha_1, 0) = 0 \end{aligned}$$

La posibilidad del segundo término de (6.14):

$$(d_2 < d_1) \text{ AND } (p_2 \leq d_2) \text{ AND } (p_1 + p_2 \leq d_1)$$

es el mínimo de la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_1 + P_2 \leq D_1)]$  y de la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_2 \leq D_2)]$ . Esto es, el mínimo de  $\alpha_2$  y  $\alpha_3$ , que es  $\alpha_2$ :

$$\begin{aligned} & \Pi \{ (D_2 < D_1) \text{ AND } (P_2 \leq D_2) \text{ AND } (P_1 + P_2 \leq D_1) \} \\ &= \min ( \Pi \{ (D_2 < D_1) \text{ AND } (P_1 + P_2 \leq D_1) \}, \Pi \{ (D_2 < D_1) \text{ AND } (P_2 \leq D_2) \} ) \\ &= \min (\alpha_2, \alpha_3) = \alpha_2 \end{aligned}$$

La posibilidad de que el sistema sea planificable es el supremo de las posibilidades de ambos términos (6.15), esto es el supremo de 0 y  $\alpha_2$ , que es  $\alpha_2$ .

$$\Pi_{planif} = \alpha_2$$

Sin embargo, si se calcula, por un lado el conjunto borroso  $C_p$  correspondiente a  $L=D_1$  y por otro lado el conjunto borroso  $C_p$  correspondiente a  $L=D_2$ , y se compara cada demanda borrosa con el valor de  $L$  correspondiente como si fueran independientes (6.17), los resultados de planificabilidad borrosa que se obtienen no coinciden con los del principio de extensión que se acaban de calcular.

$$(6.17) \Pi \{ D_1 \geq WC_1 \text{ AND } D_2 \geq WC_2 \}$$

El conjunto borroso  $C_p$  correspondiente a  $L=D_1$  es el tiempo de finalización borroso de la tarea 1  $WC_1$ . Cuando  $d_1$  es menor que  $d_2$ , ese tiempo de finalización es el tiempo borroso de ejecución de la tarea 1, que es  $P_1$ , pero truncado a la altura de  $\alpha_1$ , que es la posibilidad de  $D_1 < D_2$  (ver Figura 6.10). Cuando  $d_1$  es mayor que  $d_2$ , el tiempo de finalización  $WC_1$  es el tiempo borroso de ejecución de la tarea 1 más el de la tarea 2 (ver Figura 6.10), esto es  $P_1 + P_2$  (sin truncar, porque la posibilidad de  $D_1 > D_2$  es 1). En la Figura 6.12 se muestra el tiempo borroso de finalización  $WC_1$  resultante y su plazo correspondiente.

Comparando  $L=D_1$  con el conjunto borroso  $C_p = WC_1$  se obtiene, como se ve en la Figura 6.12, que  $\Pi(L \geq C_p) = \Pi(D_1 \geq WC_1) = \alpha_1$ .

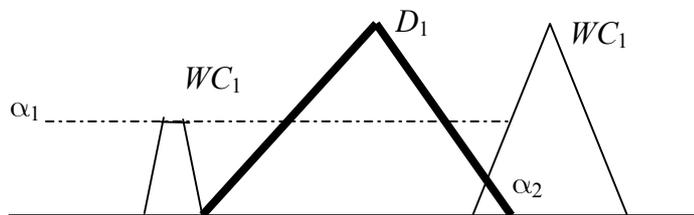


Figura 6.12: Aproximación: tiempo de finalización de la tarea 1 y comparación con su plazo

Por otro lado, el conjunto borroso  $C_p$  correspondiente a  $L=D_2$  es el tiempo de finalización borroso de la tarea 2  $WC_2$ . Cuando  $d_1$  es menor que  $d_2$ , ese tiempo de finalización es el tiempo borroso de ejecución de la tarea 2, que es  $P_1+P_2$ , pero truncado a la altura de  $\alpha_1$ , que es la posibilidad de  $D_1 < D_2$  (ver Figura 6.11). Cuando  $d_1$  es mayor que  $d_2$ , el tiempo de finalización  $WC_2$  es el tiempo borroso de ejecución  $P_2$  de la tarea 2 (ver Figura 6.11), sin truncar, ya que la posibilidad de  $D_1 > D_2$  es 1.

Se puede observar en la Figura 6.13 la comparación entre el tiempo borroso de finalización  $WC_2$  resultante y su plazo correspondiente, de forma que con  $L=D_2$ , se tiene  $\Pi(L \geq C_p) = \Pi(D_2 \geq WC_2) = \alpha_3$ .

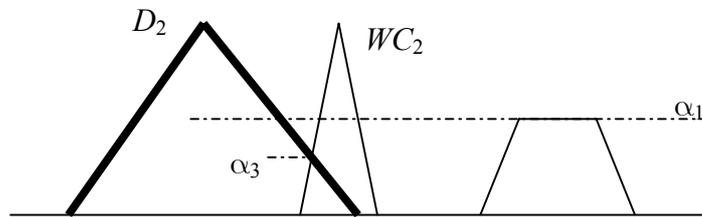


Figura 6.13: Aproximación: tiempo de finalización de la tarea 2 y comparación con su plazo

Como se debe cumplir la desigualdad para todos los valores de  $L$ , para que sea planificable se debe cumplir que  $D_1 \geq WC_1$  AND  $D_2 \geq WC_2$ , por lo que la posibilidad de que sea planificable (6.17) es:

$$\Pi_{planif} = \Pi( D_1 \geq WC_1 \text{ AND } D_2 \geq WC_2 ) = \min( \Pi(D_1 \geq WC_1), \Pi(D_2 \geq WC_2) ) = \min(\alpha_1, \alpha_3) = \alpha_3.$$

Este resultado no es exacto, ya que no coincide con el obtenido mediante el principio de extensión,  $\Pi_{planif} = \alpha_2$ .

En resumen, los resultados obtenidos por un lado mediante el principio de extensión y por otro suponiendo independencia entre  $l$  y  $c_p$  son:

Principio de extensión	Independencia
$\Pi_{planif} = \alpha_2$ $N_{planif} = 0$	$\Pi_{planif} = \alpha_3$ $N_{planif} = 0$

Se puede observar que en este caso al suponer independencia se obtiene un resultado de planificabilidad borrosa más optimista que el que se obtiene al aplicar el principio de extensión.

Veamos a continuación otro ejemplo, donde la posibilidad resultante es 1 y la necesidad es distinta de cero. Igual que en el ejemplo anterior, se observa en la Figura 6.14 que la posibilidad de que  $D_2$  sea mayor que  $D_1$  es  $\alpha_1$ .

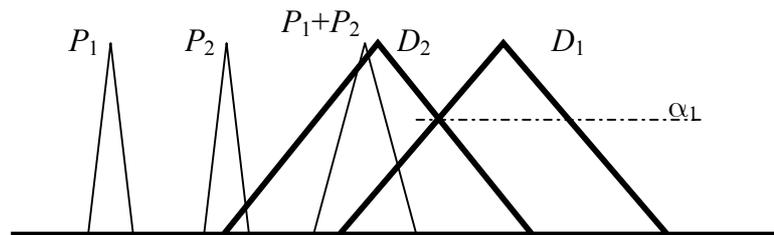


Figura 6.14: Ejemplo de tiempos de finalización de dos tareas y sus plazos

A continuación se muestra por separado cada caso de  $l$ , esto es, la comparación con cada plazo. En la Figura 6.15 se puede ver que la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 \leq D_1)]$  es  $\alpha_1$ , y que la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_1+P_2 \leq D_1)]$  es 1. Además, se puede ver que la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1 > D_1)]$  es 0. Y la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_1+P_2 > D_1)]$  es  $\alpha_2$ .

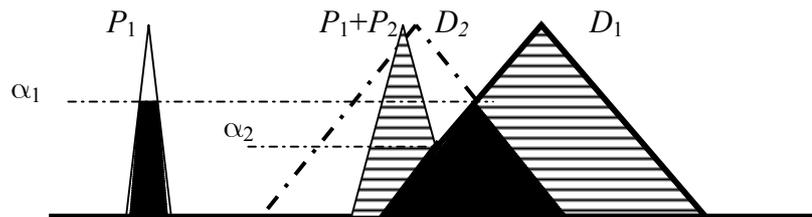


Figura 6.15: Tiempo de finalización de la tarea 1 y comparación con su plazo

Por otro lado, comparando con el otro plazo  $D_2$ , se observa en la Figura 6.16 que la posibilidad  $\Pi[(D_1 < D_2) \text{ AND } (P_1+P_2 \leq D_2)]$  es  $\alpha_1$ , y que la posibilidad  $\Pi[(D_2 < D_1) \text{ AND } (P_2 \leq D_2)]$  es 1. Además, se puede ver que la posibilidad  $\Pi[(D_1 > D_2) \text{ AND } (P_2 > D_2)]$  es  $\alpha_3$ . Y la posibilidad  $\Pi[(D_2 > D_1) \text{ AND } (P_1+P_2 > D_2)]$  es  $\alpha_2$ .

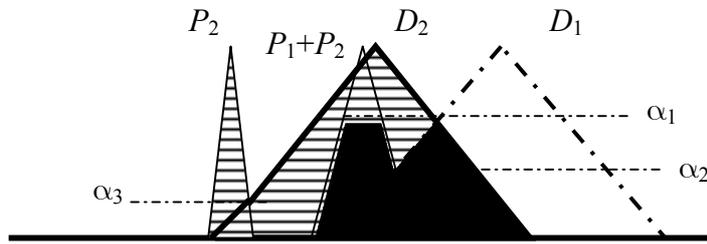


Figura 6.16: Tiempo de finalización de la tarea 2 y comparación con su plazo

En el cálculo de la posibilidad de que el sistema sea planificable, la posibilidad del primer término de (6.14),  $(d_1 < d_2) \text{ AND } (p_1 \leq d_1) \text{ AND } (p_1 + p_2 \leq d_2)$  es el mínimo de la posibilidad  $\Pi [(D_1 < D_2) \text{ AND } (P_1 \leq D_1)]$  y de la posibilidad  $\Pi [(D_1 < D_2) \text{ AND } (P_1 + P_2 \leq D_2)]$ . Esto es, el mínimo de  $\alpha_1$  y  $\alpha_1$ , que es  $\alpha_1$ :

$$\begin{aligned} & \Pi \{ (D_1 < D_2) \text{ AND } (P_1 \leq D_1) \text{ AND } (P_1 + P_2 \leq D_2) \} = \\ & = \min (\Pi \{ (D_1 < D_2) \text{ AND } (P_1 \leq D_1) \}, \Pi \{ (D_1 < D_2) \text{ AND } (P_1 + P_2 \leq D_2) \}) \\ & = \min(\alpha_1, \alpha_1) = \alpha_1 \end{aligned}$$

La posibilidad del segundo término de (6.14),  $(d_2 < d_1) \text{ AND } (p_2 \leq d_2) \text{ AND } (p_1 + p_2 \leq d_1)$  es el mínimo de la posibilidad  $\Pi [(D_2 < D_1) \text{ AND } (P_1 + P_2 \leq D_1)]$  y de la posibilidad  $\Pi [(D_2 < D_1) \text{ AND } (P_2 \leq D_2)]$ . Esto es, el mínimo de 1 y 1, que es 1:

$$\begin{aligned} & \Pi \{ (D_2 < D_1) \text{ AND } (P_2 \leq D_2) \text{ AND } (P_1 + P_2 \leq D_1) \} = \\ & = \min (\Pi \{ (D_2 < D_1) \text{ AND } (P_1 + P_2 \leq D_1) \}, \Pi \{ (D_2 < D_1) \text{ AND } (P_2 \leq D_2) \}) \\ & = \min(1, 1) = 1 \end{aligned}$$

La posibilidad de que el sistema sea planificable es el supremo de las posibilidades de ambos términos (6.15), esto es, el supremo de  $\alpha_1$  y 1, que es 1.

$$\Pi_{planif} = 1$$

La necesidad de que sea planificable es uno menos la posibilidad de que no sea planificable (6.16), esto es, uno menos la posibilidad de:

$$\{ [(P_1 > D_1) \text{ AND } (D_1 < D_2)] \text{ OR } [(P_1 + P_2 > D_1) \text{ AND } (D_1 > D_2)] \} \text{ OR } \\ \text{OR } \{ [(P_2 > D_2) \text{ AND } (D_2 < D_1)] \text{ OR } [(P_1 + P_2 > D_2) \text{ AND } (D_2 > D_1)] \}$$

Esto es:

$$N_{planif} = 1 - \sup\{0, \alpha_2, \alpha_3, \alpha_2\} = 1 - \alpha_2$$

Veamos el resultado que se habría obtenido de no tener en cuenta la dependencia entre  $C_p$  y  $L$ . Comparando  $L=D_1$  con el conjunto borroso  $C_p$  correspondiente (que es el tiempo de finalización de la tarea 1), se observa en la Figura 6.17 que  $\Pi(L \geq C_p) = \Pi(D_1 \geq WC_1) = 1$  y  $\Pi(L < C_p) = \Pi(D_1 < WC_1) = \alpha_2$ .

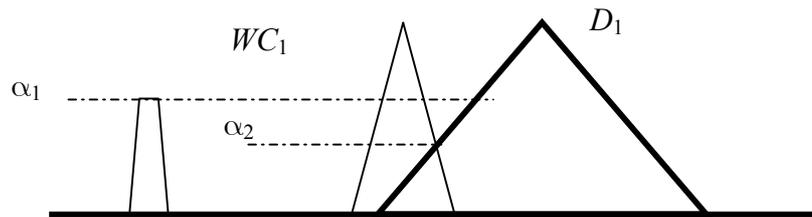


Figura 6.17: Aproximación: tiempo de finalización de la tarea 1 y comparación con su plazo

En la Figura 6.18 se compara el tiempo de finalización con el otro valor de  $L$ ,  $L=D_2$ , y se tiene que  $\Pi(L \geq C_p) = \Pi(D_2 \geq WC_2) = 1$  y  $\Pi(L < C_p) = \Pi(D_2 < WC_2) = \alpha_1$

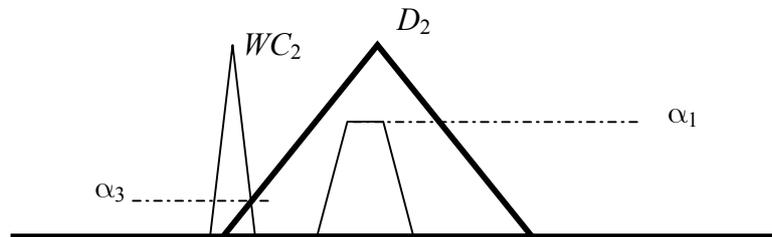


Figura 6.18: Aproximación: tiempo de finalización de la tarea 2 y comparación con su plazo

Por tanto, si no se tiene en cuenta la relación funcional existente entre la demanda del procesador y  $l$ , se tiene  $\Pi_{planif} = 1$  y  $N_{planif} = 1 - \sup\{\alpha_2, \alpha_1\} = 1 - \alpha_1$ . Estos resultados no coinciden con los que se obtienen con el principio de extensión.

En resumen, los resultados de planificabilidad borrosa obtenidos mediante el principio de extensión, y por otro lado sin tener en cuenta la relación funcional entre  $c_p$  y  $l$ , son:

Principio de extensión	Independencia
$\Pi_{planif} = 1$ $N_{planif} = 1 - \alpha_2$	$\Pi_{planif} = 1$ $N_{planif} = 1 - \alpha_1$

Se puede observar que en este caso, al suponer independencia, se obtiene un resultado de planificabilidad borrosa más pesimista que el que se obtiene al aplicar el principio de extensión.

En los ejemplos anteriores se ha mostrado que al existir dependencia entre los tiempos de finalización y los plazos no es posible calcular  $C_p$  y  $L$ , y hacer la comparación como si fueran independientes. La comparación entre ambos conjuntos borrosos no es válida para el cálculo de planificabilidad. Por tanto, no es posible extender el algoritmo basado en aritmética borrosa o en aritmética de intervalos cuando los plazos y los periodos son borrosos debido a la dependencia funcional existente entre  $c_p$  y  $l$ . Sería necesario aplicar el principio de extensión o bien encontrar algún otro procedimiento de cálculo que obtuviera los mismos resultados.

### 6.2.3. Posibilidad y necesidad total cuando existe orden fuerte en los plazos

Tal y como se ha visto antes, la posibilidad y la necesidad de que el sistema sea planificable depende del orden de los plazos absolutos, esto es, de la prioridad asignada a cada tarea en cada momento. Así, se tiene en cuenta que si el plazo 1 es menor que el plazo 2, la tarea 1 es más prioritaria. En tal caso el valor de  $C_p$  depende de dicha asignación de prioridades y se ve afectado por el valor de posibilidad de que se verifique  $D_1 < D_2$ . También se tiene en cuenta que si el plazo 1 es mayor que el plazo 2, la tarea 2 es más prioritaria y el valor de  $C_p$  depende de dicha asignación de prioridades y se ve afectado por el valor de posibilidad de que se verifique  $D_1 > D_2$ . Podemos denominar posibilidad y necesidad total a aquellas medidas que tienen en cuenta todos los órdenes posibles entre los plazos.

En el capítulo anterior, al analizar la planificabilidad de un sistema con prioridades fijas, se proponían medidas de posibilidad y de necesidad de planificabilidad dada una única asignación de prioridades fijas de las tareas. Si los plazos eran borrosos, se

suponía que se elegía una asignación y mediante los algoritmos propuestos se proporcionaban medidas de planificabilidad borrosa para ese sistema con dicha asignación. Si existía un orden fuerte entre los plazos, se proponía elegir la asignación dada por ese orden fuerte.

En lugar de esas medidas de posibilidad y de necesidad, se podrían haber dado las medidas de posibilidad y necesidad total, teniendo en cuenta todos los órdenes posibles entre los plazos, esto es, todas las asignaciones de prioridades fijas posibles, en lugar de una determinada (como por ejemplo, la asignación más posible).

Así, en el ejemplo de la Figura 6.19, habría que dar la posibilidad y la necesidad de planificabilidad, en lugar de por ejemplo para la asignación dada por el orden fuerte (tarea 2 más prioritaria), para todas las asignaciones conjuntamente, dado que cada asignación tiene un valor de posibilidad asociado.

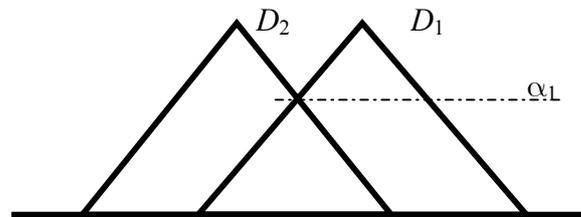


Figura 6.19: Plazos borrosos de dos tareas

Es decir, en el análisis de planificabilidad de DMS con plazos borrosos se podría tener en cuenta que el tiempo de finalización de las tareas depende de la posibilidad de las distintas asignaciones. Si un sistema está formado por dos tareas con plazos borrosos  $D_1$  y  $D_2$ , y tanto la posibilidad de  $D_1 < D_2$ , como la posibilidad de  $D_1 > D_2$  son distintas de cero, se pueden dar dos asignaciones de prioridades diferentes y, dependiendo de la asignación, el tiempo de finalización borroso de cada tarea es diferente.

Los dos ejemplos anteriores de planificación con EDF se podrían corresponder también al análisis de un sistema formado por dos tareas, planificadas mediante DMS, con plazos borrosos. Los valores de  $C_p$  analizados serían los tiempos de finalización de las dos tareas en su primera ejecución, que son los peores tiempos de respuesta cuando las prioridades son fijas y existe un instante crítico. Los valores de  $L$  analizados en dichos ejemplos serían los plazos borrosos de ambas tareas. Por tanto, la posibilidad y necesidad de que el sistema sea planificable con DMS teniendo en cuenta todas las asignaciones posibles (posibilidad y necesidad total) no se pueden calcular sin tener en cuenta la dependencia de los tiempos de finalización

con los plazos, con lo que al obtener aproximaciones borrosas de los tiempos de finalización y compararlas posteriormente con los correspondientes plazos, los resultados de planificabilidad borrosa no son exactos.

Para DMS se probó, y para EDF se indicó, que si existe orden fuerte entre los plazos, la asignación de prioridades dada por ese orden fuerte es la óptima (ver capítulo 5). Basado en esta afirmación se puede probar que si existe orden fuerte entre los plazos, la posibilidad y necesidad total de que el sistema sea planificable coincide con la posibilidad y necesidad que se obtiene para la asignación óptima.

Esto significa que si existe orden fuerte entre los plazos no es preciso tener en cuenta todas las asignaciones de prioridades posibles sino únicamente la correspondiente al orden fuerte, lo que simplifica los cálculos.

#### *Posibilidad total cuando existe orden fuerte entre los plazos*

Generalizando (6.14), la posibilidad total de que un sistema formado por  $n$  tareas sea planificable se calcula como la posibilidad:

$$(6.18) \Pi \{ \begin{aligned} & [ (o^1) \text{ AND } (WC_1^1 < D_1) \text{ AND } (WC_2^1 < D_2) \text{ AND } \dots (WC_n^1 < D_n) ] \text{ OR} \\ & \text{OR } [ (o^2) \text{ AND } (WC_1^2 < D_1) \text{ AND } (WC_2^2 < D_2) \text{ AND } \dots (WC_n^2 < D_n) ] \text{ OR} \\ & \dots \\ & \dots \text{ OR } [ (o^{np}) \text{ AND } (WC_1^{np} < D_1) \text{ AND } (WC_2^{np} < D_2) \text{ AND } \dots (WC_n^{np} < D_n) ] \end{aligned} \}$$

Donde  $wc_i^k$  es el tiempo de finalización de la tarea  $i$  dada una asignación de prioridades  $k$ , con  $i=1..n$ . Además  $o^k$  es un orden determinado de los plazos, por ejemplo  $d_1 < d_2 < d_3 < \dots < d_n$ , con  $k=1..np$  siendo  $np$  el número de permutaciones posibles en la asignación de prioridades, es decir  $np = \text{Perm}(n)$ .

Por tanto, la posibilidad total es el supremo de las posibilidades de los  $np$  términos de (6.18) asociados a los distintos órdenes  $(o^1) \dots (o^{np})$ .

Supongamos que para  $k=f$  existe orden fuerte entre los plazos borrosos y dicho orden viene dado por  $o^f$ .

Por la definición de óptimo dada cuando hay parámetros borrosos, se tiene que la posibilidad de que el sistema sea planificable siendo la asignación de prioridades la dada por el orden fuerte  $o^f$  es mayor que la posibilidad con cualquier otra asignación. O lo que es lo mismo:

$$\begin{aligned} & \Pi [(WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)] \\ & \geq \sup \{ \Pi [(WC_1^k < D_1) \text{ AND } (WC_2^k < D_2) \text{ AND } \dots (WC_n^k < D_n)] \} \quad \forall k=1..np / k \neq f \end{aligned}$$

Además, como la posibilidad del orden fuerte ( $\sigma^f$ ) es 1, se verifica:

$$\begin{aligned} & \Pi [(WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)] \\ & = \Pi [(\sigma^f) \text{ AND } (WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)] \end{aligned}$$

ya que

$$\begin{aligned} & \Pi [(\sigma^f) \text{ AND } (WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)] \\ & = \min \{ \Pi[(\sigma^f)], \Pi[(WC_1^f < D_1)], \Pi[(WC_2^f < D_2)], \dots, \Pi[(WC_n^f < D_n)] \} \\ & = \min \{ \Pi[(WC_1^f < D_1)], \Pi[(WC_2^f < D_2)], \dots, \Pi[(WC_n^f < D_n)] \} \\ & = \Pi [(WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)]. \end{aligned}$$

Por otro lado, dado que en general la posibilidad de otros órdenes distintos del orden fuerte es menor o igual que 1, se tiene:

$$\begin{aligned} & \Pi [(\sigma^k) \text{ AND } (WC_1^k < D_1) \text{ AND } (WC_2^k < D_2) \text{ AND } \dots (WC_n^k < D_n)] \\ & \leq \Pi [(WC_1^k < D_1) \text{ AND } (WC_2^k < D_2) \text{ AND } \dots (WC_n^k < D_n)] \quad \forall k=1..np / k \neq f, \end{aligned}$$

Con lo que:

$$\begin{aligned} & \Pi [(WC_1^f < D_1) \text{ AND } (WC_2^f < D_2) \text{ AND } \dots (WC_n^f < D_n)] \\ & \geq \sup \{ \Pi [(WC_1^k < D_1) \text{ AND } (WC_2^k < D_2) \text{ AND } \dots (WC_n^k < D_n)] \} \\ & \geq \sup \{ \Pi [(\sigma^k) \text{ AND } (WC_1^k < D_1) \text{ AND } (WC_2^k < D_2) \text{ AND } \dots (WC_n^k < D_n)] \} \\ & \quad \forall k=1..np / k \neq f \end{aligned}$$

Es decir, la posibilidad total de que el sistema sea planificable coincide con la posibilidad que se obtiene para la asignación óptima, ya que es el supremo de las posibilidades de los  $np$  términos de (6.18) asociados a los distintos órdenes.

### *Necesidad total cuando existe orden fuerte entre los plazos*

A continuación se va a probar que si existe orden fuerte entre los plazos, la necesidad total de que el sistema sea planificable coincide con la necesidad que se obtiene para la asignación óptima.

La necesidad de que el sistema sea planificable es uno menos la posibilidad de que no sea planificable, esto es:

$$N_{planif} = 1 - \Pi_{Nplanif}$$

Y la posibilidad de que el sistema no sea planificable es la posibilidad de que al menos una de las tareas no cumpla su plazo ( $wc_1 > d_1$ ) OR ( $wc_2 > d_2$ ) OR ... ( $wc_n > d_n$ )

Teniendo en cuenta las distintas asignaciones de prioridades, la posibilidad de que el sistema no sea planificable se puede expresar, generalizando (6.16) como:

$$\begin{aligned} \Pi_{Nplanif} = \Pi \{ & [(o^1) \text{ AND } (WC_1^1 > D_1)] \text{ OR } [(o^2) \text{ AND } (WC_1^2 > D_1)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_1^{np} > D_1)] \\ & \text{OR } [(o^1) \text{ AND } (WC_2^1 > D_2)] \text{ OR } [(o^2) \text{ AND } (WC_2^2 > D_2)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_2^{np} > D_2)] \\ & \dots \\ & \text{OR } [(o^1) \text{ AND } (WC_n^1 > D_n)] \text{ OR } [(o^2) \text{ AND } (WC_n^2 > D_n)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_n^{np} > D_n)] \} \end{aligned}$$

Hay que probar que la anterior expresión de posibilidad total de que el sistema sea no planificable es igual a la posibilidad de que el sistema sea no planificable considerando sólo el orden fuerte, esto es:

$$\begin{aligned} \Pi_{Nplanif} = & \Pi \{ [(o^1) \text{ AND } (WC_1^1 > D_1)] \text{ OR } [(o^2) \text{ AND } (WC_1^2 > D_1)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_1^{np} > D_1)] \\ & \text{OR } [(o^1) \text{ AND } (WC_2^1 > D_2)] \text{ OR } [(o^2) \text{ AND } (WC_2^2 > D_2)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_2^{np} > D_2)] \\ & \dots \\ & \text{OR } [(o^1) \text{ AND } (WC_n^1 > D_n)] \text{ OR } [(o^2) \text{ AND } (WC_n^2 > D_n)] \text{ OR } \dots [(o^{np}) \text{ AND } (WC_n^{np} > D_n)] \} \\ = \Pi \{ & [(o^f) \text{ AND } (WC_1^f > D_1)] \text{ OR } [(o^f) \text{ AND } (WC_2^f > D_2)] \text{ OR } \dots [(o^f) \text{ AND } (WC_n^f > D_n)] \} \end{aligned}$$

Para ello consideremos dos asignaciones de prioridades contiguas ( $o^f$ ) y ( $o^{f+1}$ ) correspondientes al orden fuerte y a otra asignación que se diferencia de la anterior en el intercambio de dos prioridades contiguas, las de las tareas  $i$  y  $j$ , con  $j=i+1$  siendo el orden fuerte el que viene dado por  $d_1 < d_2 < \dots < d_i < d_j < \dots < d_n$

Entre ( $o^f$ ) y ( $o^{f+1}$ ) las dos tareas que se ven afectadas, por tanto son las tareas  $i$  y  $j$ .

Y la posibilidad asociada relacionada con el cálculo de no planificabilidad es:

$$\begin{aligned} \Pi \{ & [(o^f) \text{ AND } (WC_i^f > D_i)] \text{ OR } [(o^f) \text{ AND } (WC_j^f > D_j)] \text{ OR} \\ & \text{OR } [(o^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)] \text{ OR } [(o^{f+1}) \text{ AND } (WC_j^{f+1} > D_j)] \} \end{aligned}$$

Para este cambio de prioridades se desea probar que la posibilidad máxima de los dos términos implicados es la del orden fuerte:

$$(6.19) \Pi \{ [(o^f) \text{ AND } (WC_i^f > D_i)] \text{ OR } [(o^f) \text{ AND } (WC_j^f > D_j)] \text{ OR}$$

$$\text{OR} [ (\sigma^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)] \text{ OR } [ (\sigma^{f+1}) \text{ AND } (WC_j^{f+1} > D_j)] \} \\ = \Pi \{ [ (\sigma^f) \text{ AND } (WC_i^f > D_i)] \text{ OR } [ (\sigma^f) \text{ AND } (WC_j^f > D_j)] \}$$

Prueba:

a) Supongamos en primer lugar que

$$\Pi \{ [ (\sigma^f) \text{ AND } (WC_j^f > D_j)] \} > \Pi \{ [ (\sigma^f) \text{ AND } (WC_i^f > D_i)] \}$$

esto es, de acuerdo con la Figura 6.20,  $\Pi \{ [ (\sigma^f) \text{ AND } (WC_j^f > D_j)] \} = \alpha_2$  y  $\Pi \{ [ (\sigma^f) \text{ AND } (WC_i^f > D_i)] \} = \alpha_3$  y se tiene que  $\alpha_2 > \alpha_3$ .

En este caso hay que probar, de acuerdo con (6.19), que  $\alpha_2$  es mayor que el supremo de las posibilidades de  $(\sigma^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)$  y de  $(\sigma^{f+1}) \text{ AND } (WC_j^{f+1} > D_j)$ .

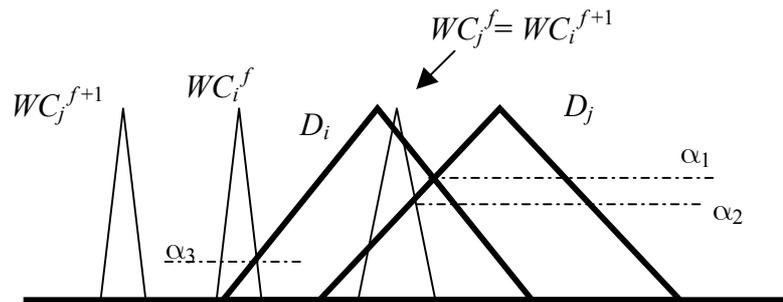


Figura 6.20: Tiempos de finalización de dos tareas  $i$  y  $j$  con dos asignaciones de prioridad  $\sigma^f$  y  $\sigma^{f+1}$  y sus plazos

En la Figura 6.21 se puede ver la comparación del tiempo de finalización de  $i$ , con su plazo con las dos asignaciones. Se muestra con relleno rallado la comparación correspondiente al orden fuerte de forma que la posibilidad de  $(\sigma^f) \text{ AND } (WC_i^f > D_i)$  se observa que es  $\alpha_3$ .

En la misma figura se muestra la comparación correspondiente a la asignación  $(\sigma^{f+1})$  con relleno sombreado. El tiempo de finalización de la tarea  $i$  en este caso es un número borroso triangular truncado a la altura de  $\alpha_1$ , que es la posibilidad de que  $d_i > d_j$ . Ese número borroso triangular es igual al tiempo borroso de finalización de la tarea  $j$  cuando la asignación de prioridades es  $(\sigma^f)$  que aparece en la Figura 6.22, es decir  $WC_j^f$ , tal y como se presentó en la prueba de la asignación de prioridades óptima borrosa en el capítulo 5. La zona de valores posibles de  $D_i$  y  $D_j$ , que cumplen

$d_i > d_j$ , también aparece sombreada en la Figura 6.21. Como se debe cumplir que  $WC_i^{f+1} > D_i$  y a la vez que  $D_i > D_j$ , se puede observar en la Figura 6.21 que es como comparar  $WC_j^f$  con  $D_j$ , en lugar de compararlo con  $D_i$ . Además, como mucho la posibilidad del término analizado  $(\sigma^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)$  es la posibilidad de  $(\sigma^{f+1})$ .

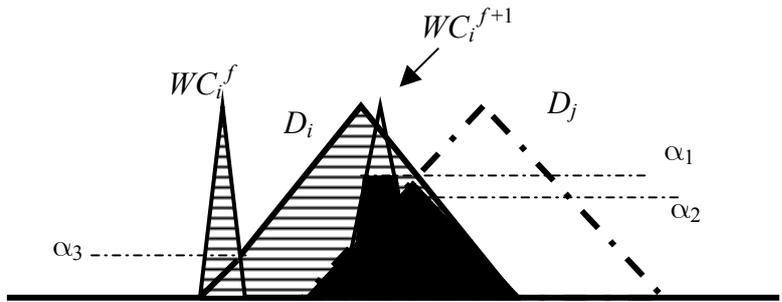


Figura 6.21: Tiempos de finalización de la tarea i con dos asignaciones de prioridad  $f$  y  $f+1$  y comparación con su plazo

Por tanto, la posibilidad del término de (6.19),  $(\sigma^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)$  es:

$$\Pi [ (WC_i^{f+1} > D_i) \text{ AND } (D_i > D_j) ] = \min \{ \Pi (WC_j^f > D_j), \Pi (D_i > D_j) \} = \min \{ \alpha_2, \alpha_1 \} \leq \alpha_2$$

Además, como  $WC_j^f = WC_j^{f+1} + P_i$ , entonces  $WC_j^f >_s WC_j^{f+1}$  (donde  $>_s$  indica fuertemente mayor), como se puede ver en la Figura 6.22. Debido a esta propiedad se tiene,  $\Pi [WC_j^f > D_j] > \Pi [WC_j^{f+1} > D_j]$ , es decir  $\alpha_2 > \Pi [WC_j^{f+1} > D_j]$ .

Y a su vez,  $\Pi [WC_j^{f+1} > D_j] > \Pi [ (WC_j^{f+1} > D_j) \text{ AND } (D_i > D_j) ]$ , con lo que  $\alpha_2 > \Pi [ (WC_j^{f+1} > D_j) \text{ AND } (D_i > D_j) ]$

Por lo que se verifica que  $\alpha_2$  es mayor o igual que el supremo de las posibilidades de  $(\sigma^{f+1}) \text{ AND } (WC_i^{f+1} > D_i)$  y de  $(\sigma^{f+1}) \text{ AND } (WC_j^{f+1} > D_j)$ , porque se ha demostrado que es mayor o igual que ambas.

Así pues, como se ha supuesto que  $\alpha_2 > \alpha_3$ , se cumple que la posibilidad de los cuatro términos indicados en (6.19) se puede calcular directamente con el orden fuerte y es  $\alpha_2$

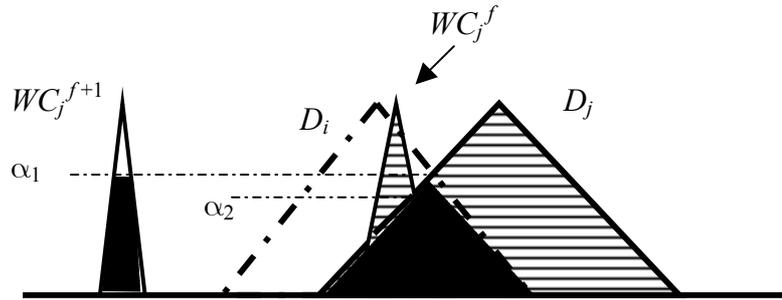


Figura 6.22: Tiempos de finalización de la tarea  $j$  con dos asignaciones de prioridad  $\sigma^f$  y  $\sigma^{f+1}$  y comparación con su plazo

b) Continuando con la prueba de (6.19), sólo resta verificar que se cumple también cuando  $\Pi \{ [( \sigma^f ) \text{ AND } ( WC_j^f > D_j ) ] \} < \Pi \{ [( \sigma^f ) \text{ AND } ( WC_i^f > D_i ) ] \}$ . Esto es, en la Figura 6.23,  $\Pi \{ [( \sigma^f ) \text{ AND } ( WC_j^f > D_j ) ] \} = \alpha_2$  y  $\Pi \{ [( \sigma^f ) \text{ AND } ( WC_i^f > D_i ) ] \} = \alpha_3$  y se tiene que  $\alpha_2 < \alpha_3$

En este caso hay que probar que  $\alpha_3$  además, es mayor que el supremo de las posibilidades de  $( \sigma^{f+1} ) \text{ AND } ( WC_i^{f+1} > D_i )$  y de  $( \sigma^{f+1} ) \text{ AND } ( WC_j^{f+1} > D_j )$ .

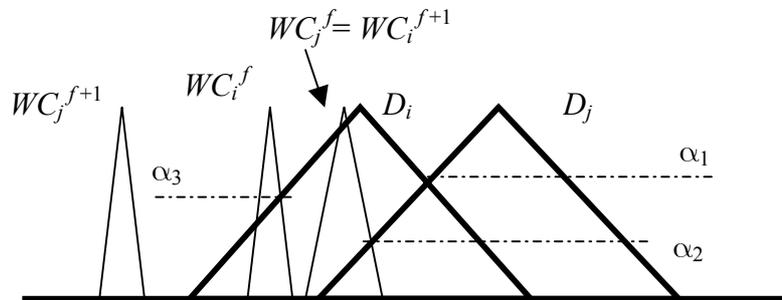


Figura 6.23: Tiempos de finalización de dos tareas  $i$  y  $j$  con dos asignaciones de prioridad  $\sigma^f$  y  $\sigma^{f+1}$  y sus plazos

De nuevo, se observa en la Figura 6.24 que  $WC_i^{f+1}$  es un número borroso triangular, que coincide con  $WC_j^f$ , truncado a la altura de  $\alpha_1$  y que aparece sombreado y además se han sombreado también los valores de los plazos que cumplen  $d_i > d_j$  y por tanto:

$$\Pi [(WC_i^{f+1} > D_i) \text{ AND } (D_i > D_j)] = \min \{ \Pi(WC_j^f > D_j), \Pi(D_i > D_j) \} = \min \{ \alpha_2, \alpha_1 \} \leq \alpha_2$$

Y se ha partido de que se verifica que  $\alpha_2 < \alpha_3$ , por lo que:

$$\Pi[(WC_i^{f+1} > D_i) \text{ AND } (D_i > D_j)] < \alpha_3$$

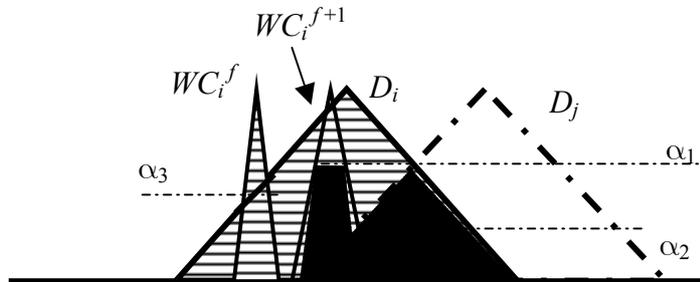


Figura 6.24: Tiempos de finalización de la tarea i con dos asignaciones de prioridad  $o^f$  y  $o^{f+1}$  y comparación con su plazo

Igual que antes, como  $WC_j^f = WC_j^{f+1} + P_i$ , entonces  $WC_j^f >_s WC_j^{f+1}$  donde  $>_s$  indica fuertemente mayor (ver la Figura 6.25). Debido a esta propiedad, se tiene  $\Pi[WC_j^f > D_j] > \Pi[WC_j^{f+1} > D_j]$ . A su vez, se verifica  $\Pi[WC_j^{f+1} > D_j] > \Pi[(WC_j^{f+1} > D_j) \text{ AND } (D_i > D_j)]$ . Esto es,  $\alpha_3 > \alpha_2 > \Pi[(WC_j^{f+1} > D_j) \text{ AND } (D_i > D_j)]$ .

Con lo que también con  $\alpha_2 < \alpha_3$  se cumple que la posibilidad de los cuatro términos de (6.19) son menores o iguales que la posibilidad con el orden fuerte, cuyo valor en el ejemplo es  $\alpha_3$

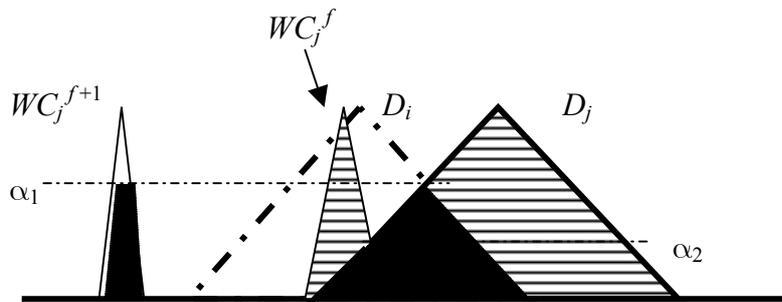


Figura 6.25: Tiempos de finalización de la tarea j con dos asignaciones de prioridad  $o^f$  y  $o^{f+1}$  y comparación con su plazo

Continuando el análisis para toda pareja de tareas con prioridades contiguas, se demuestra fácilmente que en cada intercambio de prioridades, la asignación asociada al orden fuerte proporciona el valor máximo de posibilidad de que el sistema sea no

planificable, hasta llegar a la asignación de prioridades para todas las tareas dada por dicho orden fuerte. Con lo que se ha probado que:

$$\Pi_{N_{planif}} = \Pi \{ [(o^f) \text{AND}(WC_1^f > D_1)] \text{OR} [(o^f) \text{AND}(WC_2^f > D_2)] \text{OR} \dots [(o^f) \text{AND}(WC_n^f > D_n)] \}$$

y con ello  $N_{planif} = 1 - \Pi \{ [(o^f) \text{AND}(WC_1^f > D_1)] \text{OR} [(o^f) \text{AND}(WC_2^f > D_2)] \text{OR} \dots [(o^f) \text{AND}(WC_n^f > D_n)] \}$  □

Por tanto, si existe orden fuerte en los plazos, la posibilidad y necesidad total de que el sistema sea planificable coincide con la posibilidad y necesidad que se obtiene para la asignación correspondiente al orden fuerte. De forma similar se podría plantear la prueba para EDF.

Volviendo al análisis de planificabilidad de un sistema planificado mediante el algoritmo EDF, se puede concluir que si existe orden fuerte entre los valores borrosos  $L \in \mathcal{D}^*$  entonces es posible extender el análisis de planificabilidad a plazos borrosos mediante aproximaciones con límites inferiores y superiores de los  $\alpha$ -cortes exactos.

Para ello, en el primer término de la desigualdad se mantienen los parámetros borrosos  $L$ , y en el segundo, se utilizan valores numéricos de  $l$ , y de  $d_i$ , en  $(l-d_i)/t_i$  que verifiquen el orden fuerte entre los plazos. Por ejemplo, si los plazos son borrosos y su núcleo es un conjunto unitario, y los periodos numéricos, se puede tomar el núcleo como plazo numérico.

$$\forall l \in \mathcal{D}, \forall L \in \mathcal{D}^* \quad L \geq \sum_{i=1}^n \left( \left\lfloor \frac{l-d_i}{t_i} \right\rfloor + 1 \right) \cdot P_i$$

$$L \in \mathcal{D}^* = \{ D_{i,k} \mid D_{i,k} = k \cdot t_i + D_i, D_{i,k} \leq \min(b_P, h), 1 \leq i \leq n, k \geq 0 \}$$

Veamos el mismo ejemplo presentado antes en este capítulo modelando además los plazos como números borrosos. Sea un sistema formado por tres tareas con periodos  $t_1=4, t_2=6$  y  $t_3=7$ . Los tiempos de ejecución son  $P_1=(0.8, 0.9, 1.1), P_2=(2.3, 2.4, 2.6)$  y  $P_3=(2.1, 2.2, 2.4)$ . Los plazos son  $D_1=(3.7, 3.8, 3.9), D_2=(5.8, 5.9, 6)$  y  $D_3=(5.9, 6, 6.1)$ .

Para todos los valores de  $L$  siempre existe un orden fuerte entre ellos, que es el mismo orden que se da entre los valores numéricos de  $l$  calculados a partir de los valores numéricos  $d_1=3.8, d_2=5.9$  y  $d_3=6$ . Por tanto, para el cálculo del segundo término de la desigualdad se consideran los valores numéricos de los plazos  $d_1=3.8, d_2=5.9$  y  $d_3=6$ . Así  $l \in \mathcal{D} = \{ d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_P, h), 1 \leq i \leq n, k \geq 0 \}$ . Y en

el primer término de la desigualdad está el valor borroso correspondiente  $L \in \mathcal{D}^* = \{D_{i,k} \mid D_{i,k} = k \cdot t_i + D_i, D_{i,k} \leq \min(b_P, h), 1 \leq i \leq n, k \geq 0\}$ .

$$\forall l \in \mathcal{D}, \forall L \in \mathcal{D}^* \quad L \geq \sum_{i=1}^n \left( \left\lfloor \frac{l - d_i}{t_i} \right\rfloor + 1 \right) \cdot P_i$$

De nuevo, el punto crítico para el cálculo de planificabilidad se da en  $l=48$ :

$$\Pi_{planif} = \min(\Pi(L \geq C_p)) = \Pi(L \geq C_p) \mid L = 11 \cdot t_1 + D_1 = (47.7, 47.8, 47.9).$$

$$N_{planif} = \min(N(L \geq C_p)) = N(L \geq C_p) \mid L = 11 \cdot t_1 + D_1 = (47.7, 47.8, 47.9).$$

Para  $l=48$ , se tiene  $C_p = 12 \cdot P_1 + 8 \cdot P_2 + 7 \cdot P_3 = (42.7, 45.4, 50.8)$ . Cuando  $l=48$ , entonces el valor borroso correspondiente es  $L = (47.7, 47.8, 47.9)$  y comparando ambos conjuntos borrosos se obtiene  $\Pi(L \geq C_p) = 1$ ,  $N(L \geq C_p) = 1 - 0.56 = 0.44$ , por lo que la planificabilidad del sistema es:

$$\Pi_{planif} = 1 \quad N_{planif} = 0.44$$

En este capítulo se ha demostrado que es posible extender de forma sencilla el análisis de planificabilidad de sistemas planificados mediante el algoritmo EDF en ciertos casos. Así, cuando coinciden los plazos y los periodos, se puede extender la condición de planificabilidad a tiempos de ejecución borrosos y a periodos borrosos y el cálculo se puede realizar aplicando aritmética borrosa o aritmética de intervalos, produciendo los mismos valores de posibilidad y necesidad de que el sistema sea planificable que la aplicación del principio de extensión.

Cuando los plazos son menores que los periodos, se puede extender el análisis de la misma forma si sólo los tiempos de ejecución son borrosos, porque los valores  $L \in \mathcal{D}^*$  son independientes de los tiempos de ejecución.

Sin embargo, como tanto  $L$  como  $C_p$  dependen de los plazos y de los periodos, no es posible extender el análisis a periodos y plazos borrosos aplicando sin más cualquiera de los dos métodos citados y suponiendo independencia entre  $L$  y  $C_p$ , ya que el resultado no coincide con los valores de posibilidad y de necesidad proporcionados por el principio de extensión. Únicamente en el caso de que exista orden fuerte entre los plazos absolutos es posible extender el análisis a tiempos de ejecución y plazos borrosos mediante aritmética borrosa o aritmética de intervalos.

## **7. Metodología de diseño de sistemas de tiempo real. Caso ejemplo**

En este capítulo se propone en primer lugar una extensión de la metodología de diseño de sistemas de tiempo real, cuando la incertidumbre de sus parámetros temporales se modela mediante conjuntos borrosos interpretados como distribuciones de posibilidad. Esta metodología, descrita en el apartado 7.1, se centra en la obtención de un diseño de alto nivel, orientado al análisis temporal del sistema y se obtiene a partir de los algoritmos borrosos de planificabilidad propuestos en esta tesis.

En el apartado 7.2 de este capítulo se presenta un caso ejemplo basado en el sistema de protección automática ferroviario, donde se aplica la metodología propuesta para el diseño de una parte de este sistema de tiempo real. En este caso ejemplo se muestra que se puede obtener mucha más información acerca de la planificabilidad del sistema cuando se realiza un análisis borroso que si únicamente se estudia el caso más desfavorable de planificabilidad, de manera que un diseño que sería rechazado atendiendo al análisis en el peor caso, según el análisis borroso es considerado válido, e incluso se puede mejorar aumentando la frecuencia de ejecución de una tarea crítica.

## **7.1. Extensión de la metodología de diseño de sistemas de tiempo real**

En este apartado se describen las fases de la metodología propuesta centrada en el análisis temporal cuando los parámetros temporales de las tareas se conocen de forma incierta y se modelan como conjuntos borrosos y se recogen, dentro de la fase 3 de la metodología, las extensiones borrosas del análisis de planificabilidad propuestas en esta tesis.

Las fases de la metodología de diseño de sistemas de tiempo real son:

- Descripción del entorno, del sistema, del interfaz y requisitos del sistema
- Diseño y modelado del sistema: tareas y planificación
- Análisis de planificabilidad
- Modificación del diseño, a partir de las conclusiones obtenidas de la fase anterior

Como a lo largo de toda la tesis, sólo se consideran sistemas monoprocesador, con planificador expulsor basado en prioridades.

En esta metodología no se pretende considerar de forma exhaustiva todos los casos y posibles implementaciones que se pueden dar en los sistemas de tiempo real, sino que se pretende mostrar que es posible modelar situaciones muy diversas e introducir incertidumbre en dichos modelos. De esta forma, aplicando la teoría de la posibilidad, se puede obtener más información acerca de la planificabilidad del sistema que si sólo se modela el peor caso y se pueden expresar requisitos en términos de valores de posibilidad o de necesidad, que no se pueden expresar en términos del peor caso, tal y como se ilustra en el caso ejemplo del apartado 7.2. Así, es posible obtener diseños válidos por medio del modelado de la incertidumbre que serían rechazados si únicamente se modela y analiza el caso más desfavorable de planificabilidad. Por otro lado, el cálculo de la planificabilidad borrosa del sistema mediante los algoritmos propuestos en esta tesis es más rápido que la simulación a partir de modelos probabilistas de los parámetros de las tareas.

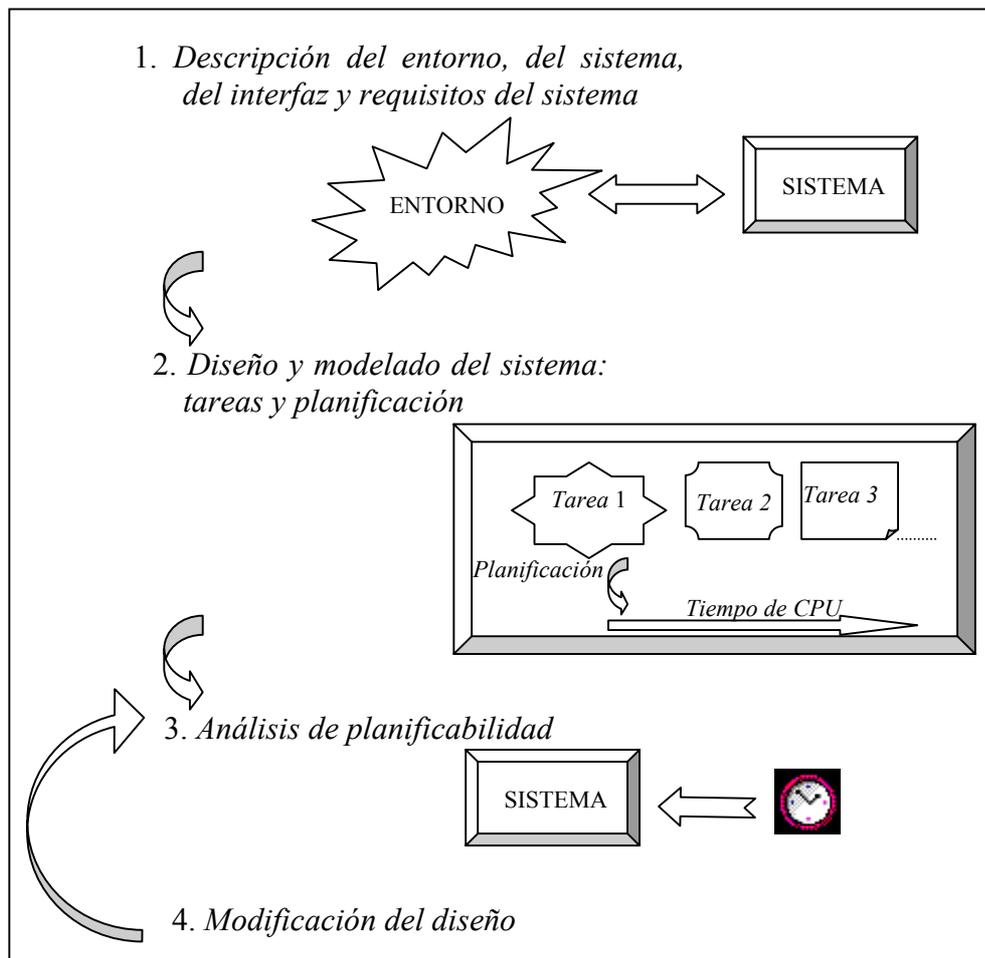


Figura 7.1: Metodología de diseño de sistemas de tiempo real

### **7.1.1. Descripción del entorno, del sistema, del interfaz y requisitos del sistema**

En la primera fase de la metodología se propone realizar una descripción del entorno en el que se encuentra el sistema y con el que se comunica, de los interfaces entre el entorno y el sistema de tiempo real que se desea diseñar y los requisitos que se exigen al sistema y a los interfaces del sistema con el entorno. Las demandas realizadas a los sistemas de tiempo real provienen de las necesidades de la aplicación y se denominan requisitos. La definición de los requisitos es una tarea que requiere un conocimiento profundo de la aplicación que se desea implantar y es de gran importancia, ya que a menudo los fallos producidos en grandes sistemas se producen por una mala definición de los mismos. Estos requisitos se pueden dividir en requisitos funcionales y requisitos no funcionales, entre los que se encuentran los requisitos temporales. En sistemas de tiempo real hay que definir de forma precisa y conjuntamente los requisitos funcionales y los no funcionales para construir la especificación del sistema ([Joseph, 1996]).

### **7.1.2. Diseño y modelado del sistema: tareas y planificación**

En esta segunda fase de la metodología hay que proponer un diseño de alto nivel (modelado de las tareas y de su planificación) del sistema que cumpla los requisitos funcionales exigidos al sistema. Cuando se disponga de dicho modelo, en la siguiente fase se verificarán los requisitos temporales impuestos al sistema en la primera fase de la metodología.

El modelado del sistema consiste en definir y modelar las tareas que lo van a componer, así como diseñar la planificación de dichas tareas.

#### **7.1.2.1. Diseño y modelado de las tareas**

Para poder cumplir con la funcionalidad requerida al sistema hay que definir las tareas que lo van a componer, proceso en el que se pueden distinguir cinco pasos:

1. Identificación de las tareas
2. Para cada tarea, determinación de si es periódica, esporádica o aperiódica
3. Para cada tarea, determinación de si es estricta, flexible o firme.

4. Identificación de los recursos compartidos y de las relaciones entre tareas
5. Modelado de los parámetros temporales de las tareas

### *1. Identificación de las tareas*

En primer lugar hay que repartir la funcionalidad requerida al sistema en las tareas que la van a llevar a cabo. Una tarea puede cumplir uno o varios requisitos funcionales, y un requisito funcional puede ser cumplimentado por la ejecución de varias tareas. Los requisitos temporales del sistema se tratarán de cumplir imponiendo restricciones temporales a las tareas diseñadas, modelando sus parámetros temporales (paso 5 dentro de la fase de diseño y modelado de las tareas).

### *2. Para cada tarea, determinación de si es periódica, esporádica o aperiódica*

Para cada tarea se debe determinar si es periódica, esporádica o aperiódica. Una tarea es periódica si el tiempo entre lanzamientos consecutivos de la tarea es constante, esporádica si existe un tiempo mínimo entre llegadas consecutivas y aperiódica si ese tiempo no está acotado (ver capítulo 2).

Dos formas posibles de implementar una tarea esporádica son, bien la utilización de una interrupción para atender a la petición y meterla en la cola de las tareas listas para ejecutarse, bien que en cada interrupción de reloj se compruebe si ha llegado una petición de dicha tarea, en cuyo caso se pone en la cola de las tareas listas. En el paso 5 de esta fase se describe el modelo asociado a cada implementación.

Por otro lado, si hay tareas aperiódicas en el sistema, uno de los mecanismos existentes para atenderlas es utilizar un servidor aperiódico. Se puede utilizar, por ejemplo, un Servidor Esporádico (“Sporadic Server” o SS) o un Servidor Aplazable (“Deferrable Server” o DS).

### *3. Para cada tarea, determinación de si es estricta, flexible o firme*

Para cada tarea también hay que especificar si se debe asegurar el cumplimiento de su restricción temporal (tarea estricta), si se permite que no lo cumpla (tarea flexible) o si tiene algún requisito estricto y otro flexible (firme). Si una tarea de tipo estricta no cumple su restricción temporal o plazo de ejecución entonces el sistema es no planificable y el diseño no es válido, con lo que habría que modificar el diseño y

volver a analizar la planificabilidad del nuevo diseño, de acuerdo con la metodología propuesta.

Es posible garantizar el cumplimiento de las restricciones temporales tanto de las tareas periódicas, como de las esporádicas, por lo que este tipo de tareas pueden tener requisitos temporales estrictos. Sin embargo, no se puede garantizar las restricciones temporales de las tareas aperiódicas, si las tuviera, con lo que este tipo de tareas serán siempre flexibles.

#### 4. Identificación de los recursos compartidos y de las relaciones entre tareas

Llegados a este punto del diseño, hay que identificar los recursos de las tareas, determinar si son compartidos por varias tareas y protegerlos. Si las tareas comparten recursos éstos deben ser protegidos frente a accesos simultáneos mediante un protocolo de sincronización y en tal caso hay que identificar dichas tareas y los recursos compartidos para calcular más adelante, una vez asignadas las prioridades, los tiempos de bloqueo que pueden sufrir las tareas.

Además hay que definir si existe relación de precedencia entre las tareas, de forma que antes de comenzar la ejecución de una tarea otra deba haber terminado. Las relaciones de precedencia se pueden modelar mediante transacciones (ver capítulo 2, apartado 2.6) formadas por un conjunto de tareas que se deben ejecutar en un orden preestablecido. Las transacciones tienen plazos asociados, que reflejan propiedades “end-to-end” o principio-a-fin.

Una transacción se puede implementar como una tarea periódica de periodo igual al de la transacción, seguida de una serie de tareas esporádicas. La tarea periódica es la primera en la relación de precedencia y cuando termina lanza la tarea esporádica que le debe seguir según dicha relación. Cuando la tarea esporádica termina, lanza la siguiente tarea esporádica, que es la tercera en la relación de precedencia, y así sucesivamente.

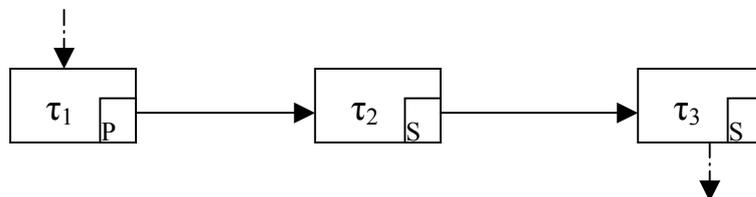


Figura 7.2: Ejemplo de transacción compuesta por tres tareas: la tarea 1 es periódica y las tareas 2 y 3 son esporádicas

### 5. Modelado de los parámetros temporales de las tareas

Es importante recordar que el modelado del sistema que se está realizando en esta fase está centrado en las características temporales del mismo. Por tanto, además de definir las tareas y su funcionalidad, hay que modelar los parámetros temporales que describen cada tarea.

Los parámetros temporales que se pueden especificar en esta fase son el tiempo de ejecución de la tarea, el plazo de ejecución, el periodo, el tiempo de utilización de un recurso compartido con otras tareas, y el “release jitter” o retraso de la activación de una tarea. Los parámetros temporales del servidor aperiódico son el periodo y la capacidad.

Si una tarea  $i$  es periódica se deben especificar, además de su tiempo de ejecución  $p_i$ , su periodo  $t_i$ , así como su plazo  $d_i$ , si lo tiene.

Si una tarea es esporádica se deben especificar su tiempo de ejecución y por lo menos el tiempo mínimo entre llegadas consecutivas. También se deben especificar su plazo si lo tiene. Estas tareas se puede modelar como tareas periódicas cuyo periodo  $t_i$  es igual al tiempo mínimo entre llegadas consecutivas. Por tanto, al analizar la planificabilidad de la tarea y del sistema se estará estudiando el caso más desfavorable al suponer que esta tarea está lista cada  $t_i$ .

Si una tarea es aperiódica se deben especificar su tiempo de ejecución, y en este caso el plazo, si lo tiene, es necesariamente flexible. Si se utiliza un servidor aperiódico del tipo Servidor Aplazable (DS) o Servidor Esporádico (SS) para atender a las tareas aperiódicas hay que especificar el periodo del servidor  $t_s$  y la capacidad  $c_s$  del mismo.

Cuando un mismo recurso es compartido por varias tareas, hay que especificar cuánto tiempo utiliza cada tarea  $i$  dicho recurso  $ru_i$ . Si además una tarea puede sufrir un retardo en su activación, hay que especificar el valor de dicho tiempo, denominado “release jitter”  $j_i$ .

Es posible modelar las dos implementaciones propuestas de una tarea esporádica mediante los parámetros que se acaban de describir. Así, si se implementa mediante una interrupción que atiende la petición de la tarea colocándola en la cola de las tareas listas para que la tarea asociada se ejecute con la prioridad que se le asigne, en el modelo aparece una nueva tarea de interrupción con un tiempo de ejecución correspondiente al tiempo de ejecución de la interrupción  $p_{int}$  y con un periodo igual al mínimo tiempo entre llegadas consecutivas de la tarea esporádica  $t_i$ . Si el diseño de la planificación que se realiza más tarde en esta fase se basa en prioridades fijas, la prioridad de esta tarea es la máxima, por ser una tarea de tipo interrupción.

Si la tarea esporádica se implementa mediante sondeo o “polling”, de forma que en cada interrupción de reloj si ha llegado una petición de la tarea esporádica se pone en la cola de las tareas listas, en el modelo se debe añadir a la tarea un “release jitter” igual al periodo de la tarea de reloj. Este periodo es el máximo retardo que puede sufrir la tarea si la petición llega justo cuando ha terminado la tarea de reloj, en cuyo caso hay que esperar un periodo completo de reloj para encolar la tarea.

Dependiendo de la información disponible se pueden representar los parámetros de forma distinta: como números clásicos correspondientes al caso más desfavorable, como distribuciones de posibilidad y como distribuciones de probabilidad.

### Parámetros numéricos

Si la tarea está en fase de diseño se deben dar al menos los tiempos más desfavorables desde el punto de vista de la planificabilidad. Así, los valores más desfavorables correspondientes a tiempo de ejecución, “release jitter”, capacidad del servidor aperiódico y tiempo de utilización de recursos son los máximos ya que cuanto mayores sean estos tiempos, mayores son los tiempos de finalización de las tareas.

Por otro lado, el peor caso para la planificabilidad del sistema se da con los valores mínimos de los periodos y de los plazos, ya que cuanto menor sea el plazo, más cerca se encuentra el tiempo de finalización de no cumplirlo, y cuanto menor sea el periodo, hay más interferencia en tareas menos prioritarias, retrasando así sus tiempos de finalización.

Para una tarea esporádica se debe dar al menos el mínimo tiempo entre llegadas, que se puede considerar como el mínimo periodo teniendo en cuenta que, en general, no se conoce con precisión y con certeza.

### Distribuciones de Posibilidad

Si se dispone de más información, se pueden modelar los parámetros correspondientes mediante conjuntos borrosos interpretados como distribuciones de posibilidad.

En general puede ser interesante modelar los tiempos de ejecución mediante conjuntos borrosos  $P_i$ , cuyo límite superior del soporte  $\overline{p}_i(0)$  es el máximo tiempo de ejecución  $p_i$  considerado antes para el caso más desfavorable.

Un periodo borroso  $T_i$  puede modelar una tarea periódica a la que se permite una pequeña variación en su periodo, o bien una tarea esporádica, en cuyo caso el límite inferior del soporte  $\underline{t}_i(0)$  es el mínimo tiempo entre llegadas consecutivas de la tarea esporádica.

El modelado borroso del plazo  $D_i$ , aparte de responder a un conocimiento incierto del mismo, puede tener relación con el tipo de tarea al que corresponde: si es una tarea estricta o una tarea flexible. Tal y como se ha dicho antes, si una tarea es estricta se debe asegurar que termina su ejecución antes del plazo correspondiente. Si el modelo en alguno de sus parámetros es borroso esto se traduce en que la posibilidad y la necesidad de que la tarea termine antes de su plazo ha de ser 1:

$$\Pi(WC_i \leq D_i) = N(WC_i \leq D_i) = 1$$

Esta condición es independiente de cómo esté modelado el plazo y de cómo sea el tiempo de finalización numérico o borroso. En principio, en este caso es más adecuado modelar el plazo como un número clásico, pero si el plazo depende de otro parámetro que también es borroso, podría ser modelado también como un número borroso (este punto se ilustra en el apartado 7.2, el caso ejemplo).

Si una tarea es flexible, se permite que en algunas ejecuciones la tarea no cumpla el plazo correspondiente. A menudo se trata de tareas con llegadas no acotadas, pero se podrían modelar también tareas esporádicas y periódicas a las que se permite cierta flexibilidad en el cumplimiento de sus plazos. Esta mayor flexibilidad definida con parámetros numéricos permitiendo sobrepasos temporales de los plazos, se puede modelar en términos borrosos permitiendo que en el peor caso la posibilidad y la necesidad de que la tarea finalice antes de su plazo sean distintas de 1. Es más apropiado modelar el plazo como un número borroso, aunque también se puede modelar como un número clásico.

Además, tanto el plazo borroso como el valor de posibilidad y necesidad de que la tarea finalice antes de su plazo pueden modelar la utilidad del resultado que proporciona la tarea: cuanto más cerca de 1 estén los valores de posibilidad y necesidad, más útil, bien por precisión del resultado, bien por ser un resultado más completo.

Mediante los algoritmos de análisis de planificabilidad borrosos propuestos en esta tesis se puede calcular la planificabilidad del sistema cuando los parámetros se modelan como números borrosos o como conjuntos borrosos no convexos, continuos o discontinuos, siempre y cuando su soporte esté acotado.

### Distribuciones de probabilidad

Si alguna tarea ya está implementada y se dispone de un dato preciso, se utilizaría dicho parámetro numérico. Si se tiene información probabilista mediante una distribución de probabilidad de un determinado tiempo, se propone en esta metodología transformar dicha distribución de probabilidad en una distribución de posibilidad. De esta forma se utiliza un modelado homogéneo de la información que facilita la interpretación de los resultados obtenidos (aunque se pierde alguna información).

La transformación propuesta es la basada en normalización, ya que se trata de una transformación muy sencilla que consiste en normalizar la distribución haciendo que el núcleo del conjunto borroso tenga posibilidad 1. Además, se ha mostrado mediante algunos ejemplos que los resultados de planificabilidad obtenidos mediante la distribución de probabilidad y mediante la distribución de posibilidad resultante de la transformación, son parecidos atendiendo a la asignación de prioridades que maximiza la planificabilidad del sistema, al igual que ocurre con la transformación basada en máxima especificidad y razón insuficiente.

La transformación es  $\pi_i = p_i / p_1$ , con  $p_1 \geq p_2 \geq \dots \geq p_n$ , que en el caso continuo se expresa como  $\pi(x) = p(x) / p_m$ , con  $p_m = \sup_x p(x)$

Por tanto, si se modela un tiempo de ejecución mediante una función de densidad de probabilidad, el número borroso triangular asociado mediante esta transformación es también triangular y escalado para que su moda tenga posibilidad 1. Ver Figura 7.3

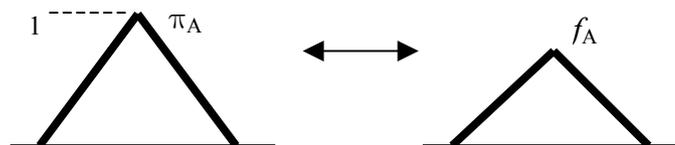


Figura 7.3: Transformación probabilidad-posibilidad basada en normalización

#### 7.1.2.2. Diseño de la planificación de las tareas

Una vez identificadas y modeladas las tareas que componen el sistema, hay que diseñar la su planificación. Dado que la planificación más apropiada se basa en la asignación de prioridades, ya que permite una mayor flexibilidad a la hora de introducir modificaciones en el sistema, ya sea añadiendo o quitando tareas, o modificando los parámetros que describen las tareas existentes, se trata por tanto de decidir si se opta por un sistema de prioridades fijas o dinámicas.

Si las prioridades son fijas y los plazos son numéricos, se propone que se asignen según el algoritmo DMS, ya que este algoritmo es óptimo en el sentido de que si existe una asignación de prioridades fijas que es planificable, la asignación dada por DMS también es planificable. Según DMS la tarea más prioritaria es la que tiene el menor plazo y la menos prioritaria la de mayor plazo. Si los plazos son borrosos y existe orden fuerte entre ellos, se propone asignar las prioridades de acuerdo con dicho orden fuerte, ya que esa asignación maximiza la posibilidad y la necesidad de que el sistema sea planificable.

Si las prioridades son dinámicas se puede utilizar el algoritmo de planificación EDF.

Es posible tener en cuenta en el modelo el tiempo que el planificador consume por el manejo de colas, poniendo una tarea a la cola de las tareas listas cuando, por ejemplo, se ha cumplido su periodo. Para ello basta con introducir una nueva tarea  $j$  en el modelo por cada tarea  $i$  periódica o esporádica con el tiempo de ejecución  $p_{cola}$  correspondiente al manejo de las colas y periodo igual al de la tarea  $i$ ,  $t_j=t_i$ .

Si se ha optado por asignar prioridades fijas a las tareas y hay recursos compartidos por varias tareas, se puede utilizar el protocolo básico de herencia de prioridades (PIP) para evitar inversiones de prioridades no acotadas. En este protocolo se utiliza un semáforo para implementar la sección crítica. Cuando el recurso está ocupado, la petición se encola de acuerdo con su prioridad. Al encolar una petición, la prioridad de la tarea que está utilizando el recurso se eleva temporalmente al valor de la prioridad encolada (hereda su prioridad). Esta herencia es la que evita que haya expulsión en la sección crítica por parte de otra tarea, lo que produciría una inversión de prioridades no acotada. Al liberar el semáforo la tarea que está al comienzo de la cola accede al recurso.

Una vez determinadas las prioridades fijas, los tiempos de utilización de los recursos y el protocolo de sincronización utilizado, se puede calcular el tiempo de bloqueo  $b_i$  que sufre cada tarea  $i$ . Si el protocolo de sincronización es PIP los tiempos de bloqueo se pueden calcular mediante el algoritmo (7.1) que permite obtener tiempos de bloqueo borrosos a partir de tiempos de utilización de recursos borrosos  $RU_i$ .

#### (7.1) Cálculo del tiempo de bloqueo con el protocolo PIP

1. Se ordenan las tareas de acuerdo con su prioridad, siendo la tarea 1 la más prioritaria.
2. Inicialización de  $B$ :  $B = 0$  (clásico)
3. Comienzo con la tarea menos prioritaria:  $i = n$
4.  $B_i = B$

5.  $B = \widetilde{\max}(RU_i, B)$
6. Si  $i = 1$ , termina el algoritmo.  
Si  $pr_{i-1} > pc_r$ , entonces  $B = 0$  (clásico)
7. Iterar:  $i = i-1$  y continúa en el paso 4.

Recordemos que  $\widetilde{\max}$  es la operación máximo borroso, y que este algoritmo se describe en el apartado 5.7 de esta tesis. Si los tiempos de utilización de recursos son números clásicos se sustituye en (7.1) el tiempo borroso de utilización de recurso  $RU_i$  por el tiempo numérico  $ru_i$  y los tiempos de bloqueo borrosos  $B_i$  y  $B$  por los tiempos numéricos  $b_i$  y  $b$  de forma que queda el algoritmo ya descrito en el capítulo 2, conocido en la literatura.

### 7.1.3. Análisis de planificabilidad.

La tercera fase de la metodología, sobre la que se ha centrado principalmente esta tesis, es la obtención de medidas borrosas de planificabilidad del sistema. En esta tesis se han definido algoritmos de cálculo de la planificabilidad borrosa basados tanto en aritmética borrosa como en aritmética de intervalos. En esta metodología se propone utilizar los algoritmos basados en aritmética de intervalos ya que su complejidad no aumenta al utilizar conjuntos borrosos no triangulares, y las operaciones elementales como la suma o la división tienen una complejidad similar.

Los casos analizados se pueden agrupar en los siguientes:

- Sistema con prioridades fijas y con plazos menores o iguales que los periodos

Se propone un algoritmo basado en aritmética de intervalos con búsqueda por bipartición, que contempla la existencia de tiempos de bloqueo y “release jitter” en las tareas, y permiten calcular la posibilidad y necesidad de planificabilidad del sistema a partir de tiempos de ejecución, plazos, periodos, tiempos de bloqueo y “release jitter” borrosos.

- Sistema con prioridades fijas y servidor aperiódico.

Cuando existen tareas aperiódicas atendidas por servidores aperiódicos del tipo esporádico y aplazable, se propone la utilización de los algoritmos de planificabilidad basados en aritmética de intervalos desarrollados en esta tesis, que extienden el análisis a tiempos de ejecución, periodos, capacidad del

servidor y periodo del servidor borrosos. En este caso analizado las tareas periódicas tienen los plazos menores o iguales que los periodos.

- Tareas con relación de precedencia.

Cuando existen tareas con relación de precedencia entre ellas se propone implementarlas de forma que la primera tarea sea periódica y el resto son esporádicas. Cada tarea esporádica es lanzada cuando finaliza la tarea anterior. En este caso se propone un análisis de planificabilidad en el apartado 7.1.3.3.

- Sistema con prioridades fijas y con plazos mayores que los periodos.

Igual que en el caso anterior, los algoritmos propuestos en esta tesis permiten el cálculo exacto de la planificabilidad borrosa del sistema. En primer lugar se presentan los algoritmos basados en aritmética de intervalos que tienen en cuenta tiempos de ejecución, plazos, tiempos de bloqueo y “release jitter” borrosos. En segundo lugar, se presentan los algoritmos basados en aritmética de intervalos que extienden el análisis de planificabilidad además de para los parámetros borrosos anteriores, para periodos borrosos. Estos algoritmos, junto con los basados en aritmética borrosa han sido desarrollados en el apartado 5.6. de esta tesis.

- Sistema con prioridades dinámicas y con plazos iguales a los periodos.

En esta tesis se ha analizado también la asignación de prioridades dinámicas de acuerdo con el algoritmo EDF. Cuando los plazos son iguales a los periodos, es posible calcular la planificabilidad borrosa a partir de tiempos de ejecución y periodos borrosos.

- Sistema con prioridades dinámicas y con plazos menores o iguales a los periodos.

Cuando los plazos son menores que los periodos se puede extender el análisis a tiempos de ejecución borrosos y se propone aplicar el algoritmo desarrollado en el apartado 6.2 basado en aritmética de intervalos. No se ha encontrado la forma de extender el análisis mediante aritmética de intervalos o aritmética borrosa cuando los plazos o los periodos son borrosos.

#### **7.1.3.1. Prioridades fijas, con plazos menores o iguales que los periodos.**

Tal y como se dijo en el capítulo 2, cuando los plazos coinciden con los periodos y no hay tiempos de bloqueo ni “release jitter”, existe una condición suficiente aunque no necesaria de planificabilidad en el análisis clásico:

$$(7.2) \frac{p_1}{t_1} + \dots + \frac{p_n}{t_n} \leq U_l(n) = n \cdot \left(2^{1/n} - 1\right)$$

donde  $n$  es el número de tareas.

En el análisis con parámetros borrosos se puede realizar esta comprobación previa, antes de aplicar el algoritmo de cálculo de tiempos de finalización. Si no hay tiempos de bloqueo ni “release jitter” y los plazos son iguales a los periodos, se aplica la desigualdad (7.2) para el peor caso como se muestra en (7.3), y si se cumple la desigualdad entonces el sistema es planificable con posibilidad y necesidad igual a 1. Si no se cumple la desigualdad para el peor caso, se aplica el algoritmo basado en aritmética de intervalos propuesto en esta tesis.

$$(7.3) \frac{\overline{p_1(0)}}{\underline{t_1(0)}} + \dots + \frac{\overline{p_n(0)}}{\underline{t_n(0)}} \leq n \cdot \left(2^{1/n} - 1\right)$$

Como ya se vio en el capítulo 5 de esta tesis, el algoritmo de análisis de planificabilidad propuesto (7.4) es válido para tareas periódicas, con plazos menores o iguales que los periodos y con prioridades fijas asignadas con cualquier criterio, incluido la asignación según RMS o DMS. Si los plazos y/o los periodos son borrosos la condición para poder aplicar el algoritmo es que los plazos sean fuertemente menores que los periodos. Si todos los parámetros son numéricos basta con aplicar el algoritmo básico clásico (7.5), donde el valor inicial de  $s$  es  $s\_in_i = wc_{i-1} + p_i$

**ALG Aritmética de Intervalos**

$\underline{wc}_0(0) = 0$

for  $i=1$  to  $n\_tasks$  do

$\underline{wc}_i(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(0); b_i = \underline{b}_i(0); j_j = j_j(0); t_j = \overline{t}_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$

$\underline{wc}_i(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(1); b_i = \underline{b}_i(1); j_j = j_j(1); t_j = \overline{t}_j(1); s\_in_i = \underline{wc}_i(0) \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$

$\overline{wc}_i(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); t_j = t_j(1); s\_in_i = \underline{wc}_i(1) \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$

$\overline{wc}_i(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); t_j = t_j(0); s\_in_i = \overline{wc}_i(1) \\ \small 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right)$

ALG ZONA

end

(7.4)  $\Pi_{planif} / N_{planif} = \min\{\Pi / N(WC_i \leq D_i)\} \quad \forall i$

Donde el algoritmo básico (ALG BASICO) es:

$$(7.5) \quad \boxed{\begin{array}{l} \text{ALG BASICO:} \\ \left\{ \begin{array}{l} s_{i,0} = s_{in_i} \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_i = s_{i,k} + j_i \\ \text{dado, } s_{i,k} = b_i + p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1} + j_h}{t_h} \right\rceil \end{array} \right. \end{array}}$$

Si por el contrario los tiempos de ejecución, los plazos, los periodos, los tiempos de bloqueo o el “release jitter” son borrosos, se puede aplicar el algoritmo (7.4) basado en aritmética de intervalos. El algoritmo no calcula las aproximaciones válidas de los tiempos de finalización borrosos sino que únicamente realiza una búsqueda por bipartición de los puntos de corte entre los tiempos de finalización y los plazos borrosos, para calcular la planificabilidad borrosa exacta del sistema.

El algoritmo ZONA (7.6) utilizado en el algoritmo basado en aritmética de intervalos, determina la zona de búsqueda del punto de corte y llama al algoritmo de búsqueda por bipartición en la zona de los límites superiores (Bipartición sup (7.8)) o de los límites inferiores (Bipartición inf (7.7)) de los  $\alpha$ -cortes de los tiempos de finalización.

$$(7.6) \quad \boxed{\begin{array}{l} \text{ALG ZONA} \\ \text{if } (\underline{d}_i(0) \geq \overline{wc}_i(0)) \quad \Pi(WC_i \leq D_i) = 1, \quad N(WC_i \leq D_i) = 1 \\ \text{if } (\overline{d}_i(0) \leq \underline{wc}_i(0)) \quad \Pi(WC_i \leq D_i) = 0, \quad N(WC_i \leq D_i) = 0 \\ \text{if } (\overline{wc}_i(1) \geq \underline{d}_i(1)) \quad N(WC_i \leq D_i) = 0 \\ \quad \text{else if } (\overline{wc}_i(1) < \underline{d}_i(1)) \quad \text{Bipartición sup} \\ \text{if } (\underline{wc}_i(1) \leq \overline{d}_i(1)) \quad \Pi(WC_i \leq D_i) = 1 \\ \quad \text{else if } (\underline{wc}_i(1) > \overline{d}_i(1)) \quad \text{Bipartición inf} \end{array}}$$



el apartado 5.2.3. Modificando el algoritmo básico como se indica en (7.9), cuando se detecta que  $s_{i,k}$  ha sobrepasado el periodo se interrumpe la iteración y se iguala  $wc_i$  a  $t_i + \varepsilon_i$ . Eso supone truncar el tiempo borroso de finalización. La comparación con el plazo sigue dando el valor exacto de posibilidad y necesidad de que la tarea sea planificable.

**ALG BASICO:**

$$(7.9) \quad \left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k} \text{ and } s_{i,k} \leq t_i(0)) \quad k = k + 1 \\ \text{if } s_{i,k} = s_{i,k-1} \text{ then } wc_i = s_{i,k} + j_i \\ \text{else } wc_i = t_i(0) + \varepsilon_i \\ \text{dado, } s_{i,k}(s_{i,k-1}) = b_i + \sum_{h=1}^i p_h \cdot \left\lceil \frac{s_{i,k-1} + j_h}{t_h} \right\rceil \end{array} \right.$$

### 7.1.3.2. Prioridades fijas y servidor aperiódico

Cuando además de tareas periódicas el sistema de tiempo real está formado por tareas aperiódicas atendidas por un servidor aperiódico de tipo esporádico (SS) o aplazable (DS), se puede analizar su planificabilidad en presencia de información incierta, como tiempos de ejecución, plazos, periodos, capacidad del servidor y periodo del servidor borrosos. El ámbito de aplicación de los algoritmos de análisis de planificabilidad desarrollados en el apartado 5.8. es para plazos de las tareas periódicas menores o iguales que los periodos, por lo que la condición en términos borrosos es que la posibilidad y necesidad de que los plazos sean mayores que los periodos son nulas.

#### Servidor Esporádico

Cuando las tareas son atendidas por un Servidor Esporádico (SS) con periodo del servidor  $t_s$  y capacidad  $c_s$ , es posible extender el análisis mediante aritmética de intervalos aplicando (7.10).

$$\begin{array}{l}
 \text{ALG Aritmética de Intervalos} \\
 \underline{wc}_0(0) = 0 \\
 \text{for } i = 1 \text{ to } n\_tasks \text{ do} \\
 \quad \underline{wc}_i(0) = \text{ALG BASICO} \left( p_j = \underline{p}_j(0); c_s = \underline{c}_s(0); t_s = \underline{t}_s(0); t_j = \underline{t}_j(0); s\_in_i = \underline{wc}_{i-1}(0) + p_i \right) \\
 \quad \underline{wc}_i(1) = \text{ALG BASICO} \left( p_j = \underline{p}_j(1); c_s = \underline{c}_s(1); t_s = \underline{t}_s(1); t_j = \underline{t}_j(1); s\_in_i = \underline{wc}_i(0) \right) \\
 \quad \overline{wc}_i(1) = \text{ALG BASICO} \left( p_j = \overline{p}_j(1); c_s = \overline{c}_s(1); t_s = \overline{t}_s(1); t_j = \overline{t}_j(1); s\_in_i = \underline{wc}_i(1) \right) \\
 \quad \overline{wc}_i(0) = \text{ALG BASICO} \left( p_j = \overline{p}_j(0); c_s = \overline{c}_s(0); t_s = \overline{t}_s(0); t_j = \overline{t}_j(0); s\_in_i = \overline{wc}_i(1) \right) \\
 \text{ALG ZONA} \\
 \text{end} \\
 (7.10) \quad \Pi_{planif} / N_{planif} = \min \{ \Pi / N(WC_i \leq D_i) \} \quad \forall i
 \end{array}$$

Donde el algoritmo básico es:

$$(7.11) \quad \text{ALG BASICO:} \quad \left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_i = s_{i,k} \\ \text{dado, } s_{i,k} = p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil + c_s \cdot \left\lceil \frac{s_{i,k-1}}{t_s} \right\rceil \end{array} \right.$$

El algoritmo ZONA (7.6), es el mismo que se mostró en el apartado de prioridades fijas y plazos menores o iguales que los periodos. Los algoritmos Bipartición Inferior (7.12) y Bipartición Superior (7.13) son:

$$\begin{aligned}
 & \textbf{Bipartición inf} \\
 & \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_i(0), \text{cota\_sup} = \underline{wc}_i(1) \\
 & \text{while } \left( \left| \underline{wc}_i(\alpha) - \bar{d}_i(\alpha) \right| > \varepsilon \right) \\
 & \quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 & \quad \underline{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(\alpha); c_s = \underline{c}_s(\alpha); t_s = \bar{t}_s(\alpha); t_j = \bar{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \hline 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 & \quad \text{if } \left( \underline{wc}_i(\alpha) < \bar{d}_i(\alpha) \right) \quad \alpha_0 = \alpha, \quad \text{cota\_inf} = \underline{wc}_i(\alpha) \\
 & \quad \text{if } \left( \underline{wc}_i(\alpha) > \bar{d}_i(\alpha) \right) \quad \alpha_1 = \alpha, \quad \text{cota\_sup} = \underline{wc}_i(\alpha) \\
 & \text{end} \\
 & \Pi(WC_i \leq D_i) = \alpha; \quad N(WC_i \leq D_i) = 0
 \end{aligned}
 \tag{7.12}$$

$$\begin{aligned}
 & \textbf{Bipartición sup} \\
 & \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \bar{wc}_i(1), \text{cota\_sup} = \bar{wc}_i(0) \\
 & \text{while } \left( \left| \bar{wc}_i(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right) \\
 & \quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 & \quad \bar{wc}_i(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \bar{p}_j(\alpha); c_s = \bar{c}_s(\alpha); t_s = \underline{t}_s(\alpha); t_j = \underline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ \hline 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
 & \quad \text{if } \left( \bar{wc}_i(\alpha) < \underline{d}_i(\alpha) \right) \quad \alpha_1 = \alpha, \quad \text{cota\_inf} = \bar{wc}_i(\alpha) \\
 & \quad \text{if } \left( \bar{wc}_i(\alpha) > \underline{d}_i(\alpha) \right) \quad \alpha_0 = \alpha, \quad \text{cota\_sup} = \bar{wc}_i(\alpha) \\
 & \text{end} \\
 & \Pi(WC_i \leq D_i) = 1; \quad N(WC_i \leq D_i) = 1 - \alpha
 \end{aligned}
 \tag{7.13}$$

Si todos los tiempos son numéricos, se aplica el algoritmo clásico (7.11) donde  $s\_in_i = wc_{i-1} + p_i$

### Servidor Aplazable

Si las tareas aperiódicas son atendidas por un Servidor Aplazable (DS), se propone utilizar el algoritmo basado en aritmética de intervalos con búsqueda por bipartición, que es el mismo que aparece en (7.10), donde el algoritmo básico es:

$$(7.14) \quad \text{ALG BASICO:} \quad \left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_i = s_{i,k} \\ \text{dado, } s_{i,k} = p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1}}{t_h} \right\rceil + c_s \cdot \left( 1 + \left\lceil \frac{s_{i,k-1} - c_s}{t_s} \right\rceil \right) \end{array} \right.$$

El algoritmo ZONA (7.6) es el mismo que para prioridades fijas y plazos menores o iguales que los periodos. Los algoritmos Bipartición Inferior (7.12) y Bipartición Superior (7.13) coinciden con los propuestos para el servidor SS. Al igual que con el servidor anterior, si todos los tiempos son numéricos, se aplica el algoritmo clásico (7.14) donde  $s\_in_i = wc_{i-1} + p_i$

### 7.1.3.3. Tareas con relación de precedencia

Si una transacción formada por  $n$  tareas con relación de precedencia se implementa de manera que la primera tarea es periódica y el resto son esporádicas y lanzadas por la tarea que les precede al finalizar (ver el apartado 2.6 de esta tesis), se puede modelar suponiendo que todas se lanzan en el mismo instante, pero que se ejecutan en el orden correcto porque las tareas predecesoras tienen prioridades más altas. Entonces, el tiempo total de respuesta de la transacción es igual al tiempo de respuesta de la tarea esporádica final, con lo que se debe cumplir que  $wc_n \leq d$ . Aquí el valor de  $wc_n$  se mide relativo al comienzo de la transacción e incluye los tiempos de finalización de las tareas anteriores.

Por tanto la transacción se modela de forma que está compuesta por  $n$  tareas con periodos iguales y prioridades consecutivas, o bien por una tarea periódica con tiempo de ejecución igual a la suma de los tiempos de ejecución de las  $n$  tareas de la transacción y de prioridad igual a la prioridad de la primera tarea de la transacción. Para realizar el análisis de planificabilidad hay que aplicar el mismo algoritmo de prioridades fijas con plazos menores que los periodos ya presentado.

### 7.1.3.4. Prioridades fijas, con plazos mayores que los periodos

Cuando los plazos son mayores que los periodos es posible que no baste con calcular la primera ejecución de cada tarea, ya que el peor tiempo de respuesta se da dentro del periodo de ocupación con lo que es preciso calcular el número de ejecuciones correspondiente dicho periodo.

Tiempos de ejecución, plazos, bloqueos y “release jitter” borrosos

El algoritmo basado en aritmética de intervalos con búsqueda por bipartición que tiene en cuenta además la existencia de tiempos de bloqueo y de “release jitter” borrosos es (7.15).

**ALG Aritmética de Intervalos**

$$\underline{wc}_{0,1}(0) = 0$$

for  $i = 1$  to  $n\_tareas$  do

$$\underline{wc}_{i,0}(0) = \underline{wc}_{i-1,1}(0); \quad n = 1$$

loop

$$\underline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(0); b_i = \underline{b}_i(0); j_j = j_j(0); s\_in_i = \underline{wc}_{i,n-1}(0) + p_i \\ 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\underline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = p_j(1); b_i = \underline{b}_i(1); j_j = j_j(1); s\_in_i = \underline{wc}_{i,n}(0) \\ 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); s\_in_i = \underline{wc}_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

$$\overline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); s\_in_i = \overline{wc}_{i,n}(1) \\ 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$$

ALG ZONA

if  $wc_{i,n}^c > n \cdot t_i$

$$n = n + 1$$

else

$$\Pi_i / N_i = \min \{ \Pi / N (WC_{i,x} \leq D_{i,x}) \} \quad \forall x \in \mathbb{N} | 1 \leq x \leq n; \quad \text{siguiente tarea}$$

end loop

end

$$\Pi_{planif} / N_{planif} = \min \{ \Pi_i / N_i \} \quad \forall i \in \mathbb{N} | 1 \leq i \leq n\_tareas$$

(7.15)

Donde  $D_{i,n} = (n-1) \cdot t_i + D_i$  y el algoritmo básico es:

**ALG BASICO:**

$$(7.16) \quad \left\{ \begin{array}{l} s_{i,0} = s\_in_i \\ \text{while } (s_{i,k-1} < s_{i,k}) \quad k = k + 1 \\ wc_{i,n} = s_{i,k} + j_i \\ \text{dado, } s_{i,k} = b_i + n \cdot p_i + \sum_{h=1}^{i-1} p_h \cdot \left\lceil \frac{s_{i,k-1} + j_h}{t_h} \right\rceil \end{array} \right.$$

Este algoritmo básico (7.16) es el que habría que aplicar si todos los parámetros son números clásicos, donde el valor inicial de  $s$  es  $s_{in_i} = wc_{i-1} + p_i$

El algoritmo ZONA (7.17) establece la zona de búsqueda por bipartición, que puede ser en la zona de límites inferiores (7.18) o de límites superiores (7.19) de los  $\alpha$ -cortes de los tiempos de finalización.

**ALG ZONA**

(7.17) if ( $\underline{d}_{i,n}(0) \geq \overline{wc}_{i,n}(0)$ )  $\Pi(WC_{i,n} \leq D_{i,n}) = 1$ ,  $N(WC_{i,n} \leq D_{i,n}) = 1$ ,  
if ( $\overline{wc}_{i,n}(0) \geq nt_i$ )  $wc_{i,n}^c = nt_i + \delta$   
else  $wc_{i,n}^c = 0$

if ( $\overline{d}_{i,n}(0) \leq \underline{wc}_{i,n}(0)$ )  $\Pi(WC_{i,n} \leq D_{i,n}) = 0$ ,  $N(WC_{i,n} \leq D_{i,n}) = 0$ ,  $wc_{i,n}^c = 0$

if ( $\overline{wc}_{i,n}(1) \geq \underline{d}_{i,n}(1)$ )  $N(WC_{i,n} \leq D_{i,n}) = 0$   
else if ( $\overline{wc}_{i,n}(1) < \underline{d}_{i,n}(1)$ ) Bipartición sup

if ( $\underline{wc}_{i,n}(1) \leq \overline{d}_{i,n}(1)$ )  $\Pi(WC_{i,n} \leq D_{i,n}) = 1$   
else if ( $\underline{wc}_{i,n}(1) > \overline{d}_{i,n}(1)$ ) Bipartición inf

Donde  $wc_{i,n}^c$  es el instante del punto de corte, si existe, entre  $WC_{i,n}$  y  $D_{i,n}$ . Se entiende por punto de corte el punto necesario para calcular los valores de posibilidad  $\Pi(WC_{i,n} \leq D_{i,n})$  y necesidad  $N(WC_{i,n} \leq D_{i,n})$ , como se indica en la Figura 7.4.

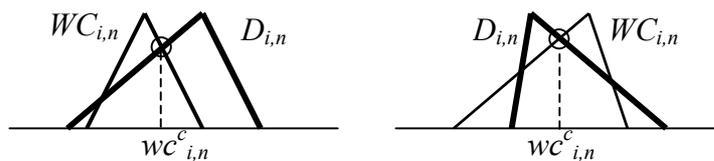


Figura 7.4: Punto de corte  $wc_{i,n}^c$  que se obtiene de la comparación entre  $WC_{i,n}$  y  $D_{i,n}$

**Bipartición inf**

$$(7.18) \quad \begin{aligned} &\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_{i,n}(0), \text{cota\_sup} = \underline{wc}_{i,n}(1) \\ &\text{while } \left( \left| \underline{wc}_{i,n}(\alpha) - \overline{d}_{i,n}(\alpha) \right| > \varepsilon \right) \\ &\quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\ &\quad \underline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = p_j(\alpha); b_i = \underline{b}_i(\alpha); j_j = j_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\ &\quad \text{if } \left( \underline{wc}_{i,n}(\alpha) < \overline{d}_{i,n}(\alpha) \right) \quad \alpha_0 = \alpha, \text{cota\_inf} = \underline{wc}_{i,n}(\alpha) \\ &\quad \text{if } \left( \underline{wc}_{i,n}(\alpha) > \overline{d}_{i,n}(\alpha) \right) \quad \alpha_1 = \alpha, \text{cota\_sup} = \underline{wc}_{i,n}(\alpha) \\ &\text{end} \\ &\Pi(WC_{i,n} \leq D_{i,n}) = \alpha; \quad N(WC_{i,n} \leq D_{i,n}) = 0; \quad wc_{i,n}^c = \underline{wc}_{i,n}(\alpha) \end{aligned}$$

**Bipartición sup**

$$(7.19) \quad \begin{aligned} &\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{wc}_{i,n}(1), \text{cota\_sup} = \overline{wc}_{i,n}(0) \\ &\text{while } \left( \left| \overline{wc}_{i,n}(\alpha) - \underline{d}_{i,n}(\alpha) \right| > \varepsilon \right) \\ &\quad \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\ &\quad \overline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(\alpha); b_i = \overline{b}_i(\alpha); j_j = \overline{j}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\ &\quad \text{if } \left( \overline{wc}_{i,n}(\alpha) < \underline{d}_{i,n}(\alpha) \right) \quad \alpha_1 = \alpha, \text{cota\_inf} = \overline{wc}_{i,n}(\alpha) \\ &\quad \text{if } \left( \overline{wc}_{i,n}(\alpha) > \underline{d}_{i,n}(\alpha) \right) \quad \alpha_0 = \alpha, \text{cota\_sup} = \overline{wc}_{i,n}(\alpha) \\ &\text{end} \\ &\Pi(WC_{i,n} \leq D_{i,n}) = 1; \quad N(WC_{i,n} \leq D_{i,n}) = 1 - \alpha; \quad wc_{i,n}^c = \overline{wc}_{i,n}(\alpha) \end{aligned}$$

### Periodos borrosos

La extensión del análisis de planificabilidad a periodos borrosos requiere trabajar con los tiempos de respuesta de las tareas en lugar de con los tiempos de finalización para evitar dependencias funcionales entre parámetros.

El algoritmo basado en aritmética de intervalos, para plazos mayores que los periodos, con tiempos de ejecución, plazos, tiempos de bloqueo, “release jitter” y periodos borrosos es:

$$\begin{aligned}
& \text{ALG Aritmética de Intervalos} \\
& \underline{wc}_{0,1}(0) = 0 \\
& \text{for } i = 1 \text{ to } n\_tarefas \text{ do} \\
& \quad \underline{wc}_{i,0}(0) = \underline{wc}_{i-1,1}(0) ; \quad n = 1 \\
& \quad \text{loop} \\
& \quad \quad \underline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \underline{p}_j(0); b_i = \underline{b}_i(0); j_j = \underline{j}_j(0); t_j = \underline{t}_j(0); s\_in_i = \underline{wc}_{i,n-1}(0) + p_i \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
& \quad \quad \underline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \underline{p}_j(1); b_i = \underline{b}_i(1); j_j = \underline{j}_j(1); t_j = \underline{t}_j(1); s\_in_i = \underline{wc}_{i,n}(0) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
& \quad \quad \overline{wc}_{i,n}(1) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(1); b_i = \overline{b}_i(1); j_j = \overline{j}_j(1); t_j = \overline{t}_j(1); s\_in_i = \overline{wc}_{i,n}(1) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
& \quad \quad \overline{wc}_{i,n}(0) = \text{ALG BASICO} \left( \begin{array}{l} p_j = \overline{p}_j(0); b_i = \overline{b}_i(0); j_j = \overline{j}_j(0); t_j = \overline{t}_j(0); s\_in_i = \overline{wc}_{i,n}(1) \\ 1 \leq j \leq i \quad 1 \leq j \leq i \end{array} \right) \\
& \quad \quad \text{ALG ZONA} \\
& \quad \quad \text{if } [e_{i,n}^c > \underline{t}_i(0)] \quad n = n + 1 \\
& \quad \quad \text{else } \Pi_i / N_i = \min \{ \Pi / N (E_{i,x} \leq D_i) \} \quad \forall x \in \mathbb{N} \mid 1 \leq x \leq n ; \quad \text{siguiente tarea} \\
& \quad \quad \text{end loop} \\
& \text{end} \\
& \Pi_{planif} / N_{planif} = \min \{ \Pi_i / N_i \} \quad \forall i \in \mathbb{N} \mid 1 \leq i \leq n\_tarefas
\end{aligned}
\tag{7.20}$$

El algoritmo básico es el mismo que aparece en (7.16), y el algoritmo ZONA que establece la zona de búsqueda por bipartición es:

$$\begin{aligned}
& \text{ALG ZONA} \\
& \underline{e}_{i,n}(0) = \underline{wc}_{i,n}(0) - (n-1)\underline{t}_i(0) ; \quad \overline{e}_{i,n}(1) = \overline{wc}_{i,n}(1) - (n-1)\overline{t}_i(1) \\
& \underline{e}_{i,n}(1) = \underline{wc}_{i,n}(1) - (n-1)\underline{t}_i(1) ; \quad \overline{e}_{i,n}(0) = \overline{wc}_{i,n}(0) - (n-1)\overline{t}_i(0) \\
& \text{if } (\underline{d}_i(0) \geq \underline{e}_{i,n}(0)) \quad \Pi(E_{i,n} \leq D_i) = 1, \quad N(E_{i,n} \leq D_i) = 1, \\
& \quad \quad \text{if } [\overline{e}_{i,n}(0) > \underline{t}_i(0)] \quad e_{i,n}^c = \underline{t}_i(0) + \delta \\
& \quad \quad \text{else } e_{i,n}^c = 0 \\
& \text{if } (\overline{d}_i(0) \leq \underline{e}_{i,n}(0)) \quad \Pi(E_{i,n} \leq D_i) = 0, \quad N(E_{i,n} \leq D_i) = 0, \quad e_{i,n}^c = 0 \\
& \text{if } (\overline{e}_{i,n}(1) \geq \underline{d}_i(1)) \quad N(E_{i,n} \leq D_i) = 0 \\
& \quad \quad \text{else if } (\overline{e}_{i,n}(1) < \underline{d}_i(1)) \quad \text{Bipartición sup} \\
& \text{if } (\underline{e}_{i,n}(1) \leq \overline{d}_i(1)) \quad \Pi(E_{i,n} \leq D_i) = 1 \\
& \quad \quad \text{else if } (\underline{e}_{i,n}(1) > \overline{d}_i(1)) \quad \text{Bipartición inf}
\end{aligned}
\tag{7.21}$$

Los algoritmos de búsqueda por bipartición en la zona de límites inferiores (7.22) y de límites superiores (7.23) de los  $\alpha$ -cortes de los tiempos de finalización son:

**Bipartición inf**

$\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{wc}_{i,n}(0), \text{cota\_sup} = \underline{wc}_{i,n}(1)$

while  $\left( \left| \underline{e}_{i,n}(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right)$

(7.22)  $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$

$\underline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = \underline{p}_j(\alpha); b_i = \underline{b}_i(\alpha); j_j = \underline{j}_j(\alpha); t_j = \underline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$

if  $\left( \underline{e}_{i,n}(\alpha) < \underline{d}_i(\alpha) \right) \alpha_0 = \alpha, \text{cota\_inf} = \underline{wc}_{i,n}(\alpha)$

if  $\left( \underline{e}_{i,n}(\alpha) > \underline{d}_i(\alpha) \right) \alpha_1 = \alpha, \text{cota\_sup} = \underline{wc}_{i,n}(\alpha)$

end

$\Pi(E_{i,n} \leq D_i) = \alpha; \quad N(E_{i,n} \leq D_i) = 0; \quad e_{i,n}^c = \underline{e}_{i,n}(\alpha)$

**Bipartición sup**

$\alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{wc}_{i,n}(1), \text{cota\_sup} = \overline{wc}_{i,n}(0)$

while  $\left( \left| \overline{e}_{i,n}(\alpha) - \underline{d}_i(\alpha) \right| > \varepsilon \right)$

(7.23)  $\alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2}$

$\overline{wc}_{i,n}(\alpha) = \text{ALG BASICO} \left( \begin{matrix} p_j = \overline{p}_j(\alpha); b_i = \overline{b}_i(\alpha); j_j = \overline{j}_j(\alpha); t_j = \overline{t}_j(\alpha); s\_in_i = \text{cota\_inf} \\ 1 \leq j \leq i & 1 \leq j \leq i \end{matrix} \right)$

if  $\left( \overline{e}_{i,n}(\alpha) < \underline{d}_i(\alpha) \right) \alpha_1 = \alpha, \text{cota\_inf} = \overline{wc}_{i,n}(\alpha)$

if  $\left( \overline{e}_{i,n}(\alpha) > \underline{d}_i(\alpha) \right) \alpha_0 = \alpha, \text{cota\_sup} = \overline{wc}_{i,n}(\alpha)$

end

$\Pi(E_{i,n} \leq D_i) = 1; \quad N(E_{i,n} \leq D_i) = 1 - \alpha; \quad e_{i,n}^c = \overline{e}_{i,n}(\alpha)$

### 7.1.3.5. Prioridades dinámicas. EDF

#### Plazos iguales a los periodos

Cuando las tareas son periódicas con prioridades dinámicas asignadas según EDF, si los plazos son iguales a los periodos es posible extender el análisis de planificabilidad a tiempos de ejecución y periodos borrosos.

La planificabilidad borrosa es:

$$(7.24) \quad \Pi_{planif} = \Pi \left( \sum_{i=1}^{n\_tar} \frac{P_i}{T_i} \leq 1 \right) \quad N_{planif} = N \left( \sum_{i=1}^{n\_tar} \frac{P_i}{T_i} \leq 1 \right)$$

De esta forma se compara el número borroso que se obtiene de la utilización total del procesador con la cota superior que puede tomar dicha utilización, que es un número clásico.

Si existen tareas estrictas y los valores de posibilidad y necesidad de que el sistema sea planificable son distintos de 1, es preciso aplicar el algoritmo basado en la demanda del procesador (7.25) para determinar si las tareas estrictas verifican sus plazos con posibilidad y necesidad igual a 1.

### Plazos menores que los periodos

El primer paso consiste en comprobar  $\Pi(\sum_{i=1}^{n_{tar}} \frac{P_i}{T_i} \leq 1)$  y  $N(\sum_{i=1}^{n_{tar}} \frac{P_i}{T_i} \leq 1)$ , ya que si son cero, la posibilidad y necesidad de que el sistema sea planificable es cero (por ser una condición necesaria pero no suficiente). En caso contrario, para tiempos de ejecución borrosos se puede extender el análisis de planificabilidad mediante aritmética de intervalos (7.25).

	<b>ALG Aritmética de Intervalos</b>
	$wc_0(0) = 0$
	$\forall l_{i,k} \in D$
(7.25)	$\underline{c}_p(0) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(0); \quad \underline{c}_p(1) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(1)$
	$\overline{c}_p(1) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(1); \quad \overline{c}_p(0) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(0)$
	ALG ZONA
	end
	$\Pi_{planif} / N_{planif} = \min \{ \Pi(C_p(l_{i,k}) \leq l_{i,k}) / N(C_p(l_{i,k}) \leq l_{i,k}) \} \quad \forall i, k$

donde  $D = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(b_p, h), 1 \leq i \leq n, k \geq 0\}$ , y  $b_p$  es el periodo de ocupación y  $h$  es el hiperperiodo. El periodo de ocupación, que en este caso es un valor numérico, se calcula como:

$$\begin{array}{l}
 \text{Periodo de ocupación} \\
 l^* = \sum_{i=1}^n p_i ; \quad l' = w_p(l^*) \\
 h = \text{lcm}(t_1, t_2, \dots, t_n) \\
 (7.26) \quad \text{while } (l' \neq l^*) \text{ and } (l' \leq h) \{ \\
 \quad l^* = l' \\
 \quad l' = w_p(l^*) \} \\
 \text{if } (l' \leq h) \quad b_p = l^* \\
 \text{else } b_p = \infty \\
 \text{donde } w_p(l^*) = \sum_{i=1}^n \left\lceil \frac{l^*}{t_i} \right\rceil \cdot p_i
 \end{array}$$

El algoritmo que determina la zona de búsqueda por bipartición del punto de corte entre  $C_p$  y  $l_{i,k}$  es:

$$\begin{array}{l}
 \text{ALG ZONA} \\
 \text{if } (l_{i,k} \geq \overline{c}_p(0)) \quad \Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1, \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 1, \\
 (7.27) \quad \text{if } (l_{i,k} \leq \underline{c}_p(0)) \quad \Pi(C_p(l_{i,k}) \leq l_{i,k}) = 0, \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 0, \\
 \text{if } (\overline{c}_p(1) \geq l_{i,k}) \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 0 \\
 \quad \text{else if } (\overline{c}_p(1) < l_{i,k}) \quad \text{Bipartición sup} \\
 \text{if } (\underline{c}_p(1) \leq l_{i,k}) \quad \Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1 \\
 \quad \text{else if } (\underline{c}_p(1) > l_{i,k}) \quad \text{Bipartición inf}
 \end{array}$$

Y los algoritmos de búsqueda en la zona de límites inferiores (7.28) y límites superiores (7.29) de los  $\alpha$ -cortes de  $C_p$  son:

$$\begin{array}{l}
 \text{Bipartición inf} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \underline{c}_p(0), \text{cota\_sup} = \underline{c}_p(1) \\
 \text{while } \left( \left| \underline{c}_p(\alpha) - l_{i,k} \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \underline{c}_p(\alpha) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \underline{p}_j(\alpha) \\
 \text{if } \left( \underline{c}_p(\alpha) < l_{i,k} \right) \quad \alpha_0 = \alpha, \quad \text{cota\_inf} = \underline{c}_p(\alpha) \\
 \text{if } \left( \underline{c}_p(\alpha) > l_{i,k} \right) \quad \alpha_1 = \alpha, \quad \text{cota\_sup} = \underline{c}_p(\alpha) \\
 \text{end} \\
 \Pi(C_p(l_{i,k}) \leq l_{i,k}) = \alpha; \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 0
 \end{array}
 \tag{7.28}$$

$$\begin{array}{l}
 \text{Bipartición sup} \\
 \alpha_0 = 0, \alpha_1 = 1, \alpha = 0, \text{cota\_inf} = \overline{c}_p(1), \text{cota\_sup} = \overline{c}_p(0) \\
 \text{while } \left( \left| \overline{c}_p(\alpha) - l_{i,k} \right| > \varepsilon \right) \\
 \alpha = \alpha_0 + \frac{\alpha_1 - \alpha_0}{2} \\
 \overline{c}_p(\alpha) = \sum_{j=1}^n \left( \left\lfloor \frac{l_{i,k} - d_j}{t_j} \right\rfloor + 1 \right) \cdot \overline{p}_j(\alpha) \\
 \text{if } \left( \overline{c}_p(\alpha) < l_{i,k} \right) \quad \alpha_1 = \alpha, \quad \text{cota\_inf} = \overline{c}_p(\alpha) \\
 \text{if } \left( \overline{c}_p(\alpha) > l_{i,k} \right) \quad \alpha_0 = \alpha, \quad \text{cota\_sup} = \overline{c}_p(\alpha) \\
 \text{end} \\
 \Pi(C_p(l_{i,k}) \leq l_{i,k}) = 1; \quad N(C_p(l_{i,k}) \leq l_{i,k}) = 1 - \alpha
 \end{array}
 \tag{7.29}$$

Si todos los parámetros son numéricos hay que comprobar que se verifica:

$$(7.30) \quad \forall l \in \mathcal{D} \quad l \geq \sum_{i=1}^n \left( \left\lfloor \frac{l - d_i}{t_i} \right\rfloor + 1 \right) \cdot p_i$$

donde  $\mathcal{D} = \{d_{i,k} \mid d_{i,k} = k \cdot t_i + d_i, d_{i,k} \leq \min(B_P, h), 1 \leq i \leq n, k \geq 0\}$

No se ha encontrado una extensión del análisis de planificabilidad basada en aritmética borrosa o en aritmética de intervalos cuando los plazos y los periodos son

borrosos debido a la dependencia funcional existente entre  $c_p$  y  $l$ , por lo que se hace necesario recurrir al principio de extensión.

#### 7.1.4. Modificación del diseño

Una vez concluido el análisis de planificabilidad del sistema hay que analizar los resultados y, si procede, proponer cambios en el modelo, para volver a realizar el análisis de planificabilidad del nuevo modelo, en la fase 3 de la metodología. Al final de la fase 3, se pueden presentar dos situaciones diferentes: que el sistema no sea planificable, con lo que hay que proponer cambios con el fin de conseguir un diseño de sistema que cumpla las restricciones temporales, o que el sistema sea planificable, en cuyo caso se puede estudiar la introducción de alguna mejora en el modelo.

##### *Sistema no planificable*

Cuando el sistema no es planificable se puede modificar el criterio de asignación de prioridades si se considera que puede mejorar la planificabilidad del sistema. Si en el modelo analizado en la fase anterior los plazos eran numéricos y se habían asignado las prioridades fijas de acuerdo con el algoritmo DMS, no habrá mejora alguna en la planificabilidad del sistema si se cambian las prioridades estáticas, ya que DMS es óptimo. En cambio, sí se podría analizar el modelo basado en prioridades dinámicas asignadas mediante el algoritmo EDF, ya que éste mejora la planificabilidad del sistema. Sin embargo, si hay tareas estrictas y tareas flexibles y con la asignación DMS una tarea estricta no es planificable, habría que intercambiar su prioridad con una tarea flexible de mayor prioridad según DMS.

Otra posibilidad es introducir más información acerca de los parámetros. Es posible que exista información actualizada acerca de los parámetros numéricos, que habría que modificar en el modelo. Si se dispone de más información se podría modelar alguno de los parámetros mediante distribuciones de posibilidad, o incluso, en caso de tener datos estadísticos podría modelarse alguno de ellos mediante distribuciones de probabilidad. Con todo ello se analizaría la planificabilidad del sistema en la fase 3 de la metodología.

Si las modificaciones anteriores no son posibles, o bien ya se han introducido y a pesar de ello el sistema continúa siendo no planificable, hay que explorar otras modificaciones que conllevan desventajas. Se puede aumentar el periodo de alguna tarea, perdiéndose funcionalidad y/o requisitos temporales del sistema, lo que reduce la interferencia de dicha tarea en las menos prioritarias, disminuyendo sus tiempos de

respuesta. Por otro lado, una disminución del tiempo de ejecución de alguna tarea, permitiría mejorar la planificabilidad del sistema a costa de empeorar la precisión del resultado de la tarea y la funcionalidad del sistema. De forma similar, si se reduce el número de tareas no críticas a ejecutar, mejoraría la planificabilidad empeorando el cumplimiento de requisitos funcionales no críticos.

### *Sistema planificable*

Otra posibilidad es que, aún cumpliendo todas las restricciones temporales, se pueda modificar el diseño con el fin de mejorar sus prestaciones funcionales o sus tiempos de respuesta sin que se vea comprometido el cumplimiento de las restricciones temporales. Para ello se puede analizar la reducción del periodo de alguna tarea si interesa que se ejecute más a menudo, o aumentar el tiempo de ejecución si con ello se consigue mayor precisión en el resultado, o aumentar el número de tareas a ejecutar ampliando de esta forma la funcionalidad del sistema con tareas no previstas inicialmente.

En resumen, se ha presentado en este apartado una metodología de diseño de sistemas de tiempo real centrada en el objetivo de lograr que se respeten las restricciones temporales, en la que los parámetros temporales se pueden modelar mediante números borrosos que se interpretan como distribuciones de posibilidad. Esta metodología se aplica al caso ejemplo que se describe en el siguiente apartado, donde se modelan el sistema y sus parámetros en primer lugar mediante números clásicos y después se introducen modificaciones para lograr que el sistema sea planificable consistentes en cambiar la asignación de prioridades fijas de las tareas y modelar algunos parámetros mediante conjuntos borrosos.

## 7.2. Caso ejemplo

### 7.2.1. Descripción del entorno, del sistema, del interfaz y requisitos del sistema

En este apartado se presenta un caso ejemplo basado en un sistema ferroviario de protección automática (ATP: Automatic Train Protection). El objetivo es, partiendo de la descripción temporal del sistema, proponer un modelo del mismo y realizar el análisis temporal. En primer lugar se han escogido números clásicos para modelar los parámetros temporales del sistema. Los números se han escogido para que el ejemplo sea ilustrativo y se supondrán adimensionales. Los valores numéricos escogidos representan el peor caso para cada uno de los parámetros. El análisis temporal correspondiente da el resultado de planificabilidad del sistema en el caso más desfavorable.

En segundo lugar se propone una definición más realista (al modelar la incertidumbre) de los parámetros, mediante números borrosos y se realiza el análisis borroso de planificabilidad.

#### Sistema de protección automática

El sistema de protección automática de un tren es un sistema de seguridad que se ocupa de mantener el nivel de integridad del sistema, asegurando el correcto funcionamiento, y en caso de fallo detiene al tren, que es su estado seguro. Es el que maneja las restricciones de velocidad en cada tramo de vía, el estado de las puertas y de los frenos, y lleva cuenta de la posición y velocidad del tren en cada momento. Este sistema supervisa en todo momento la circulación del tren, ya sea manual o automática y, si detecta un fallo grave, un sobrepaso de la velocidad máxima o pérdida de comunicación, detiene al tren. Existe redundancia para aumentar la fiabilidad y la disponibilidad del sistema (ver Figura 7.5).

El sistema de protección automática del tren tiene una parte embarcada en el tren que consta del módulo principal, el módulo de posición y velocidad, el módulo de comunicaciones, el módulo de entradas y salidas discretas, y el módulo de recepción de códigos de velocidad.

El módulo principal es el que controla al resto de módulos, vigilando si su funcionamiento es correcto y en caso contrario enviando la orden de terminar a todos los módulos.

El módulo de posición y velocidad mide la posición y velocidad del tren en cada momento y mide distancias recorridas bajo petición, a partir del instante en que recibe la petición.

El módulo de comunicaciones envía y recibe información de los sistemas redundantes con el fin de comparar dicha información, que debe coincidir en todo momento.

El módulo de entradas y salidas discretas se ocupa del control de las puertas y de los frenos: en caso de fallo la salida que indican es la correspondiente a puertas cerradas y freno activado.

El módulo de recepción de códigos de velocidad recibe la señal modulada, que es transmitida por la vía, la decodifica, y el código resultante lo envía al módulo principal. En este ejemplo se ha escogido el módulo de recepción de códigos de velocidad y se va a modelar una de sus partes, ya que se trata de un sistema de tiempo real, con la suficiente riqueza funcional como para presentar un ejemplo ilustrativo.

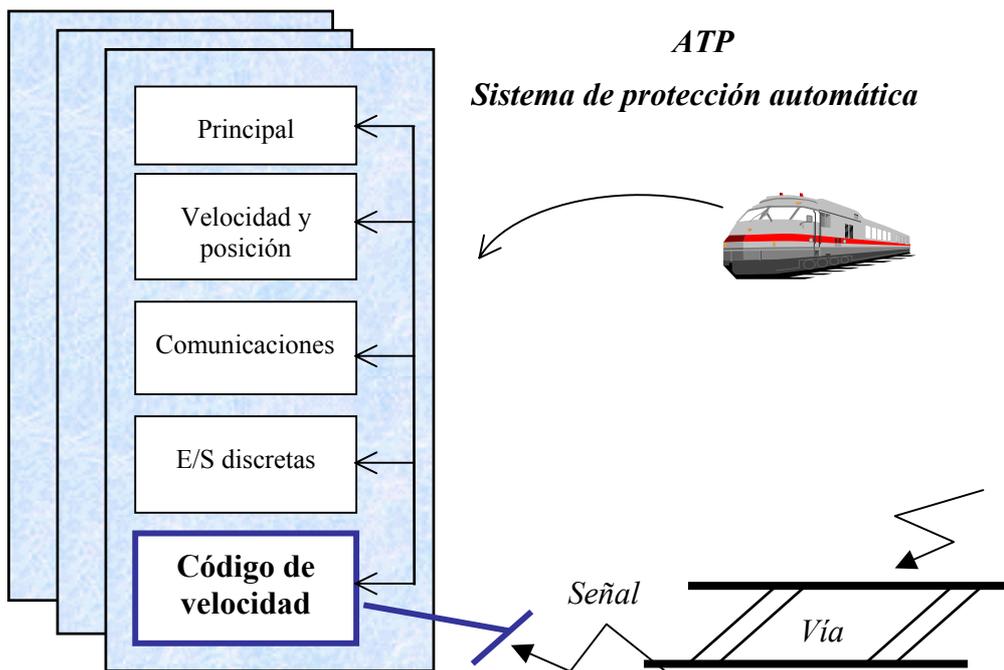


Figura 7.5: Sistema de protección automática embarcado

**Módulo de recepción de códigos de velocidad. Procesador principal.**

El módulo de recepción de códigos de velocidad del ATP contiene dos procesadores. Por un lado, el procesador principal recibe el espectro de la señal que le envía el procesador secundario (ver la Figura 7.6). Por otro, el procesador secundario recibe la señal modulada captada por la antena del tren y obtiene el espectro en frecuencia de dicha señal, que envía al procesador principal.

A partir del espectro el sistema correspondiente al procesador principal tiene que determinar los valores de portadora y moduladora de la señal. Esa pareja (portadora, moduladora) define una velocidad máxima permitida del tren que tiene un código asociado, y el código de velocidad es enviado por el procesador principal de este módulo de código de velocidad, al módulo principal. Si hay pérdida de la señal, se cuenta el espacio recorrido por el tren y si recorre un cierto umbral de distancia sin señal (hueco máximo), el sistema da la alarma al módulo principal, que se ocupa de enviar la orden de detención del tren. Valores por debajo sí están permitidos porque hay zonas sin señal, como por ejemplo la unión entre tramos de vía.

La cuenta del espacio recorrido la realiza el módulo de velocidad y posición del tren y comienza cuando el módulo de código de velocidad realiza una petición. Al llegar al umbral definido como distancia máxima sin señal, o hueco máximo, el módulo de velocidad y posición del tren envía un evento al módulo de código de velocidad.

En este caso ejemplo se va a analizar el sistema correspondiente al procesador principal.

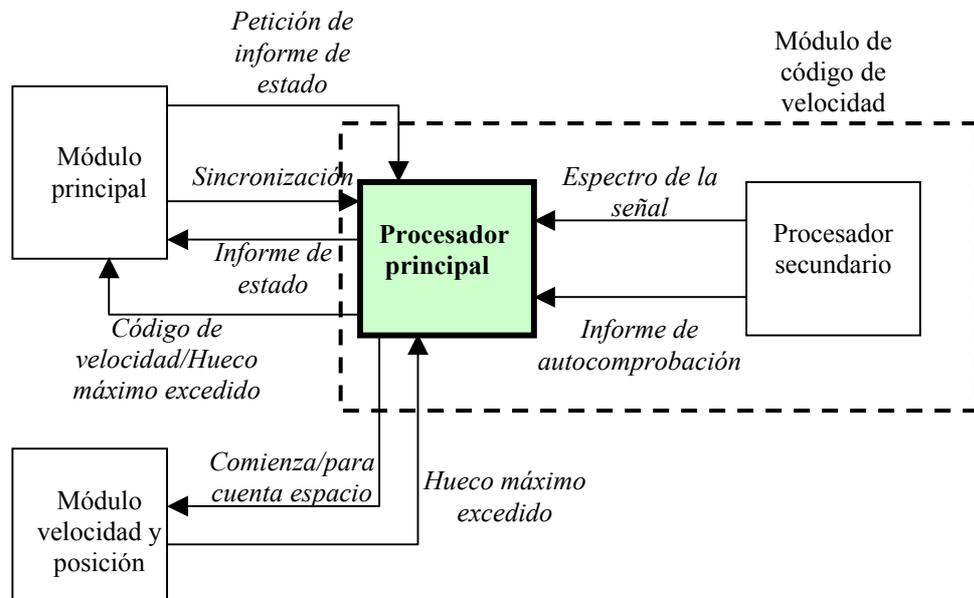


Figura 7.6: Interfaces del procesador principal del módulo de código de velocidad

En lo que sigue se usarán datos temporales medidos en unidades de tiempo (u.t.).

El análisis de la señal y obtención del código de velocidad se debe ejecutar cada 65 u.t., para que, en caso de pérdida de la señal, se detecte como máximo en ese tiempo y se pueda cumplir el requisito de distancia de frenado del tren desde el momento en el que se perdió la señal. Esta función es crítica, por lo que debe finalizar antes del lanzamiento de la siguiente ejecución.

Además, el sistema correspondiente al procesador principal del módulo de código de velocidad debe ejecutar funciones de autocomprobación periódicamente cada 110 u.t., salvo una de ellas que comprueba la zona de memoria en la que trabaja la tarea de análisis de la señal que se ejecuta con un periodo mayor que la anterior, 220 u.t., para que interfiera menos en la función crítica de análisis.

Del mismo modo, el procesador secundario realiza también funciones de autocomprobación como muy pronto cada 110 u.t., con lo que el procesador principal (que es el sistema analizado) debe recoger los informes de dicho procesador secundario.

Procedentes del módulo principal, llegan a este sistema (procesador principal del módulo de código de velocidad) dos tipos de eventos: eventos de sincronización, para sincronizar el módulo de código de velocidad, y eventos de petición de estado del módulo, para que el módulo de código de velocidad informe al módulo principal del estado en que se encuentra, y le envíe los informes de autocomprobación que estén disponibles.

La sincronización se debe ejecutar lo antes posible, y no debe finalizar más allá de 10 u.t. después de recibido el evento. El tiempo mínimo entre llegadas consecutivas del evento es 55 u.t.. El tiempo mínimo entre llegadas consecutivas del evento de petición de informe de estado del módulo es 110 u.t..

El requisito, tanto para la tarea de funciones de autocomprobación (sin memoria compartida) como la de recepción de informes del procesador secundario, es que debe ser posible que exista algún informe de cada una disponible antes de la llegada de la siguiente petición de estado por parte del módulo principal. La tarea de autocomprobación que comparte memoria con la tarea de análisis de la señal, al tener un periodo mayor, debe terminar antes de su periodo de 220 u.t., para que la frecuencia de envío de informes sea razonable.

Sólo se permite que exista una interrupción en el sistema que se va a diseñar ya que al ser un sistema crítico de seguridad se busca que sea lo más determinista posible.

## 7.2.2. Modelo de las tareas con tiempos numéricos y planificación del sistema

En este apartado se va a modelar el sistema y sus tareas con parámetros numéricos para realizar el análisis de planificabilidad en el peor caso, para una asignación de prioridades fijas definida de acuerdo con el algoritmo DMS. A partir de los resultados obtenidos se observará que puede resultar interesante modificar el diseño cambiando las prioridades de dos tareas. Así pues, en el siguiente apartado se realizará el análisis de planificabilidad con esa nueva asignación de prioridades y se obtendrán conclusiones.

### 7.2.2.1. Diseño de las tareas

De acuerdo con la descripción del sistema y de sus requisitos funcionales y temporales realizada en la fase anterior de la metodología, se pueden identificar las siguientes tareas:

- Tarea de análisis de la señal y obtención del código de velocidad
- Tarea de sincronización
- Tarea de autocomprobación
- Tarea de autocomprobación en zona de memoria compartida
- Tarea de envío de informe de estado al módulo principal
- Tarea de recepción de informes del procesador secundario
- Además, se precisa otra tarea que se ocupa de actualizar el reloj del sistema

Las tareas que se ejecutan a petición del módulo principal, la sincronización y el envío de informe de estado, pueden considerarse como esporádicas. Las tareas de análisis de la señal y obtención del código de velocidad, de ejecución de funciones de autocomprobación, de recogida de informes de autocomprobación del otro procesador y la tarea de reloj pueden considerarse como periódicas.

Dado que se está presentando el análisis de planificabilidad que se realizaría durante la fase de diseño de un sistema de tiempo real, la incertidumbre está presente en el conocimiento que se tiene acerca del sistema. Como primera aproximación, los parámetros del modelo utilizado son números clásicos por lo que deben representar el caso más desfavorable desde el punto de vista de la planificabilidad del sistema. Así, para los tiempos de ejecución se deben proporcionar los máximos tiempo de

ejecución que se consideran posibles, y lo mismo con los tiempos de utilización de la memoria compartida y el “release jitter”. Por otro lado, el caso más desfavorable se da para los mínimos valores de los periodos y de los plazos. Como resultado del análisis se obtiene la planificabilidad del sistema en el peor caso.

#### *Tarea de sincronización*

A partir de los requisitos del sistema, la tarea de sincronización se puede considerar como una tarea esporádica con tiempo mínimo entre eventos consecutivos de 55 u.t., con lo que se propone modelarla como una tarea periódica de periodo  $t_{sinc}=55$  u.t. Como debe finalizar como muy tarde 10 u.t. después del lanzamiento de la tarea, el plazo de ejecución es  $d_{sinc}=10$  u.t. Se estima que el tiempo máximo de ejecución es  $p_{sinc} = 0.1$  u.t.

#### *Tarea de envío de informe de estado al módulo principal*

Teniendo en cuenta la descripción de los requisitos del sistema, la tarea de envío de informe de estado al módulo principal se puede considerar esporádica con tiempo mínimo entre eventos consecutivos 110 u.t., con lo que se propone modelarla como una tarea periódica de periodo  $t_{inf\_est}=110$  u.t.. El plazo de esta tarea es igual a su periodo  $d_{inf\_est} = t_{inf\_est} = 110$  u.t. y se requiere que lo cumpla porque esta tarea es estricta. El máximo tiempo de ejecución se estima que es  $p_{inf\_est} = 8$  u.t.

#### *Tarea de análisis de la señal y obtención del código de velocidad*

La tarea de análisis de la señal y obtención del código de velocidad puede considerarse crítica y periódica con  $t_{cod}=65$  u.t., y por ser crítica es también estricta, con plazo  $d_{cod} = t_{cod}=65$  u.t. El máximo tiempo de ejecución de la tarea de análisis es  $p_{cod} = 40$  u.t.

#### *Tarea de autocomprobación*

Para cumplir los requisitos del sistema, se considera que la tarea de autocomprobación es periódica, con  $t_{test} = 110$  u.t. El plazo de esta tarea es  $d_{test} = t_{inf\_est} = 110$  u.t., ya que se requiere que la posibilidad o la probabilidad de que haya algún informe disponible antes de la llegada de la siguiente petición de estado por parte del módulo principal ha de ser mayor que cero. Es decir, no se considera un

diseño válido si resulta imposible tener un informe disponible cuando llegue la siguiente petición. Es una tarea firme ya que existe flexibilidad en el cumplimiento del plazo, pero se debe asegurar que no es totalmente imposible el envío de un informe. Su máximo tiempo de ejecución se estima que es  $p_{test} = 20$  u.t.

#### *Tarea de autocomprobación en zona de memoria compartida*

De la descripción de los requisitos del sistema se extrae que la tarea de autocomprobación en zona de memoria compartida se puede considerar periódica de periodo  $t_{test\_b} = 220$  u.t. El plazo es igual al periodo  $d_{test\_b} = t_{test\_b} = 220$  u.t. y aunque es una tarea no crítica se requiere que esta tarea finalice antes de su plazo.

Se considerará que en el peor de los casos el tiempo de utilización de la zona de memoria por parte de esta función de autocomprobación es  $ru_{test\_b} = 2$  y el tiempo de ejecución es  $p_{test\_b} = 5$ .

#### *Tarea de recepción de informes del procesador secundario*

La tarea de recepción de informes del procesador secundario se puede considerar como una tarea periódica con periodo  $t_{inf\_sec} = 110$  u.t. El plazo es  $d_{inf\_sec} = t_{inf\_est} = 110$  u.t., ya que se requiere que sea posible que haya algún informe disponible antes de la llegada de la siguiente petición de estado por parte del módulo principal. Al igual que ocurría con la tarea de autocomprobación, la tarea de recepción de informes del procesador secundario es una tarea firme. El máximo tiempo de ejecución que se considera posible es  $p_{inf\_sec} = 1$  u.t.

#### *Tarea de reloj*

La tarea de reloj se ejecuta a nivel de interrupción y es periódica con  $t_{clk} = 5$  u.t.. Su tiempo de ejecución máximo se considera que es  $p_{clk} = 0.05$  u.t.

Una vez identificadas las tareas del sistema necesarias para cumplir los principales requisitos funcionales del sistema, se analizan otras cuestiones como el modelado de la función de recepción de hueco máximo excedido, cómo afecta al modelo la comunicación con el procesador secundario, la decisión de cómo implementar las tareas esporádicas y cómo afecta al modelo, y el modelado del tiempo que se consume por manejo de colas.

### Recepción del evento de hueco máximo excedido

Para el modelado y análisis temporal del sistema no hay que tener en cuenta la posible interferencia de esta función en las tareas identificadas previamente. La razón es que si llega este evento, es único ya que el efecto del evento de hueco máximo excedido y el correspondiente informe al módulo principal es que los módulos son cerrados y el sistema deja de ejecutar sus tareas. Externamente, si se confirma dicha alarma, el tren es detenido. Por tanto, no hace falta modelar esta tarea. Solamente sería necesario tenerla en cuenta si se desease analizar el tiempo de respuesta del sistema desde la llegada del evento hasta que informa al módulo principal, en el peor de los casos.

### Modelado de la comunicación con el procesador secundario

Hay dos tareas que requieren recepción de información del procesador secundario: la tarea de análisis y código de velocidad, que recibe el espectro de la señal generado por el procesador secundario, y la tarea de recepción de informes de autocomprobación de dicho procesador. La comunicación entre ambos se realiza a través de una memoria compartida, protegida mediante un semáforo, por lo que ambas tareas pueden ver retrasada su ejecución si la memoria está siendo utilizada por el otro procesador, precisamente mientras escribe los datos que esperan las tareas. Esos retardos, que son los tiempos de escritura del espectro y del informe, se pueden modelar como retraso de la activación de las tareas o “release jitter”. El “release jitter” de la tarea de códigos de velocidad se estima que es  $j_{cod}=1$  u.t. y el de recogida de informes  $j_{inf\_sec}=0.5$  u.t.

### Implementación y modelado de las tareas esporádicas

Tal y como se ha descrito en la metodología, una forma de implementar las tareas esporádicas es considerar que el evento llega mediante una interrupción y en tiempo de interrupción se encola, con un tiempo asociado  $p_{int}$ . La tarea asociada se ejecutará con la prioridad que se le asigne, por lo que en el análisis temporal hay que añadir a todas las tareas la posible interferencia por la ejecución del manejador de interrupciones. Así, en la siguiente ecuación iterativa para el cálculo del tiempo de finalización:

$$s_{i,k}(s_{i,k-1}) = p_i + \sum_{j \in hp(i)} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil + p_{clk} \cdot \left\lceil \frac{s_{i,k-1}}{t_{clk}} \right\rceil + \sum_{g \in esp} p_{int} \cdot \left\lceil \frac{s_{i,k-1}}{t_g} \right\rceil$$

se tiene que  $esp$  es el conjunto de tareas esporádicas lanzadas por interrupciones y  $p_{int}$  es el tiempo de interrupción (suponiendo que es constante para todas las interrupciones)

Sin embargo, dado que por seguridad del sistema existe la restricción de que sólo puede haber una interrupción, que obviamente tiene que ser la interrupción del reloj de periodo  $t_{clk}$ , no se puede implementar la llegada de tareas esporádicas mediante interrupción.

Se decide por tanto utilizar la segunda implementación, que consiste en que durante la interrupción de reloj se comprueba si ha llegado el evento de la tarea esporádica. En el peor de los casos el evento llegaría justo cuando ha terminado de ejecutarse la tarea de interrupción de reloj, con lo cual tendría que esperar un periodo completo de reloj para ser encolado. Este retardo se puede modelar como un “release jitter”  $j=t_{clk}$ . Por tanto, el “release jitter” de la tarea de sincronización es  $j_{sinc} = 5$  u.t., y el de la tarea de envío de informe de estado  $j_{inf\_est} = 5$  u.t..

### Manejo de cola

Para tener en cuenta el retardo por la manipulación de la cola que realiza el manejador de interrupciones de reloj, hay que considerar las sobrecargas causadas por cada tarea periódica y esporádica. Este retardo puede modelarse mediante dos parámetros:  $p_{clk}$ , la sobrecarga de cada interrupción supuesto que no se mueve ninguna tarea de la cola, y  $p_{cola}$  que es el coste de mover una tarea de la cola de retardo a la cola de ejecución. La ecuación del análisis temporal queda como sigue:

$$s_{i,k}(s_{i,k-1}) = b_i + p_i + \sum_{j \in hp(i)} p_j \cdot \left\lceil \frac{s_{i,k-1}}{t_j} \right\rceil + p_{clk} \cdot \left\lceil \frac{s_{i,k-1}}{t_{clk}} \right\rceil + \sum_{f \in psts} p_{cola} \cdot \left\lceil \frac{s_{i,k-1}}{t_f} \right\rceil$$

donde  $hp(i)$  es el conjunto de tareas más prioritarias que  $i$ , y  $psts$  es el conjunto de tareas periódicas y esporádicas.

Se considerará que para este sistema  $p_{cola}=0.05$  u.t.

### 7.2.2.2. Diseño de la planificación de tareas

El siguiente paso de esta segunda fase (diseño y modelado del sistema) de la metodología consiste en escoger una asignación de prioridades de acuerdo con el criterio del tamaño del plazo de ejecución, esto es según DMS ya que se trata de un

algoritmo de planificación óptimo. Por tanto, se decide asignar prioridades fijas según el tamaño de los plazos de forma que la más prioritaria es la tarea de reloj y después, en el orden que se indica: sincronización, análisis de la señal y obtención y envío del código de velocidad, envío de informe de estado al módulo principal, ejecución de rutinas de autocomprobación, recepción de informes de autocomprobación del otro procesador y ejecución de la rutina de autocomprobación en zona de memoria compartida por la tarea de códigos de velocidad.

### Tiempos de bloqueo

Como ya se ha dicho, la tercera tarea (análisis de la señal y código de velocidad) y la séptima (tarea de autocomprobación en zona de memoria compartida) comparten zona de memoria. Así, la tarea más prioritaria de las dos puede sufrir bloqueo, pero también pueden sufrir bloqueo, dada la asignación de prioridades elegida, todas aquellas tareas que, a pesar de no compartir zona de memoria con las anteriores, tienen una prioridad menor que la tarea de análisis de la señal y código de velocidad.

Se selecciona el protocolo de sincronización PIP para implementar la sección crítica, por lo que, aplicando el algoritmo de cálculo del bloqueo  $b$  a partir de los tiempos numéricos de utilización de recursos compartidos (versión numérica de (7.1)), se obtiene  $b=2$  para todas las tareas, desde la tarea de códigos de velocidad, hasta la tarea de recogida de informes del procesador secundario, ambas incluidas.

Así pues, el cálculo de los tiempos de bloqueo es:

$$n = 7$$

$$i = n = 7: b_7 = b_{test\_b} = b = 0. \text{ Como } ru_7 = ru_{test\_b} = 2 > b, \text{ entonces } b = 2$$

$$i = 6: b_6 = b_{inf\_sec} = b = 2. \text{ Como } ru_6 = ru_{inf\_sec} = 0 < b, \text{ entonces } b = 2$$

$$i = 5: b_5 = b_{test} = b = 2. \text{ Como } ru_5 = ru_{test} = 0 < b, \text{ entonces } b = 2$$

$$i = 4: b_4 = b_{inf\_est} = b = 2. \text{ Como } ru_4 = ru_{inf\_est} = 0 < b, \text{ entonces } b = 2$$

$$i = 3: b_3 = b_{cod} = b = 2. \text{ } ru_3 = ru_{cod} \neq 0. \text{ Como } pr_{i-1} > pc_r = pc_3 = pr_{cod} \text{ entonces } b = 0$$

$$i = 2: b_2 = b_{sinc} = b = 0.$$

$$i = 1: b_1 = b_{clk} = b = 0.$$

A continuación se resumen los parámetros de las tareas del sistema modeladas con la asignación de prioridades seleccionada. De acuerdo con lo descrito anteriormente, los parámetros temporales que caracterizan a las tareas se deben haber escogido estimando los casos más desfavorables. Para los valores de tiempos de ejecución, bloqueos y “release jitter”, los peores tiempos corresponden a los valores máximos, mientras que para los plazos y los periodos de las tareas esporádicas los valores más desfavorables son los mínimos.

	<i>Periodo</i>	<i>Tiempo de ejecución</i>	<i>Release jitter</i>	<i>Tiempo de bloqueo</i>	<i>Plazo</i>
Reloj	$t_{clk}=5$	$p_{clk}=0.05,$ $p_{cola}=0.05$			$d_{clk}=0.1$
Sincronización	$t_{sinc}=55$	$p_{sinc}=0.1$	$j_{sinc}=t_{clk}=5$		$d_{sinc}=10$
Análisis de señal y código de velocidad	$t_{cod}=65$	$p_{cod}=40$	$j_{cod}=1$	$b_{cod}=2$	$d_{cod}=65$
Envío de informe de estado al módulo principal	$t_{inf\_est}=110$	$p_{inf\_est}=8$	$j_{inf\_est} = t_{clk}$ $=5$	$b_{inf\_est}=2$	$d_{inf\_est} =$ $t_{inf\_est}=110$
Funciones de autocomprobación	$t_{test}=110$	$p_{test}=20$		$b_{test}=2$	$d_{test}=110$
Recepción de informe de autocomprobación del procesador secundario	$t_{inf\_sec}=110$	$p_{inf\_sec}=1$	$j_{inf\_sec}=0.5$	$b_{inf\_sec}=2$	$d_{inf\_sec}=110$
Función de autocomprobación en zona de memoria compartida	$t_{test\_b}=220$	$p_{test\_b}=5$			$d_{test\_b}=220$

### 7.2.2.3. Análisis de planificabilidad

A partir del modelo del sistema con parámetros numéricos se puede realizar el análisis de planificabilidad aplicando (7.5) para determinar si el sistema es o no planificable. Para ello todas las tareas deben respetar sus plazos, incluso las tareas firmes ya que su requisito es que debe haber algún informe disponible en el momento del correspondiente plazo. Como en este caso sólo se conocen los tiempos máximos de ejecución, la conclusión es que el tiempo de finalización de dichas tareas debe estar antes del plazo.

La sobrecarga causada por el manejo de la cola se denominará en adelante  $M$  y viene dado en la iteración  $k$  del cálculo del tiempo de finalización por la siguiente expresión:

$$M = \sum_{f \in \text{pts}} p_{cola} \cdot \left\lceil \frac{S_{k-1}}{t_f} \right\rceil = p_{cola} \cdot \left\{ \left\lceil \frac{S_{k-1}}{t_{sinc}} \right\rceil + \left\lceil \frac{S_{k-1}}{t_{cod}} \right\rceil + \left\lceil \frac{S_{k-1}}{t_{inf\_est}} \right\rceil + \left\lceil \frac{S_{k-1}}{t_{test}} \right\rceil + \left\lceil \frac{S_{k-1}}{t_{inf\_sec}} \right\rceil + \left\lceil \frac{S_{k-1}}{t_{test\_b}} \right\rceil \right\}$$

### Tarea de reloj

Como la tarea de reloj es la más prioritaria, su tiempo de finalización coincide con su tiempo de ejecución. Se considera que primero se ejecuta esta tarea y que luego se realizan las funciones de manejo de cola en tiempo de interrupción de reloj, por lo que el tiempo  $M$  correspondiente a esas funciones no retrasa la finalización de la tarea de reloj. Por tanto, se tiene que el tiempo de finalización es  $wc_{clk} = p_{clk} = 0.05$

### Sincronización

El cálculo del tiempo de finalización de la tarea de sincronización en la iteración  $k$  debe tener en cuenta la interferencia de la tarea de reloj y el tiempo de manejo de colas, y viene dado por:

$$s_k(s_{k-1}) = p_{sinc} + p_{clk} \cdot \left\lceil \frac{S_{k-1}}{t_{clk}} \right\rceil + M$$

$$M = \sum_{f \in \text{pts}} p_{cola} \cdot \left\lceil \frac{S_{k-1}}{t_f} \right\rceil$$

$p_{sinc} = 0.1; j_{sinc} = 5; d_{sinc} = 10; p_{clk} = 0.05; t_{clk} = 5$
$s_0 = p_{clk} + p_{sinc} = 0.05 + 0.1 = 0.15$
$s_1 = 0.1 + 0.05 \cdot \lceil 0.15/5 \rceil + 0.3 = 0.45$
$s_2 = 0.1 + 0.05 \cdot \lceil 0.45/5 \rceil + 0.3 = 0.45 = s_1$
$wc_{sinc} = s_2 + j_{sinc} = 0.45 + 5 = 5.45$

Por tanto  $wc_{sinc} = 5.45$ . Como  $wc_{sinc} < d_{sinc}$  el peor tiempo de respuesta de la tarea es menor que el plazo, y por tanto la tarea es planificable.

### Análisis de señal y código de velocidad

Para el cálculo del tiempo de finalización de esta tarea hay que tener en cuenta, además de la interferencia de tareas más prioritarias y el tiempo de manejo de cola, el tiempo de bloqueo que sufre por parte de tareas menos prioritarias. Se tiene:

$$s_k(s_{k-1}) = b_{cod} + p_{cod} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{cod} = 40; b_{cod} = 2; j_{cod} = 1; d_{cod} = 65; p_{sinc} = 0.1; t_{sinc} = 55; j_{sinc} = 5; p_{clk} = 0.05; t_{clk} = 5$
$s_0 = p_{clk} + p_{sinc} + p_{cod} = 0.05 + 0.1 + 40 = 40.15$
$s_1 = 2 + 40 + 0.05 \cdot \lceil 40.15/5 \rceil + 0.1 \cdot \lceil 45.15/55 \rceil + 0.3 = 42.85$ $s_2 = 2 + 40 + 0.05 \cdot \lceil 42.85/5 \rceil + 0.1 \cdot \lceil 42.85/55 \rceil + 0.3 = 42.85 = s_1$
$wc_{cod} = s_2 + j_{cod} = 42.85 + 1 = 43.85$

Por tanto, el tiempo de finalización es  $wc_{cod} = 43.85$ . Como  $wc_{cod} < d_{cod}$  el peor tiempo de respuesta es menor que el plazo y en consecuencia, la tarea es planificable.

### Envío de informe de estado al módulo principal

De forma similar, esta tarea también sufre interferencia por parte de tareas más prioritarias, y retraso debido al manejo de colas y bloqueo. Por ello se tiene:

$$s_k(s_{k-1}) = b_{inf\_est} + p_{inf\_est} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{inf\_est}=8; j_{inf\_est}=5; b_{inf\_est}=2; d_{inf\_est}=110; p_{cod}=40; t_{cod}=65; j_{cod}=1; p_{sinc}=0.1;$ $t_{sinc}=55; j_{sinc}=5; p_{clk}=0.05; t_{clk}=5$
$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} = 0.05 + 0.1 + 40 + 8 = 48.15$
$s_1 = 2 + 8 + 0.05 \lceil 48.15/5 \rceil + 0.1 \lceil 53.15/55 \rceil + 40 \lceil 49.15/65 \rceil + 0.3 = 50.9$
$s_2 = 2 + 8 + 0.05 \lceil 50.9/5 \rceil + 0.1 \lceil 55.9/55 \rceil + 40 \lceil 51.9/65 \rceil + 0.3 = 51.05$
$s_3 = 2 + 8 + 0.05 \lceil 51.05/5 \rceil + 0.1 \lceil 56.05/55 \rceil + 40 \lceil 52.05/65 \rceil + 0.3 = 51.05 = s_2$
$wC_{inf\_est} = s_3 + j_{inf\_est} = 51.05 + 5 = 56.05$

Finalmente se tiene que el tiempo de finalización vale  $wC_{inf\_est} = 56.05$ .

Como  $wC_{inf\_est} < d_{inf\_est}$  esta tarea también es planificable.

### Funciones de autocomprobación

En este apartado se han modelado los parámetros temporales para el peor caso, lo que no permite introducir el requisito firme según el cual debe ser posible tener un informe disponible antes del plazo de la tarea. Si se verifica el plazo en el peor caso, se tiene certeza de que todos los informes están disponibles en ese instante, pero si no se verifica el plazo, no se puede determinar si es posible que exista algún informe disponible, con lo que el diseño debería ser rechazado.

El cálculo del tiempo de finalización de esta tarea periódica, que se encarga de comprobar el propio sistema, viene dado por:

$$s_k(s_{k-1}) = b_{test} + n \cdot p_{test} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil + p_{inf\_est} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_est}}{t_{inf\_est}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{test}=20; b_{test}=2; d_{test}=110; t_{test}=110; p_{inf\_est}=8; j_{inf\_est}=5; t_{inf\_est}=110; p_{cod}=40; t_{cod}=65; j_{cod}=1; p_{sinc}=0.1; t_{sinc}=55; j_{sinc}=5; p_{clk}=0.05; t_{clk}=5$
$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} = 0.05 + 0.1 + 40 + 8 + 20 = 68.15$
$n=1$
$s_1 = 2 + 20 + 0.05 \lceil 68.15/5 \rceil + 0.1 \lceil 73.15/55 \rceil + 40 \lceil 69.15/65 \rceil + 8 \lceil 73.15/110 \rceil + 0.4 = 111.3$ $s_2 = 2 + 20 + 0.05 \lceil 111.3/5 \rceil + 0.1 \lceil 116.3/55 \rceil + 40 \lceil 112.3/65 \rceil + 8 \lceil 116.3/110 \rceil + 0.05 \cdot 12 = 120.05$ $s_3 = 2 + 20 + 0.05 \lceil 120.05/5 \rceil + 0.1 \lceil 125.05/55 \rceil + 40 \lceil 121.05/65 \rceil + 8 \lceil 125.05/110 \rceil + 0.05 \cdot 12 = 120.15$ $s_4 = 2 + 20 + 0.05 \lceil 120.15/5 \rceil + 0.1 \lceil 125.15/55 \rceil + 40 \lceil 121.15/65 \rceil + 8 \lceil 125.15/110 \rceil + 0.05 \cdot 12 = 120.15 = s_3$
$w_{C_{test,1}} = s_4 = 120.15$
Como $s_4 > t_{test}$ , $n=2$
$s_4 = s_4 + p_{test} = 140.15$ $s_5 = 2 + 2 \cdot 20 + 0.05 \lceil 140.15/5 \rceil + 0.1 \lceil 145.15/55 \rceil + 40 \lceil 141.15/65 \rceil + 8 \lceil 145.15/110 \rceil + 0.05 \cdot 13 = 180.4$ $s_6 = 2 + 2 \cdot 20 + 0.05 \lceil 180.4/5 \rceil + 0.1 \lceil 185.4/55 \rceil + 40 \lceil 181.4/65 \rceil + 8 \lceil 185.4/110 \rceil + 0.05 \cdot 14 = 180.95$ $s_7 = 2 + 2 \cdot 20 + 0.05 \lceil 180.95/5 \rceil + 0.1 \lceil 185.95/55 \rceil + 40 \lceil 181.95/65 \rceil + 8 \lceil 185.95/110 \rceil + 0.05 \cdot 14 = 180.95 = s_6$
$w_{C_{test,2}} = s_7 = 180.95$ y $w_{C_{test,2}} < 2 \cdot t_{test}$

El tiempo de finalización de la tarea de autocomprobación en su primera ejecución es  $w_{C_{test,1}} = 120.15$ . Como se puede observar, en este caso el tiempo de finalización no respeta el plazo  $w_{C_{test,1}} > d_{test}$  con lo que la tarea no es planificable.

En el cálculo de este tiempo de finalización se ha detectado que se ha sobrepasado el periodo  $s_4 > t_{test}$  con lo que el periodo de ocupación no ha terminado. Con el fin de

obtener el peor tiempo de respuesta de la tarea, se continúa con el cálculo de la segunda ejecución.

Así, el tiempo de finalización de la tarea de autocomprobación en su segunda ejecución es  $w_{C_{test,2}} = 180.95$ . Como  $w_{C_{test,2}} < t_{test} + d_{test} = 220$ , el tiempo de respuesta en esta segunda ejecución es menor que el plazo. Además ha terminado el periodo de ocupación, ya que  $w_{C_{test,2}} < 2 \cdot t_{test}$

Como la tarea no cumple el plazo en su primera ejecución no se puede determinar si se cumple el requisito firme y en consecuencia no es planificable.

### Recepción de informe de autocomprobación del procesador secundario

Al igual que con la tarea de autocomprobación, si se verifica el plazo en el peor caso, se tiene certeza de que todos los informes están disponibles en ese instante, pero si no se verifica el plazo, no se puede determinar si es posible que exista algún informe disponible, y el diseño no puede ser considerado válido.

El tiempo de finalización de la tarea periódica encargada de recibir los informes de autocomprobación ejecutados por el procesador secundario viene dado por:

$$s_k(s_{k-1}) = b_{inf\_sec} + n \cdot p_{inf\_sec} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil$$

$$+ p_{inf\_est} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_est}}{t_{inf\_est}} \right\rceil + p_{test} \cdot \left\lceil \frac{s_{k-1}}{t_{test}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{inf\_sec} = 1; b_{inf\_sec} = 2; j_{inf\_sec} = 0.5; t_{inf\_sec} = 110; d_{inf\_sec} = 110; p_{test} = 20; t_{test} = 110; p_{inf\_est} = 8; j_{inf\_est} = 5; t_{inf\_est} = 110; p_{cod} = 40; t_{cod} = 65; j_{cod} = 1; p_{sinc} = 0.1; t_{sinc} = 55; j_{sinc} = 5; p_{clk} = 0.05; t_{clk} = 5$
$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} + p_{inf\_sec} = 0.05 + 0.1 + 40 + 8 + 20 + 1 = 69.15$
$n=1$
$s_1 = 2 + 1 + 0.05 \lceil 69.15/5 \rceil + 0.1 \lceil 74.15/55 \rceil + 40 \lceil 70.15/65 \rceil + 8 \lceil 74.15/110 \rceil + 20 \lceil 69.15/110 \rceil + 0.4 = 112.3$
$s_2 = 2 + 1 + 0.05 \lceil 112.3/5 \rceil + 0.1 \lceil 117.3/55 \rceil + 40 \lceil 113.3/65 \rceil + 8 \lceil 117.3/110 \rceil + 20 \lceil 112.3/110 \rceil + 0.05 \cdot 12 = 141.05$
$s_3 = 2 + 1 + 0.05 \lceil 141.05/5 \rceil + 0.1 \lceil 146.05/55 \rceil + 40 \lceil 142.05/65 \rceil + 8 \lceil 146.05/110 \rceil + 20 \lceil 141.05/110 \rceil + 0.05 \cdot 13 = 181.4$
$s_4 = 2 + 1 + 0.05 \lceil 181.4/5 \rceil + 0.1 \lceil 186.4/55 \rceil + 40 \lceil 182.4/65 \rceil + 8 \lceil 186.4/110 \rceil + 20 \lceil 181.4/110 \rceil + 0.05 \cdot 14 = 181.95$
$s_5 = 2 + 1 + 0.05 \lceil 181.95/5 \rceil + 0.1 \lceil 186.95/55 \rceil + 40 \lceil 182.95/65 \rceil + 8 \lceil 186.95/110 \rceil + 20 \lceil 181.95/110 \rceil + 0.05 \cdot 14 = 181.95 = s_4$
$wc_{inf\_sec,1} = s_5 + j_{inf\_sec} = 181.95 + 0.5 = \mathbf{182.45}$
Como $s_5 > t_{inf\_sec}$ , $n=2$
$s_5 = s_5 + p_{inf\_sec} = 182.95$
$s_6 = 2 + 2 \cdot 1 + 0.05 \lceil 182.95/5 \rceil + 0.1 \lceil 187.95/55 \rceil + 40 \lceil 183.95/65 \rceil + 8 \lceil 187.95/110 \rceil + 20 \lceil 182.95/110 \rceil + 0.05 \cdot 14 = 182.95 = s_5$
$wc_{inf\_sec,2} = s_6 + j_{inf\_sec} = 182.95 + 0.5 = \mathbf{183.45}$ y $wc_{inf\_sec,2} < 2 \cdot t_{inf\_sec}$

El tiempo de finalización de la tarea en su primera ejecución es  $wc_{inf\_sec,1} = 182.45$ . Se observa que este tiempo de finalización no respeta el plazo  $wc_{inf\_sec,1} > d_{inf\_sec}$  con lo que la tarea no es planificable.

Durante el cálculo del tiempo de finalización en la primera ejecución de la tarea, se detecta que  $s_5 > t_{inf\_sec}$  lo que significa que el periodo de ocupación no ha terminado. Con el fin de obtener el peor tiempo de respuesta de la tarea, se sigue calculando la segunda ejecución. Con  $n=2$ , se obtiene el tiempo de finalización de la tarea en su segunda ejecución, que vale  $wc_{inf\_sec,2} = 183.45$ . Como  $wc_{inf\_sec,2} < t_{inf\_sec} + d_{inf\_sec} = 220$ , la tarea respeta el plazo en su segunda ejecución. Además ha terminado el periodo de ocupación, ya que  $wc_{inf\_sec,2} < 2 \cdot t_{inf\_sec}$ .

### Función de autocomprobación en zona de memoria compartida

Esta tarea es la tarea periódica de comprobación que comparte zona de memoria con la tarea de código de velocidad, provocando bloqueos a las tareas más prioritaria que ella misma hasta la tarea de código de velocidad incluida.

$$s_k(s_{k-1}) = p_{test\_b} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil$$

$$+ p_{inf\_est} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_est}}{t_{inf\_est}} \right\rceil + p_{test} \cdot \left\lceil \frac{s_{k-1}}{t_{test}} \right\rceil + p_{inf\_sec} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_sec}}{t_{inf\_sec}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{test\_b}=5; d_{test\_b}=220; p_{inf\_sec}=1; j_{inf\_sec}=0.5; t_{inf\_sec}=110; p_{test}=20; t_{test}=110; p_{inf\_est}=8; j_{inf\_est}=5; t_{inf\_est}=110; p_{cod}=40; t_{cod}=65; j_{cod}=1; p_{sinc}=0.1; t_{sinc}=55; j_{sinc}=5; p_{clk}=0.05; t_{clk}=5$

$$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} + p_{inf\_sec} + p_{test\_b} = 0.05 + 0.1 + 40 + 8 + 20 + 1 + 5 = 74.15$$

$$s_1 = 5 + 0.05 \cdot \lceil 74.15/5 \rceil + 0.1 \cdot \lceil 79.15/55 \rceil + 40 \cdot \lceil 75.15/65 \rceil + 8 \cdot \lceil 79.15/110 \rceil + 20 \cdot \lceil 74.15/110 \rceil + 1 \cdot \lceil 74.65/110 \rceil + 0.4 = 115.35$$

$$s_2 = 5 + 0.05 \cdot \lceil 115.35/5 \rceil + 0.1 \cdot \lceil 120.35/55 \rceil + 40 \cdot \lceil 116.35/65 \rceil + 8 \cdot \lceil 120.35/110 \rceil + 20 \cdot \lceil 115.35/110 \rceil + 1 \cdot \lceil 115.85/110 \rceil + 0.05 \cdot 12 = 145.1$$

$$s_3 = 5 + 0.05 \cdot \lceil 145.1/5 \rceil + 0.1 \cdot \lceil 150.1/55 \rceil + 40 \cdot \lceil 146.1/65 \rceil + 8 \cdot \lceil 150.1/110 \rceil + 20 \cdot \lceil 145.1/110 \rceil + 1 \cdot \lceil 145.6/110 \rceil + 0.05 \cdot 13 = 185.45$$

$$s_4 = 5 + 0.05 \cdot \lceil 185.45/5 \rceil + 0.1 \cdot \lceil 190.45/55 \rceil + 40 \cdot \lceil 186.45/65 \rceil + 8 \cdot \lceil 190.45/110 \rceil + 20 \cdot \lceil 185.45/110 \rceil + 1 \cdot \lceil 185.95/110 \rceil + 0.05 \cdot 14 = 186$$

$$s_5 = 5 + 0.05 \cdot \lceil 186/5 \rceil + 0.1 \cdot \lceil 191/55 \rceil + 40 \cdot \lceil 187/65 \rceil + 8 \cdot \lceil 191/110 \rceil + 20 \cdot \lceil 186/110 \rceil + 1 \cdot \lceil 186.5/110 \rceil + 0.05 \cdot 14 = 186 = s_4$$

$$wc_{test\_b} = s_5 = 186$$

Por tanto, el tiempo de finalización es  $wc_{test\_b} = 186$ . Como  $wc_{test\_b} < d_{test\_b}$  la tarea es planificable.

**7.2.2.4. Resumen y conclusiones**

En la siguiente tabla se resumen los tiempos de finalización calculados previamente. Se resaltan en **negrita** aquellos que no respetan el plazo correspondiente.

Tarea de reloj	$wc_{clk} = 0.05$
Sincronización	$wc_{sinc} = 5.45 < d_{sinc}$
Análisis de señal y código de velocidad	$wc_{cod} = 43.85 < d_{cod}$
Envío de informe de estado al módulo principal	$wc_{inf\_est} = 56.05 < d_{inf\_est}$
Funciones de autocomprobación	$wc_{test,1} = \mathbf{120.15} > d_{test}$ $wc_{test,2} = 180.95 < t_{test} + d_{test}$
Recepción de informe de autocomprobación del procesador secundario	$wc_{inf\_sec,1} = \mathbf{182.45} > d_{inf\_sec}$ $wc_{inf\_sec,2} = 183.45 < t_{inf\_sec} + d_{inf\_sec}$
Función de autocomprobación en zona de memoria compartida	$wc_{test\_b} = 186 < d_{test\_b}$

Se observa que tanto la tarea de funciones de autocomprobación como la de recepción de informe de autocomprobación del procesador secundario son no planificables ya que no respetan el plazo en el peor caso y no se puede determinar por tanto si es posible que tengan algún informe disponible en ese instante. Atendiendo a los tiempos de ejecución de las dos tareas no planificables, en el siguiente apartado se propone intercambiar las prioridades de ambas tareas con el fin de mejorar la planificabilidad de la tarea de recepción de informe de autocomprobación del procesador secundario, a pesar de que con dicho cambio la tarea de autocomprobación seguiría siendo no planificable.

### 7.2.3. Modificación del diseño: nueva asignación de prioridades

#### *Nueva asignación de prioridades*

Analizando los resultados anteriores se deduce que podría ser favorable para la tarea de recepción de informe del procesador secundario intercambiar la prioridad con la tarea de autocomprobación ya que el tiempo de ejecución  $p_{inf\_sec}$  es mucho más pequeño que  $p_{test}$ , y con la asignación anterior la tarea de recepción de informes se ve muy retrasada debido a que deben finalizar dos ejecuciones de la tarea de autocomprobación antes de poder ejecutarse. Si se intercambia la prioridad de las tareas no planificables del caso anterior, la tarea de autocomprobación continuará siendo no planificable en su primera ejecución, ya que su tiempo de finalización será mayor que 120.15 al disminuir su prioridad. A pesar de eso se han calculado los resultados para dicha asignación de prioridades con el fin de comprobar cómo queda la tarea de recepción de informe de autocomprobación del procesador secundario.

Tanto las tareas con prioridad mayor que estas dos tareas, como la tarea de prioridad menor, no se ven afectadas por el intercambio de prioridades y sus tiempos de respuesta son los mismos que en el caso anterior. Así pues, con la nueva asignación la quinta tarea por orden de prioridad es la de recepción de informe de autocomprobación del procesador secundario y la sexta es la correspondiente a las funciones de autocomprobación.

#### 7.2.3.1. Análisis de planificabilidad con la nueva asignación

##### **Recepción de informe de autocomprobación del procesador secundario**

Con la nueva asignación de prioridades esta tarea se ve afectada por la interferencia de las más prioritarias que son: reloj, sincronización, obtención y envío del código de velocidad, y envío de informe de estado al módulo principal. El tiempo de finalización viene dado por:

$$s_k(s_{k-1}) = b_{inf\_sec} + p_{inf\_sec} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil \\ + p_{inf\_est} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_est}}{t_{inf\_est}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left\lceil \frac{s_{k-1}}{t_f} \right\rceil$$

$p_{inf\_sec} = 1; b_{inf\_sec} = 2; j_{inf\_sec} = 0.5; t_{inf\_sec} = 110; d_{inf\_sec} = 110; p_{inf\_est} = 8; j_{inf\_est} = 5; t_{inf\_est} = 110; p_{cod} = 40; t_{cod} = 65; j_{cod} = 1; p_{sinc} = 0.1; t_{sinc} = 55; j_{sinc} = 5; p_{clk} = 0.05; t_{clk} = 5$
$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} = 0.05 + 0.1 + 40 + 8 + 1 = 49.15$
$s_1 = 2 + 1 + 0.05 \cdot \lceil 49.15/5 \rceil + 0.1 \cdot \lceil 54.15/55 \rceil + 40 \cdot \lceil 50.15/65 \rceil + 8 \cdot \lceil 54.15/110 \rceil + 0.3 = 51.9$
$s_2 = 2 + 1 + 0.05 \cdot \lceil 51.9/5 \rceil + 0.1 \cdot \lceil 56.9/55 \rceil + 40 \cdot \lceil 52.9/65 \rceil + 8 \cdot \lceil 56.9/110 \rceil + 0.3 = 52.05$
$s_3 = 2 + 1 + 0.05 \cdot \lceil 52.05/5 \rceil + 0.1 \cdot \lceil 57.05/55 \rceil + 40 \cdot \lceil 53.05/65 \rceil + 8 \cdot \lceil 57.05/110 \rceil + 0.3 = 52.05 = s_2$
$wc_{inf\_sec} = s_3 + j_{inf\_sec} = 52.55$

Con esta nueva asignación de prioridades, el tiempo de finalización de la tarea de recepción de informes del procesador secundario es  $wc_{inf\_sec} = 52.55$ . En este caso, a diferencia de lo obtenido con la otra asignación de prioridades, como  $wc_{inf\_sec} < d_{inf\_sec}$ , la tarea de recepción de informe del procesador secundario es planificable. Como respeta el plazo en el caso más desfavorable, verifica el requisito firme ya que no sólo es posible que tenga un informe disponible en el plazo sino que se asegura que todos los informes están disponibles en ese instante.

Además el periodo de ocupación ha terminado ya que  $wc_{inf\_sec} < t_{inf\_sec}$ , con lo que no es necesario calcular posteriores ejecuciones de la tarea, ya que el peor tiempo de respuesta se da en la primera.

### Funciones de autocomprobación

Con la nueva asignación de prioridades esta tarea se ve afectada por la interferencia de las más prioritarias que son: reloj, sincronización, obtención y envío del código de velocidad, envío de informe de estado al módulo principal, y recepción del informe del procesador secundario.

$$s_k(s_{k-1}) = b_{test} + n \cdot p_{test} + p_{clk} \cdot \left\lceil \frac{s_{k-1}}{t_{clk}} \right\rceil + p_{sinc} \cdot \left\lceil \frac{s_{k-1} + j_{sinc}}{t_{sinc}} \right\rceil + p_{cod} \cdot \left\lceil \frac{s_{k-1} + j_{cod}}{t_{cod}} \right\rceil + p_{inf\_est} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_est}}{t_{inf\_est}} \right\rceil + p_{inf\_sec} \cdot \left\lceil \frac{s_{k-1} + j_{inf\_sec}}{t_{inf\_sec}} \right\rceil + M$$

$$M = \sum_{f \in pts} p_{cola} \cdot \left[ \frac{s_{k-1}}{t_f} \right]$$

$p_{test} = 20; b_{test} = 2; t_{test} = 110; d_{test} = 110; p_{inf\_sec} = 1; j_{inf\_sec} = 0.5; t_{inf\_sec} = 110; p_{inf\_est} = 8; j_{inf\_est} = 5; t_{inf\_est} = 110; p_{cod} = 40; t_{cod} = 65; j_{cod} = 1; p_{sinc} = 0.1; t_{sinc} = 55; j_{sinc} = 5; p_{clk} = 0.05; t_{clk} = 5$

$$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 40 + 8 + 1 + 20 = 69.15$$

$$s_1 = 2 + 20 + 0.05 \lceil 69.15/5 \rceil + 0.1 \lceil 74.15/55 \rceil + 40 \lceil 70.15/65 \rceil + 8 \lceil 74.15/110 \rceil + 1 \lceil 69.55/110 \rceil + 0.4 = 112.3$$

$$s_2 = 2 + 20 + 0.05 \lceil 112.3/5 \rceil + 0.1 \lceil 117.3/55 \rceil + 40 \lceil 113.3/65 \rceil + 8 \lceil 117.3/110 \rceil + 1 \lceil 112.8/110 \rceil + 0.05 \cdot 12 = 122.05$$

$$s_3 = 2 + 20 + 0.05 \lceil 122.05/5 \rceil + 0.1 \lceil 127.05/55 \rceil + 40 \lceil 123.05/65 \rceil + 8 \lceil 127.05/110 \rceil + 1 \lceil 122.55/110 \rceil + 0.05 \cdot 12 = 122.15$$

$$s_4 = 2 + 20 + 0.05 \lceil 122.15/5 \rceil + 0.1 \lceil 127.15/55 \rceil + 40 \lceil 123.15/65 \rceil + 8 \lceil 127.15/110 \rceil + 1 \lceil 122.65/110 \rceil + 0.05 \cdot 12 = 122.15 = s_3$$

$$wc_{test,1} = 122.15$$

Como  $wc_{test,1} > t_{test}$ ,  $n=2$

$$s_4 = s_4 + p_{test} = 142.15$$

$$s_5 = 2 + 2 \cdot 20 + 0.05 \lceil 142.15/5 \rceil + 0.1 \lceil 147.15/55 \rceil + 40 \lceil 143.15/65 \rceil + 8 \lceil 147.15/110 \rceil + 1 \lceil 142.65/110 \rceil + 0.05 \cdot 13 = 182.4$$

$$s_6 = 2 + 2 \cdot 20 + 0.05 \lceil 182.4/5 \rceil + 0.1 \lceil 187.4/55 \rceil + 40 \lceil 183.4/65 \rceil + 8 \lceil 187.4/110 \rceil + 1 \lceil 182.9/110 \rceil + 0.05 \cdot 14 = 182.95$$

$$s_7 = 2 + 2 \cdot 20 + 0.05 \lceil 182.95/5 \rceil + 0.1 \lceil 187.95/55 \rceil + 40 \lceil 183.95/65 \rceil + 8 \lceil 187.95/110 \rceil + 1 \lceil 183.45/110 \rceil + 0.05 \cdot 14 = 182.95 = s_6$$

$$wc_{test,2} = s_7 = 182.95 \text{ y } wc_{test,2} < 2 \cdot t_{test}$$

Con esta asignación de prioridades, el tiempo de finalización de la tarea de autocomprobación en su primera ejecución es  $wc_{test,1} = 122.15$ . Tal y como ya se había predicho, el tiempo de respuesta es mayor que con la otra asignación de prioridades (120.15).

Como  $wc_{test,1} > d_{test}$ , no se puede verificar si la tarea cumple su requisito firme, con lo que este diseño no es válido. Además,  $wc_{test,1} > t_{test}$  por lo que el periodo de ocupación no ha terminado y se continúa el cálculo para la segunda ejecución de la tarea  $n=2$ .

El tiempo de finalización de la tarea de autocomprobación en su primera ejecución es  $wc_{test,2} = 182.95$ . Como  $wc_{test,2} < t_{test} + d_{test} = 220$ , la segunda ejecución de la tarea respeta el plazo. Además ha terminado el periodo de ocupación, ya que  $wc_{test,2} < 2 \cdot t_{test}$  y por tanto no es necesario calcular ejecuciones posteriores de esta tarea.

### 7.2.3.2. Resumen y conclusiones

Resumiendo, los tiempos de finalización con la nueva asignación de prioridades son los siguientes:

Tarea de reloj	$wc_{clk} = 0.05$
Sincronización	$wc_{sinc} = 5.45 < d_{sinc}$
Análisis de señal y código de velocidad	$wc_{cod} = 43.85 < d_{cod}$
Envío de informe de estado al módulo principal	$wc_{inf\_est} = 56.05 < d_{inf\_est}$
Recepción de informe de autocomprobación del procesador secundario	$wc_{inf\_sec} = 52.55 < d_{inf\_sec}$
Funciones de autocomprobación	$wc_{test,1} = 122.15 > d_{test}$ $wc_{test,2} = 182.95 < t_{test} + d_{test}$
Función de autocomprobación en zona de memoria compartida	$wc_{test\_b} = 186 < d_{test\_b}$

Por tanto, con esta asignación de prioridades la tarea de autocomprobación continúa siendo no planificable, pero la tarea de recepción de informe del procesador secundario pasa a ser planificable. A pesar de que el sistema sigue siendo no planificable, queda confirmado que esta asignación de prioridades es mejor ya que la tarea de autocomprobación tiene un tiempo de finalización sólo ligeramente mayor que antes, y por otro lado se ha conseguido que el tiempo de finalización de la tarea de recepción de informes del procesador secundario sea mucho menor.

En cualquier caso, dado que el sistema no es planificable se considera que este diseño no es válido. Si únicamente se realiza un modelado del peor caso, habría que recortar la funcionalidad del sistema, bien reduciendo el tiempo de ejecución de una

tarea crítica, por ejemplo eliminando alguna función de autocomprobación no crítica, o aumentando el periodo de una tarea, o cualquier otra modificación que llevara a un diseño planificable, aunque sin cumplir todos los requisitos iniciales del sistema. En lugar de esto, se propone modelar la incertidumbre existente en el conocimiento de algunos parámetros temporales mediante distribuciones de posibilidad, y expresar el requisito firme de las tareas de autocomprobación y de recepción de informes del procesador secundario como una medida de posibilidad.

#### 7.2.4. Modificación del diseño: introducción de incertidumbre en algún plazo

Se puede tener en cuenta que en realidad los plazos de las tareas de recepción de informes del procesador secundario y las funciones de autocomprobación están asociados al comienzo de la tarea de envío del informe de estado al módulo principal, es decir, en una ejecución  $i$  de ambas tareas, éstas deben finalizar durante el periodo de envío de informes  $i-1$ , antes de que se lance la tarea de envío de informes en la ejecución  $i$ . Por tanto,  $d_{inf\_sec} = d_{lest} = t_{inf\_est}$

De acuerdo con la fase 2 de la metodología propuesta, como la tarea de envío de informes al módulo principal es esporádica, se puede modelar el periodo, y por tanto los anteriores plazos como números borrosos cuyo valor más pequeño del soporte es 110 que es el mínimo tiempo entre llegadas de la tarea esporádica. Ese valor mínimo que se considera posible es precisamente el valor numérico utilizado en el modelo anterior para realizar el análisis de planificabilidad en el peor caso. Por tanto, se modela el periodo como un número borroso triangular (110, 115, 116) tal y como se muestra en la Figura 7.7. A las tareas firmes (autocomprobación y recepción de informes del procesador secundario) se les exigirá que tengan algún informe disponible para enviar antes de la llegada de la siguiente petición de informe. En términos posibilistas, para las tareas estrictas la posibilidad y la necesidad de que la tarea finalice antes de su plazo han de ser 1, sea borroso o numérico el plazo. Para las tareas firmes, se permiten valores de posibilidad menores que 1, pero deben ser mayores que 0. Si son mayores que cero se puede interpretar que es posible que finalicen la tarea de autocomprobación o la tarea de recepción de informe del procesador secundario antes de que llegue el siguiente evento de petición de informe de estado, lo que se considera suficiente para estas tareas, dado que son tareas no críticas y con requisitos temporales firmes y no estrictos.

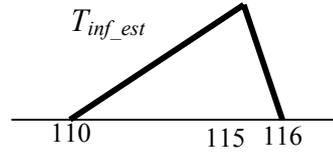


Figura 7.7: Modelo borroso del periodo de la tarea esporádica: envío de informe de estado al módulo principal

Si se conserva el modelo numérico para el periodo de la tarea esporádica pero se introduce flexibilidad en los plazos asociados a dicho periodo, se tiene que  $t_{inf\_est} = 110$  y  $D_{inf\_sec} = D_{test} = (110, 115, 116)$ .

#### 7.2.4.1. Análisis de planificabilidad

Dado que los tiempos de ejecución de las tareas, los periodos, los tiempos de bloqueo y los valores de “release jitter” no se han visto modificados, los tiempos de finalización son los mismos que los obtenidos en el apartado anterior, y simplemente hay que comparar esos tiempos con los plazos borrosos para obtener las medidas de posibilidad y necesidad de planificabilidad del sistema. Se presentan los resultados para la segunda asignación de prioridades que se estudió en el caso clásico ya que era más favorable en cuanto al número de tareas planificables.

$$wc_{clk} = 0.05 \text{ y } d_{clk} = 0.1 \quad \Rightarrow \quad \Pi(wc_{clk} < d_{clk}) = N(wc_{clk} < d_{clk}) = 1$$

$$wc_{sinc} = 5.45 \text{ y } d_{sinc} = 10 \quad \Rightarrow \quad \Pi(wc_{sinc} < d_{sinc}) = N(wc_{sinc} < d_{sinc}) = 1$$

$$wc_{cod} = 43.85 \text{ y } d_{cod} = 65 \quad \Rightarrow \quad \Pi(wc_{cod} < d_{cod}) = N(wc_{cod} < d_{cod}) = 1$$

$$wc_{inf\_est} = 56.05 \text{ y } d_{inf\_est} = 110 \quad \Rightarrow \quad \Pi(wc_{inf\_est} < d_{inf\_est}) = N(wc_{inf\_est} < d_{inf\_est}) = 1$$

$$wc_{inf\_sec} = 52.55 \text{ y } D_{inf\_sec} = (110, 115, 116) \Rightarrow \Pi(wc_{inf\_sec} < D_{inf\_sec}) = N(wc_{inf\_sec} < D_{inf\_sec}) = 1$$

$$wc_{test,1} = 122.15 \text{ y } D_{test} = (110, 115, 116) \Rightarrow \Pi(wc_{test,1} < D_{test}) = N(wc_{test,1} < D_{test}) = 0 \text{ ya que } 122.15 \text{ es fuertemente mayor que } D_{test}$$

$$wc_{test,2} = 182.95 \text{ y } t_{test} + D_{test} = (220, 225, 226) \Rightarrow \Pi(wc_{test,2} < t_{test} + D_{test}) = N(wc_{test,2} < t_{test} + D_{test}) = 1$$

$$wc_{test\_b} = 186 \text{ y } d_{test\_b} = 220 \quad \Rightarrow \quad \Pi(wc_{test\_b} < d_{test\_b}) = N(wc_{test\_b} < d_{test\_b}) = 1$$

#### 7.2.4.2. Resumen y conclusiones

La conclusión de la aplicación de la fase 3 de la metodología propuesta es la misma que con los plazos numéricos: el sistema es no planificable debido a que la tarea de autoverificación es no planificable, con lo que habría que rediseñar el sistema. Sin embargo, mediante un modelo más detallado y realista que el peor caso de algunos parámetros se puede obtener más información del análisis de planificabilidad. Si con los nuevos modelos, que incluyen el caso más desfavorable y otros más favorables, se obtiene que las tareas estrictas están garantizadas en el peor caso, y las firmes tienen un valor de posibilidad de que sean planificables mayor que cero, entonces el diseño es válido de acuerdo con los requisitos del sistema.

#### 7.2.5. Modificación del diseño: plazos, tiempos de ejecución y tiempos de utilización de memoria borrosos

Mediante el modelado borroso de algunos parámetros se presenta de forma más completa la información incierta disponible acerca de los mismos. De esta forma es posible obtener los valores de planificabilidad borrosa del sistema en lugar de únicamente la planificabilidad en el caso más desfavorable. De momento se mantiene en este apartado el análisis con el mínimo tiempo entre llegadas de las tareas esporádicas es 55 u.t., con lo que  $t_{sinc}=55$  y  $t_{inf\_est}=110$ , aunque sí se introduce flexibilidad en los plazos  $d_{inf\_sec}$  y  $d_{test}$  para modelar que son tareas firmes. El tiempo de utilización de la zona de memoria compartida por la tarea menos prioritaria de autoverificación se modela como un número borroso triangular  $RU = (1.8, 1.9, 2)$ , con lo que los tiempos de bloqueo pasan a ser números borrosos iguales a  $RU$ .

Tarea reloj:

Los parámetros temporales de la tarea de reloj son:

$$t_{clk}=5, p_{clk}=0.05, p_{cola}=0.05, d_{clk}=0.05$$

Esta tarea es estricta, por tanto se requiere que  $\Pi(wc_{clk} < d_{clk}) = N(wc_{clk} < d_{clk}) = 1$

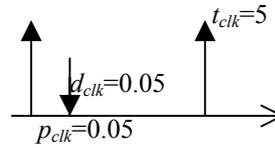


Figura 7.8: Tarea de reloj

Sincronización:

Los parámetros temporales de la tarea de sincronización son:

$$t_{sinc}=55, p_{sinc}=0.1, j_{sinc}=t_{clk}=5, d_{sinc}=10$$

Esta tarea también es estricta, por lo que se requiere que  $\Pi(wc_{sinc} < d_{sinc}) = N(wc_{sinc} < d_{sinc})=1$

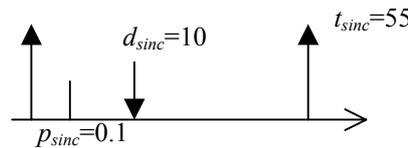


Figura 7.9: Tarea de sincronización

Análisis de señal y código de velocidad:

Los parámetros temporales de la tarea de análisis de señal y código de velocidad son:

$$t_{cod}=65, P_{cod}=(38, 39, 40), j_{cod}=1, B_{cod}=(1.8, 1.9, 2), d_{cod}=65$$

Nótese que se conservan los peores tiempos de ejecución y de bloqueo. Esta tarea es estricta, por lo que se requiere que  $\Pi(WC_{cod} < d_{cod}) = N(WC_{cod} < d_{cod})=1$ . Como el tiempo de ejecución y el bloqueo son borrosos, el tiempo de finalización de esta tarea también lo es.

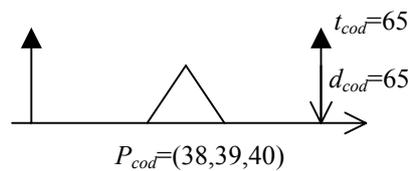


Figura 7.10: Tarea de análisis de la señal y envío del código de velocidad

Envío de informe de estado al módulo principal:

Los parámetros temporales de la tarea de envío de informe de estado al módulo principal son:

$$t_{inf\_est}=110, P_{inf\_est}=(6.9, 7, 8), j_{inf\_est}= t_{clk} =5, B_{inf\_est}=(1.8, 1.9, 2), d_{inf\_est}=t_{inf\_est}= 110$$

Nótese que se conservan los peores tiempos de ejecución y de bloqueo. Esta tarea es estricta, por tanto se requiere que  $\Pi(WC_{inf\_est} < d_{inf\_est}) = N(WC_{inf\_est} < d_{inf\_est})=1$ . De momento, el periodo y el plazo de esta tarea esporádica se modelan con un valor numérico.

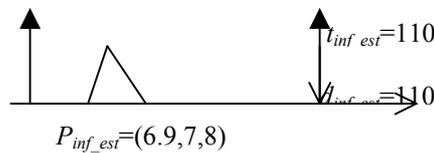


Figura 7.11: Tarea de envío de informe de estado al módulo principal

Recepción de informe de test del procesador secundario:

Los parámetros temporales de la tarea de recepción de informe de test del procesador secundario son:

$$t_{inf\_sec}=110, P_{inf\_sec}=(0.7, 0.8, 1), j_{inf\_sec}=0.5, B_{inf\_sec}=(1.8, 1.9, 2), D_{inf\_sec}=(110, 115, 116)$$

Nótese que se conservan los peores tiempos de ejecución, de bloqueo y del plazo. Esta tarea es firme, por lo que únicamente se requiere que  $\Pi(WC_{inf\_sec} < D_{inf\_sec}) > 0$ . Esto es así porque se considera que si se cumple  $\Pi(WC_{inf\_sec} < D_{inf\_sec}) > 0$ , pero no se cumple  $\Pi(WC_{inf\_sec} < D_{inf\_sec}) = N(WC_{inf\_sec} < D_{inf\_sec})=1$ , entonces puede que se haya recibido parte de los informes del procesador secundario en el instante de plazo y por tanto será posible enviar una parte de los informes al módulo principal. Pero si  $\Pi(WC_{inf\_sec} < D_{inf\_sec}) = 0$  se considera que el diseño no es adecuado ya que es seguro que no hay ningún informe disponible del procesador secundario.

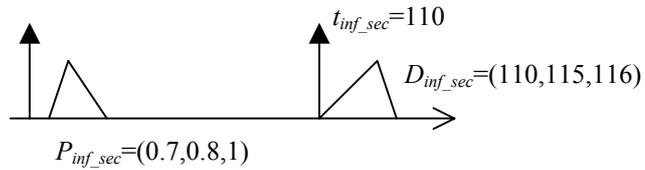


Figura 7.12: Tarea de recepción de informes del procesador secundario

Funciones de autocomprobación:

Los parámetros temporales de la tarea de autocomprobación son:

$t_{test}=110$ ,  $P_{test} = \{ 0.1 / [17, 19), 0.6 / [19, 19.5), 0.9 / [19.5, 20), 1 / 20 \}$  (ver Figura 7.13),  $B_{test}=(1.8, 1.9, 2)$ ,  $D_{test} = (110, 115, 116)$

Nótese que se conservan los peores tiempos de ejecución, de bloqueo y del plazo. Esta tarea se puede considerar como una ejecución incremental ya que está compuesta de varias funciones de autocomprobación: en el instante 17 alguna función ha terminado y se modela de forma que en ese instante la tarea total se ha ejecutado con una posibilidad 0.1. Desde 17 u.t. hasta 19 u.t. se ejecutan otras y no es hasta ese instante 19 que podemos decir que ese conjunto de funciones ha terminado, con lo que la tarea total se ha ejecutado durante el intervalo con la misma posibilidad 0.1 y en 19 u.t. pasa a ser 0.6. Desde 19 u.t. hasta 19.5 u.t. se ejecuta otro conjunto de funciones que no podemos decir que han finalizado hasta 19.5 u.t., por tanto la tarea total se ha ejecutado durante el intervalo con la misma posibilidad 0.6 y en 19.5 u.t. pasa a ser 0.9. Entre 19.5 u.t. y 20 u.t. se ejecutan las funciones restantes y se sabe que en 20 u.t. han finalizado. Por tanto la tarea total se ha ejecutado durante el intervalo con la misma posibilidad 0.9 y en 20 u.t. pasa a ser 1.

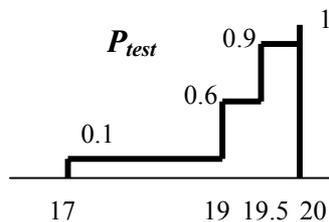


Figura 7.13: Modelo borroso del tiempo de ejecución de la tarea de autocomprobación

Esta tarea es firme, y se requiere que sea posible tener un informe disponible en el instante en el que llegue la siguiente petición de informe de estado. Por

tanto la condición que se debe verificar es  $\Pi(WC_{test} < D_{test}) > 0$ . Esto es así porque se considera que si se cumple que  $\Pi(WC_{test} < D_{test}) > 0$ , pero no se cumple  $\Pi(WC_{test} < D_{test}) = N(WC_{test} < D_{test}) = 1$ , entonces se puede dar la situación admisible de que se hayan ejecutado parte de las funciones de autocomprobación en el instante de plazo y por tanto será posible enviar una parte de los informes al módulo principal. Pero si  $\Pi(WC_{test} < D_{test}) = 0$  se considera que el diseño no es adecuado ya que es seguro que no habrá ningún informe de autocomprobación disponible.

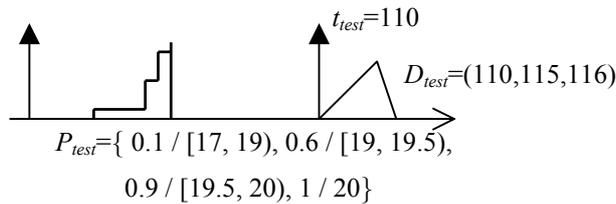


Figura 7.14: Tarea de autocomprobación

Función de autocomprobación en zona de memoria compartida:

Los parámetros temporales de la tarea de autocomprobación en zona de memoria compartida son:

$$t_{test\_b}=220, p_{test\_b}= 5 \text{ y } d_{test\_b}=220$$

Esta tarea es estricta, por tanto se requiere que  $\Pi(WC_{test\_b} < d_{test\_b}) = N(WC_{test\_b} < d_{test\_b})=1$



Figura 7.15: Tarea de autocomprobación en zona de memoria compartida

**7.2.5.1. Análisis de planificabilidad**

**Tarea de recepción de informes del procesador secundario más prioritaria que la tarea de autocomprobación**

En primer lugar, dentro de la fase 3 de la metodología, se va a analizar la planificabilidad del sistema dada la asignación de prioridades que permitía que la tarea de recepción de informes del procesador secundario fuera planificable (por ser más prioritaria que la tarea de autocomprobación). Toda tarea que es planificable en el caso más desfavorable analizado en el modelo clásico es planificable con posibilidad y necesidad igual a 1 en el modelo borroso, ya que se conservan los peores tiempos. A continuación se resumen en una tabla los parámetros del modelo.

Reloj	$t_{clk}=5$	$p_{clk}=0.05,$ $p_{cola}=0.05$			$d_{clk}=0.05$
Sincronización	$t_{sinc}=55$	$p_{sinc}=0.1$	$j_{sinc}=t_{clk}=5$		$d_{sinc}=10$
Análisis de señal y código de velocidad	$t_{cod}=65$	$P_{cod}=(38,39, 40)$	$j_{cod}=1$	$B_{cod}=(1.8,1.9,2)$	$d_{cod}=65$
Envío de informe de estado al módulo principal	$t_{inf\_est}=110$	$P_{inf\_est}=(6.9,7,8)$	$j_{inf\_est}=t_{clk}=5$	$B_{inf\_est}=(1.8,1.9,2)$	$d_{inf\_est}=t_{inf\_est}=110$
Recepción de informe de autocomprobación del procesador secundario	$t_{inf\_sec}=110$	$P_{inf\_sec}=(0.7, 0.8, 1)$	$j_{inf\_sec}=0.5$	$B_{inf\_sec}=(1.8,1.9,2)$	$D_{inf\_sec}=(110,115, 116)$
Funciones de autocomprobación	$t_{test}=110$	$P_{test} = \{0.1/[17,19), 0.6/[19,19.5), 0.9/[19.5,20), 1/20\}$		$B_{test}=(1.8,1.9,2)$	$D_{test}=(110, 115, 116)$
Función de autocomprobación en zona de memoria compartida	$t_{test\_b}=220$	$p_{test\_b}=5$			$d_{test\_b}=220$

Todas las tareas que eran planificables en el modelo clásico (la tarea de reloj, sincronización, análisis de la señal, informe de estado, recepción de informes del

procesador secundario y autocomprobación en zona de memoria compartida), son también planificables con el modelo borroso con posibilidad y necesidad iguales a 1. Recordemos que en el modelo borroso los valores máximos del soporte de los tiempos de ejecución y tiempos de bloqueo borrosos coinciden con los tiempos del modelo clásico y lo mismo sucede con los valores mínimos de los plazos borrosos. Por tanto:

$$\Pi(wc_{clk} < d_{clk}) = N(wc_{clk} < d_{clk}) = 1$$

$$\Pi(wc_{sinc} < d_{sinc}) = N(wc_{sinc} < d_{sinc}) = 1$$

$$\Pi(WC_{cod} < d_{cod}) = N(WC_{cod} < d_{cod}) = 1$$

$$\Pi(WC_{inf\_est} < d_{inf\_est}) = N(WC_{inf\_est} < d_{inf\_est}) = 1$$

$$\Pi(WC_{inf\_sec} < D_{inf\_sec}) = N(WC_{inf\_sec} < D_{inf\_sec}) = 1$$

$$\Pi(WC_{test,2} < t_{test} + D_{test}) = N(WC_{test,2} < t_{test} + D_{test}) = 1$$

$$\Pi(WC_{test\_b} < d_{test\_b}) = N(WC_{test\_b} < d_{test\_b}) = 1$$

Falta por calcular la posibilidad y necesidad para la tarea de autocomprobación en su primera ejecución:  $\Pi(WC_{test,1} < D_{test})$ ,  $N(WC_{test,1} < D_{test})$ .

Para ello se va a utilizar el método de bipartición del algoritmo basado en aritmética de intervalos para prioridades fijas y plazos mayores que los periodos (7.15), (7.16), (7.17), (7.18) y (7.19). Por tanto no se obtendrá el tiempo borroso de finalización, sino únicamente el punto de corte entre el tiempo de finalización borroso y el plazo, con el cual se calcula la posibilidad y la necesidad de que la tarea finalice antes de su plazo.

$\overline{wc}_{test,1}(0)$	$\overline{p}_{cod}(0)=40$ , $\overline{p}_{inf\_est}(0)=8$ , $\overline{p}_{inf\_sec}(0)=1$ , $\overline{p}_{test}(0)=20$ , $\overline{b}_{test}(0)=2$ , $\underline{d}_{test}(0)=110$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 40 + 8 + 1 + 20 = 69.15$
	$s_1 = 112.3$ , $s_2 = 122.05$ , $s_3 = 122.15$ , $s_4 = s_3$
	$\overline{wc}_{test,1}(0) = 122.15$ $\overline{wc}_{test,1}(0) > \underline{d}_{test}(0)$

$\overline{WC}_{test,1}(1)=$ $\overline{WC}_{test,1}(1)$	$\overline{p_{cod}}(1) = \overline{p_{cod}}(1)=39, \overline{p_{inf\_est}}(1) = \overline{p_{inf\_est}}(1)=7, \overline{p_{inf\_sec}}(1)=\overline{p_{inf\_sec}}(1)=0.8,$ $\overline{p_{test}}(1) = \overline{p_{test}}(1)=20, \overline{b_{test}}(1) = \overline{b_{test}}(1)=1.9, \overline{d_{test}}(1)=\overline{d_{test}}(1)=115$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 39 + 7 + 0.8 + 20 = 66.95$
	$s_1 = 109, s_2 = 116.5, s_3 = 117.6, s_4 = s_3$
	$\overline{WC}_{test,1}(1) = \overline{WC}_{test,1}(1) = 117.6 > \overline{d_{test}}(1) = \overline{d_{test}}(1) = 115$
$\overline{WC}_{test,1}(0)$	$\overline{p_{cod}}(0) = 38, \overline{p_{inf\_est}}(0) = 6.9, \overline{p_{inf\_sec}}(0) = 0.7, \overline{p_{test}}(0) = 17, \overline{b_{test}}(0) = 1.8,$ $\overline{d_{test}}(0) = 116$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 38 + 6.9 + 0.7 + 17 = 62.75$
	$s_1 = 65.6, s_2 = 103.7, s_3 = 104.05, s_4 = s_3$
	$\overline{WC}_{test,1}(0) = 104.05 < \overline{d_{test}}(0) = 116$ . Zona de búsqueda por bipartición es la de los límites inferiores de los $\alpha$ -cortes del tiempo de finalización.
$\overline{WC}_{test,1}(0.5)$	$\overline{p_{cod}}(0.5) = 38.5, \overline{p_{inf\_est}}(0.5) = 6.95, \overline{p_{inf\_sec}}(0.5) = 0.75, \overline{p_{test}}(0.5) = 19,$ $\overline{b_{test}}(0.5) = 1.85, \overline{d_{test}}(0.5) = 115.5$
	$s_0 = 65.35, s_1 = 106.85, s_2 = 114.3, s_3 = 115.3, s_4 = 115.35, s_5 = s_4,$
	$\overline{WC}_{test,1}(0.5) = 115.35 < \overline{d_{test}}(0.5) = 115.5$ . Por tanto el punto de corte se encuentra entre 0.5 y 1
$\overline{WC}_{test,1}(0.75)$	$\overline{p_{cod}}(0.75) = 38.75, \overline{p_{inf\_est}}(0.75) = 6.975, \overline{p_{inf\_sec}}(0.75) = 0.775, \overline{p_{test}}(0.75) = 19.5,$ $\overline{b_{test}}(0.75) = 1.875, \overline{d_{test}}(0.75) = 115.75$
	$s_0 = 66.15, s_1 = 107.93, s_2 = 115.4, s_3 = 116.48, s_4 = s_3$
	$\overline{WC}_{test,1}(0.75) = 116.48 > \overline{d_{test}}(0.75) = 115.75$ . Por tanto el punto de corte se encuentra entre 0.5 y 0.75

$WC_{test,1}(0.625)$	$\underline{p}_{cod}(0.625) = 38.625, \underline{p}_{inf\_est}(0.625) = 6.9625, \underline{p}_{inf\_sec}(0.625) = 0.7625,$ $\underline{p}_{test}(0.625) = 19.5, \underline{b}_{test}(0.625) = 1.8625$
	$s_0 = 66, s_1 = 107.64, s_2 = 115.1, s_3 = 116.16, s_4 = s_3$
	$\underline{wc}_{test,1}(0.625) = 116.16 > \overline{d}_{test}(0.625) = 115.625$ . Por tanto el punto de corte se encuentra entre 0.5 y 0.625

Considerando que la precisión es suficiente diremos que el punto de corte se encuentra aproximadamente en  $\alpha=0.6$  con lo que  $\Pi(WC_{test,1} < D_{test}) = 0.6$  y  $N(WC_{test,1} < D_{test}) = 0$ .

Por tanto, la planificabilidad borrosa del sistema es:

$$\Pi_{planif} = \min\{ \Pi(wc_{clk} < d_{clk}), \Pi(wc_{sinc} < d_{sinc}), \Pi(WC_{cod} < d_{cod}), \Pi(WC_{inf\_est} < d_{inf\_est}), \\ \Pi(WC_{inf\_sec} < D_{inf\_sec}), \Pi(WC_{test,1} < D_{test}), \Pi(WC_{test,2} < t_{test} + D_{test}), \Pi(WC_{test\_b} < d_{test\_b}) \}$$

$$N_{planif} = \min\{ N(wc_{clk} < d_{clk}), N(wc_{sinc} < d_{sinc}), N(WC_{cod} < d_{cod}), N(WC_{inf\_est} < d_{inf\_est}), \\ N(WC_{inf\_sec} < D_{inf\_sec}), N(WC_{test,1} < D_{test}), N(WC_{test,2} < t_{test} + D_{test}), N(WC_{test\_b} < d_{test\_b}) \}$$

$\Pi_{planif} = 0.6 \quad N_{planif} = 0$
---

Es importante destacar que si sólo se hubiera realizado el análisis de planificabilidad a partir del modelo clásico, se hubiera descartado este diseño por no ser planificable, pero gracias a este resultado del análisis borroso se puede considerar que el diseño es válido: todas las tareas estrictas son planificables con posibilidad y necesidad 1, el sistema es planificable con posibilidad 0.6 y ese valor es el valor de planificabilidad de una tarea firme, con lo que se considera aceptable.

### 7.2.5.2. Tarea de recepción de informes del procesador secundario menos prioritaria que la tarea de autocomprobación

Si en lugar de tener esta asignación de prioridades se hubiera utilizado la asignación inicial, en la que la tarea de recepción de informes del procesador secundario es menos prioritaria que la tarea de autocomprobación, los valores de planificabilidad borrosa serían menores que en el caso anterior, tal y como se muestra a continuación. Aplicando (7.15), (7.16), (7.17), (7.18) y (7.19) se tiene:

$\overline{WC}_{inf\_sec,1}(0)$	$\overline{p_{cod}}(0)=40, \overline{p_{inf\_est}}(0)=8, \overline{p_{test}}(0)=20, \overline{b_{inf\_sec}}(0)=2, \overline{p_{inf\_sec}}(0)=0.8, \overline{d_{inf\_sec}}(0)=110$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} + p_{inf\_sec} = 0.05 + 0.1 + 40 + 8 + 20 + 1 = 69.15$
	$s_1 = 112.3, s_2 = 141.05, s_3 = 181.4, s_4 = 181.95, s_5 = s_4$
	$\overline{WC}_{inf\_sec,1}(0) = 182.45 > \overline{d_{inf\_sec}}(0) = 110$
$\frac{\overline{WC}_{inf\_sec,1}(1)}{\overline{WC}_{inf\_sec,1}(1)}$	$\overline{p_{cod}}(1) = \overline{p_{cod}}(1)=39, \overline{p_{inf\_est}}(1) = \overline{p_{inf\_est}}(1)=7, \overline{p_{test}}(1) = \overline{p_{test}}(1)=20, \overline{b_{inf\_sec}}(1) = \overline{b_{inf\_sec}}(1)=1.9, \overline{p_{inf\_sec}}(1)=\overline{p_{inf\_sec}}(1)=0.8, \overline{d_{inf\_sec}}(1)=\overline{d_{inf\_sec}}(1)=115$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} + p_{inf\_sec} = 0.05 + 0.1 + 39 + 7 + 20 + 0.8 = 66.95$
	$s_1 = b_{inf\_sec} + p_{inf\_sec} + p_{clk} \cdot \lceil s_0/t_{clk} \rceil + p_{sinc} \cdot \lceil (s_0+j_{sinc})/t_{sinc} \rceil + p_{cod} \cdot \lceil (s_0+j_{cod})/t_{cod} \rceil + p_{inf\_est} \cdot \lceil (s_0+j_{inf\_est})/t_{inf\_est} \rceil + p_{test} \cdot \lceil s_0/t_{test} \rceil + M = 109$ $s_2 = 116.5, s_3 = 136.8, s_4 = 176.05, s_5 = 176.6, s_6 = s_5$
	$\overline{WC}_{inf\_sec,1}(1) = \overline{WC}_{inf\_sec,1}(1) = 177.1 < \overline{d_{inf\_sec}}(1) = \overline{d_{inf\_sec}}(1) = 115$
$\overline{WC}_{inf\_sec,1}(0)$	$\overline{p_{cod}}(0)=38, \overline{p_{inf\_est}}(0)=6.9, \overline{p_{test}}(0)=17, \overline{b_{inf\_sec}}(0)=1.8, \overline{p_{inf\_sec}}(0)=0.7, \overline{d_{inf\_sec}}(0)=116$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{test} + p_{inf\_sec} = 0.05 + 0.1 + 38 + 6.9 + 17 + 0.7 = 62.75$
	$s_1 = b_{inf\_sec} + p_{inf\_sec} + p_{clk} \cdot \lceil s_0/t_{clk} \rceil + p_{sinc} \cdot \lceil (s_0+j_{sinc})/t_{sinc} \rceil + p_{cod} \cdot \lceil (s_0+j_{cod})/t_{cod} \rceil + p_{inf\_est} \cdot \lceil (s_0+j_{inf\_est})/t_{inf\_est} \rceil + p_{test} \cdot \lceil s_0/t_{test} \rceil + M = 1.8 + 0.7 + 0.05 \cdot \lceil 62.75/5 \rceil + 0.1 \cdot \lceil 62.75/55 \rceil + 38 \cdot \lceil 62.75/65 \rceil + 6.9 \cdot \lceil 62.75/110 \rceil + 17 \cdot \lceil 62.75/110 \rceil + 0.35 = 65.6$ $s_2 = 103.7, s_3 = 104.05, s_4 = s_3$
	$\overline{WC}_{inf\_sec,1}(0) = 104.55 < \overline{d_{inf\_sec}}(0) = 116$ . Por tanto la zona de búsqueda por bipartición es la de los límites inferiores del tiempo de finalización.

$\underline{WC}_{inf\_sec,1} (0.5)$	$\underline{p}_{cod} (0.5) = 38.5, \underline{p}_{inf\_est} (0.5) = 6.95, \underline{p}_{test} (0.5) = 19, \underline{b}_{inf\_sec} (0.5) = 1.85,$ $\underline{p}_{inf\_sec} (0.5) = 0.75, \overline{d}_{inf\_sec} (0.5) = 115.5$
	$s_0 = 65.35, s_1 = 106.85, s_2 = 114.3, s_3 = 133.55, s_4 = 172.3, s_5 = 172.85, s_6 = s_5$
	$\underline{WC}_{inf\_sec,1} (0.5) = 173.35 > \overline{d}_{inf\_sec} (0.5) = 115.5$ . Por tanto el punto de corte se encuentra entre 0 y 0.5
$\underline{WC}_{inf\_sec,1} (0.25)$	$\underline{p}_{cod} (0.25) = 38.25, \underline{p}_{inf\_est} (0.25) = 6.925, \underline{p}_{test} (0.25) = 19, \underline{b}_{inf\_sec} (0.25) = 1.825,$ $\underline{p}_{inf\_sec} (0.25) = 0.725, \overline{d}_{inf\_sec} (0.25) = 115.75$
	$s_0 = 65.05, s_1 = 106.28, s_2 = 113.7, s_3 = 132.95, s_4 = 171.45, s_5 = 172, s_6 = s_5$
	$\underline{WC}_{inf\_sec,1} (0.25) = 172.5 > \overline{d}_{inf\_sec} (0.25) = 115.75$ . Por tanto el punto de corte se encuentra entre 0 y 0.25
$\underline{WC}_{inf\_sec,1} (0.125)$	$\underline{p}_{cod} (0.125) = 38.125, \underline{p}_{inf\_est} (0.125) = 6.9125, \underline{p}_{test} (0.125) = 19,$ $\underline{b}_{inf\_sec} (0.125) = 1.8125, \underline{p}_{inf\_sec} (0.125) = 0.7125, \overline{d}_{inf\_sec} (0.125) = 115.875$
	$s_0 = 64.9, s_1 = 105.89, s_2 = 113.4, s_3 = 132.65, s_4 = 171.03, s_5 = 171.58, s_6 = s_5$
	$\underline{WC}_{inf\_sec,1} (0.125) = 172.08 > \overline{d}_{inf\_sec} (0.125) = 115.875$ . Por tanto el punto de corte se encuentra entre 0 y 0.125
$\underline{WC}_{inf\_sec,1} (0.0625)$	$\underline{p}_{cod} (0.0625) = 38.0625, \underline{p}_{inf\_est} (0.0625) = 6.90625, \underline{p}_{test} (0.0625) = 17,$ $\underline{b}_{inf\_sec} (0.0625) = 1.80625, \underline{p}_{inf\_sec} (0.0625) = 0.70625, \overline{d}_{inf\_sec} (0.0625) = 115.9375$
	$s_0 = 64.63125, s_1 = 103.74, s_2 = 104.19, s_3 = s_2$
	$\underline{WC}_{inf\_sec,1} (0.0625) = 104.69 < \overline{d}_{inf\_sec} (0.0625) = 115.9375$ . Por tanto el punto de corte se encuentra entre 0.0625 y 0.125

Por tanto, para una precisión en  $\alpha$  de 0.0625 se tiene que el punto de corte se encuentra aproximadamente en  $\alpha=0.0625$  con lo que  $\Pi(WC_{inf\_sec,1} < D_{inf\_sec}) = 0.0625$  y  $N(WC_{inf\_sec,1} < D_{inf\_sec}) = 0$ . La planificabilidad de la tarea de autocomprobación es mayor que con la asignación anterior ya que se ejecuta con más prioridad, con lo que  $\Pi(WC_{test,1} < D_{test}) \geq 0.6$  y  $N(WC_{test,1} < D_{test}) \geq 0$

La planificabilidad del sistema es el mínimo de las planificabilidades borrosas del sistema, por lo que, cuando la tarea de recepción de informes del procesador secundario es menos prioritaria que la tarea de autocomprobación, queda finalmente:

$$\Pi_{planif} = 0.0625 \quad N_{planif} = 0$$

### 7.2.5.3. Resumen y conclusiones

En el análisis clásico realizado en el apartado 7.2.2.3., donde la tarea de recepción de informes del procesador secundario es menos prioritaria que la tarea de autocomprobación, sólo se puede decir de la tarea de autocomprobación y de la tarea de recepción de informes del procesador secundario que, dada la información disponible del peor caso, ambas son no planificables. En el análisis realizado con el modelo borroso ambas tareas tienen un valor de planificabilidad borrosa distinto de cero, ya que en el modelo se introduce información adicional al peor caso estudiado en el análisis clásico, pero el de la tarea de recepción de informes es muy pequeño (0.0625) debido al retraso que sufre por la ejecución de la tarea de autocomprobación.

Con la otra asignación (donde la tarea de recepción de informes del procesador secundario es más prioritaria que la tarea de autocomprobación), el análisis clásico muestra que la tarea de recepción de informes sí es planificable y que únicamente la tarea de autocomprobación no lo es, dado el peor caso. Cuando se utilizan distribuciones de posibilidad en el modelado de la incertidumbre se puede expresar el requisito de las tareas firmes en términos de medidas de posibilidad, y a partir de los parámetros temporales borrosos se obtiene que la posibilidad de que el sistema sea planificable es 0.6.

Por tanto, con cualquiera de las dos asignaciones, cuando se analiza el sistema en el caso más desfavorable únicamente se puede concluir que no es planificable, mientras que con el modelo borroso los requisitos firmes de las tareas de autocomprobación y recepción de informes del procesador secundario se pueden expresar con medidas de posibilidad de manera que la posibilidad de que estas tareas respeten sus plazos debe ser mayor que cero. La posibilidad de que el sistema sea planificable es distinta de 0 en ambos casos, gracias a la información adicional introducida en el modelo borroso del sistema. En consecuencia se puede ver que es preferible la asignación de prioridades donde la tarea de recepción de informes del procesador secundario es más prioritaria que la tarea de autocomprobación, ya que su valor de planificabilidad borrosa es mayor. Este diseño es válido ya que todas las tareas estrictas son planificables con posibilidad y necesidad igual a 1 y la tarea con posibilidad igual a 0.6 es una tarea firme.

## 7.2.6. Modificación del modelo: periodo de la tarea de envío de informes de estado borroso

En el apartado anterior se han modelado mediante conjuntos borrosos algunos tiempos de ejecución de las tareas y algunos plazos, mientras que el periodo de la tarea de envío de informes al módulo principal, que es una tarea esporádica, se ha modelado para el caso más desfavorable, esto es, para el tiempo mínimo entre activaciones consecutivas, esto es 110u.t.

En este apartado se introduce información adicional en el modelado del periodo de esta tarea con el fin de comprobar si las medidas de posibilidad y necesidad de planificabilidad del sistema se ven afectadas. Así pues, ahora se considera que el periodo es un número triangular borroso  $T_{inf\_est} = (110, 115, 116)$  de manera que se conserva el caso más desfavorable modelado anteriormente que es 110 u.t., tal y como se describe en el apartado 7.2.4. Nótese que coincide con los plazos borrosos de las tareas de recepción de informes del procesador secundario y de autocomprobación. En lo que sigue sólo se analiza la asignación de prioridades donde la tarea de recepción de informes del procesador secundario es más prioritaria que la tarea de autocomprobación, ya que su valor de planificabilidad borrosa es mayor.

### 7.2.6.1. Análisis de planificabilidad

La respuesta temporal de esta tarea teniendo en cuenta el modelo más detallado del periodo de la tarea de envío de informes al módulo principal afecta significativamente al resultado de  $\Pi(WC_{test,l} < D_{test})$  y  $N(WC_{test,l} < D_{test})$ .

Cuando la tarea de recepción de informes del procesador secundario es más prioritaria que la tarea de autocomprobación, el análisis de planificabilidad para el caso más desfavorable del periodo de la tarea de envío de informes al módulo principal  $t_{inf\_est} = 110$  concluye que todas las tareas son planificables con posibilidad y necesidad igual a 1, salvo la tarea de autocomprobación. Por tanto, al flexibilizar el modelo de ese periodo  $T_{inf\_est} = (110, 115, 116)$ , las tareas que eran totalmente planificables en el peor caso, lo siguen siendo con el nuevo modelo, y sólo es preciso volver a analizar la planificabilidad de la tarea de autocomprobación.

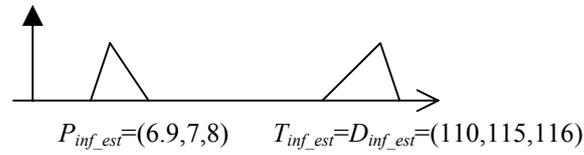


Figura 7.16: Tarea de envío de informe de estado al módulo principal con periodo y plazo borrosos

El modelo del sistema queda reflejado en la siguiente tabla:

Reloj	$t_{clk}=5$	$p_{clk}=0.05,$ $p_{cola}=0.05$			$d_{clk}=0.05$
Sincronización	$t_{sinc}=55$	$p_{sinc}=0.1$	$j_{sinc} =$ $t_{clk}=5$		$d_{sinc}=10$
Análisis de señal y código de velocidad	$t_{cod}=65$	$P_{cod}=(38,39, 40)$	$j_{cod}=1$	$B_{cod} =$ $(1.8,1.9,2)$	$d_{cod}=65$
Envío de informe de estado al módulo principal	$T_{inf\_est} =$ $(110,115,116)$	$P_{inf\_est}=(6.9,7,8)$	$j_{inf\_est} = t_{clk}$ $=5$	$B_{inf\_est} =$ $(1.8,1.9,2)$	$D_{inf\_est} =$ $T_{inf\_est} =$ $(110,115,116)$
Recepción de informe de auto-comprobación del procesador secundario	$t_{inf\_sec}=110$	$P_{inf\_sec} =$ $(0.7,0.8, 1)$	$j_{inf\_sec}=0.5$	$B_{inf\_sec} =$ $(1.8,1.9,2)$	$D_{inf\_sec} =$ $(110,115, 116)$
Funciones de auto-comprobación	$t_{test}=110$	$P_{test} =$ $\{0.1/[17,19),$ $0.6/[19,19.5),$ $0.9/[19.5,20),1/2$ $0\}$		$B_{test} =$ $(1.8,1.9,2)$	$D_{test} =$ $(110,115, 116)$
Función de auto-comprobación en zona de memoria compartida	$t_{test\_b}=220$	$p_{test\_b}=5$			$d_{test\_b}=220$

De nuevo se aplica el algoritmo de cálculo de planificabilidad borrosa basado en aritmética de intervalos (algoritmos (7.15), (7.16), (7.17), (7.18) y (7.19)).

$\overline{WC}_{test,1}(0)$	$\overline{p_{cod}}(0)=40, \overline{p_{inf\_est}}(0)=8, \overline{t_{inf\_est}}(0)=110, \overline{p_{inf\_sec}}(0)=1, \overline{p_{test}}(0)=20, \overline{b_{test}}(0)=2,$ $\overline{d_{test}}(0)=110$
	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 40 + 8 + 1 + 20 = 69.15$ $s_1 = 112.3, s_2 = 122.05, s_3 = 122.15, s_4 = s_3$
	$\overline{WC}_{test,1}(0) = 122.15 > \overline{d_{test}}(0) = 110$
$\overline{WC}_{test,1}(1)$	$\overline{p_{cod}}(1) = \overline{p_{cod}}(1)=39, \overline{p_{inf\_est}}(1) = \overline{p_{inf\_est}}(1)=7, \overline{t_{inf\_est}}(1) = \overline{t_{inf\_est}}(1)=115,$ $\overline{p_{inf\_sec}}(1)=\overline{p_{inf\_sec}}(1)=0.8, \overline{p_{test}}(1) = \overline{p_{test}}(1)=20, \overline{b_{test}}(1) = \overline{b_{test}}(1)=1.9,$ $\overline{d_{test}}(1)=\overline{d_{test}}(1)=115$
$\overline{WC}_{test,1}(1)$	$s_0 = p_{clk} + p_{sinc} + p_{cod} + p_{inf\_est} + p_{inf\_sec} + p_{test} = 0.05 + 0.1 + 39 + 7 + 0.8 + 20 = 66.95$ $s_1 = 109, s_2 = 109.5, s_3 = s_2$
	$\overline{WC}_{test,1}(1) = \overline{WC}_{test,1}(1) = 109.5 < \overline{d_{test}}(1) = \overline{d_{test}}(1) = 115$ . Así pues, la zona de búsqueda por bipartición es en los límites superiores de los $\alpha$ -cortes de los tiempos de finalización.
$\overline{WC}_{test,1}(0.5)$	$\overline{p_{cod}}(0.5)=39.5, \overline{p_{inf\_est}}(0.5)=7.5, \overline{t_{inf\_est}}(0.5)=112.5, \overline{p_{inf\_sec}}(0.5)=0.9,$ $\overline{p_{test}}(0.5)=20, \overline{b_{test}}(0.5)=1.95$
	$s_0 = 68.05, s_1 = 110.65, s_2 = 119.75, s_3 = 119.85, s_4 = s_3$
	$\overline{WC}_{test,1}(0.5) = 119.85 > \overline{d_{test}}(0.5) = 112.5$ . Por tanto el punto de corte se encuentra entre 0.5 y 1
$\overline{WC}_{test,1}(0.75)$	$\overline{p_{cod}}(0.75)=39.25, \overline{p_{inf\_est}}(0.75)=7.25, \overline{t_{inf\_est}}(0.75)=113.75, \overline{p_{inf\_sec}}(0.75)=0.85,$ $\overline{p_{test}}(0.75)=20, \overline{b_{test}}(0.75)=1.925$
	$s_0 = 67.5, s_1 = 109.83, s_2 = 118.43, s_3 = 118.73, s_4 = s_3$
	$\overline{WC}_{test,1}(0.75) = 118.73 > \overline{d_{test}}(0.75) = 113.75$ . Por tanto el punto de corte se encuentra entre 0.75 y 1

$WC_{test,1}$ (0.875)	$\overline{p_{cod}}(0.875)=39.125,$ $\overline{p_{inf\_est}}(0.875)=7.125,$ $\overline{t_{inf\_est}}(0.875)=114.375,$ $\overline{p_{inf\_sec}}(0.875)=0.825,$ $\overline{p_{test}}(0.875)=20,$ $\overline{b_{test}}(0.875)=1.9125,$ $\overline{d_{test}}(0.875)=114.375$
	$s_0 = 67.225, s_1 = 109.41, s_2 = 117.04, s_3 = 118.16, s_4 = s_3$
	$WC_{test,1}(0.875) = 118.16 > \overline{d_{test}}(0.875)=114.375.$ Por tanto el punto de corte se encuentra entre 0.875 y 1
$WC_{test,1}$ (0.9375)	$\overline{p_{cod}}(0.9375)=39.0625,$ $\overline{p_{inf\_est}}(0.9375)=7.0625,$ $\overline{t_{inf\_est}}(0.9375)=114.6875,$ $\overline{p_{inf\_sec}}(0.9375)=0.8125,$ $\overline{p_{test}}(0.9375)=20,$ $\overline{b_{test}}(0.9375)=1.90625,$ $\overline{d_{test}}(0.9375)=114.6875$
	$s_0 = 67.0875, s_1 = 109.21, s_2 = 109.71, s_3 = 117.58, s_4 = 117.88, s_5 = s_4$
	$WC_{test,1}(0.9375) = 117.88 > \overline{d_{test}}(0.9375)=114.6875.$ Por tanto el punto de corte se encuentra entre 0.9375 y 1

Por tanto, para una precisión en  $\alpha$  de 0.0625 se tiene que el punto de corte se encuentra aproximadamente en  $\alpha=1$  con lo que la posibilidad y la necesidad son  $\Pi(WC_{test,1} < D_{test})=1$  y  $N(WC_{test,1} < D_{test})=0$ .

Gracias al análisis borroso se puede ver que los valores de posibilidad y necesidad de que el sistema sea planificable son:

$$\boxed{\Pi_{planif}=1 \quad N_{planif}=0}$$

**7.2.6.2. Resumen y conclusiones**

El resultado que se obtuvo anteriormente con el tiempo mínimo entre llegadas  $t_{inf\_est}=110$  de la tarea de envío de informe de estado ( $\Pi_{planif}=0.6$  y  $N_{planif}=0$ ) era más pesimista que el resultado que se obtiene utilizando el modelo borroso del periodo de dicha tarea  $T_{inf\_est} = (110,115,116)$ .

Se considera que el diseño es válido porque todas las tareas estrictas son planificables con posibilidad y necesidad 1 y el valor de planificabilidad del sistema con necesidad 0, viene de una tarea firme.

### 7.2.7. Resumen de resultados de planificabilidad

Resumiendo, estos son los resultados de planificabilidad borrosa obtenidos en los distintos casos analizados con modelos borrosos de los parámetros.

En el análisis con tiempos de ejecución y tiempos de bloqueo numéricos, y plazos borrosos  $D_{inf\_sec} = (110,115,116)$  y  $D_{test} = (110,115, 116)$ , el periodo de la tarea de envío de informe de estado era también numérico y los resultados de planificabilidad borrosa para cada una de las dos asignaciones de prioridad analizadas fueron:

$t_{inf\_est} = 110$	$pr_{inf\_sec} > pr_{test}$	$\Pi_{planif} = 0$	$N_{planif} = 0$
	$pr_{test} > pr_{inf\_sec}$	$\Pi_{planif} = 0$	$N_{planif} = 0$

En el análisis con tiempos de ejecución, tiempos de bloqueo y plazos borrosos, se supuso primero que el periodo de la tarea de envío de informe de estado era numérico, y después se modeló como un número borroso. En el caso de periodo numérico se comparó el resultado obtenido para dos asignaciones de prioridades, con la tarea de recepción de informes del procesador secundario más prioritaria que la de autocomprobación ( $pr_{inf\_sec} > pr_{test}$ ), y al contrario. Como el resultado era sensiblemente mejor en el caso de  $pr_{inf\_sec} > pr_{test}$ , el análisis con periodo borroso sólo se realizó para dicha asignación de prioridad.

Los parámetros modelados como conjuntos borrosos, tanto con  $t_{inf\_est} = 110$  como con  $T_{inf\_est} = (110, 115, 116)$  son:

$$P_{cod} = (38,39, 40), B_{cod} = (1.8,1.9,2)$$

$$P_{inf\_est} = (6.9,7,8), B_{inf\_est} = (1.8,1.9,2), D_{inf\_est} = (110,115,116)$$

$$P_{inf\_sec} = (0.7,0.8, 1), B_{inf\_sec} = (1.8,1.9,2), D_{inf\_sec} = (110,115, 116)$$

$$P_{test} = \{0.1/[17,19), 0.6/[19,19.5), 0.9/[19.5,20), 1/20\},$$

$$B_{test} = (1.8,1.9,2), D_{test} = (110,115, 116)$$

$t_{inf\_est} = 110$	$pr_{inf\_sec} > pr_{test}$	$\Pi_{planif} = 0.6$	$N_{planif} = 0$
	$pr_{test} > pr_{inf\_sec}$	$\Pi_{planif} = 0.0625$	$N_{planif} = 0$
$T_{inf\_est} = (110, 115, 116)$	$pr_{inf\_sec} > pr_{test}$	$\Pi_{planif} = 1$	$N_{planif} = 0$

Por tanto, si sólo se hubiera realizado el análisis de planificabilidad para el caso más desfavorable, el diseño presentado en este caso ejemplo se hubiera descartado, por no ser planificable. No hubiera quedado más remedio que tratar de reducir el tiempo de respuesta de la tarea no planificable, por ejemplo reduciendo el número de funciones de autocomprobación a ejecutar, o aumentando significativamente su periodo de ejecución. Sin embargo, al introducir un modelado realista acerca de los parámetros en forma de números borrosos, se ha llegado a la conclusión de que el diseño presentado es válido y no es necesario realizar cambio alguno.

**7.2.8. Modificación del diseño: mejora del modelo a partir del análisis borroso de planificabilidad**

Tal y como se acaba de explicar, el diseño del sistema es válido ya que verifica los requisitos especificados, y no es necesario perder funcionalidad reduciendo el número de funciones de autocomprobación a ejecutar. Pero cabría preguntarse si es posible introducir algún cambio que mejore la funcionalidad del sistema sin empeorar los valores borrosos de planificabilidad, es decir, sin aumentar el tiempo de respuesta de la tarea de autocomprobación.

Algunas alternativas serían aumentar el número de informes recibidos del procesador secundario, o reducir el periodo de envío de informe de estado al módulo principal. Pero no debemos olvidar que la tarea más importante es la de análisis de la señal y envío de código de velocidad ya que es la encargada de recibir la consigna de velocidad a la que debe viajar el tren en cada momento. Si se pierde la señal, esta tarea debe detectarlo y dar la alarma para que el tren se detenga. El periodo de esta tarea es el tiempo entre dos envíos consecutivos de consignas de velocidad al módulo principal, por lo que si se pierde la señal, ese periodo es el tiempo máximo que tarda en detectar que se ha perdido la señal. Si se pierde la señal inmediatamente después de que se haya activado la tarea de análisis de la señal y envío de códigos de

velocidad, hasta que no se vuelva a activar después de un tiempo  $t_{cod}$ , esta tarea no detectará que no hay señal disponible.

Por tanto, si se reduce el periodo de esta tarea se consigue que el máximo tiempo de detección de pérdida de la señal se reduzca de igual forma, y con ello también disminuye el tiempo y distancia de frenado del tren desde el instante en el que se pierde la señal.

Se puede comprobar fácilmente que es posible reducir el periodo de la tarea de análisis del espectro desde  $t_{cod} = 65$  hasta  $t_{cod} = 56$  (un 13.8%) sin que cambie la planificabilidad del sistema, que seguiría siendo  $\Pi_{planif}=1$ ,  $N_{planif}=0$ . La reducción de dicho periodo hace que el tiempo de respuesta del sistema en caso de pérdida de código de velocidad sea menor ya que desde que se pierde el código hasta que se detecta pasa menos tiempo, reduciendo así el tiempo y la distancia de frenado del tren. Si se reduce más el periodo de dicha tarea, la tarea de autocomprobación se hace no planificable.

### 7.2.9. Conclusiones

Si únicamente se hubiera realizado el análisis clásico basado en los peores tiempos, la conclusión hubiera sido que el sistema, tal y como está planteado no es planificable y habría que recortar la funcionalidad del sistema, por ejemplo ejecutando menos funciones de autocomprobación. Esto se debe a que no es posible expresar el requisito de las tareas firmes en términos del caso más desfavorable. Para poder expresar el requisito de que el diseño no es válido si es imposible tener un informe disponible en el momento en el que llegue la siguiente petición de estado, es preciso modelar la incertidumbre.

El requisito de las tareas firmes se podría haber expresado en términos de medidas de probabilidad, y se podrían haber estimado funciones de densidad de probabilidad de los parámetros temporales. Sin embargo, por un lado consideramos que cuando no existe información estadística acerca de los parámetros, la forma natural de modelar la incertidumbre es mediante distribuciones de posibilidad, y por otro lado, los algoritmos propuestos en esta tesis para el análisis de planificabilidad borrosa son más rápidos de aplicar que realizar la simulación del sistema a partir de dichas funciones estimadas.

En este caso ejemplo se han empleado distribuciones de posibilidad para modelar la incertidumbre existente en el conocimiento de algunos parámetros temporales, y el requisito de las tareas firmes se ha expresado de forma que se exige que para esas

tareas la posibilidad de que respeten sus plazos ha de ser mayor que cero. Gracias al análisis borroso, realizado a partir del modelado borroso de los parámetros, no sólo se ha comprobado que el diseño es válido, sino que además se ha podido ajustar el periodo de la tarea de análisis de la señal (sin recortar la funcionalidad del sistema) para mejorar el tiempo de respuesta del sistema en caso de pérdida de señal, reduciendo así el tiempo de frenado del tren desde el instante en el que se pierde la señal.



## **8. Conclusiones, aportaciones y futuros desarrollos**

## 8.1. Conclusiones

Los sistemas de tiempo real son sistemas que deben realizar las tareas dentro de los plazos especificados en los requisitos para que su funcionamiento sea considerado correcto. Por ello el análisis de planificabilidad en el diseño de sistemas de tiempo real es fundamental ya que comprueba el cumplimiento de las restricciones temporales. Los análisis de planificabilidad clásicos que se encuentran en la literatura para sistemas monoprocesador con planificador expulsor basado en prioridades, parten del supuesto de que los parámetros temporales que caracterizan a las tareas del sistema se conocen de forma precisa. Lamentablemente, esto no suele ocurrir en los sistemas reales, especialmente en las fases de diseño, por lo que en tal caso, la opción clásica tomada en la literatura consiste en modelar el caso más desfavorable para cada uno de los parámetros y analizar la planificabilidad del sistema en esa situación. Si se obtiene que el sistema es planificable porque respeta todos los plazos de ejecución de las tareas, entonces se puede afirmar que el sistema será planificable en cualquier otra situación más favorable. Por otro lado, si se obtiene que el sistema es no planificable en el caso más desfavorable, no se puede afirmar nada sobre la planificabilidad del sistema en cualquier otra situación. Se trata por tanto de un análisis muy parcial y poco realista que se ocupa únicamente del peor caso.

Por tanto, con el fin de obtener modelos más realistas y de poder establecer requisitos temporales no expresables en términos del caso más desfavorable, es necesario el modelado de la incertidumbre existente en las primeras fases de desarrollo de los sistemas de tiempo real. Para el modelado de la incertidumbre se pueden utilizar modelos probabilistas o modelos posibilistas. La teoría de la posibilidad y la teoría de los conjuntos borrosos fue presentada en los años sesenta y desde entonces se ha ido aplicando a multitud de campos, tanto a nivel teórico como en la industria. Modela un tipo de incertidumbre distinta de la probabilidad y tiene sus propias reglas para combinar la información incierta y para extender funciones clásicas a argumentos borrosos, así como sus propias medidas de incertidumbre sobre la veracidad de una proposición, entre las que se encuentran las medidas de posibilidad y de necesidad. Las funciones reales se extienden a parámetros borrosos aplicando el principio de extensión, cuya utilización práctica directa está muy restringida debido al alto coste computacional que conlleva.

En la fase de diseño de un sistema de tiempo real no suele haber datos estadísticos o medidas acerca de los parámetros temporales de las tareas que lo forman, por lo que en esta tesis se propone un modelo más realista del sistema basado en la utilización de conjuntos borrosos interpretados como distribuciones de posibilidad, para lo que se ha redefinido el concepto de planificabilidad. Cuando los parámetros temporales de las tareas se modelan como distribuciones de posibilidad se ha propuesto medir la

planificabilidad del sistema con un grado de posibilidad y un grado de necesidad de que sea planificable. Las distribuciones de posibilidad permiten modelar tanto el caso más desfavorable como otros valores de los parámetros que son más favorables para la planificabilidad del sistema y que también se consideran posibles, de manera que se puede garantizar la planificabilidad de las tareas estrictas en el peor caso. Los resultados que se obtienen son más complejos pero aportan más información que el análisis del peor caso.

Se han encontrado muy pocos estudios en la literatura que extiendan el análisis de planificabilidad a parámetros borrosos, y casi en ninguno de ellos se ha aplicado la teoría de la posibilidad sin introducir simplificaciones en el cálculo borroso o sin definir otras medidas, diferentes de las propuestas por la teoría de la posibilidad, del cumplimiento de una restricción temporal borrosa. En algunos estudios se ha aplicado aritmética borrosa sin considerar que, al ser una simplificación del principio de extensión, no siempre produce los mismos resultados que la aplicación del principio de extensión de la teoría de la posibilidad y se han introducido simplificaciones en las operaciones borrosas.

En esta tesis se ha demostrado que es posible realizar un análisis riguroso de planificabilidad partiendo de información borrosa, de acuerdo con la teoría de la posibilidad, pero de forma más rápida que si se aplica directamente el principio de extensión. Todos los algoritmos alternativos de planificabilidad borrosa propuestos están basados en aritmética borrosa y aritmética de intervalos, y permiten obtener aproximaciones válidas de los conjuntos borrosos que se obtendrían al aplicar el principio de extensión. Una aproximación válida de un conjunto borroso se ha definido como cualquier conjunto borroso que tenga los mismos límites inferiores y superiores de los  $\alpha$ -cortes, y verifica que la posibilidad y necesidad de la comparación con otro conjunto son las mismas que las que se obtienen al comparar el conjunto borroso inicial. Gracias a esta propiedad, los resultados de planificabilidad borrosa que se obtienen mediante los algoritmos alternativos propuestos son iguales que los que se tienen al aplicar el principio de extensión al análisis de planificabilidad del sistema.

Además se ha mostrado que los algoritmos propuestos son también aplicables cuando los parámetros se modelan como conjuntos borrosos no convexos, y que se puede hacer todo el estudio considerando siempre aproximaciones válidas convexas de los conjuntos de partida.

Se ha considerado interesante comparar los resultados de planificabilidad con parámetros modelados como conjuntos borrosos y con parámetros modelados mediante distribuciones de probabilidad. Por un lado, los resultados son similares atendiendo a la asignación de prioridades que maximiza la planificabilidad del sistema y, por otro lado, el cálculo de la planificabilidad mediante los algoritmos

propuestos en esta tesis para parámetros modelados como conjuntos borrosos es significativamente más rápido.

Los algoritmos de análisis de planificabilidad desarrollados en esta tesis han permitido establecer una metodología de diseño de sistemas de tiempo real cuyas fases son:

1. Descripción del entorno, del sistema, del interfaz y requisitos del sistema.
2. Diseño y modelado del sistema: tareas y planificación.
3. Análisis de planificabilidad.
4. Modificación del diseño (vuelta a fase 3).

Una vez realizada la modificación del diseño, en la fase 4 de la metodología, con el fin de mejorar el sistema, se vuelve a la fase 3 para realizar el análisis de planificabilidad del sistema con esa modificación. El trabajo de esta tesis se ha centrado en el desarrollo de la fase de análisis de la planificabilidad, donde se han propuesto extensiones del análisis de planificabilidad en muy diversas situaciones, como por ejemplo, en sistemas con prioridades fijas y plazos menores o iguales que los periodos, con prioridades fijas y plazos mayores que los periodos, con tareas aperiódicas atendidas por servidores aperiódicos, con recursos compartidos y con retardos en la activación de las tareas, y con prioridades dinámicas asignadas según el algoritmo EDF. Hay que destacar que no se trata de un estudio exhaustivo de todas las posibles soluciones de diseño de un sistema de tiempo real, aunque se muestra un grupo muy diverso de soluciones.

Así pues, como conclusiones principales de esta tesis se tiene que:

- gracias a la aplicación de la teoría de la posibilidad al modelado de las tareas en el proceso de diseño de sistemas de tiempo real monoprocesador con planificador expulsor basado en prioridades, se pueden obtener diseños válidos que habrían sido rechazados si únicamente se hubiese realizado el análisis de planificabilidad en el caso más desfavorable
- es posible aplicar de forma rigurosa la teoría de la posibilidad de manera efectiva y ágil en el diseño de sistemas gracias a los algoritmos propuestos, alternativos a la aplicación del principio de extensión
- el modelado basado en distribuciones de posibilidad tiene ventajas frente al modelado mediante funciones de densidad de probabilidad, ya que por un lado el primero es la forma natural de modelar la incertidumbre cuando no existe

información estadística en las primeras fases de diseño y, por otro lado, la aplicación de los algoritmos propuestos de análisis de planificabilidad borrosa es más rápida que la simulación que se debería realizar a partir de las funciones de densidad de probabilidad estimadas.

La aplicación de la metodología propuesta a un caso ejemplo basado en un sistema real perteneciente al sistema de protección automática ferroviario ha mostrado las siguientes ventajas:

1. El análisis clásico basado en el peor caso es insuficiente.

En este ejemplo, el diseño inicial donde se modela el sistema para el caso más desfavorable tiene que ser rechazado ya que con cualquier asignación de prioridades fijas siempre hay alguna tarea que no respeta su plazo de ejecución. Si únicamente se pudiera realizar este tipo de análisis clásico basado en el peor caso, habría que modificar drásticamente el diseño cambiando los requisitos del sistema, por ejemplo eliminando alguna tarea no crítica, o recortando su funcionalidad para reducir el tiempo de ejecución.

2. El modelado borroso es más realista.

En lugar de recortar la funcionalidad del sistema, se propone modelar la incertidumbre existente en el conocimiento de las tareas. Por otro lado, existen requisitos del sistema no expresables en términos del peor caso, como ocurre con el requisito de las tareas firmes del caso ejemplo, y que pueden ser expresados modelando el grado en el que se permite que la tarea sea planificable.

Dado que en la fase de diseño de un sistema en general no hay información estadística acerca de los parámetros temporales del sistema, el modelado borroso de alguno de los parámetros a partir del conocimiento de los expertos resulta una alternativa natural. En el ejemplo se han modelado como números borrosos triangulares algunos de los tiempos de ejecución de las tareas, el tiempo de utilización de la zona de memoria compartida, y alguno de los plazos. Otro de los tiempos de ejecución se ha modelado como un conjunto borroso no triangular. Todos ellos han modelado el peor caso analizado anteriormente y otros valores más favorables para la planificabilidad del sistema. El requisito de las tareas firmes se expresa en este contexto permitiendo que la planificabilidad de la tarea tenga un valor de posibilidad simplemente mayor que cero.

3. Permite garantizar la planificabilidad de tareas estrictas.

Para que un diseño que incluya incertidumbre en el modelado de los parámetros sea considerado válido, la condición que se debe exigir a las tareas estrictas es que la posibilidad y la necesidad de que respeten sus plazos sean igual a 1. Esta

condición no se exige para tareas flexibles y firmes. Con estas condiciones el modelado borroso permite garantizar la planificabilidad de las tareas estrictas.

4. El resultado de planificabilidad obtenido proporciona más información.

Si se introduce incertidumbre en el modelado de los parámetros y se realiza el análisis borroso se llega a la conclusión de que el diseño es adecuado a la especificación del sistema ya que, aunque en el peor caso no se respetan todos los plazos, todas las tareas estrictas los respetan con posibilidad y necesidad igual a 1. Por ejemplo, la tarea firme que no respeta su plazo en el peor caso es planificable con un valor de posibilidad mayor que cero, que es la condición que se exige para que sea admisible el diseño ya que indica que es posible que esa tarea ejecute parte de las funciones de auto comprobación antes de la finalización de su plazo.

5. Introducción de mejoras en la funcionalidad del sistema.

Es posible introducir mejoras en el sistema sin empeorar los valores de posibilidad y necesidad de planificabilidad obtenidos. Así, en el ejemplo, reduciendo el periodo de una de las tareas se pueden conseguir mejores tiempos de detección de fallos y por tanto, de frenado del tren, sin empeorar la posibilidad y la necesidad de que el sistema sea planificable, y quedando garantizada la planificabilidad de las tareas estrictas.

## **8.2. Aportaciones**

Se destacan como principales aportaciones de esta tesis la extensión de la metodología de diseño de sistemas de tiempo real con parámetros borrosos y la extensión del análisis de planificabilidad de sistemas de tiempo real a parámetros borrosos de forma sencilla y obteniendo los mismos resultados de posibilidad y necesidad que si se aplicara el principio de extensión de la teoría de la posibilidad.

Se pueden agrupar las aportaciones de esta tesis en conceptos y propiedades matemáticas, algoritmos de análisis de planificabilidad borrosa, extensión de la metodología de diseño de sistemas de tiempo real y aportaciones en el campo de la teoría de la evidencia.

## 8.2.1. Conceptos y propiedades matemáticas

### Planificabilidad borrosa

Se ha propuesto en esta tesis una definición de planificabilidad borrosa basada en las medidas de posibilidad y de necesidad de que las tareas finalicen antes de sus plazos de ejecución. Así, la posibilidad de que el sistema sea planificable se calcula como el mínimo de las posibilidades de que los tiempos de finalización borrosos de las tareas estén antes de los plazos correspondientes y la necesidad de que el sistema sea planificable se calcula como el mínimo de las necesidades de que los tiempos de finalización estén antes de sus plazos.

Cuando el análisis de planificabilidad clásico se basa en la utilización del procesador, la planificabilidad borrosa se define como la posibilidad y la necesidad de que se verifique la cota del tiempo de utilización. Cuando el análisis de planificabilidad clásico se basa en el criterio de demanda del procesador, se define la planificabilidad del sistema como el mínimo valor de posibilidad y de necesidad de que se verifiquen las desigualdades asociadas a cada uno de los plazos absolutos de las tareas en las sucesivas ejecuciones.

### Aproximación válida de un conjunto borroso

Para calcular la planificabilidad borrosa de un sistema se han propuesto mecanismos alternativos a la aplicación del principio de extensión de la teoría de la posibilidad, más eficientes que éste. La condición que deben verificar es que proporcionen los mismos resultados de posibilidad y de necesidad de que el sistema sea planificable que la aplicación del principio de extensión. Para ello los conjuntos borrosos que proporcionan estos métodos alternativos deben ser aproximaciones válidas de los que proporciona el principio de extensión. Se denomina en esta tesis aproximación válida de un conjunto borroso a todo conjunto borrosos que tenga los mismos límites inferiores y superiores de los  $\alpha$ -cortes que el original. Estas aproximaciones se denominan válidas porque, tal y como se ha mostrado en el Anexo B, los valores de posibilidad y de necesidad que se obtienen al comparar la aproximación válida con cualquier otro número, clásico o borroso, o con otro conjunto borroso no convexo son los mismos que si se compara el conjunto borroso original. Por tanto, los valores que se obtienen de planificabilidad borrosa mediante aproximaciones válidas coinciden con los que se obtienen al aplicar el principio de extensión.

### **Extensión de funciones**

Se han determinado las propiedades matemáticas que debe verificar una función para que su extensión mediante aritmética borrosa y aritmética de intervalos sea una aproximación válida de la que proporciona el principio de extensión.

En efecto, en esta tesis se ha mostrado que se puede extender una función, continua o discontinua (con un número infinito de discontinuidades y numerables), mediante aritmética de intervalos y obtener una aproximación válida del resultado proporcionado por el principio de extensión si verifica que es monótona en los parámetros modelados mediante conjuntos borrosos.

Por otro lado, para extender una función mediante aritmética de intervalos y obtener una aproximación válida debe ser siempre no creciente o siempre no decreciente en cada operación parcial que involucre a los parámetros modelados mediante conjuntos borrosos.

### **Asignación óptima de prioridades**

Cuando los parámetros temporales que describen a las tareas se modelan como distribuciones de posibilidad, el concepto de asignación óptima de prioridades se ha redefinido según las medidas de posibilidad y de necesidad de planificabilidad del sistema, de forma que diremos que una asignación es óptima cuando maximiza la posibilidad y la necesidad de que el sistema sea planificable.

Además se ha mostrado que cuando existe un orden fuerte entre los plazos borrosos la asignación dada por dicho orden fuerte es óptima.

### **8.2.2. Extensión de la metodología de diseño de sistemas de tiempo real con incertidumbre**

Se ha extendido la metodología de diseño de sistemas de tiempo real teniendo en cuenta el modelado de la incertidumbre en los parámetros temporales del sistema mediante distribuciones de posibilidad. El análisis de planificabilidad del sistema, a realizar como parte de esta metodología, proporciona una medida de posibilidad y una medida de necesidad de que el sistema sea planificable y permite garantizar la planificabilidad de las tareas estrictas en el caso más desfavorable.

La metodología propuesta permite imponer requisitos no expresables en términos del peor caso y aprovechar así la flexibilidad que introducen, de manera que un diseño

que debería ser rechazado atendiendo únicamente al peor caso, mediante el modelado de la incertidumbre se puede considerar válido. Por otro lado, se propone que la forma de introducir la incertidumbre sea mediante distribuciones de posibilidad ya que normalmente no hay información estadística acerca de los parámetros temporales del sistema, y por otro lado, el proceso iterativo de diseño en sus sucesivas modificaciones resulta más ágil si se emplean los algoritmos propuestos a partir del modelado borroso, que si hay que realizar una simulación por Montecarlo a partir de un modelo probabilista estimado de los parámetros, o un análisis estadístico directo.

Además este modelado más detallado y realista del sistema que el que se tiene en el caso más desfavorable permite también introducir mejoras en la funcionalidad del sistema aprovechando la información disponible.

La metodología propuesta se ha aplicado a un caso ejemplo de un sistema de tiempo real situado en el sistema de protección automática ferroviario, mostrando la flexibilidad de modelado que se pueden obtener al introducir la incertidumbre en el modelo del sistema y en su análisis de planificabilidad.

### **8.2.3. Análisis de planificabilidad**

Se ha mostrado que en muchos algoritmos de análisis de planificabilidad existentes en la literatura, muchos de ellos consistentes en calcular los tiempos de finalización, se cumplen las propiedades necesarias para poder aplicar aritmética borrosa y aritmética de intervalos con los mismos resultados que la aplicación del principio de extensión. Por ello se han propuesto algoritmos de planificabilidad borrosa basados en aritmética borrosa y en aritmética de intervalos para prioridades fijas, con plazos menores, iguales o mayores que los periodos, con recursos de datos compartidos, con “release jitter” o retraso de la activación de las tareas, y con dos tipos de servidores de tareas aperiódicas. Además, se ha desarrollado el algoritmo de cálculo borroso del tiempo de bloqueo a partir de los tiempos de utilización del recurso compartido cuando el protocolo utilizado es el PIP (protocolo de herencia de prioridades). Los algoritmos permiten calcular la planificabilidad cuando se tienen modelos borrosos de los tiempos de ejecución de las tareas, los plazos de ejecución, los periodos, los tiempos de bloqueo, los “release jitter”, así como la capacidad del servidor aperiódico y su periodo.

#### **Prioridades fijas**

Así, los algoritmos propuestos de análisis de planificabilidad de sistemas de tiempo real basados en prioridades fijas mediante aritmética borrosa y mediante aritmética de intervalos contemplan los siguientes casos:

- Plazos menores, iguales o mayores que los periodos
- Recursos de datos compartidos
- Retardos en la activación de las tareas o “release jitter”
- Servidores de tareas aperiódicas: servidor esporádico y servidor aplazable

Se han introducido mejoras en la inicialización del algoritmo de análisis de planificabilidad borrosa basado en aritmética de intervalos, para reducir el tiempo de cálculo de la planificabilidad borrosa del sistema.

Una vez desarrollados los algoritmos de análisis de planificabilidad borrosa para sistemas con prioridades fijas, se ha presentado una comparación entre el análisis de planificabilidad a partir de tiempos de ejecución borrosos calculado mediante aritmética de intervalos, y el análisis a partir de tiempos de ejecución modelados como distribuciones de probabilidad. Para ello se han escogido números borrosos triangulares y las distribuciones de probabilidad se han obtenido a partir de los números borrosos mediante la transformación basada en normalización que conserva la forma triangular, y mediante la transformación basada en máxima especificidad y razón insuficiente. La comparación se ha realizado teniendo en cuenta:

- La mejor asignación de prioridades fijas para cada modelo
- El grado de planificabilidad de la mejor asignación en relación con las otras asignaciones
- La rapidez del método cuantificada en número de llamadas al algoritmo básico

La conclusión que se ha obtenido es que la mejor asignación coincidía en la mayoría de los casos, y en aquellos en los que no coincidían las asignaciones elegidas por el método de posibilidad y por los métodos de probabilidad tenían un grado de planificabilidad significativamente mayor al resto de las asignaciones. Además el cálculo basado en posibilidad es mucho más rápido que los basados en los dos modelos probabilistas.

En esta tesis se ha hecho un estudio comparativo de las extensiones propuestas en esta tesis con las presentadas por Litoiu y Tadei. Los autores contemplan únicamente la extensión para tiempos de ejecución y plazos borrosos en el caso de que los plazos sean menores o iguales a los periodos y no estudian la existencia de “release jitter” ni servidores aperiódicos. Además, no miden la planificabilidad a través de la posibilidad y la necesidad de que los tiempos de finalización estén antes de los correspondientes plazos, sino que definen unas funciones particulares que denominan de satisfacción. Otra diferencia con los análisis propuestos en esta tesis es que en el

algoritmo de Litoiu y Tadei, basado únicamente en aritmética borrosa, se introduce una simplificación en el cociente borroso, que es sustituido por un cociente clásico para el peor caso, obteniendo, por tanto, un resultado de planificabilidad más pesimista que el que se tendría de aplicar el principio de extensión. Se pueden encontrar ejemplos en los que aplicando el algoritmo de Litoiu y Tadei se concluye que el sistema es completamente no planificable y mientras que aplicando el principio de extensión se tiene que el sistema es casi completamente planificable.

### **Prioridades dinámicas**

Se ha extendido también el análisis de planificabilidad para sistemas planificados mediante el algoritmo EDF basado en prioridades dinámicas. Se han propuesto extensiones, basadas en aritmética borrosa y en aritmética de intervalos, de la ecuación clásica de planificabilidad basada en la utilización del procesador, cuando los plazos son iguales a los periodos. En estos casos se calcula la planificabilidad borrosa tanto para tiempos de ejecución borrosos como para periodos borrosos.

Por otro lado, se ha probado que para este mismo algoritmo de planificación EDF es posible extender las ecuaciones de planificabilidad cuando los plazos son menores que los periodos, mediante aritmética borrosa y aritmética de intervalos, sólo para tiempos de ejecución borrosos, pero no es posible hacerlo para plazos o periodos borrosos obteniendo los mismos resultados que la aplicación del principio de extensión.

Todos los algoritmos propuestos en esta tesis para el análisis de planificabilidad de sistemas basados en prioridades fijas o en prioridades dinámicas asignadas según EDF son aplicables también si los parámetros se modelan mediante conjuntos borrosos no convexos.

### **8.2.4. Teoría de la evidencia**

En esta tesis también se ha propuesto una representación gráfica del cuerpo de evidencia conjunto. Esta representación permite mostrar los cuerpos de evidencia consonantes asociados a dos conjuntos borrosos que modelan dos parámetros, el cuerpo de evidencia conjunto, y la forma de calcular la distribución de posibilidad o plausibilidad conjunta. Además, permite calcular la distribución de posibilidad o plausibilidad de cualquier función que relacione los dos parámetros de partida. Se ha estudiado el caso de tener correlación positiva perfecta entre los cuerpos de evidencia, correlación negativa perfecta e independencia y se ha propuesto una interpretación para cada una de estas tres correlaciones.

Este método gráfico se ha aplicado al cálculo de tiempos de finalización borrosos en sistemas de tiempo real formados por dos tareas con prioridades fijas.

### **8.3. Futuros desarrollos**

Tal y como ya se ha dicho, la extensión de la metodología de diseño de sistemas de tiempo real propuesta en esta tesis no abarca de manera exhaustiva todas las soluciones de diseño de un sistema de tiempo real. Así pues, la continuación natural de los estudios de esta tesis llevan a su ampliación en el campo el análisis de las relaciones de precedencia, que en esta tesis se ha tratado de forma muy somera, los “offsets” o desfases como método de análisis de la planificabilidad, el análisis de otros algoritmos basados en prioridades dinámicas como el algoritmo LSF (“Least Slack First”: primero el menos holgado), y el análisis con otros servidores de tareas aperiódicas. Así mismo se podría aplicar el análisis borroso de planificabilidad al caso de tener plazos internos, que describen de forma realista los bloques en que se dividen las tareas y a qué bloques afectan los plazos.

Litoiu y Tadei ([Litoiu, Tadei, 1997]) han propuesto un algoritmo de búsqueda de la asignación de prioridades fijas que maximizan la medida de satisfacción por ellos definida cuando los plazos son borrosos. En lugar de maximizar dicha medida, en futuros desarrollos se podría obtener un método de búsqueda rápida de la asignación óptima que maximice la posibilidad y la necesidad de que el sistema sea planificable no sólo para plazos borrosos sino también para otros parámetros.

En el diseño de los sistemas reales, algunas tareas pueden estar en fases de desarrollo más avanzadas que otras, e incluso estar terminadas y probadas por ser componentes reutilizados de otros sistemas. De esta forma, es posible que existan datos estadísticos acerca de los parámetros de algunas tareas. Por esta razón, uno de los estudios a realizar en el futuro es el análisis de planificabilidad de sistemas de tiempo real partiendo de información mixta, de manera que algunos parámetros estén modelados mediante funciones de densidad de probabilidad y otros mediante conjuntos borrosos.

Además, se pueden aplicar los resultados del análisis de planificabilidad borrosa al diseño de un planificador que sea capaz de aprovechar el modelado de la incertidumbre para la toma de decisiones en tiempo real acerca de la tarea que debe ser ejecutada en cada momento, por ejemplo, mediante un test de aceptación en tiempo de ejecución.

Los estudios realizados en esta tesis han estado centrados en el análisis de planificabilidad de sistemas monoprocesador, por lo que sería también de gran interés estudiar la aplicabilidad de los resultados obtenidos al análisis de sistemas distribuidos.

Por último señalar que los algoritmos propuestos para el análisis de planificabilidad borrosa se podrían integrar en el entorno de desarrollo de sistemas digitales basado en el lenguaje MODUS (lenguaje de modelado unificado basado en simulación), desarrollado en el Instituto de Investigación Tecnológica para el modelado de sistemas digitales [González Arechavala 2003].



## **A. ANEXO: Representación gráfica del cuerpo de evidencia**

El objetivo de este anexo es proponer una representación gráfica de un cuerpo de evidencia conjunto que modela el conocimiento incierto de dos argumentos. Mediante dicha representación se puede calcular la distribución de plausibilidad de una función real de dos argumentos.

Esta representación tiene puntos en común con la propuesta en [Wang et al., 1991] y [Tan et al., 1993] asociada a la teoría de las sombras y que se describe en este anexo, pero la representación aquí propuesta permite obtener gráficamente la extensión de funciones a argumentos inciertos.

Se comparan las distribuciones de plausibilidad conjuntas asociadas a tres cuerpos de evidencia conjuntos, con las distribuciones de posibilidad conjuntas obtenidas en teoría de la posibilidad mediante tres t-normas aplicadas a dos conjuntos borrosos. Así mismo se comparan las distribuciones de plausibilidad obtenidas al extender una función real de dos argumentos a parámetros inciertos.

Esta representación gráfica se aplica en esta tesis al análisis de planificabilidad de sistemas de tiempo real, en la obtención del tiempo de finalización borroso de una tarea a partir de los tiempos de ejecución borrosos. Además, esta representación permite obtener conclusiones interesantes para acelerar el cálculo de la planificabilidad mediante el análisis de intervalos.

## A.1. Teoría de las sombras

En [Wang et al., 1991], [Tan et al., 1993] los autores proponen una forma de representación de un conjunto aleatorio y su “sombra” que es el conjunto borroso asociado. Los distintos tipos de relación entre dos conjuntos aleatorios se pueden representar en el cuadrado unidad, de forma que para cada par  $(s,t)$  en  $[0,1]^2$  representa un elemento de cada conjunto aleatorio. Proponen un método gráfico de cálculo de la unión y la intersección de dos conjuntos borrosos a partir de la distribución de probabilidad conjunta en  $[0,1]^2$ , y analizan el caso de tener correlación positiva perfecta en dicha distribución, independencia y correlación negativa perfecta.

Un conjunto aleatorio  $\xi$  finito en  $U$  se puede definir como un par  $(F,m)$  donde  $F$  es una familia finita de subconjuntos diferentes de  $U$  ( $F \subseteq \mathcal{P}(U)$ ) y  $m$  es una aplicación  $m: F \rightarrow [0,1]$ , tal que  $\sum_{A \in F} m(A) = 1$ .  $F$  es el soporte del conjunto aleatorio y  $m$  es una asignación básica de probabilidad, por lo que se pueden relacionar los conjuntos aleatorios con la teoría de la evidencia.

Expresado de otra forma, dado un espacio de probabilidad  $(\Omega, \mathcal{A}, P)$  y una estructura hipermedible  $(\mathcal{P}(U), \mathcal{B})$  en un universo  $U$ , se define un conjunto aleatorio en  $U$  a la función  $\xi: \Omega \rightarrow \mathcal{P}(U)$ , que es  $\mathcal{A}$ - $\mathcal{B}$  medible.

La función de cobertura de un conjunto aleatorio  $\xi$ , denominada  $\hat{\xi}$ , se define como la probabilidad de  $\omega$  para la que  $u \in \xi(\omega)$ , esto es,  $\hat{\xi}: U \rightarrow [0,1]$ , donde

$$\hat{\xi}(u) = P(\omega \mid u \in \xi(\omega)) \quad \forall u \in U, \omega \in \Omega, \xi(\omega) \in \mathcal{P}(U)$$

Cualquier conjunto borroso en un universo  $U$  se puede ver como la función de cobertura de algunos conjuntos aleatorios en  $U$ . Así,  $\hat{\xi}$  representa un conjunto borroso  $A$  en  $U$  y se tiene:

$$\hat{\xi} = A$$

El espacio  $\Omega$  se puede reducir y representar mediante  $[0,1]$ , con lo que el espacio de probabilidad resultante queda  $([0,1], \mathcal{B}, m)$ , donde  $\mathcal{B}$  es un campo de Borel en  $[0,1]$  y  $m$  la medida de Lebesgue. Con lo que se tiene  $\xi: [0,1] \rightarrow \mathcal{P}(U)$  definido por:

$$\xi: \lambda \rightarrow A_\lambda, \lambda \in [0,1]$$

donde  $A_\lambda$  es el  $\lambda$ -corte del conjunto borroso  $A$ , esto es,  $A_\lambda = \{u \in U \mid A(u) \geq \lambda\}$

En la teoría de “falling shadows” o de las sombras, esos conjuntos aleatorios se denominan nubes y el conjunto borroso se denomina sombra (“falling shadow”) de las nubes [Wang et al., 1991]. Supuesto que el sol está en la vertical, cuanto más gruesa sea la nube más oscura será la sombra de la misma proyectada sobre el suelo (ver Figura A.1).

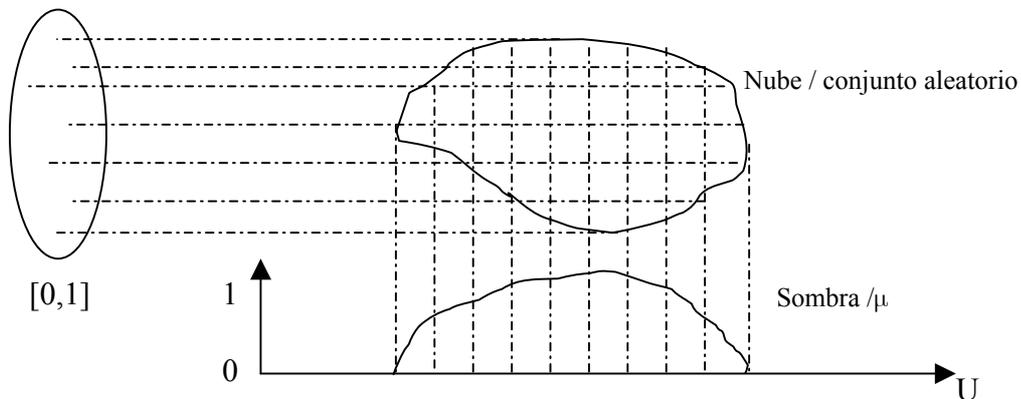


Figura A.1: Representación de la nube (conjunto aleatorio) y su sombra (conjunto borroso)

Dado un subconjunto borroso cualquiera  $A$  en  $U$ , existen infinitos conjuntos aleatorios cuya función de cobertura es  $A$ , y por tanto, existen infinitas nubes cuya sombra es  $A$ .

Si se define  $\xi(s)=A_s$ , con  $s \in [0,1]$ , donde  $A_s$  es el  $\alpha$ -corte de nivel  $s$  de  $A$ ,  $A_s = \{u \in U \mid \mu_A(u) \geq s\}$ , entonces  $\xi$  es un conjunto aleatorio en  $[0,1]$ . Ese conjunto aleatorio es la nube-corte de  $A$ .  $\xi: s \rightarrow A_s, s \in [0,1]$

Sean dos subconjuntos borrosos  $A$  y  $B$  en  $U$ , cuyas nubes-corte son respectivamente  $\xi_1$  y  $\xi_2$ . Se puede considerar un conjunto aleatorio en el universo bidimensional  $U \times U$ , y redefinir los conjuntos aleatorios a partir de la distribución conjunta como  $\xi: [0,1]^2 \rightarrow \mathcal{P}(U)$  de forma que  $\xi_1$  y  $\xi_2$  son las proyecciones:

$$\xi_1: (s,t) \rightarrow s \rightarrow \xi_1(s)=A_s$$

$$\xi_2: (s,t) \rightarrow t \rightarrow \xi_2(t)=B_t$$

Con lo que para cada par  $(s,t) \in [0,1]^2$  hay dos conjuntos clásicos  $A_s$  y  $B_t$  asociados (ver Figura A.2).

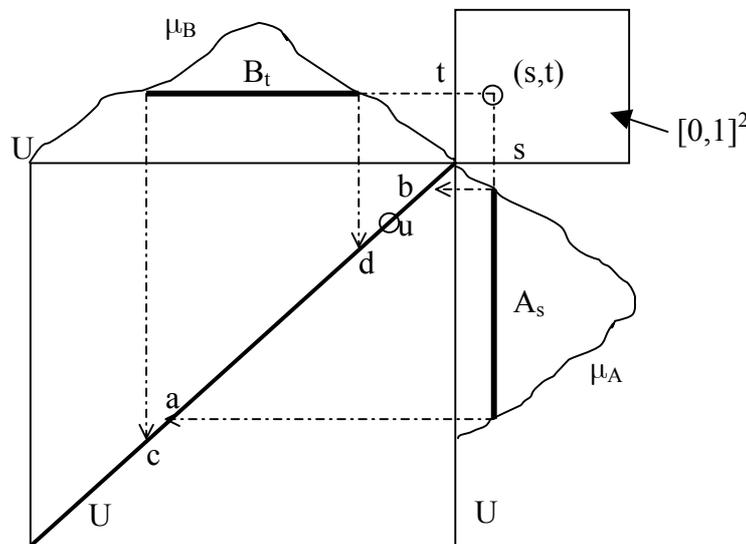


Figura A.2: Representación de dos nubes y sus sombras

Los distintos tipos de relación entre  $\xi_1$  y  $\xi_2$  se pueden representar en el cuadrado unidad, de forma que para cada par  $(s,t)$  en  $[0,1]^2$  se tiene  $\xi_1(s)=A_s$  y  $\xi_2(t)=B_t$ . Se pueden proyectar las imágenes de  $\xi_1$  y  $\xi_2$  en la diagonal principal de  $U^2$  (ver Figura

A.2):  $A_s = (a,b)$  y  $B_t = (c,d)$ . Ambos son conjuntos clásicos y se puede operar con ellos para obtener la unión y la intersección en la diagonal de  $U^2$ . Debido a la equivalencia entre la diagonal de  $U^2$  y  $U$ , los conjuntos resultantes son subconjuntos de  $U$ . Al variar el par  $(s,t)$ , los conjuntos resultantes también varían, formando conjuntos aleatorios en  $U$ . De esta forma, se puede conseguir la unión y la intersección de conjuntos aleatorios  $\xi_1$  y  $\xi_2$  como:

$$[\xi_1 \cup \xi_2](s,t) = A_s \cup B_t$$

$$[\xi_1 \cap \xi_2](s,t) = A_s \cap B_t$$

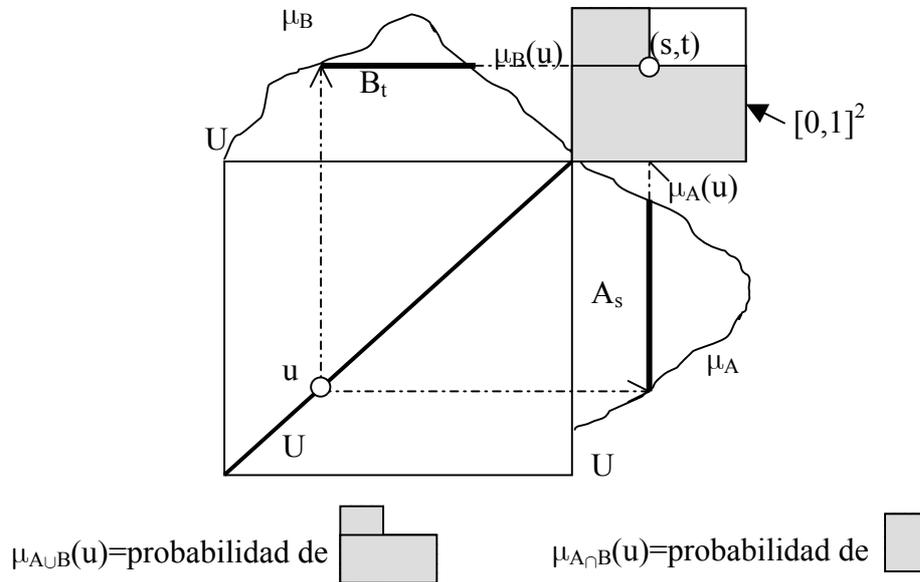


Figura A.3: Cálculo de la unión e intersección de conjuntos borrosos mediante la teoría de las sombras

Además, la unión y la intersección de dos subconjuntos borrosos  $A$  y  $B$  se puede definir como las sombras de los conjuntos aleatorios  $\xi_1 \cup \xi_2$  y  $\xi_1 \cap \xi_2$ :

$$\mu_{A \cup B}(u) = p\{(s,t) | u \in \xi_1 \cup \xi_2(s,t)\} = p\{(s,t) | u \in A_s \cup B_t\}$$

$$\mu_{A \cap B}(u) = p\{(s,t) | u \in \xi_1 \cap \xi_2(s,t)\} = p\{(s,t) | u \in A_s \cap B_t\}$$

O lo que es lo mismo:

$$\mu_{A \cup B}(u) = p \{ [0, \mu_A(u)] \times [0, 1] \cup [0, 1] \times [0, \mu_B(u)] \}$$

$$\mu_{A \cap B}(u) = p \{ [0, \mu_A(u)] \times [0, 1] \cap [0, 1] \times [0, \mu_B(u)] \}$$

Un método para calcular  $\mu_{A \cup B}(u)$  y  $\mu_{A \cap B}(u)$ , se muestra en la Figura A.3 y dependen de la distribución de probabilidad conjunta en  $[0, 1]^2$ .

Existen infinitas distribuciones conjuntas de probabilidad en  $[0, 1]^2$  cuyas distribuciones marginales son uniformes. Las más típicas son (ver Figura A.4):

- La correlación positiva perfecta, que se corresponde a una distribución conjunta distribuida uniformemente en la diagonal principal de  $[0, 1]^2$ .
- La correlación negativa perfecta, que se corresponde a una distribución conjunta distribuida uniformemente en la anti-diagonal de  $[0, 1]^2$ .
- Independencia, que se corresponde a una distribución conjunta distribuida uniformemente en todo el cuadrado unidad  $[0, 1]^2$ .

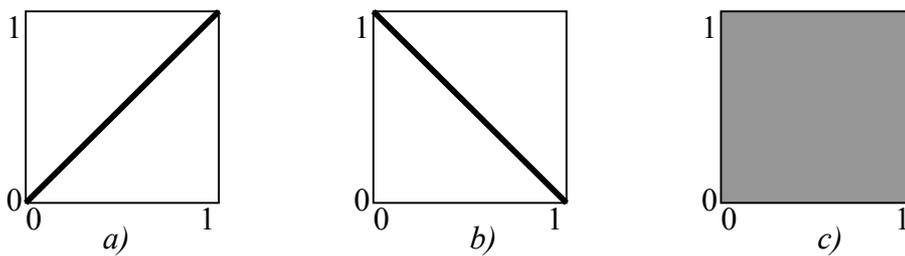


Figura A.4: a) Correlación positiva perfecta. b) Correlación negativa perfecta. c) Independencia

Otras distribuciones conjuntas con distribuciones marginales uniformes aparecen en [Wang et al., 1991].

Se puede calcular la unión y la intersección de dos subconjuntos borrosos A y B para las correlaciones citadas anteriormente.

#### Correlación positiva perfecta:

Como se puede ver en la Figura A.5, a), cuando la correlación es positiva perfecta se verifica:

$$\mu_{A \cup B}(u) = p \{ [0, \mu_A(u)] \times [0, 1] \cup [0, 1] \times [0, \mu_B(u)] \} = p(OI \cup OJ)$$

$$= \begin{cases} p(OI) = \mu_A(u) & \text{si } \mu_A(u) \geq \mu_B(u) \\ p(OJ) = \mu_B(u) & \text{si } \mu_B(u) \geq \mu_A(u) \end{cases}$$

$$= \max ( \mu_A(u), \mu_B(u) )$$

Por otro lado, para la intersección se tiene:

$$\mu_{A \cap B}(u) = p\{ [0, \mu_A(u)] \times [0, 1] \cap [0, 1] \times [0, \mu_B(u)] \} = p(OI \cap OJ)$$

$$= \begin{cases} p(OJ) = \mu_B(u) & \text{si } \mu_A(u) \geq \mu_B(u) \\ p(OI) = \mu_A(u) & \text{si } \mu_B(u) \geq \mu_A(u) \end{cases}$$

$$= \min ( \mu_A(u), \mu_B(u) )$$

#### Correlación negativa perfecta:

Como se puede ver en la Figura A.5, b), cuando la correlación es negativa perfecta la unión e intersección dependen de si el punto está por debajo o por encima de la antidiagonal:

$$\mu_{A \cup B}(u) = p\{ [0, \mu_A(u)] \times [0, 1] \cup [0, 1] \times [0, \mu_B(u)] \} = p(SI \cup RJ)$$

$$= \begin{cases} p(SI+RJ) = \mu_A(u) + \mu_B(u) & \text{si } \mu_A(u) + \mu_B(u) \leq 1 \\ p(SR) = 1 & \text{si } \mu_A(u) + \mu_B(u) \geq 1 \end{cases}$$

$$= \min ( \mu_A(u) + \mu_B(u), 1 )$$

Por otro lado, para la intersección se tiene:

$$\mu_{A \cap B}(u) = p\{ [0, \mu_A(u)] \times [0, 1] \cap [0, 1] \times [0, \mu_B(u)] \} = p(SI \cap RJ)$$

$$= \begin{cases} p(\emptyset) = 0 & \text{si } \mu_A(u) + \mu_B(u) \leq 1 \\ p(JI) = \mu_A(u) - (1 - \mu_B(u)) & \text{si } \mu_A(u) + \mu_B(u) \geq 1 \end{cases}$$

$$= \max ( \mu_A(u) + \mu_B(u) - 1, 0 )$$

Independencia:

Como se puede ver en la Figura A.5, c), en caso de independencia en la distribución conjunta de probabilidad, se tiene:

$$\mu_{A \cup B}(u) = p\{ [0, \mu_A(u)] \times [0, 1] \cup [0, 1] \times [0, \mu_B(u)] \} = p(\text{OSCD} \cup \text{OEFR}) = p(\text{OSCD}) + p(\text{OEFR}) - p(\text{OEGD}) = \mu_A(u) + \mu_B(u) - \mu_A(u) \cdot \mu_B(u)$$

Por otro lado, la función de pertenencia de la intersección de A y B se puede calcular como:

$$\mu_{A \cap B}(u) = p\{ [0, \mu_A(u)] \times [0, 1] \cap [0, 1] \times [0, \mu_B(u)] \} = p(\text{OSCD} \cap \text{OEFR}) = p(\text{OEGD}) = \mu_A(u) \cdot \mu_B(u)$$

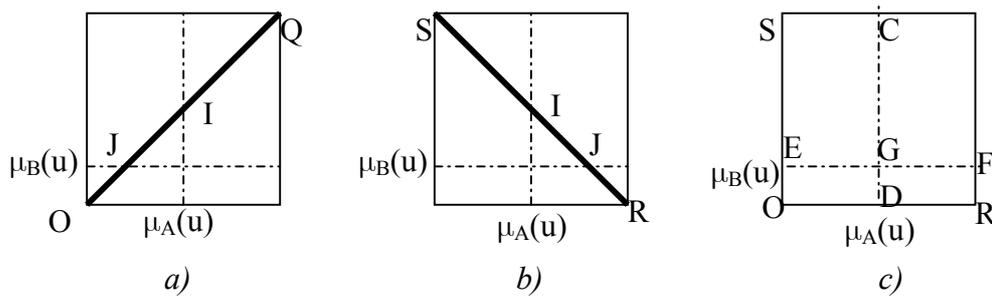


Figura A.5: Cálculo de  $A \cup B$  y  $A \cap B$  con correlación: a) positiva perfecta, b) negativa perfecta, c) independencia

Por tanto, mediante la representación gráfica de las tres principales distribuciones conjuntas en el cuadrado unidad, se prueba que :

- Cuando la correlación es positiva perfecta  $\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$  y  $\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$ , que son el resultado de las operaciones borrosas unión e intersección con la t-conorma máximo y t-norma mínimo.
- Cuando la correlación es negativa perfecta  $\mu_{A \cup B}(u) = \min\{\mu_A(u) + \mu_B(u), 1\}$  y  $\mu_{A \cap B}(u) = \max\{\mu_A(u) + \mu_B(u) - 1, 0\}$ , que son el resultado de las operaciones borrosas unión e intersección con la t-conorma y t-norma de Lukasiewicz.

- En el caso de independencia,  $\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u) \cdot \mu_B(u)$  y  $\mu_{A \cap B}(u) = \mu_A(u) \cdot \mu_B(u)$ , que son el resultado de las operaciones borrosas unión e intersección con la t-conorma suma y t-norma producto.

## A.2. Representación gráfica

En este apartado se propone una representación gráfica del cuerpo de evidencia asociado a la distribución de posibilidad de un conjunto borroso, así como la representación de un cuerpo de evidencia conjunto y los dos cuerpos de evidencia marginales. Se propone un método para calcular gráficamente la distribución de plausibilidad conjunta asociada a dicho cuerpo de evidencia conjunto y la distribución de plausibilidad de cualquier función de dos argumentos a partir del cuerpo de evidencia conjunto de esos dos argumentos.

Este método, a diferencia del propuesto en la teoría de las sombras se basa en la representación de los cuerpos de evidencia y no de conjuntos aleatorios, ya que el objetivo es la obtención de distribuciones de plausibilidad o de posibilidad y con ellas de medidas de plausibilidad y de posibilidad de funciones de dos argumentos, cuyos argumentos se conocen de forma incierta. Por tanto, resulta más apropiado partir de una representación de la información en forma de cuerpos de evidencia, aunque ambas teorías tienen puntos en común.

Además, la teoría de las sombras se centra únicamente en la obtención de la unión e intersección de conjuntos borrosos. En la representación que se propone a continuación se puede obtener la distribución de plausibilidad conjunta de un cuerpo de evidencia conjunto (y por tanto de sus conjuntos borrosos asociados marginales) y la distribución de plausibilidad de cualquier función de dos argumentos. Esto nos permitirá utilizar este tipo de representación para analizar la extensión de funciones a argumentos borrosos, a partir de sus cuerpos de evidencia y de la relación existente entre ellos.

Para lograr este objetivo, se ha optado, tal y como se describirá en los próximos apartados, por representar los límites inferiores de los  $\alpha$ -cortes en lugar del valor de  $\alpha$  (que es el elemento equivalente representado en la teoría de las sombras).

El método gráfico propuesto consiste en representar, para dos números borrosos triangulares  $A$  y  $B$ :

- La asignación básica conjunta en función de los límites inferiores  $l(A_j)$  y  $l(B_k)$  de los elementos focales (o  $\alpha$ -cortes)  $A_j$  y  $B_k$  en los ejes  $x$  e  $y$ , en lugar de los correspondientes  $\alpha_A^j$  y  $\alpha_B^k$ . Cada par  $(l(A_j), l(B_k))$  del gráfico representa un elemento focal conjunto  $C_i$ . El dominio de los límites inferiores de los  $\alpha$ -cortes de un número borroso  $A$  lo denominamos  $U_{l(A)}$ , por lo que esta representación se realiza en el dominio de  $U_{l(A)} \times U_{l(B)}$
- Los elementos focales conjuntos  $C_i = A_j \times B_k$  en el dominio de los soportes de  $A$  y  $B$ , esto es  $]A[ \times ]B[$ , donde  $]A[$  es el soporte de  $A$ .

### A.2.1. Distribución conjunta de posibilidad. Teoría de Posibilidad.

En este apartado se van a obtener aplicando la teoría de la posibilidad, las distribuciones conjuntas de posibilidad combinando la información borrosa de partida mediante las t-normas mínimo, producto y Lukasiewicz. Estas distribuciones de posibilidad se compararán posteriormente con las distribuciones de plausibilidad o posibilidad obtenidas combinando los cuerpos de evidencia dadas distintas correlaciones entre dichos cuerpos de evidencia.

Cuando las fuentes de información son completamente fiables la distribución conjunta de plausibilidad/posibilidad de  $A$  y  $B$  se obtiene por consenso conjuntivo [Dubois, Prade, 1991a], [Dubois, Prade, 1988b]. Las t-normas modelan ese tipo de consenso.

En teoría de la posibilidad, las distribuciones de posibilidad asociadas a dos conjuntos borrosos  $A$  ( $\pi_A(a): \mathbb{R} \rightarrow [0,1]$ ) y  $B$  ( $\pi_B(b): \mathbb{R} \rightarrow [0,1]$ ) puede ser combinada seleccionando la t-norma apropiada para agregarlas conjuntamente. La distribución de posibilidad conjunta  $\pi_{A \times B}$  de ( $A$  “and”  $B$ ) viene dada por:

$$\pi_{A \times B}((a_1, b_1)) = T(\pi_A(a_1), \pi_B(b_1))$$

donde  $a_i \in \mathbb{R}$ ,  $b_i \in \mathbb{R}$ , y  $T$  es una t-norma.

Por tanto, la distribución de posibilidad conjunta  $\pi_{A \times B}$  de ( $A$  “and”  $B$ ) con la t-norma *min* es:

$$(A.1) \quad \pi_{A \times B}((a_1, b_1)) = \min(\pi_A(a_1), \pi_B(b_1)) \quad \forall (a_1, b_1) \in \mathbb{R} \times \mathbb{R}$$

La distribución de posibilidad conjunta de ( $A$  “and”  $B$ ) con la t-norma producto es:

$$(A.2) \pi_{A \times B}((a_1, b_1)) = \pi_A(a_1) \cdot \pi_B(b_1) \quad \forall (a_1, b_1) \in \mathbb{R} \times \mathbb{R}$$

La distribución de posibilidad conjunta  $\pi_{A \times B}$  de ( $A$  “and”  $B$ ) con la t-norma de Lukasiewicz se puede expresar como:

$$(A.3) \pi_{A \times B}((a_1, b_1)) = \max(0, \pi_A(a_1) + \pi_B(b_1) - 1) \quad \forall (a_1, b_1) \in \mathbb{R} \times \mathbb{R}$$

A continuación se va a mostrar un ejemplo de distribución de posibilidad conjunta obtenida mediante las tres t-normas citadas. Sean dos números borrosos triangulares definidos por sus vértices  $A=(1,2,3)$  y  $B=(1,3,5)$ . La Figura A.6 muestra las distribuciones de posibilidad de  $A$  y de  $B$  y la distribución conjunta de posibilidad  $\pi_{A \times B}$  para las t-normas: *min*, producto y Lukasiewicz aplicando (A.1), (A.2) y (A.3) respectivamente.

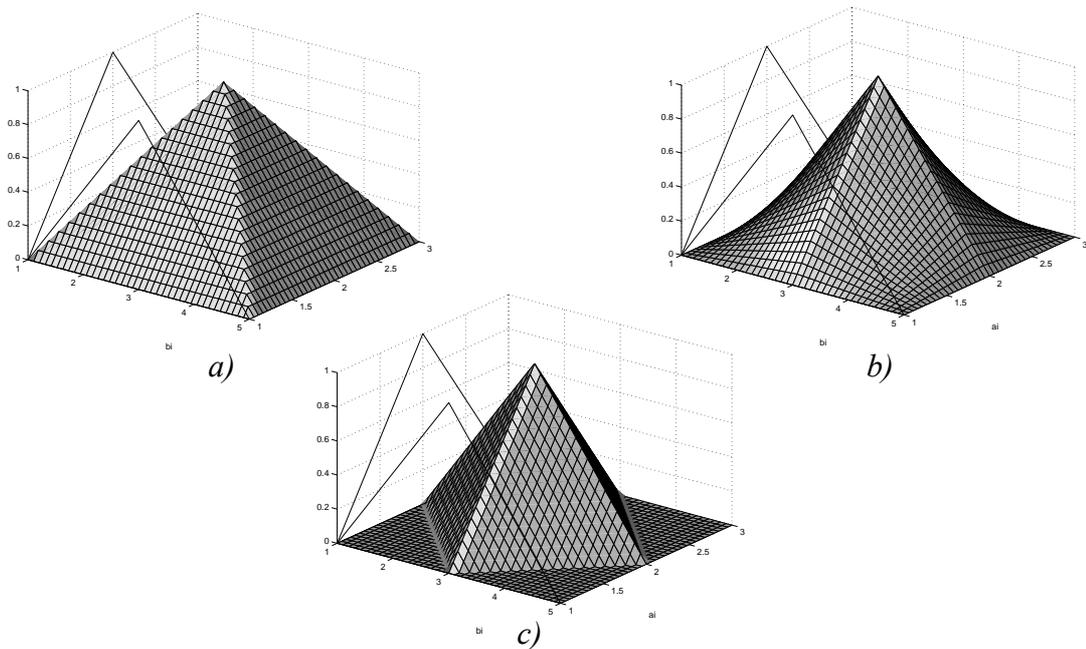


Figura A.6: Distribución conjunta de posibilidad con tres t-normas:  
 a) mínimo, b) producto, c) Lukasiewicz

Tal y como se ve en la Figura A.6, la t-norma *min* proporciona el resultado menos específico [Klir, Yuan, 1995].

Si  $A$  y  $B$  son conjuntos borrosos que modelan la misma variable  $x$ , entonces  $a_i$  y  $b_i$  son iguales y  $a_i = b_i = x$ . Se puede obtener una distribución de posibilidad de la misma, cortando las superficies mostradas anteriormente con el plano vertical que pasa por  $a_i = b_i$ , como se ve en la Figura A.7. Es decir, se obtiene la intersección  $A \cap B$  de los dos conjuntos borrosos.

El resultado es el mismo que si se calcula la intersección como  $T(\pi_A(x), \pi_B(x)) \quad \forall x \in \mathbb{R}$ , tal y como se ve en la Figura A.8.

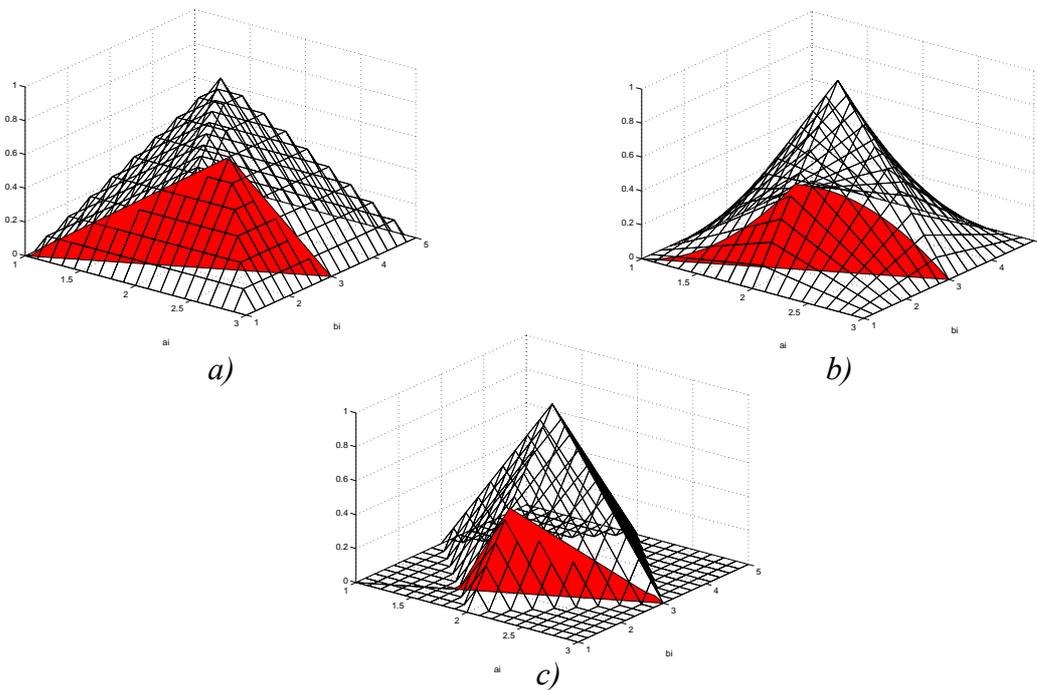


Figura A.7: Distribución de posibilidad de  $A$  and  $B$  cuando  $a_i = b_i$ , con tres  $t$ -normas:  
a) mínimo, b) producto, c) Lukasiewicz

En teoría de la evidencia, la información disponible se expresa mediante la asignación básica conjunta. Se van a estudiar a continuación tres asignaciones básicas conjuntas y a comparar las distribuciones de plausibilidad resultantes con las tres distribuciones de posibilidad obtenidas previamente en teoría de la posibilidad con las tres  $t$ -normas. En el siguiente apartado se describe el método gráfico propuesto para obtener la distribución de plausibilidad conjunta de  $A$  y  $B$  a partir de los cuerpos de evidencia.

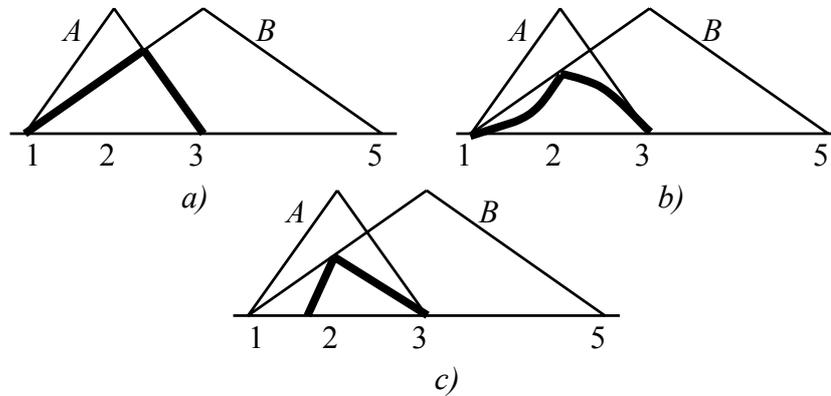


Figura A.8: Distribución de posibilidad de A and B cuando  $a_i=b_i$ , con tres t-normas: a) mínimo, b) producto, c) Lukasiewicz

### A.2.2. Distribución conjunta de plausibilidad. Teoría de la Evidencia

Supongamos que se conoce la asignación básica conjunta  $m_C(C_i)$  definida en el producto cartesiano  $X \times Y$ ,  $m_C(C_i) = m_{A \times B}(A_j \times B_k)$  tal que  $m_{A \times B}: \mathcal{P}(X \times Y) \rightarrow [0,1]$  y  $(F_A, m_A)$  y  $(F_B, m_B)$  son las proyecciones del cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$  en  $X$  e  $Y$  respectivamente, también denominados cuerpos de evidencia marginales [Klir, Folger, 1988].

El valor de plausibilidad conjunta de cualquier par  $(a,b)$  se puede obtener aplicando la definición de medida de plausibilidad  $Pl(B) = \sum_{A_i \cap B \neq \emptyset} m(A_i)$  [Wang et al., 1991]

como:

$$Pl(\{(a,b)\}) = \sum_{(a,b) \in C_i} m_{A \times B}(C_i)$$

donde los elementos focales conjuntos  $C_i$  se obtienen como el producto cartesiano de los elementos focales de los cuerpos de evidencia marginales, esto es,  $C_i = A_j \times B_k$ .

Si el cuerpo de evidencia conjunto es consonante, esta medida de plausibilidad se denomina medida de posibilidad. Suponiendo que los cuerpos de evidencia marginales son consonantes, los elementos focales  $A_j$  y  $B_k$  asociados a  $C_i$  tal que  $C_i =$

$A_j \times B_k$  son los  $\alpha$ -cortes de los conjuntos borrosos  $A$  y  $B$  correspondientes a  $\alpha_A^j$  y  $\alpha_B^k$ . Por tanto, cada elemento focal conjunto  $C_i$  se puede caracterizar mediante una pareja  $(\alpha_A^j, \alpha_B^k)$  y la asignación básica conjunta se puede expresar en términos de correlación existente entre  $\alpha_A$  y  $\alpha_B$ , con lo que  $m_{A \times B}(\alpha_A, \alpha_B)$  está definida en  $[0, 1]^2$ . Se puede interpretar como la probabilidad  $P(A_j \times B_k)$  del elemento focal conjunto o la probabilidad del par  $(\alpha_A^j, \alpha_B^k)$ .

A continuación, por sencillez, se consideran únicamente números borrosos triangulares. Los gráficos que se obtienen son fáciles de interpretar, aunque las conclusiones son aplicables a cualquier tipo de número borroso. No se supone la simetría de los números borrosos.

El método gráfico propuesto consiste en representar los límites inferiores  $l(A_j)$  y  $l(B_k)$  de los elementos focales  $A_j$  y  $B_k$  en los ejes  $x$  e  $y$ , en lugar de los correspondientes  $\alpha_A^j$  y  $\alpha_B^k$  (que son los que se representan en la teoría de las sombras [Wang et al., 1991]). Cada par  $(l(A_j), l(B_k))$  del gráfico representa un elemento focal conjunto  $C_i$ . El dominio de los límites inferiores de los  $\alpha$ -cortes de un número borroso  $A$  lo denominamos  $U_{l(A)}$ .

Dado un cuerpo de evidencia, se puede calcular la distribución de plausibilidad y esa distribución se puede interpretar como la función de pertenencia de un conjunto borroso:  $\mu(x) = Pl(\{x\}) = \sum_{A_i \cap \{x\} \neq \emptyset} m(A_i) = \sum_{x \in A_i} m(A_i)$ . Si la asignación básica es continua, entonces se puede expresar como:

$$\mu(x) = Pl(\{x\}) = \int_{x \in A_i} m(A_i) \cdot d\delta_{A_i}$$

donde  $\delta_{A_i}$  es una distancia entre intervalos compactos de la recta real. Esa distancia se define como:

$$\delta(A, B) = \sup \{ ds(a_{inf}, b_{inf}), ds(a_{sup}, b_{sup}) \}$$

donde  $a_{inf}$  es un número real que se define como el límite inferior del intervalo  $a_{inf} = \inf_{x \in A} x$  y  $a_{sup}$  es un número real que se define como el límite superior del intervalo  $a_{sup} = \sup_{x \in A} x$ . Así,  $ds(a_{inf}, b_{inf})$  es la distancia euclídea entre dos números  $a_{inf}$  y  $b_{inf}$ .

La medida  $\delta(A, B)$  verifica las propiedades de una distancia:

1.  $\delta(A, B) = 0$  si, y sólo si,  $A = B$

$$2. \delta(A, B) + \delta(A, C) \geq \delta(B, C)$$

1. Si  $\delta(A, B) = 0$ , se verifica que  $ds(a_{inf}, b_{inf}) = 0$  y  $ds(a_{sup}, b_{sup}) = 0$  y como los intervalos son compactos y los límites coinciden, entonces  $A = B$ .
2. 
$$\begin{aligned} \delta(A, B) + \delta(A, C) &= \sup\{ds(a_{inf}, b_{inf}), ds(a_{sup}, b_{sup})\} + \sup\{ds(a_{inf}, c_{inf}), ds(a_{sup}, c_{sup})\} \\ &= \sup\{ ds(a_{inf}, b_{inf}) + ds(a_{inf}, c_{inf}), ds(a_{inf}, b_{inf}) + ds(a_{sup}, c_{sup}), ds(a_{sup}, b_{sup}) + ds(a_{inf}, c_{inf}), \\ &\quad ds(a_{sup}, b_{sup}) + ds(a_{sup}, c_{sup}) \} \\ &\geq \sup\{ds(b_{inf}, c_{inf}), ds(a_{inf}, b_{inf}) + ds(a_{sup}, c_{sup}), ds(a_{sup}, b_{sup}) + ds(a_{inf}, c_{inf}), ds(b_{sup}, c_{sup})\} \\ &\geq \sup\{ds(b_{inf}, c_{inf}), ds(b_{sup}, c_{sup})\} = \delta(B, C) \end{aligned}$$

Si el cuerpo de evidencia es consonante, suponiendo que se ordenan los elementos focales de forma que  $A_1 \subset A_2 \subset \dots \subset A_n$ , entonces la función de pertenencia se puede calcular como:

$$\mu(x) = \pi(x) = \int_{A_n}^{A_j} m(A_i) \cdot d\delta_{A_i}$$

siendo los elementos focales que van de  $A_n$  a  $A_j$  todos aquellos que contienen a  $x$ .

Si se escoge como representante de cada elemento focal  $A_i$  su límite inferior  $l(A_i)$ , se puede expresar la asignación básica en función de dichos representantes  $m(l(A_i))$  y calcular así la función de pertenencia:

$$(A.4) \mu(x) = \pi(x) = \int_{A_n}^{A_j} m(A_i) \cdot d(A_i) = \int_{l(A_n)}^{l(A_j)} m(l(A_i)) \cdot d(l(A_i))$$

Por tanto, dada una función de pertenencia de un conjunto borroso, o una distribución de posibilidad, se puede calcular la asignación básica del cuerpo de evidencia consonante asociado derivando la distribución de posibilidad como:

$$m(l(A_i)) = \frac{d(\pi(x))}{d(l(A_i))}$$

y los elementos focales de dicho cuerpo de evidencia son los  $\alpha$ -cortes del conjunto borroso de forma que  $A_1 \subset A_2 \subset \dots \subset A_n$  y  $A_1$  es el  $\alpha$ -corte con  $\alpha = 1$ . Cuando  $A$  es un número borroso triangular  $l(A_j)$  está uniformemente distribuido en  $U_{l(A_i)}$ . Es decir, la asignación básica de los elementos focales representada en función de  $l(A_j)$  es una distribución uniforme, tal y como se ve en la Figura A.9, ya que la derivada de la

distribución de posibilidad en el dominio de los límites inferiores es una constante. El área completa bajo  $m_A(l(A_i))$  es 1.

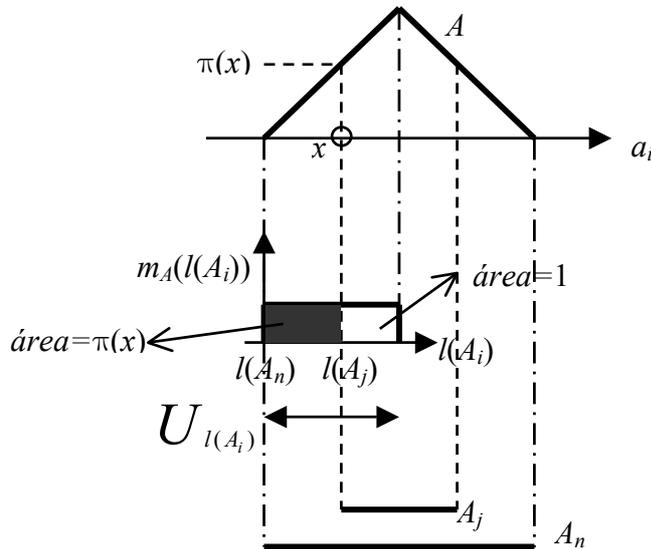


Figura A.9: Número borroso triangular y asignación básica del cuerpo de evidencia consonante asociado

Así, en la Figura A.9, la posibilidad de  $x$  es el área marcada entre  $l(A_n)$  y  $l(A_j)$ , de acuerdo con (A.4), ya que los elementos focales del cuerpo de evidencia consonante asociado (o  $\alpha$ -cortes del conjunto borroso) que van desde  $A_n$  hasta  $A_j$  contienen a  $x$ . De la misma forma, se podrían haber escogido como representantes de los elementos focales los límites superiores de dichos intervalos y la representación sería similar.

A continuación se muestran tres asignaciones básicas conjuntas según la relación existente entre  $\alpha_A$  y  $\alpha_B$ : correlación positiva perfecta, independencia y correlación negativa perfecta. Los resultados de distribución de plausibilidad se comparan con los resultados obtenidos por teoría de la posibilidad con las t-normas: mínimo, producto y Lukasiewicz.

**Correlación positiva perfecta**

Cuando existe correlación positiva perfecta entre  $\alpha_A$  y  $\alpha_B$ , la asignación básica conjunta  $m_{A \times B}(\alpha_A, \alpha_B)$  se distribuye uniformemente en la diagonal principal  $\alpha_A = \alpha_B$  del cuadrado unidad  $[0,1]^2$  [Wang et al., 1991], [Tan et al., 1993]. Ver Figura A.10.

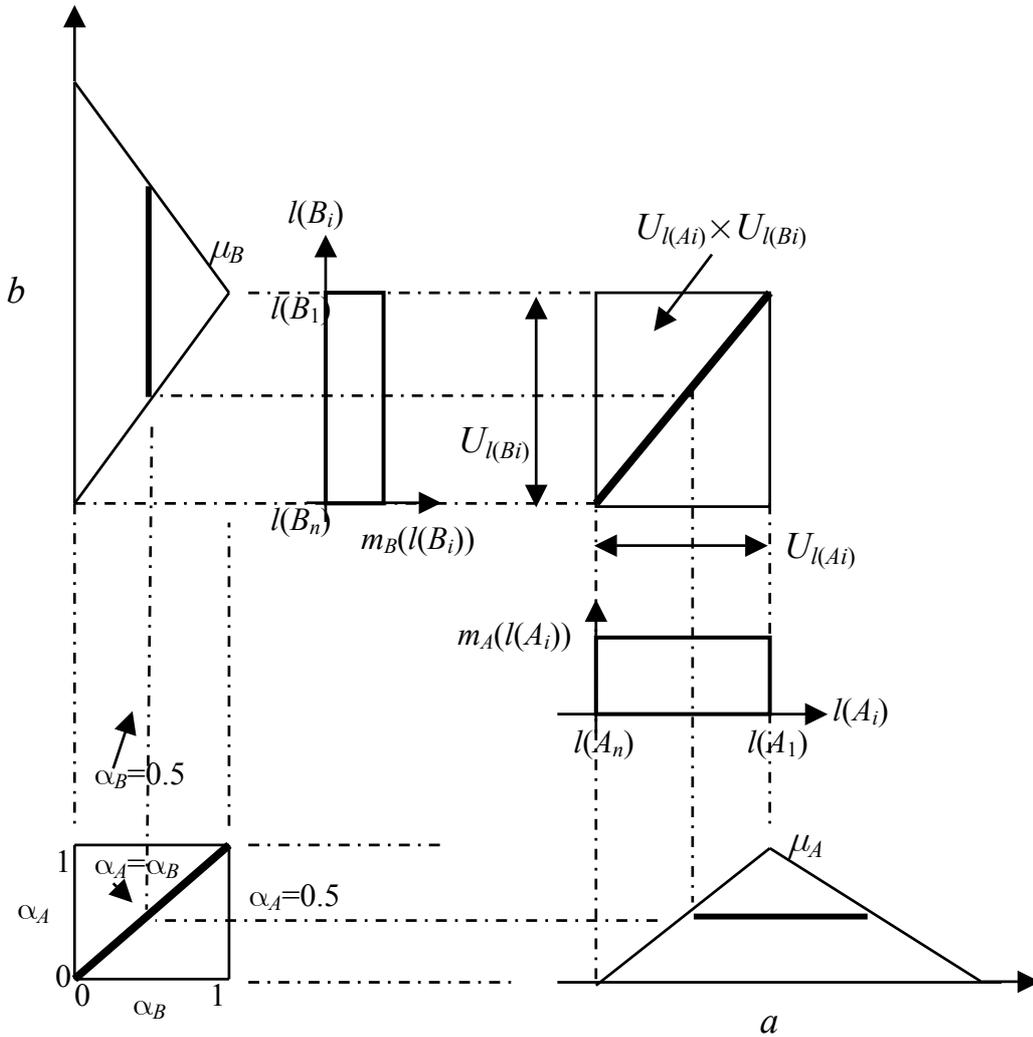


Figura A.10: Correlación positiva perfecta

Si los cuerpos de evidencia marginales son consonantes y las distribuciones de posibilidad asociadas a dichos cuerpos de evidencia son triangulares, tenemos que las asignaciones básicas  $m_A(l(A_i))$  y  $m_B(l(B_i))$  son distribuciones uniformes en  $l(A_i)$  y  $l(B_i)$  respectivamente (tal y como se vio en la Figura A.9). Además, se tiene que la

diagonal principal  $\alpha_A = \alpha_B$  del cuadrado unidad  $[0,1]^2$  se corresponde con la diagonal principal de  $U_{l(A_i)} \times U_{l(B_i)}$ :

Para números borrosos triangulares, se puede expresar  $\alpha$  en función de los límites inferiores de los  $\alpha$ -cortes como:

$$\alpha_A = (a - l(A_n)) / (l(A_1) - l(A_n)) \quad \text{y} \quad \alpha_B = (b - l(B_n)) / (l(B_1) - l(B_n))$$

Con lo que se tiene que la recta  $\alpha_A = \alpha_B$  es:

$$b = ( (a - l(A_n)) * (l(B_1) - l(B_n)) / (l(A_1) - l(A_n)) ) + l(B_n) \quad \text{que es la diagonal principal de } U_{l(A_i)} \times U_{l(B_i)}. \text{ Ver Figura A.10.}$$

Y la distribución en la diagonal principal de  $U_{l(A_i)} \times U_{l(B_i)}$  cuyas proyecciones son uniformes es a su vez una distribución uniforme. Ver Figura A.12.

Por ejemplo, sean dos números borrosos triangulares definidos por sus vértices  $A=(1,2,3)$  y  $B=(1,3,5)$ . En la Figura A.11 se muestran las asignaciones básicas de los cuerpos de evidencia consonantes asociados con  $A$  y  $B$ , en función de los límites inferiores de los elementos focales. Entre ellos existe correlación positiva perfecta entre  $\alpha_A$  y  $\alpha_B$ , esto es  $\alpha_A = \alpha_B$ . La asignación básica conjunta se puede obtener mediante el método de Montecarlo generando un número aleatorio uniformemente distribuido entre 0 y 1. Para el valor de  $\alpha \in (0,1]$ , se obtiene el elemento focal de  $A$  correspondiente y para el mismo valor de  $\alpha$  se obtiene el elemento focal de  $B$  correspondiente.

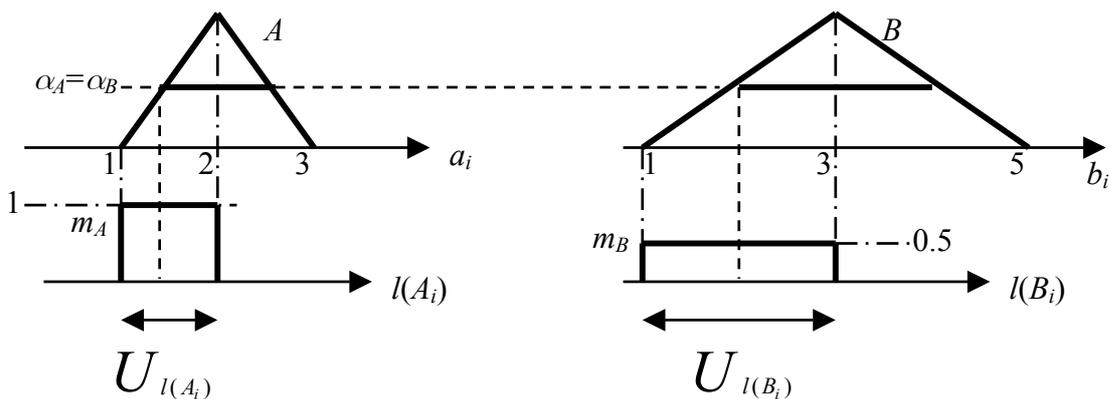


Figura A.11: Asignaciones básicas de los cuerpos de evidencia consonantes asociados con  $A$  y  $B$

La asignación básica conjunta de este ejemplo con correlación positiva perfecta está distribuida uniformemente en la diagonal principal del dominio de los límites inferiores  $U_{I(A_i)} \times U_{I(B_i)}$ , tal y como aparece en la Figura A.12. En dicha figura, cada punto de la diagonal principal es el representante de un elemento focal conjunto, concretamente el límite inferior izquierdo.

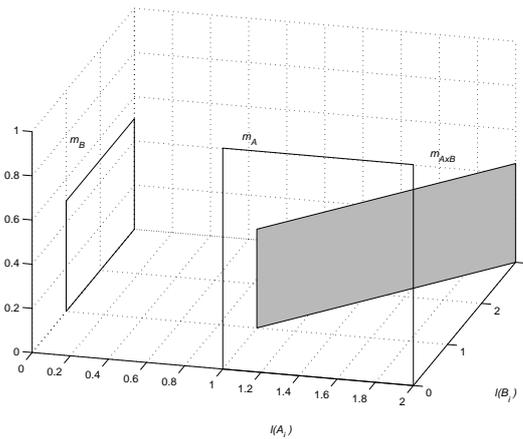


Figura A.12: Asignación básica conjunta cuando  $\alpha_A = \alpha_B$

La Figura A.13 muestra en este caso los elementos focales conjuntos completos, cuyo límite inferior izquierdo se desplaza a lo largo de la diagonal. Los elementos focales conjuntos son productos Cartesianos  $C_i = A_j \times B_k$  y en la Figura A.13 se dibujan en el dominio de los soportes  $]A[ \times ]B[$ , donde  $]A[$  es el soporte de  $A$ . El límite inferior izquierdo del elemento focal conjunto es el representante de dicho elemento focal en el gráfico de límites inferiores de la Figura A.12. Se observa en la Figura A.13 que los elementos focales conjuntos están anidados y, por tanto, las medidas de plausibilidad son medidas de posibilidad. Para variables discretas, la posibilidad de  $(a_1, b_1)$  se calcula sumando la asignación básica de cada elemento focal conjunto que contenga el par  $(a_1, b_1)$ . En este ejemplo  $(a_1, b_1)$  vale  $(1.8, 2.6)$ . Como los elementos focales conjuntos están anidados, se pueden numerar de forma que  $C_1 \subset C_2 \subset \dots \subset C_n$ . Se puede ver en la Figura A.13 que todos los elementos focales conjuntos desde  $C_i$  hasta  $C_n$  contienen el par  $(a_1, b_1)$ , por tanto la posibilidad de  $(a_1, b_1)$  expresado en continuo es:

$$\pi((a_1, b_1)) = \int_{(a_1, b_1) \in C_i} m_{A \times B}(C_i) \cdot d\delta_{C_i}$$

donde  $\delta_{C_i}$  es la distancia entre los elementos focales definidos anteriormente. La distancia  $\delta(C_j, C_s)$  siendo  $C_j$  y  $C_s$  dos elementos focales conjuntos rectangulares se define como:

$$\delta(C_j, C_s) = \sup \left\{ d(C_{sup}^j, C_{sup}^s), d(C_{inf}^j, C_{inf}^s), d(C_{der}^j, C_{der}^s), d(C_{izq}^j, C_{izq}^s) \right\}$$

donde  $d(I_1, I_2)$  es la distancia entre dos intervalos definida posteriormente y  $C_{izq}^j$  es un intervalo correspondiente al lado izquierdo del elemento focal rectangular  $C_j$ ,  $C_{der}^j$  es un intervalo correspondiente al lado derecho de  $C_j$ ,  $C_{inf}^j$  es un intervalo correspondiente al lado inferior de  $C_j$ , y  $C_{sup}^j$  es un intervalo correspondiente al lado superior de  $C_j$ , esto es:

$$C_{izq}^j = \{(x, y) \mid x = \inf(x_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{der}^j = \{(x, y) \mid x = \sup(x_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{inf}^j = \{(x, y) \mid y = \inf(y_0)\}_{(x_0, y_0) \in C_j}$$

$$C_{sup}^j = \{(x, y) \mid y = \sup(y_0)\}_{(x_0, y_0) \in C_j}$$

Además se define la distancia entre dos intervalos como:

$$d(I_1, I_2) = \sup \left\{ \inf_{\substack{a \in I_1 \\ b \in I_2}} ds(a, b) \right\}$$

donde  $ds(a, b)$  es la distancia euclídea entre dos números reales.

$\delta_{C_i}$  cumple las propiedades de una distancia:

1.  $\delta(C_j, C_s) = 0$  si, y sólo si,  $C_j = C_s$
2.  $\delta(C_j, C_s) + \delta(C_j, C_t) \geq \delta(C_s, C_t)$

1. Si la distancia entre dos elementos focales es nula  $\delta(C_j, C_s)=0$ , se verifica que  $d(C_{sup}^j, C_{sup}^s)=0$ ,  $d(C_{inf}^j, C_{inf}^s)=0$ ,  $d(C_{der}^j, C_{der}^s)=0$ ,  $d(C_{izq}^j, C_{izq}^s)=0$  por lo que todos los lados de los elementos focales conjuntos  $C_j$  y  $C_s$  coinciden y con ello se tiene  $C_j=C_s$ .

$$\begin{aligned}
 & 2. \delta(C_j, C_s) + \delta(C_j, C_t) = \sup \{ d(C_{sup}^j, C_{sup}^s), d(C_{inf}^j, C_{inf}^s), d(C_{der}^j, C_{der}^s), d(C_{izq}^j, C_{izq}^s) \} + \\
 & \quad \sup \{ d(C_{sup}^j, C_{sup}^t), d(C_{inf}^j, C_{inf}^t), d(C_{der}^j, C_{der}^t), d(C_{izq}^j, C_{izq}^t) \} \\
 & = \sup \{ d(C_{sup}^j, C_{sup}^s) + d(C_{sup}^j, C_{sup}^t), d(C_{sup}^j, C_{sup}^s) + d(C_{inf}^j, C_{inf}^t), \\
 & \quad d(C_{sup}^j, C_{sup}^s) + d(C_{der}^j, C_{der}^t), d(C_{sup}^j, C_{sup}^s) + d(C_{izq}^j, C_{izq}^t), \\
 & \quad d(C_{inf}^j, C_{inf}^s) + d(C_{sup}^j, C_{sup}^t), d(C_{inf}^j, C_{inf}^s) + d(C_{inf}^j, C_{inf}^t), \\
 & \quad d(C_{inf}^j, C_{inf}^s) + d(C_{der}^j, C_{der}^t), d(C_{inf}^j, C_{inf}^s) + d(C_{izq}^j, C_{izq}^t), \\
 & \quad d(C_{der}^j, C_{der}^s) + d(C_{sup}^j, C_{sup}^t), d(C_{der}^j, C_{der}^s) + d(C_{inf}^j, C_{inf}^t), \\
 & \quad d(C_{der}^j, C_{der}^s) + d(C_{der}^j, C_{der}^t), d(C_{der}^j, C_{der}^s) + d(C_{izq}^j, C_{izq}^t), \\
 & \quad d(C_{izq}^j, C_{izq}^s) + d(C_{sup}^j, C_{sup}^t), d(C_{izq}^j, C_{izq}^s) + d(C_{inf}^j, C_{inf}^t), \\
 & \quad d(C_{izq}^j, C_{izq}^s) + d(C_{der}^j, C_{der}^t), d(C_{izq}^j, C_{izq}^s) + d(C_{izq}^j, C_{izq}^t) \} \\
 & \geq \sup \{ d(C_{sup}^j, C_{sup}^s) + d(C_{sup}^j, C_{sup}^t), d(C_{inf}^j, C_{inf}^s) + d(C_{inf}^j, C_{inf}^t), \\
 & \quad d(C_{der}^j, C_{der}^s) + d(C_{der}^j, C_{der}^t), d(C_{izq}^j, C_{izq}^s) + d(C_{izq}^j, C_{izq}^t) \} \\
 & \geq \sup \{ d(C_{sup}^s, C_{sup}^t), d(C_{inf}^s, C_{inf}^t), d(C_{der}^s, C_{der}^t), d(C_{izq}^s, C_{izq}^t) \} = \delta(C_s, C_t)
 \end{aligned}$$

Esto se verifica porque  $d(C_{sup}^j, C_{sup}^s) + d(C_{sup}^j, C_{sup}^t) \geq d(C_{sup}^s, C_{sup}^t)$ , y de forma equivalente para los lados inferior, izquierdo y derecho de los elementos focales conjuntos, ya que  $d(I_1, I_2)$  es una medida de distancia entre dos intervalos [del Castillo, 1980].

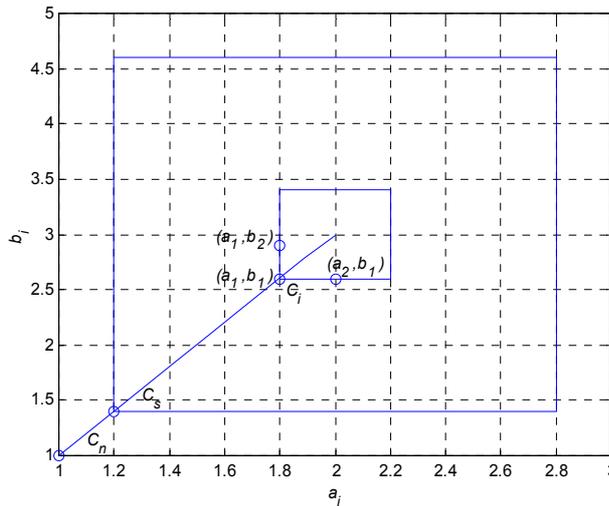


Figura A.13: Elementos focales del cuerpo de evidencia conjunto cuando  $\alpha_A = \alpha_B$

Tanto la asignación básica conjunta  $m_{A \times B}$  como la posibilidad de  $(a_1, b_1)$   $\pi((a_1, b_1))$  aparecen en la Figura A.14. En continuo, se puede calcular  $\pi((a_1, b_1))$  como el área sombreada, donde la posición del punto  $(a_1, b_1)$  y del par  $(l(A_r), l(B_s))$  que representa el elemento focal conjunto  $C_i$  coincide, ambos están en  $(1.8, 2.6)$ .

Como dicho punto pertenece a la recta definida por  $\alpha_A = \alpha_B$ , se cumple que  $\pi((a_1, b_1)) = \pi_A(a_1) = \pi_B(b_1)$  (que en este ejemplo vale 0.8).

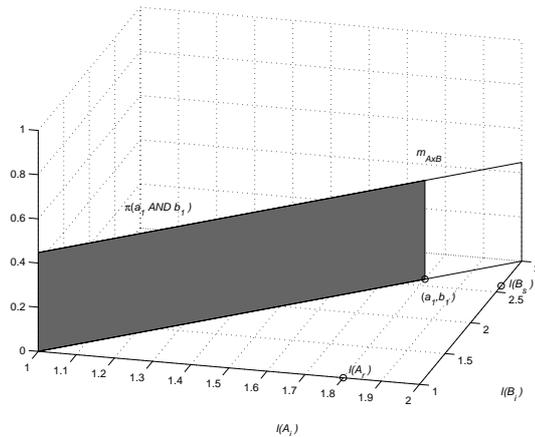


Figura A.14: Posibilidad de  $(a_1, b_1)$  cuando  $\alpha_A = \alpha_B$

Para un punto  $(a_2, b_1)$  del cuadrante inferior izquierdo situado por debajo de la diagonal correspondiente a  $\alpha_A = \alpha_B$ , se tiene que  $\alpha_A > \alpha_B$ , o lo que es lo mismo,  $\pi_A(a_2) > \pi_B(b_1)$ . Los elementos focales conjuntos que contienen al par  $(a_2, b_1)$  son los mismos que contienen a  $(a_1, b_1)$ , con lo que,

$$\text{Para } \pi_A(a_2) > \pi_B(b_1), \text{ se tiene que } \pi((a_2, b_1)) = \pi((a_1, b_1)) = \pi_B(b_1)$$

Por otro lado, para un punto  $(a_1, b_2)$  del cuadrante inferior izquierdo situado por encima de la diagonal correspondiente a  $\alpha_A = \alpha_B$ , se tiene que  $\alpha_B > \alpha_A$ , o lo que es lo mismo,  $\pi_A(a_1) < \pi_B(b_2)$ . Los elementos focales conjuntos que contienen al par  $(a_1, b_2)$  son los mismos que contienen a  $(a_1, b_1)$ , con lo que,

$$\text{Para } \pi_A(a_1) < \pi_B(b_2), \text{ se tiene que } \pi((a_1, b_2)) = \pi((a_1, b_1)) = \pi_A(a_1)$$

Por tanto, se concluye que:

$$\pi((a_i, b_j)) = \min(\pi_A(a_i), \pi_B(b_j))$$

En el ejemplo de la Figura A.13, se tiene:

$$\pi((a_2, b_1)) = \min(\pi_A(2), \pi_B(2.6)) = \min(1, 0.8) = 0.8$$

$$\pi((a_1, b_2)) = \min(\pi_A(1.8), \pi_B(2.9)) = \min(0.8, 0.95) = 0.8$$

Mediante esta representación hemos llegado al resultado ya conocido [Wang et al., 1991] de que las distribuciones conjuntas de posibilidad obtenidas por teoría de la evidencia con  $\alpha_A = \alpha_B$  por un lado, y por teoría de la posibilidad con la t-norma *min* por otro lado, coinciden.

### Relación de independencia

Cuando  $\alpha_A$  y  $\alpha_B$  son independientes [Wang et al., 1991], [Tan et al., 1993], están uniformemente distribuidas en el cuadrado unidad  $[0,1]^2$ . Pasando al dominio de los límites inferiores, es equivalente a decir que  $(l(A_j), l(B_k))$  está uniformemente distribuida en todo el dominio  $U_{l(A_i)} \times U_{l(B_i)}$

Esta asignación básica conjunta se puede obtener, mediante el método de Montecarlo, generando dos números aleatorios uniformemente distribuidos entre 0 y 1 e independientes entre sí. Para el primer valor de  $\alpha \in (0,1]$ , se obtiene el elemento focal de  $A$  correspondiente y para el segundo valor de  $\alpha$  se obtiene el elemento focal de  $B$  correspondiente. En la Figura A.15 se muestra la asignación básica conjunta de los cuerpos de evidencia de  $A$  y de  $B$  que aparecen en la Figura A.11, cuando  $\alpha_A$  y  $\alpha_B$  son independientes.

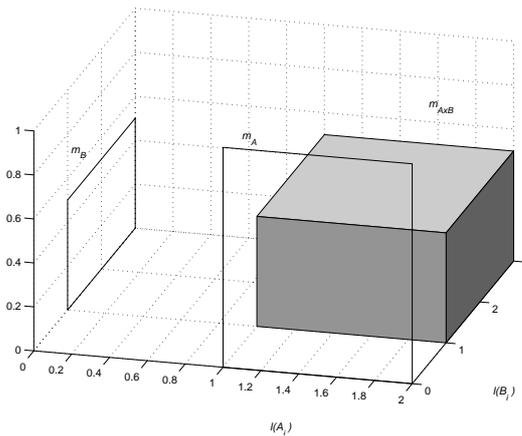


Figura A.15: Asignación básica conjunta cuando  $\alpha_A$  y  $\alpha_B$  son independientes

La asignación básica conjunta coincide con el resultado de combinar los cuerpos de evidencia de  $A$  y de  $B$  o cuerpos de evidencia marginales  $(F_A, m_A)$  y  $(F_B, m_B)$  mediante la regla de Dempster [Klir, Folger, 1988]. En este caso se dice que los cuerpos de evidencia marginales son no interactivos, ya que los elementos focales conjuntos son productos cartesianos de los elementos focales de cuerpos de evidencia marginales  $A_i \times B_i$  y la asignación básica conjunta se calcula como el producto de las asignaciones marginales.  $m_{A \times B}(A_i \times B_i) = m_A(A_i) \cdot m_B(B_i)$

Este concepto de no interacción se basa en la regla de combinación de Dempster:

$$m_{1,2}(D) = \frac{\sum_{E \cap F = D} m_1(E) \cdot m_2(F)}{1 - K}$$

para  $D \neq \emptyset$ , donde  $K = \sum_{E \cap F = \emptyset} m_1(E) \cdot m_2(F)$ , con  $m_{1,2}(\emptyset) = 0$

En la Figura A.16 se muestran cuatro elementos focales  $C_i, C_j, C_k, C_s$  del cuerpo de evidencia conjunto. Como se puede ver, no están anidados por lo que las medidas de plausibilidad asociadas a este cuerpo de evidencia no son medidas de posibilidad. Los límites inferiores izquierdos de los elementos focales conjuntos que en la Figura A.16 se marcan con un punto, son los representantes de los elementos focales,  $(I(A_i), I(B_j))$  en la Figura A.15.

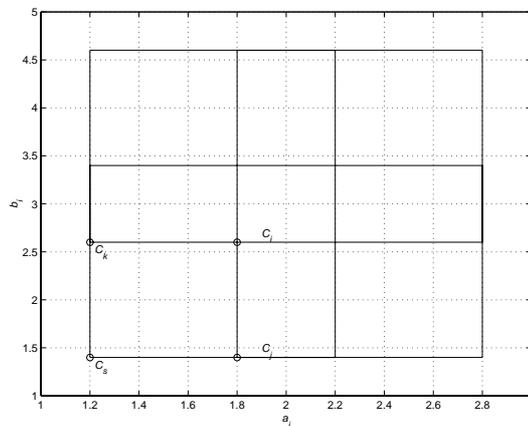


Figura A.16: Elementos focales conjuntos cuando  $\alpha_A$  y  $\alpha_B$  son independientes

Para calcular la plausibilidad de  $(a_1 \text{ AND } b_1)$  hay que considerar todos los elementos focales conjuntos que contienen el par  $(a_1, b_1)$ . Los elementos focales conjuntos representados por los puntos situados en la base del volumen de la Figura A.17,

contienen  $(a_1, b_1)$ , siendo en este ejemplo  $(a_1, b_1) = (1.3, 2.2)$ . Expresado para variables continuas, la plausibilidad de ese punto es el volumen marcado en la Figura A.17.

$$\begin{aligned}
 Pl((a_1, b_1)) &= \int_{C_i \cap (a_1, b_1) \neq \emptyset} m_{A \times B}(C_i) \cdot d\delta_{C_i} = \\
 &= \int_{A_i=A_j}^{A_n} \int_{B_i=B_k}^{B_n} m_A(A_i) \cdot m_B(B_i) \cdot d\delta_{A_i} \cdot d\delta_{B_i} = \\
 &= \int_{A_i=A_j}^{A_n} m_A(A_i) \cdot d\delta_{A_i} \cdot \int_{B_i=B_k}^{B_n} m_B(B_i) \cdot d\delta_{B_i} = \pi_A(a_1) \cdot \pi_B(b_1)
 \end{aligned}$$

donde  $\delta_{C_i}$  es la medida definida entre elementos focales conjuntos como:

$$\delta(C_j, C_s) = \sup \left\{ d(C_{sup}^j, C_{sup}^s), d(C_{inf}^j, C_{inf}^s), d(C_{der}^j, C_{der}^s), d(C_{izq}^j, C_{izq}^s) \right\}$$

y donde  $\delta_{A_i}$  (respectivamente  $\delta_{B_i}$ ) se refiere a la distancia entre elementos focales de  $A$  (respectivamente  $B$ ) definida anteriormente en este apartado.

Vemos que la plausibilidad de  $(a_1, b_1)$  es el producto de posibilidades  $\pi_A(a_1)$ ,  $\pi_B(b_1)$ , y por tanto, la distribución conjunta de plausibilidad en este caso coincide con la distribución conjunta de posibilidad obtenida por teoría de la posibilidad cuando la t-norma utilizada es el producto.

Tal y como se mostrará más tarde, la diferencia entre combinar la información mediante teoría de la evidencia y combinarla mediante teoría de la posibilidad está en el cuerpo de evidencia conjunto que subyace. El de la teoría de la posibilidad es consonante, mientras que el de la teoría de la evidencia no lo es.

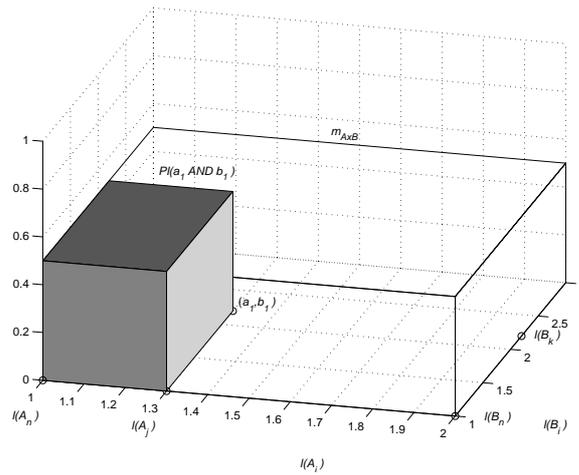


Figura A.17: Plausibilidad de  $(a_1, b_1)$  cuando  $\alpha_A$  y  $\alpha_B$  son independientes.

**Correlación negativa perfecta**

Cuando existe correlación negativa perfecta entre  $\alpha_A$  y  $\alpha_B$  [Wang et al., 1991], [Tan et al., 1993] la asignación básica conjunta es una distribución uniforme en la anti-diagonal del cuadrado unidad, esto es, en la recta  $\alpha_A=1-\alpha_B$ .

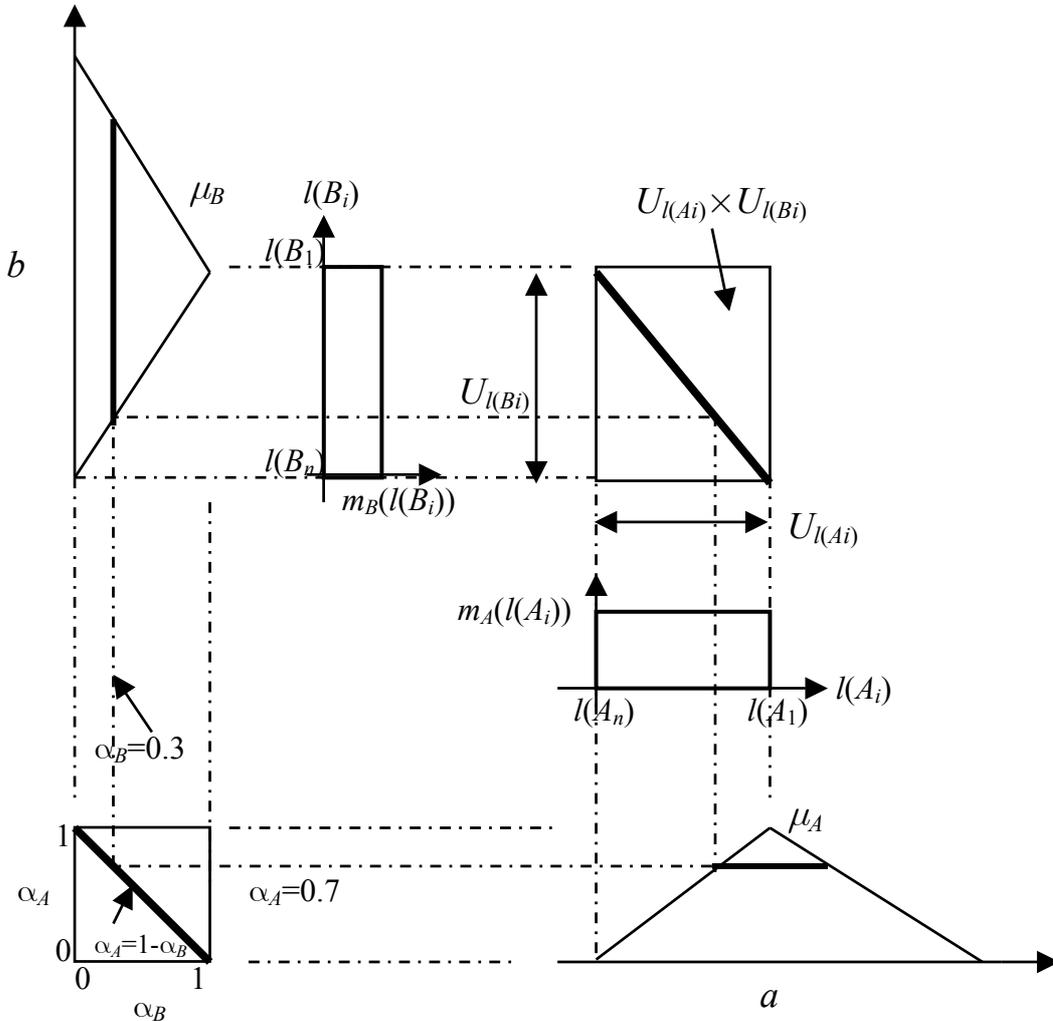


Figura A.18: Correlación negativa perfecta

La anti-diagonal del cuadrado unidad se corresponde con la anti-diagonal de  $U_{l(A_i)} \times U_{l(B_i)}$ :

Para números borrosos triangulares, se puede expresar  $\alpha$  en función de los límites inferiores de los  $\alpha$ -cortes como:

$$\alpha_A = (a - l(A_n)) / (l(A_1) - l(A_n)) \text{ y } \alpha_B = (b - l(B_n)) / (l(B_1) - l(B_n))$$

Con lo que se tiene que la recta  $\alpha_A = 1 - \alpha_B$  es:

$$b = (l(B_1) - l(B_n)) * [ 1 - (a - l(A_n)) / (l(A_1) - l(A_n)) ] + l(B_n) \text{ que es la anti-diagonal de } U_{l(A_i)} \times U_{l(B_i)} \text{ (ver Figura A.18).}$$

Esta asignación básica conjunta se puede obtener mediante el método de Montecarlo, generando un número aleatorio uniformemente distribuido entre 0 y 1. Para ese valor de  $\alpha \in (0, 1]$ , se obtiene el elemento focal de  $A$  correspondiente y con  $1 - \alpha$  se obtiene el elemento focal de  $B$  correspondiente.

En la Figura A.19 se muestra la asignación básica conjunta de los cuerpos de evidencia de  $A$  y de  $B$  que aparecen en la Figura A.11, cuando existe correlación negativa perfecta entre  $\alpha_A$  y  $\alpha_B$ .

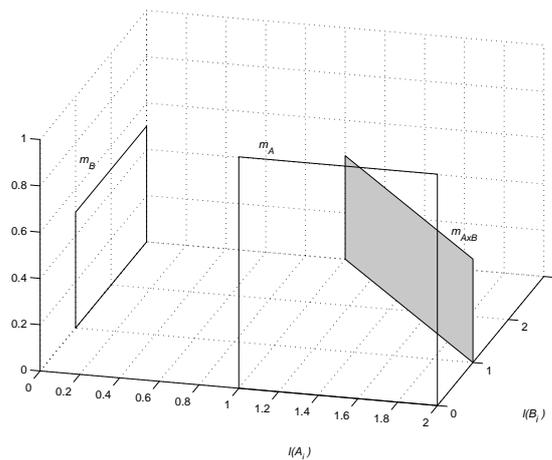


Figura A.19: Asignación básica conjunta cuando  $\alpha_A = 1 - \alpha_B$

En este caso los elementos focales conjuntos no están anidados, tal y como se observa en la Figura A.20, donde se muestran dos elementos focales conjuntos  $C_k$  y  $C_j$ . Los vértices de los elementos focales se desplazan a lo largo de las líneas discontinuas. El cuerpo de evidencia no es consonante y, por tanto, las medidas de plausibilidad no son medidas de posibilidad. En esta figura se puede ver que la plausibilidad de  $(a_1, b_1)$ , con  $(a_1, b_1) = (1.3, 1.5)$  es 0, ya que ningún elemento focal del cuerpo de evidencia conjunto contiene dicho punto. Para calcular la plausibilidad de  $(a_2, b_2) = (1.8, 2.6)$ , se observa que todos los elementos focales conjuntos que van de  $C_j$  a  $C_k$  contienen dicho punto, con lo que expresado en variables continuas se tiene:

$$Pl((a_2, b_2)) = \int_{C_i=C_k}^{C_j} m_{AxB}(C_i) \cdot d\delta_{C_i}$$

que puede interpretarse como el área sombreada en la Figura A.21.

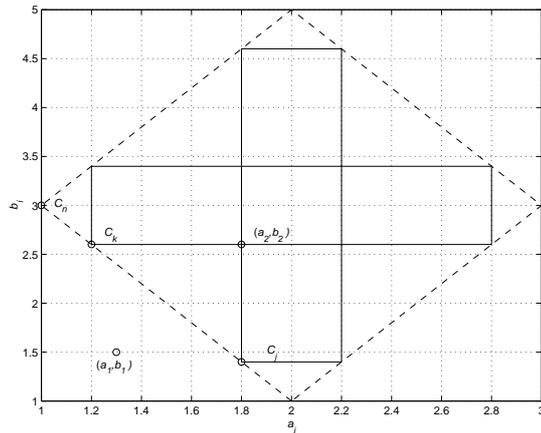


Figura A.20: Elementos focales del cuerpo de evidencia conjunto cuando  $\alpha_A=1-\alpha_B$

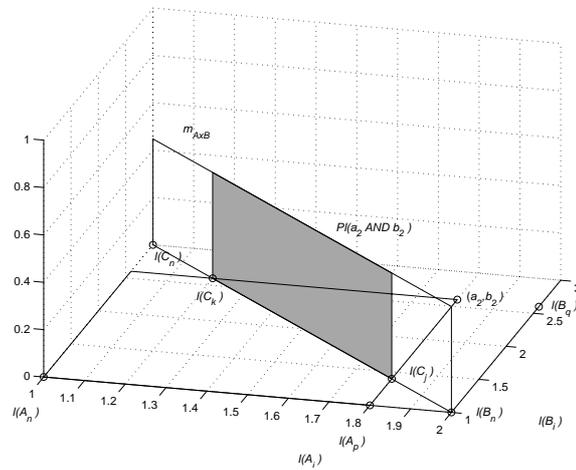


Figura A.21: Plausibilidad de  $(a_2, b_2)$  cuando  $\alpha_A=1-\alpha_B$

Se puede expresar la medida de plausibilidad en términos de los cuerpos de evidencia de partida como:

$$\begin{aligned}
 Pl((a_2, b_2)) &= \int_{C_i=C_k}^{C_j} m_{AxB}(C_i) \cdot d\delta_{C_i} = \\
 &= \int_{C_i=C_n}^{C_j} m_{AxB}(C_i) \cdot d\delta_{C_i} - \int_{C_i=C_n}^{C_k} m_{AxB}(C_i) \cdot d\delta_{C_i} = \\
 &= \int_{A_i=A_p}^{A_n} m_A(A_i) \cdot d\delta_{A_i} - \int_{B_i=B_1}^{B_2} m_B(B_i) \cdot d\delta_{B_i} = \\
 &= \int_{A_i=A_p}^{A_n} m_A(A_i) \cdot d\delta_{A_i} - \left( 1 - \int_{B_i=B_1}^{B_2} m_B(B_i) \cdot d\delta_{B_i} \right) = \\
 &= \pi_A(a_2) - 1 + \pi_B(b_2)
 \end{aligned}$$

Para el par anterior  $(a_1, b_1)$ ,  $Pl(a_1, b_1)=0$  y  $\pi_A(a_1)+\pi_B(b_1)-1 < 0$ . Por tanto en general, para cualquier par  $(a_i, b_j)$ , se cumple:

$$Pl((a_i, b_j)) = \max ( 0, \pi_A(a_i)+\pi_B(b_j)-1)$$

que es precisamente la t-norma de Lukasiewicz. La distribución conjunta de plausibilidad obtenida en teoría de la evidencia con  $\alpha_A=1-\alpha_B$  coincide con la distribución conjunta de posibilidad con la t-norma de Lukasiewicz. De nuevo, la diferencia se encuentra en el cuerpo de evidencia que subyace: en teoría de la posibilidad el cuerpo de evidencia subyacente es consonante, mientras que en teoría de la evidencia el cuerpo de evidencia subyacente de esa misma distribución de plausibilidad es no consonante.

### A.2.3. Principio de extensión

Dos posibles formas de extender funciones reales con argumentos borrosos son mediante la teoría de la posibilidad y mediante la teoría de la evidencia, combinando los cuerpos de evidencia asociados a los conjuntos borrosos. En la teoría de la posibilidad se enuncia el principio de extensión [Dubois, Prade, 1991a]:

$$\pi_c(c) = \sup_{c=f(a,b)} (T(\pi_A(a), \pi_B(b)))$$

siendo  $T$  una t-norma.

Por otro lado, dado un cuerpo de evidencia conjunto  $(F_{A \times B}, m_{A \times B})$ , se puede extender una función de números reales  $c=f(a,b)$  de forma que la distribución de plausibilidad resultante de la extensión de  $f$  es:

$$Pl(c) = \sum_{\substack{(a_1, b_1) \in C_i \\ OR (a_2, b_2) \in C_i \\ \dots OR (a_n, b_n) \in C_i \\ \forall (a_j, b_j) \mid c = f(a_j, b_j)}} m_{A \times B}(C_i)$$

Supongamos que existen dos puntos  $(a_1, b_1)$  y  $(a_2, b_2)$  tales que  $f(a_1, b_1) = f(a_2, b_2) = c$ . Según la teoría de la posibilidad, la posibilidad de un conjunto de puntos  $\{(a_1, b_1), (a_2, b_2)\}$  es el supremo de las posibilidades de los puntos del conjunto. Así, el principio de extensión queda expresado en este caso como:

$$\pi(c) = \pi((a_1, b_1) \cup (a_2, b_2)) = \sup(\pi(a_1, b_1), \pi(a_2, b_2)) = \max(\pi(a_1, b_1), \pi(a_2, b_2))$$

donde la posibilidad conjunta  $\pi(a_i, b_i)$ , se obtiene combinando las de partida con una t-norma.

En teoría de la evidencia la extensión de funciones reales con argumentos inciertos se calcula a partir del cuerpo de evidencia conjunto, previamente obtenido a partir de los conjuntos borrosos. Supongamos de nuevo que existen dos puntos  $(a_1, b_1)$  y  $(a_2, b_2)$  tales que  $f(a_1, b_1) = f(a_2, b_2) = c$ . Entonces, la plausibilidad de  $c$  es la plausibilidad de  $(a_1, b_1) \text{OR} (a_2, b_2)$ . Dado un cuerpo de evidencia  $(F_{A \times B}, m_{A \times B})$ , la medida de plausibilidad de un conjunto formado por dos puntos  $(a_1, b_1), (a_2, b_2)$  se calcula:

$$Pl((a_1, b_1) \cup (a_2, b_2)) = \sum_{\substack{(a_1, b_1) \in C_i \\ or \\ (a_2, b_2) \in C_i}} m_{A \times B}(C_i)$$

esto es, intervienen en la fórmula todos aquellos elementos focales conjuntos que contienen al menos uno de los dos puntos del conjunto.

La Figura A.22.a) muestra en línea gruesa los representantes de los elementos focales conjuntos que contienen  $(a_1, b_1)$  o  $(a_2, b_2)$ , cuando  $\alpha_A = \alpha_B$ . En este caso, la plausibilidad de la unión de los dos puntos es la plausibilidad de  $(a_2, b_2)$ , que es precisamente el supremo de ambas plausibilidades. Además, por ser cuerpos de evidencia consonantes las plausibilidades son posibilidades, con lo que en este caso:

$$\pi((a_1, b_1) \cup (a_2, b_2)) = Pl((a_1, b_1) \cup (a_2, b_2)) = \max(Pl(a_1, b_1), Pl(a_2, b_2)) = \max(\pi(a_1, b_1), \pi(a_2, b_2))$$

El resultado obtenido es el mismo que por teoría de la posibilidad, porque el cuerpo de evidencia subyacente es también el mismo.

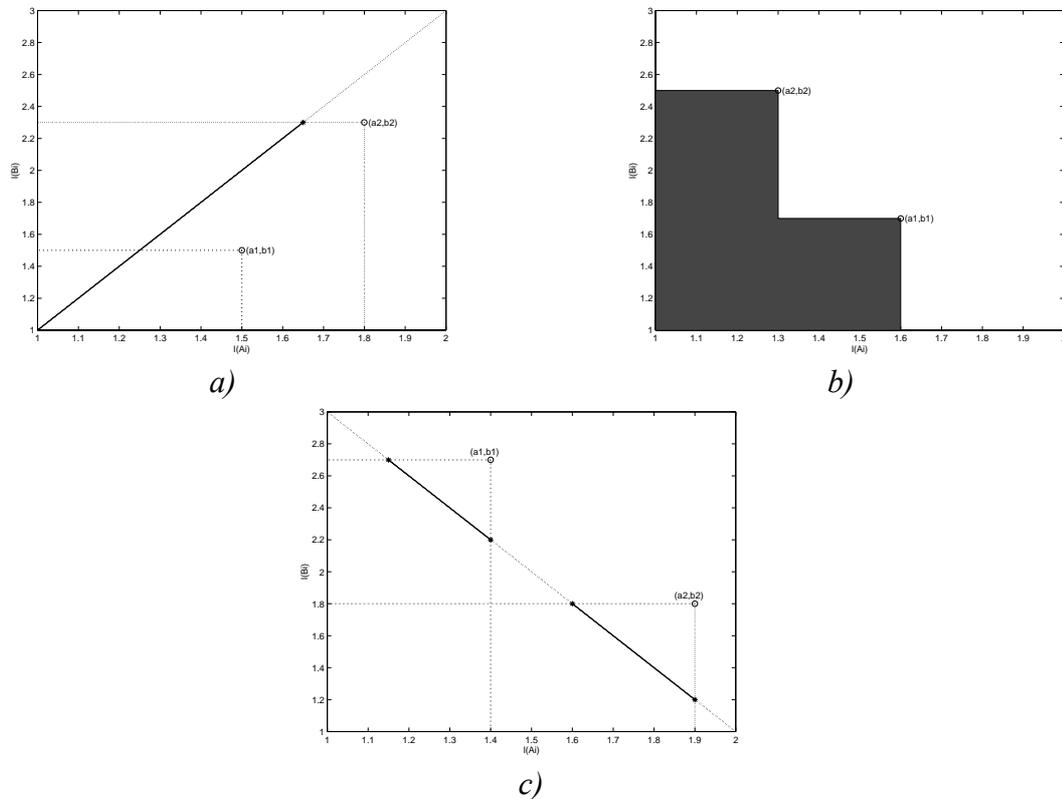


Figura A.22: Elementos focales conjuntos que contienen  $(a_1, b_1)$  o  $(a_2, b_2)$ .  
 a)  $\alpha_A = \alpha_B$ , b) independencia, c)  $\alpha_A = 1 - \alpha_B$

La Figura A.22.b) muestra los elementos focales conjuntos cuando  $\alpha_A$  y  $\alpha_B$  son independientes. Todos los elementos focales cuyos representantes se encuentran en el área sombreada contienen al menos uno de los dos puntos y deben considerarse en el cálculo de la plausibilidad. Para variables continuas, la plausibilidad es la integral de la distribución uniforme  $m_{A \times B}$  a lo largo del área sombreada, es decir, la medida de plausibilidad es el volumen cuya base está sombreada. Como en general este volumen es mayor o igual que el volumen asociado a  $Pl((a_1, b_1))$ , o a  $Pl((a_2, b_2))$ , se verifica:

$$Pl((a_1, b_1) \cup (a_2, b_2)) \geq \max(Pl((a_1, b_1)), Pl((a_2, b_2)))$$

La Figura A.22.c) muestra los elementos focales conjuntos que contienen al menos uno de los dos puntos  $(a_1, b_1)$ ,  $(a_2, b_2)$  cuando  $\alpha_A = 1 - \alpha_B$ . Para variables continuas, la plausibilidad es el área cuya base se marca con línea gruesa, y en general es mayor o igual que las medidas de plausibilidad individuales de cada punto. Por tanto, de nuevo la plausibilidad de la unión es mayor o igual que el supremo de ambas.

Se ha demostrado gráficamente el hecho ya conocido de que la correspondencia entre teoría de la evidencia y teoría de la posibilidad no existe al aplicar el principio de extensión [Dubois, Prade, 1989a]. Sólo en el caso de correlación positiva perfecta, donde el cuerpo de evidencia conjunto es consonante, el resultado coincide. Los elementos focales conjuntos definidos como los  $\alpha$ -cortes de las distribuciones de posibilidad conjuntas con *min*, producto y Lukasiewicz, se muestran en la Figura A.26. Se puede ver que los elementos focales correspondientes a *min* coinciden con los de correlación positiva perfecta.

### Suma de dos números borrosos

A continuación se presenta como ejemplo de aplicación del principio de extensión la extensión de la función suma con argumentos borrosos. Es la operación básica que aparece para el cálculo de los tiempos de finalización de las tareas en los sistemas de tiempo real. Éstos se pueden expresar como sumas de los tiempos de ejecución con discontinuidades.

Así, la extensión de la operación suma de dos números reales  $a$  y  $b$ ,  $r=a+b$ , al modelarlos mediante números borrosos  $A$  y  $B$  se puede expresar como  $R=A+B$ . El resultado borroso se puede obtener calculando la posibilidad/plausibilidad de la unión de los puntos de la recta  $a_j + b_k = r_i$ , para todo  $r_i \in \mathbb{R}$ .

#### Teoría de la Posibilidad

Aplicando el principio de extensión con la t-norma  $T$  (A.5), la posibilidad de  $r_i$  se calcula como el máximo de las posibilidades de los pares  $(a_j, b_k)$  que cumplen  $a_j + b_k = r_i$ . Gráficamente en la Figura A.23, hay que coger el máximo valor de la sección definida por el corte de la distribución de posibilidad conjunta y el plano vertical que pasa por  $a_j + b_k = r_i$ .

$$(A.5) \pi_R(r_i) = \sup_{r_i = a_j + b_k} T\{\pi_A(a_j), \pi_B(b_k)\}$$

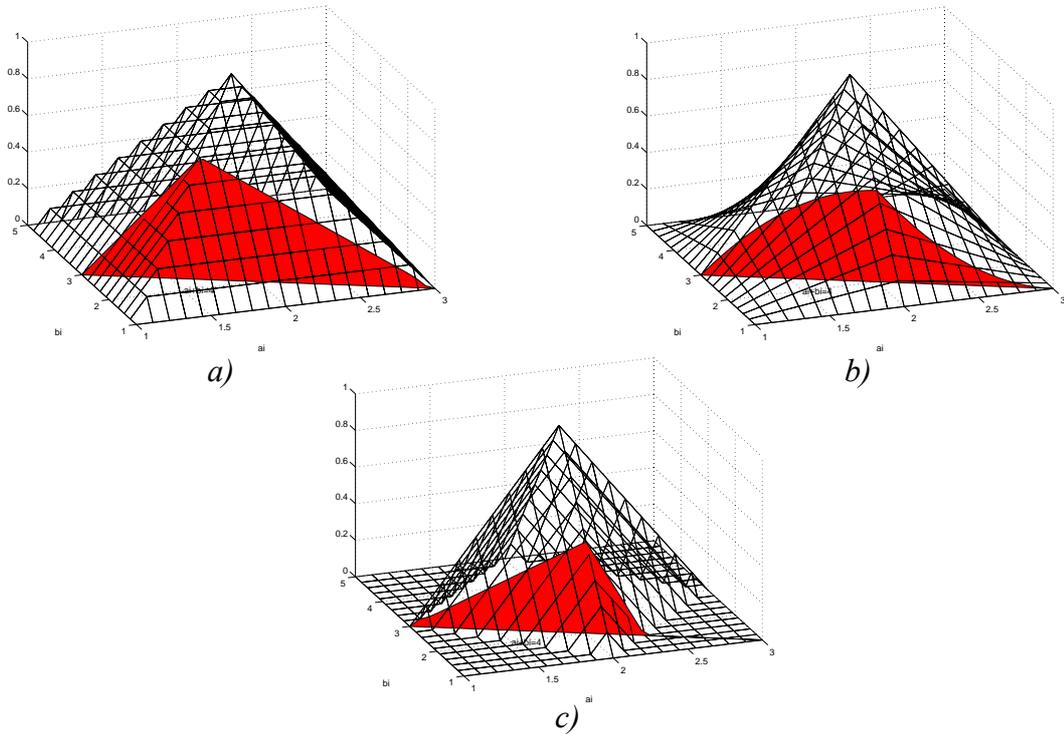


Figura A.23: Corte definido por  $a_j + b_k = \text{constante}$  con tres t-normas:  
 a) mínimo, b) producto, c) Lukasiewicz

Teoría de la Evidencia

Dada la recta definida como  $a_j + b_k = \text{constante} = r_i$ , la plausibilidad de  $r_i$  se calcula sumando (o bien integrando, para variables continuas) la asignación básica conjunta de todos los elementos focales que contengan al menos un punto de dicha recta.

$$Pl(r_i) = \sum_{\substack{(a_j, b_k) \in C_i \\ | r_i = a_j + b_k}} m_{A \times B}(C_i)$$

La Figura A.24.a) marca los representantes de dichos elementos focales cuando la correlación entre  $\alpha_A$  y  $\alpha_B$  es positiva perfecta, esto es  $\alpha_A = \alpha_B$ . En el ejemplo de la Figura A.24.a), la posibilidad de  $r_i=3$  (en línea discontinua) coincide con la posibilidad del punto definido por el corte entre la diagonal principal y  $a_j + b_k = 3$ . Es decir, la posibilidad de  $r_i=3$  es la integral de la asignación básica conjunta en el tramo grueso de la diagonal principal.

La Figura A.24.b) marca los representantes de los elementos focales que contienen al menos un punto de la recta  $r_i=3$  cuando  $\alpha_A$  y  $\alpha_B$  son independientes. La medida de plausibilidad de  $r_i=3$  es el volumen cuya base está sombreada.

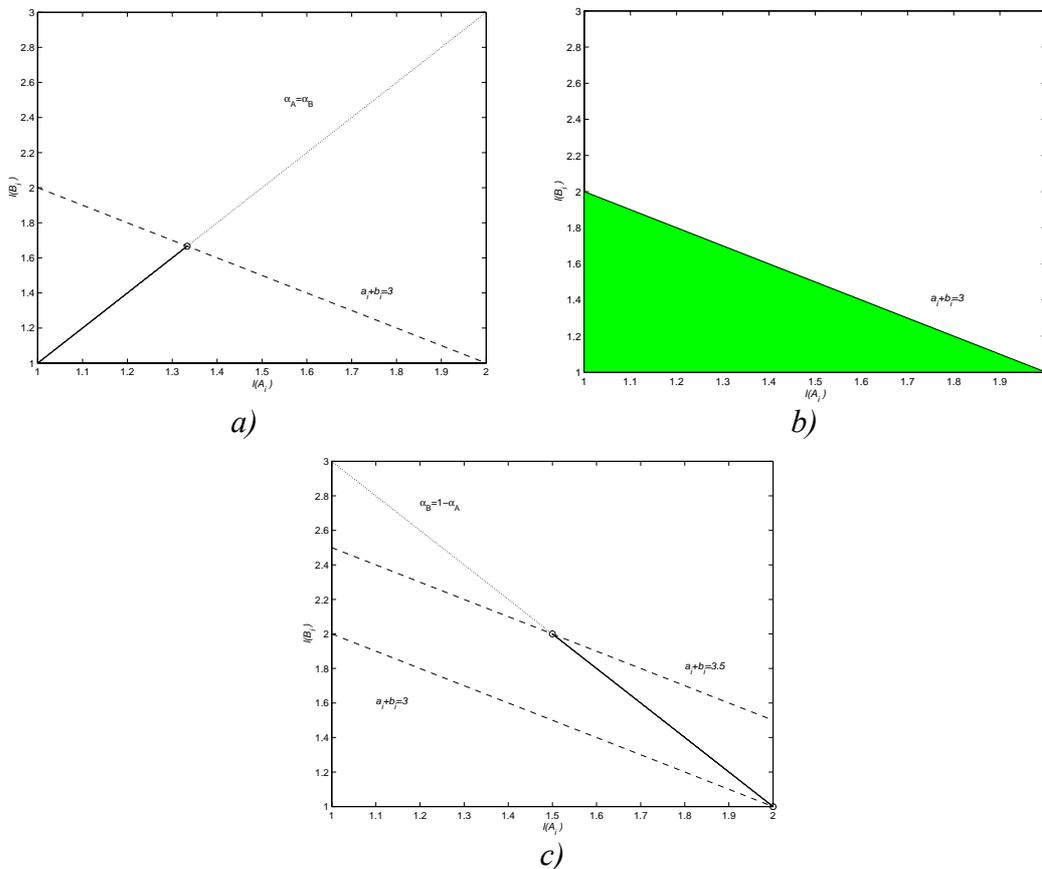


Figura A.24: Elementos focales conjuntos que contienen algún punto de  $a_j + b_k = constante$ :  
 a)  $\alpha_A = \alpha_B$ , b) independencia, c)  $\alpha_A = 1 - \alpha_B$

Tal y como se muestra en la Figura A.24.c), donde la correlación entre  $\alpha_A$  y  $\alpha_B$  es  $\alpha_A = 1 - \alpha_B$ , ningún elemento focal conjunto contiene punto alguno de la recta  $a_j + b_k = 3$ . Esto quiere decir que la plausibilidad de  $r_i = 3$  es nula. La Figura A.24.c) muestra también en línea gruesa los representantes de los elementos focales que intervienen en el cálculo de la plausibilidad de  $r_i = 3.5$ .

Podemos comprobar de nuevo que la plausibilidad de la unión de varios puntos (todos los puntos pertenecientes a  $a_j + b_k = constante$ ) es mayor o igual que el máximo de las plausibilidades de los puntos:

$$Pl(r_i) \geq \sup_{a_j + b_k = r_i} (Pl(a_j, b_k))$$

Excepto con  $\alpha_A = \alpha_B$  y la t-norma del mínimo que dan el mismo resultado, en general la teoría de la evidencia da distribuciones menos específicas que la teoría de la posibilidad.

A continuación se va a mostrar el cálculo de la suma borrosa mediante cálculo de intervalos y se verá que se puede interpretar tanto desde el punto de vista de la teoría de la posibilidad como desde la teoría de la evidencia.

### *Suma borrosa como suma de intervalos*

Cuando la t-norma utilizada es el *min*, las operaciones de aritmética borrosa suma, resta, multiplicación y división de conjuntos borrosos se pueden expresar como operaciones de aritmética de intervalos, donde los intervalos son los  $\alpha$ -cortes de los conjuntos borrosos. A continuación, se va a analizar gráficamente de nuevo la suma de dos números borrosos con el fin de ilustrar, que por teoría de la evidencia se llega también a la operación de intervalos, que coinciden con los  $\alpha$ -cortes.

En teoría de la posibilidad, cuando la t-norma aplicada es el mínimo, el principio de extensión de la función suma de dos números,

$$\pi_R(r) = \sup_{r=a+b} (\min(\pi_A(a), \pi_B(b)))$$

puede expresarse también mediante representación de intervalos de la siguiente forma [Nakamura, 1993]:

$$[\underline{r}, \bar{r}](\alpha) = [\underline{a} + \underline{b}, \bar{a} + \bar{b}](\alpha)$$

donde  $[\underline{r}, \bar{r}](\alpha)$  es un  $\alpha$ -corte de  $R$ . Esto significa que sólo hay que calcular la suma de las parejas  $(a, b)$  con el mismo  $\alpha$  para obtener un  $r$  con ese mismo valor de  $\alpha$ . Se reduce así considerablemente el tiempo de cálculo de la operación suma.

Veamos el cálculo de la suma por teoría de la evidencia, utilizando la representación gráfica propuesta, cuando la correlación es  $\alpha_A = \alpha_B$ . En la Figura A.25 se representan, tanto los elementos focales conjuntos  $C_i$  como la función suma a calcular. Los elementos focales conjuntos son rectángulos con los vértices situados a lo largo de las diagonales. Los pares  $(a, b)$  con el mismo  $\alpha$  son los puntos situados en dichas diagonales.

La posibilidad de cada recta  $a_j + b_k = \text{constante} = r_i$  es la suma de la asignación básica de los  $C_i$  que contienen al menos un punto de la recta. Esto es, en la Figura

A.25, todos los elementos focales desde  $C_n$  hasta  $C_i$ . Esta es precisamente la posibilidad del par  $(a_1, b_1)$  que es el punto de la recta  $a_j + b_k = \text{constante} = r_i$  que cumple que  $a$  y  $b$  tienen el mismo  $\alpha_t$ . Para calcular la posibilidad de  $r_p$  (ver Figura A.25) hay que considerar los mismos elementos focales de  $C_n$  a  $C_i$ , obteniéndose el mismo  $\alpha_t$ . Como  $R$  es consonante,  $r_i$  y  $r_p$  son los límites del  $\alpha_t$ -corte de  $R$ ,  $\alpha_t$  es el valor de posibilidad del par  $(a_2, b_2)$  y la posibilidad tanto de  $a_2$  como de  $b_2$  es también  $\alpha_t$ . Por tanto,  $[a_1, a_2]$  es el  $\alpha_t$ -corte de  $A$  (respectivamente  $B$ ).

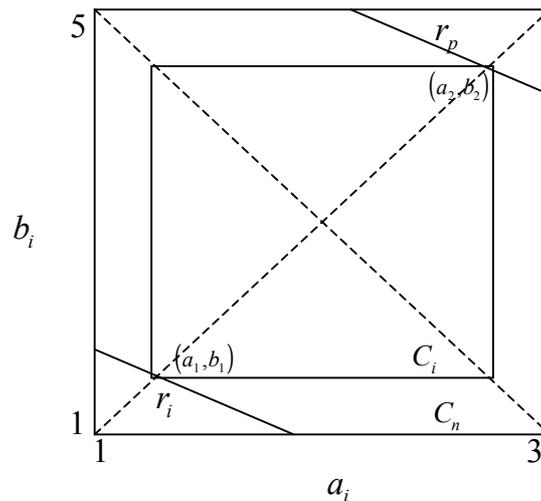


Figura A.25: Representación de la función  $r = a+b$  así como de los elementos focales conjuntos.

Se pueden encontrar algunos estudios sobre aritmética borrosa con diversas t-normas en [Mesiar, 1997], [De Baets, Markova-Stupnanova, 1997], [Markova-Stupnanova, 1997] y [Klir, 1997].

#### A.2.4. Interpretación de la relación entre $\alpha_A$ y $\alpha_B$

En [Kruse et al., 1994] se presenta una interpretación interesante de la aplicación de cada una de las tres t-normas mencionadas antes en la combinación de información por teoría de la posibilidad. Se tienen dos cubos de  $4 \times 4 \times 4$  casillas cada uno. Cada casilla puede ser gris o transparente y el grado de gris (grado de pertenencia) de una casilla del plano  $4 \times 4$  de proyección al proyectar una luz vertical depende del número de casillas grises en esa columna. Hay cinco posibles grados de grises en la proyección. Al combinar los cubos se obtiene otro cubo donde cada casilla es gris si las mismas casillas de los dos cubos de partida son ambas grises. En el cubo

resultante el grado de gris de la proyección depende de los grados de grises de los cubos de partida y de la información disponible acerca de la posición de casillas grises en cada cubo.

Si se sabe que en cada columna de ambos cubos las casillas transparentes están por encima de las grises, entonces el grado de gris de la proyección resultante se calcula con la t-norma mínimo aplicada a las proyecciones de los cubos de partida. Si se sabe que para cada columna del cubo 1 con las casillas transparentes por encima de las grises, la correspondiente columna del cubo 2 tiene las casillas transparentes por debajo de las grises, entonces el grado de gris de la proyección resultante se calcula con la t-norma de Lukasiewicz. Si la posición de casillas grises es independiente en cada cubo, entonces el grado de gris de la proyección resultante se calcula con la t-norma producto. Para más detalle ver [Kruse et al., 1994].

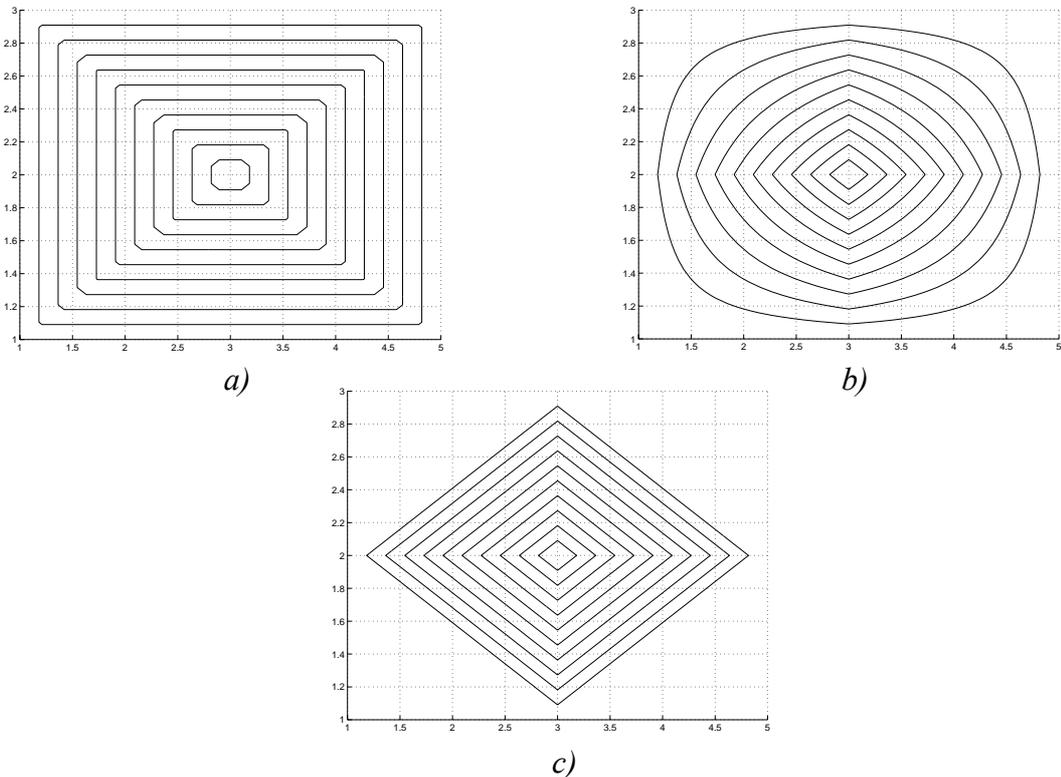


Figura A.26: Elementos focales conjuntos con tres t-normas: a) min, b) producto, c) Lukasiewicz

A continuación se va a presentar una nueva interpretación de las tres asignaciones básicas conjuntas analizadas anteriormente en teoría de la evidencia, considerando los elementos focales provenientes de los  $\alpha$ -cortes como observaciones imprecisas de una variable expresadas mediante intervalos.

En la Figura A.26 se muestran los elementos focales del cuerpo de evidencia consonante asociado a las distribuciones conjuntas de posibilidad con tres t-normas: mínimo, producto y Lukasiewicz (Figura A.6). Se obtienen como secciones horizontales de las superficies a distintos valores de  $\alpha$ , ya que coinciden con los  $\alpha$ -cortes de las distribuciones conjuntas.

Tal y como se ve en la Figura A.26, salvo en el caso del *min* los elementos focales conjuntos que subyacen en la distribución obtenida por teoría de la posibilidad no son rectángulos, esto es, no resultan de productos Cartesianos de los intervalos de partida  $A_i$  y  $B_j$ . Por tanto, no pueden ser interpretados como una distribución obtenida a partir de observaciones imprecisas de  $A$  y de  $B$  [Dubois, Prade, 1991b].

Cuando los números borrosos  $A$  y  $B$  están contruidos a partir de observaciones imprecisas [Dubois, Prade, 1986a] se puede aplicar la teoría de la evidencia. La asignación básica conjunta puede modelar la información disponible acerca de la forma en que se han obtenido los intervalos  $A_i$  y  $B_j$ . A continuación se propone una interpretación para los tres casos analizados en las secciones previas.

Por un lado, supongamos que el experimento realizado consiste en preguntar a una serie de expertos qué intervalo puede definir a dichos concepto impreciso. Entonces:

- *Correlación positiva perfecta*: El mismo experto da intervalos  $A_i$  y  $B_j$  para las variables  $a$  y  $b$  (obviamente, utilizando el mismo criterio).
- *Independencia*: Los expertos que dan los intervalos  $A_i$  para  $a$  son distintos que los que dan los intervalos  $B_j$  para  $b$  (y por tanto utilizan criterios independientes).
- *Correlación negativa perfecta*: En este caso, existe una precisión total que debe ser compartida por las observaciones de  $a$  y  $b$ . Por ejemplo, un único experto, situado entre dos objetos tiene que estimar la altura de ambos ( $A$  y  $B$ ). Cuanto más cerca se encuentre del primero, con más precisión estimará su altura, pero será menos preciso en la estimación de la segunda.

Por otro lado, supongamos que el experimento consiste en realizar una serie de medidas con instrumentos que tienen errores de medida.

- *Correlación positiva perfecta*: Se utiliza el mismo instrumento de medida para medir  $a$  y  $b$ .
- *Independencia*: Se utilizan instrumentos de medida diferentes para medir  $a$  y  $b$ , con principios de medida independientes.

- *Correlación negativa perfecta*: Si se realizan dos medidas con un número total de bits fijo (precisión total fija), cuanto más precisa sea la primera medida (más bits utilizados), menos precisa será la segunda.

En conclusión, en este anexo se ha propuesto una representación gráfica del cuerpo de evidencia conjunto asociado a dos argumentos inciertos. Mediante la aplicación de la representación es posible extender funciones de dos argumentos a valores borrosos y obtener la distribución de plausibilidad de la imagen de dichos argumentos borrosos a través de la función. En el dominio de los elementos focales completos  $A_j \times B_k$  se representa la función completa y los elementos focales conjuntos. En el dominio de los límites inferiores de los elemento focales  $U_{I(A_i)} \times U_{I(B_i)}$  se representa la asignación básica conjunta del cuerpo de evidencia y se representa el área o el volumen que hay que calcular para obtener la plausibilidad de cada imagen de la función.



## **B. Anexo: Comparación de dos conjuntos borrosos**

En este anexo se analiza la comparación entre dos conjuntos borrosos proporcionando índices que miden el grado en que un conjunto se puede considerar menor o igual que otro, (o si modelan fechas temporales, el grado en que una fecha está cronológicamente antes que la otra).

En el apartado B.1. se definen los conceptos de periodos de tiempo antes y después de una fecha borrosa  $A$  como los intervalos borrosos,  $(-\infty, A]$ ,  $(-\infty, A[$ ,  $]A, +\infty)$ , y  $]A, +\infty)$ .

En el apartado B.2. se definen algunos de los índices de ordenación de fechas, entre los que destacan los índices de posibilidad y de necesidad de que una fecha  $A$  esté antes que otra  $B$ ,  $\Pi(A \leq B)$ ,  $\Pi(A < B)$ ,  $N(A \leq B)$  y  $N(A < B)$ . Estos índices son los utilizados en esta tesis para medir la planificabilidad del sistema ya que no interesa tanto la decisión de una ordenación determinada (que es el resultado obtenido al aplicar otros índices, tal i como se ve en el apartado B.2.2.), sino conservar toda la información sobre el orden relativo entre los tiempos de finalización y los plazos borrosos.

Por último, en el apartado B.3. se muestra que los valores de posibilidad y necesidad de que un conjunto borroso sea menor o igual que otro coinciden si se sustituyen los conjuntos por otros con los mismos límites inferiores y superiores de los  $\alpha$ -cortes. Este resultado es de gran importancia en esta tesis porque establece la condición que deben cumplir los tiempos de finalización borrosos aproximados, tal y como se indica en el capítulo 3 calculables de forma más sencilla que el principio de extensión, para que la planificabilidad borrosa coincida con la obtenida mediante el principio de extensión.

## B.1. Definición de periodos de tiempo antes y después de una fecha

La distribución de posibilidad  $\pi_A$  de una fecha  $A$  es  $\pi_A: \mathbb{R}^+ \rightarrow [0,1]$  una función definida de la escala de tiempo al intervalo  $[0,1]$ , y delimita el conjunto borroso de los valores de tiempo más o menos posibles de  $A$ . En [Dubois, Prade, 1989b] la fecha es definida como un conjunto borroso unimodal, es decir, convexo.

En la Figura B.1 se muestran los conjuntos borrosos  $]A, +\infty)$ ,  $]A, +\infty)$ ,  $(-\infty, A]$  y  $(-\infty, A[$  que se pueden obtener a partir de la fecha borrosa  $A$ .

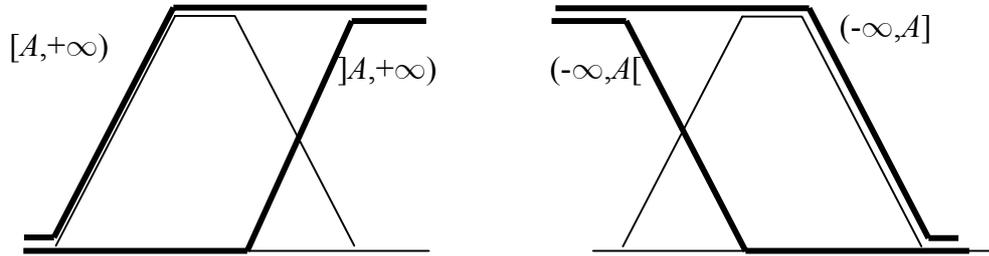


Figura B.1: Representación de los tiempos antes y después de una fecha  $A$ .

- 1) El conjunto borroso  $[A, +\infty)$  de los tiempos que están posiblemente después que la fecha  $A$ , se define por la siguiente función de pertenencia:

$$\mu_{[A, +\infty)}(t) = \sup_{s \leq t} \pi_A(s) = \Pi_A((-\infty, t])$$

- 2) El conjunto borroso  $]A, +\infty)$  de los tiempos que están necesariamente estrictamente después que la fecha  $A$ , se define por la siguiente función de pertenencia:

$$\mu_{]A, +\infty)}(t) = \inf_{s \geq t} (1 - \pi_A(s)) = N_A((-\infty, t])$$

- 3) El conjunto borroso  $(-\infty, A]$  de los tiempos que están posiblemente antes que la fecha  $A$ , se define por la siguiente función de pertenencia:

$$\mu_{(-\infty, A]}(t) = \sup_{s \geq t} \pi_A(s) = \Pi_A([t, +\infty))$$

- 4) El conjunto borroso  $(-\infty, A[$  de los tiempos que están necesariamente estrictamente antes que la fecha  $A$ , se define por la siguiente función de pertenencia:

$$\mu_{(-\infty, A[}(t) = \inf_{s \leq t} (1 - \pi_A(s)) = N_A([t, +\infty))$$

## B.2. Ordenación de fechas

Para calcular la medida de posibilidad o de necesidad de que una fecha tenga lugar antes que otra, se puede aplicar el principio de extensión, o bien calcularlas como la altura de la intersección de determinados conjuntos borrosos.

Se definen los índices  $\Pi(A \leq B)$ ,  $\Pi(A < B)$ ,  $N(A \leq B)$  y  $N(A < B)$ , como ([Dubois, Prade, 1989b]):

$$\Pi(A \leq B) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t)) \quad \Pi(A < B) = \sup_{s < t} \min(\pi_A(s), \pi_B(t))$$

$$N(A \leq B) = 1 - \sup_{s > t} \min(\pi_A(s), \pi_B(t)) \quad N(A < B) = 1 - \sup_{s \geq t} \min(\pi_A(s), \pi_B(t))$$

Si  $A$  o  $B$  son continuos, entonces se tiene:

$$\Pi(A \leq B) = \Pi(A < B) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t))$$

$$N(A \leq B) = N(A < B) = 1 - \sup_{s > t} \min(\pi_A(s), \pi_B(t))$$

Concretamente si para cada valor de  $x \in \mathbb{R}$ , o bien  $A$  es continuo en  $x$ , o bien  $B$  es continuo en  $x$ , entonces se cumple que  $\Pi(A \leq B) = \Pi(A < B)$  y  $N(A \leq B) = N(A < B)$  (ver la Figura B.2).

En este caso  $\Pi(A \leq B) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t)) = \Pi(A < B) = \sup_{s < t} \min(\pi_A(s), \pi_B(t)) = \pi_A(x_1)$ .

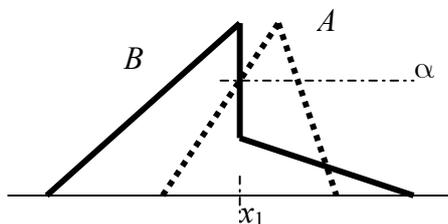


Figura B.2: Comparación de dos números borrosos

### B.2.1. Índices de ordenación

En [Dubois, Prade, 1983] se definen otros dos índices, además de los anteriores, necesarios para tener un análisis completo de la posición relativa de dos distribuciones de posibilidad, aunque son menos utilizados en la literatura. Los índices (descritos en [Dubois, Prade, 1983]) para determinar la posición de un número borroso  $A$  con respecto a otro número borroso  $B$  que se toma como referencia, son:

$$(B.1) \Pi_B([A, +\infty)) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t)) = \Pi(A \leq B)$$

$$(B.2) \Pi_B(]A, +\infty)) = \sup_t \inf_{\substack{s \\ s \geq t}} \min(1 - \pi_A(s), \pi_B(t))$$

$$(B.3) N_B([A, +\infty)) = \inf_t \sup_{\substack{s \\ s \leq t}} \max(\pi_A(s), 1 - \pi_B(t))$$

$$(B.4) N_B(]A, +\infty)) = 1 - \sup_{t \leq s} \min(\pi_A(s), \pi_B(t)) = N(A < B)$$

Nótese que los índices  $\Pi_B([A, +\infty))$  y  $N_B(]A, +\infty))$  corresponden a las definiciones previas de  $\Pi(A \leq B)$  y  $N(A < B)$ .

El primer índice  $\Pi_B([A, +\infty))$  se refiere al grado en que el conjunto borroso  $B \cap [A, +\infty)$  de números mayores o iguales que  $A$ , dado que están restringidos por  $B$ , es distinto de cero. El segundo índice  $\Pi_B(]A, +\infty))$  se refiere al grado en que el conjunto borroso  $B \cap ]A, +\infty)$  de números estrictamente mayores que  $A$ , dado que están restringidos por  $B$ , es distinto de cero. Por tanto, el primer índice proporciona el grado de posibilidad de la proposición “ $x$  es mayor o igual que  $A$ ” dado que “ $x$  es  $B$ ”, y el segundo índice proporciona el grado de posibilidad de la proposición “ $x$  es estrictamente mayor que  $A$ ” dado que “ $x$  es  $B$ ”, esto es:

$$\Pi_B([A, +\infty)) = \text{Pos}(x \geq A \mid x \text{ es } B)$$

$$\Pi_B(]A, +\infty)) = \text{Pos}(x > A \mid x \text{ es } B)$$

De forma similar, el tercer índice  $N_B([A, +\infty))$  se refiere al grado de inclusión del conjunto borroso  $B$  en  $[A, +\infty)$  y el cuarto índice  $N_B(]A, +\infty))$  se refiere al grado de inclusión del conjunto borroso  $B$  en  $]A, +\infty)$ . El índice  $N_B([A, +\infty))$  proporciona el grado de necesidad de la proposición “ $x$  es mayor o igual que  $A$ ” dado que “ $x$  es  $B$ ”,

y  $N_B ([A, +\infty))$  proporciona el grado de necesidad de la proposición “ $x$  es estrictamente mayor que  $A$ ” dado que “ $x$  es  $B$ ”, esto es:

$$N_B ([A, +\infty)) = \text{Nec}(x \geq A \mid x \text{ es } B)$$

$$N_B (]A, +\infty)) = \text{Nec}(x > A \mid x \text{ es } B)$$

Dado este primer conjunto de cuatro índices, se pueden construir otros tres conjuntos formados también por cuatro índices, cambiando  $+\infty$  por  $-\infty$  y/o cambiando  $B$  por  $A$ .

Los cuatro índices son independientes esto es, no se pueden calcular unos a partir de los otros, aunque existen relaciones entre ellos se pueden expresar como:

$$\Pi_B ([A, +\infty)) \geq \max(\Pi_B (]A, +\infty)), N_B ([A, +\infty))$$

$$\min(\Pi_B (]A, +\infty)), N_B ([A, +\infty)) \geq N_B (]A, +\infty))$$

Además  $\Pi_B (]A, +\infty))$  puede ser mayor o menor que  $N_B ([A, +\infty))$ .

#### Índices relacionados con $\Pi_B ([A, +\infty))$ y $N_B (]A, +\infty))$ :

A partir de (B.1) y teniendo en cuenta que  $\overline{[A, +\infty)} = (-\infty, A[$  (donde el complemento de  $[A, +\infty)$  se expresa como  $\overline{[A, +\infty)}$ ), se puede ver que  $\Pi_B ([A, +\infty)) = \Pi_A ((-\infty, B]) = 1 - N_B ((-\infty, A]) = 1 - N_A (]B, +\infty))$

Además se puede deducir que  $N_B (]A, +\infty)) = 1 - \Pi_A (]B, +\infty)) = N_A((-\infty, B]) = 1 - \Pi_B((-\infty, A])$

Por tanto, todos los índices que se obtienen de intercambiar  $+\infty$  y  $-\infty$ , y/o  $A$  y  $B$ , son redundantes respecto al conjunto  $\{ \Pi_B ([A, +\infty)), N_B (]A, +\infty)) \}$

Otras relaciones existentes entre  $\Pi_B ([A, +\infty))$  y  $N_B (]A, +\infty))$  son:

$$\Pi_B ([A, +\infty)) < 1 \Rightarrow N_B (]A, +\infty)) = 0$$

$$N_B (]A, +\infty)) > 0 \Rightarrow \Pi_B ([A, +\infty)) = 1$$

**Indices relacionados con  $\Pi_B ([A, +\infty))$** 

Por la relación entre las medidas de posibilidad y de necesidad se cumple:

$$\Pi_B ([A, +\infty)) = 1 - N_B ((-\infty, A])$$

$$\Pi_A ([B, +\infty)) = 1 - N_A ((-\infty, B])$$

Además, se verifica las siguientes desigualdades:

$$\Pi_B ([A, +\infty)) \leq N_A ((-\infty, B])$$

$$\Pi_B ([A, +\infty)) + \Pi_A ([B, +\infty)) \leq 1$$

Tal y como se afirma en [Dubois, Prade, 1983], en determinadas condiciones de continuidad las desigualdades se convierten en igualdades.

**Indices relacionados con  $N_B ([A, +\infty))$** 

En este caso se verifica:

$$N_B ([A, +\infty)) = 1 - \Pi_B ((-\infty, A])$$

$$N_A ([B, +\infty)) = 1 - \Pi_A ((-\infty, B])$$

$$N_B ([A, +\infty)) \geq \Pi_A ((-\infty, B])$$

$$N_B ([A, +\infty)) + N_A ([B, +\infty)) \geq 1$$

También en este caso, en determinadas condiciones de continuidad (ver [Dubois, Prade, 1983]) las últimas desigualdades pasan a ser igualdades.

**B.2.2. Otros índices**

Existen numerosas propuestas en la literatura para la ordenación de números borrosos, dentro de la teoría de la posibilidad.

La propuesta de Jain, resumida en [Dubois, Prade, 1983] consiste en enlazar cada número borroso  $N_i$  con una meta borrosa  $G$ , donde  $G$  es un conjunto borroso de la escala de ordenación que se considera, cuyo significado es “tan grande como se pueda”. Por ejemplo, si la escala de ordenación es  $[0, a]$ , entonces  $G$  es tal que:

$$\forall x \in [0,a], \mu_G(x) = (x/a)^p \text{ para cierto } p > 0$$

Los números borrosos se ordenan de acuerdo con la consistencia de  $N_i$  con  $G$ :

$$c_i = \sup \min ( \mu_G(x), \mu_{N_i}(x) )$$

El problema de este índice es que los resultados obtenidos pueden ser poco intuitivos tal y como se muestra en la Figura B.3. Según el índice de Jain,  $N_1$  es mayor que  $N_2$ , a pesar de que en este caso parece más intuitivo el resultado contrario.

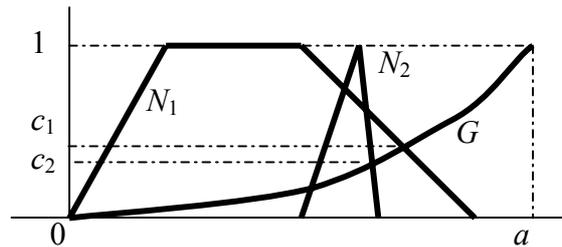


Figura B.3: Comparación de dos números borrosos mediante el índice de Jain

Tanto Dubois y Prade como Freeling han sugerido la utilización de los operadores máximo y mínimo extendidos. Sin embargo,  $\widetilde{\max}(N_1, N_2)$  puede no ser igual ni a  $N_1$  ni a  $N_2$  con lo que en ese caso no sería posible discriminar entre ambos (ver Figura B.4 a)). Además puede ser que  $N_1$  y  $N_2$  estén muy próximos de manera que intuitivamente no esté clara la discriminación como ocurre por ejemplo en la Figura B.4 b), donde se tiene  $\widetilde{\max}(N_1, N_2) = N_2$ .

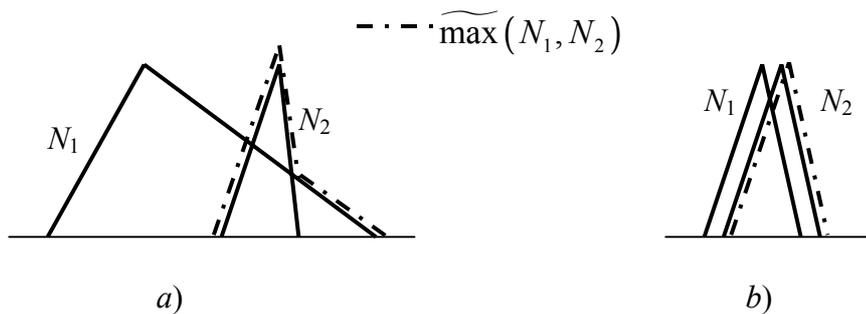


Figura B.4: Máximo extendido de dos números borrosos

El procedimiento propuesto por Yager para la ordenación se basa en el cálculo de la distancia de Hamming entre números borrosos con soportes acotados:

$$\Delta(N_1, N_2) = \int_{-\infty}^{+\infty} |\mu_{N_1}(u) - \mu_{N_2}(u)| du$$

El propósito de Yager es muy específico y diferente del que aquí nos interesa. Yager pretende ordenar valores de verdad borrosos vistos como conjuntos borrosos del intervalo unidad, en términos de su proximidad (según la distancia de Hamming) al valor borroso de verdad estándar “verdad”  $\mu_{\text{verdad}}(u) = u \quad \forall u \in [0,1]$ . La distancia de Hamming se puede ver como un índice probabilista de comparación entre números borrosos ([Dubois, Prade, 1983]).

Baas y Kwakernaak ([Baas, Kwakernaak, 1977]) propusieron extender la desigualdad  $u_1 \geq u_2$  a números borrosos como:

$$(B.5) \quad d(N_1 \geq N_2) = \sup_{u_1 \geq u_2} \min(\mu_{N_1}(u_1), \mu_{N_2}(u_2))$$

Se puede ver que este índice es el presentado anteriormente como  $\Pi_{N_1}([N_2, +\infty))$ .

Los autores extienden (B.5) a  $n$  números borrosos. El problema de este índice es que en ciertos casos puede ser insuficiente en la discriminación. Este es el índice utilizado en [Buckley, 1985] para discriminar entre elementos descritos por expertos mediante diversos criterios.

Baas y Kwakernaak ([Baas, Kwakernaak, 1977]) también definieron un índice de preferencia  $P_i$  asociado con  $N_i$ , pero al final  $P_i$  debe ser ordenado con lo que el problema queda sin resolver.

Watson et al. ([Watson et al., 1979]) analizaron el problema de ordenación como modelado de la implicación:

$$O_1 \text{ es } N_1 \text{ y } O_2 \text{ es } N_2 \quad \Rightarrow \quad O_1 \text{ se prefiere frente a } O_2$$

La función de pertenencia del primer término es:

$$\mu_x(u_1, u_2) = \min(\mu_{N_1}(u_1), \mu_{N_2}(u_2))$$

y la función de pertenencia del segundo término es:

$$\mu_y(u_1, u_2) = \begin{cases} 1 & \text{si } u_1 > u_2 \\ 0 & \text{si } u_1 \leq u_2 \end{cases}$$

La implicación propuesta es  $a \rightarrow b = \max(1-a, b)$  y el índice  $d'$  se define como:

$$\begin{aligned} d'(N_1 > N_2) &= \inf_{u_1, u_2} \max(1 - \mu_X(u_1, u_2), \mu_Y(u_1, u_2)) \\ &= \inf_{u_1 \leq u_2} [1 - \min(\mu_{N_1}(u_1), \mu_{N_2}(u_2))] \\ &= 1 - \Pi_{N_2}([N_1, +\infty)) \end{aligned}$$

Por tanto se puede ver que  $d'$  no es un índice nuevo sino otro cálculo de la aproximación propuesta por Baas y Kwakernaak. El índice de ordenación de Baas y Kwakernaak es el más natural según los principios básicos de la teoría de la posibilidad, a pesar de su falta de poder de discriminación en algunos casos. Además, en [Buckley, 1985] se presentan las interesantes propiedades que posee.

Baldwin y Guild modificaron el índice  $d$  de Baas y Kwakernaak con el fin de mejorar su poder de discriminación, sustituyendo  $u_1 \geq u_2$  por una relación borrosa de ordenación  $R$ .

$$d_R(N_1 \geq N_2) = \sup_{u_1, u_2} \min(\mu_{N_1}(u_1), \mu_{N_2}(u_2), \mu_R(u_1, u_2))$$

De manera similar, se podría suavizar la relación  $Y$  en el índice de Watson.

Tsukamoto et al. proponen índices muy relacionados con los propuestos por Dubois y Prade, partiendo de que un intervalo borroso se puede ver como la intersección de dos conjuntos borrosos en forma de  $S$ :  $M = M_L \cap M_R$ . Sin embargo, Tsukamoto et al. no interpretan los resultados dentro de la teoría de la posibilidad ni demuestran que sea un conjunto completo de índices (ver [Dubois, Prade, 1983]).

Liou y Wang ([Liou, Wang, 1992]) proponen un método de ordenación de varios números borrosos basado en el valor integral.

En Bortolan y Degani ([Bortolan, Degani, 1985]) hacen una revisión interesante acerca de los métodos de ordenación de conjuntos borrosos, llegando a la conclusión de que en casos en los que la decisión de ordenación depende del contexto, los mejores índices son los de Dubois y Prade ya que estos índices no fuerzan ningún orden específico sino que describen la situación de forma clara, permitiendo al experto tomar la decisión que considere adecuada.

Debido a lo citado anteriormente, en esta tesis se trabaja con los índices de Dubois y Prade, ya que no interesa tanto la decisión de una ordenación determinada, sino conservar toda la información sobre el orden relativo entre dos conjuntos borrosos.

### B.3. Posibilidad y necesidad de que una fecha esté antes que otra

Esta información sobre el orden relativo entre dos números es completa si se expresa mediante los cuatro índices de Dubois y Prade ([Dubois, Prade, 1983]). Sin embargo, nos hemos centrado en los dos más importantes, que son suficientes para la ordenación de fechas, tal y como se afirma en [Dubois, Prade, 1989b].

Para ello se presentan posibles formas de calcular los índices considerados  $\Pi(A \leq B)$  y  $N(A \leq B)$ , al comparar conjuntos convexos y al comparar conjuntos no convexos y se llega a la conclusión de que cuando se comparan conjuntos borrosos no convexos, los valores de posibilidad y necesidad son los mismos que los que se obtienen cuando se comparan otros conjuntos borrosos con los mismos límites de los  $\alpha$ -cortes que los anteriores.

#### B.3.1. Conjuntos convexos

En este apartado se analiza la forma de calcular la posibilidad y la necesidad de que un conjunto borrosos convexo  $A$  sea menor o igual que otro conjunto borroso convexo  $B$ .

##### 1. $\Pi(A \leq B)$

La posibilidad de que  $A$  sea menor o igual que  $B$  se define como:

$$\Pi(A \leq B) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t))$$

Se cumple que la posibilidad de  $A$  menor o igual que  $B$  es igual a la altura del conjunto borroso  $[A, B]$ , que a su vez es la intersección de  $[A, +\infty)$  con  $(-\infty, B]$ ,  $[A, +\infty) \cap (-\infty, B]$ .

El intervalo borroso  $[A, B]$  se define como:

$$\forall t \in T, \mu_{[A, B]}(t) = \sup_{s \leq t \leq s'} \min(\pi_A(s), \pi_B(s'))$$

La altura del intervalo borroso  $[A, B]$  es el supremo en  $t$  de su función de pertenencia  $\mu_{[A, B]}(t)$ :

$$\begin{aligned} \forall t \in T, \sup_t \mu_{[A,B]}(t) &= \sup_t \sup_{s \leq t \leq s'} \min(\pi_A(s), \pi_B(s')) \\ &= \sup_{s \leq s'} \min(\pi_A(s), \pi_B(s')) = \Pi(A \leq B) \end{aligned}$$

Tal y como se muestra a continuación, este mismo resultado se puede obtener calculando la altura del conjunto borroso  $[A, +\infty) \cap B$  ([Dubois, Prade, 1983]):

Recordemos la definición del conjunto borroso  $[A, +\infty)$ :

$$\mu_{[A, +\infty)}(t) = \sup_{s \leq t} \pi_A(s)$$

Aplicando la t-norma *min* para el cálculo de la intersección  $[A, +\infty) \cap B$  se tiene:

$$\mu_{[A, +\infty) \cap B}(y) = \min(\mu_{[A, +\infty)}(y), \pi_B(y)) = \min\left(\sup_{x \leq y} \pi_A(x), \pi_B(y)\right)$$

La altura del conjunto borroso resultante  $[A, +\infty) \cap B$  se puede calcular como el supremo de su función de pertenencia:

$$\begin{aligned} \forall y \in T, \sup_y \mu_{[A, +\infty) \cap B}(y) &= \sup_y \min\left(\sup_{x \leq y} \pi_A(x), \pi_B(y)\right) = \\ &= \sup_{x \leq y} \min(\pi_A(x), \pi_B(y)) = \Pi(A \leq B) \end{aligned}$$

El resultado obtenido es por tanto el mismo que el que se obtiene al calcular la altura del intervalo borroso  $[A, B]$ .

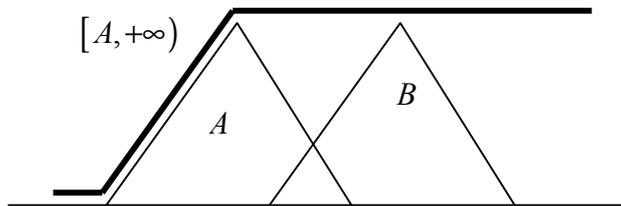


Figura B.5: Cálculo de  $\Pi(A \leq B)$  como intersección de  $[A, +\infty)$  y  $B$

En el ejemplo de la Figura B.5, se tiene que  $[A, +\infty) \cap B = B$ , con lo que  $\Pi(A \leq B) = 1$

2.  $N(A \leq B)$

La necesidad de que un conjunto borroso  $A$  sea menor o igual que otro conjunto borroso  $B$  se puede definir como:

$$N(A \leq B) = 1 - \sup_{s>t} \min(\pi_A(s), \pi_B(t)) = 1 - \Pi(A > B)$$

Tal y como se afirma en [Dubois, Prade, 1989b], si  $\pi_A(s)$ ,  $\pi_B(t)$  son continuas, entonces se tiene:

$$\Pi(A \leq B) = \Pi(A < B)$$

$$N(A \leq B) = N(A < B)$$

Por tanto,  $N(A \leq B) = 1 - \Pi(A > B) = 1 - \Pi(A \geq B)$ . El valor de  $\Pi(A > B)$ , o lo que es lo mismo de  $\Pi(A \geq B)$  se puede calcular como la altura del conjunto borroso  $(-\infty, A] \cap [B, +\infty)$ , o bien como la altura del conjunto borroso  $(-\infty, A] \cap B$  ([Dubois, Prade, 1983]).

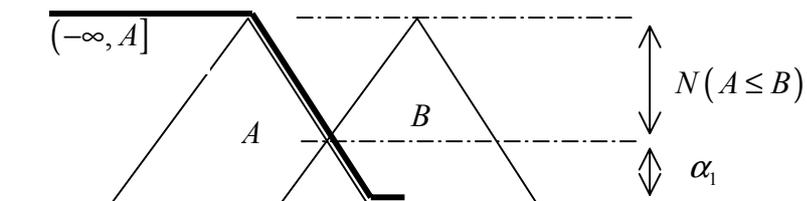


Figura B.6: Cálculo de  $N(A \leq B)$  a partir de  $(-\infty, A]$  y B

En el ejemplo de la Figura B.6, la altura de  $(-\infty, A] \cap B$  es  $\alpha_1$ , con lo que  $\Pi(A > B) = \alpha_1$ , y  $N(A \leq B) = 1 - \Pi(A > B) = 1 - \alpha_1$ . También se puede calcular el valor de  $N(A \leq B)$  como la altura del conjunto borroso  $]A, B[$  ([Dubois, Prade, 1989b]) de forma que:

$$]A, B[ = ]A, +\infty) \cap (-\infty, B[ = \overline{(-\infty, A]} \cap \overline{[B, +\infty)} = \overline{((-\infty, A] \cup [B, +\infty))}$$

La altura del conjunto borroso  $]A, B[$  es:

$$\begin{aligned} \sup ]A, B[ &= \sup ]A, +\infty) \cap (-\infty, B[ = \inf ]A, +\infty) \cup (-\infty, B[ = \\ &= 1 - \sup \{ \overline{]A, +\infty)} \cap \overline{(-\infty, B[} \} = 1 - \sup \{ (-\infty, A] \cap [B, +\infty) \} \end{aligned}$$

Como se puede observar, se llega de esta manera al mismo resultado que antes.

En realidad, para que se cumpla la igualdad  $\Pi(A \leq B) = \Pi(A < B)$  es suficiente con que una de las dos distribuciones  $\pi_A(s)$ ,  $\pi_B(t)$  sea continua (ver Figura B.7).

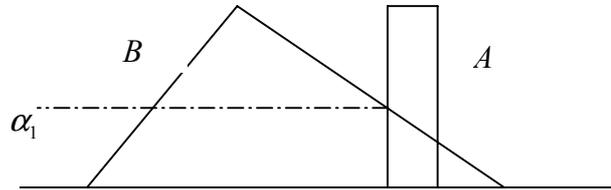


Figura B.7: Comparación de dos números borrosos, siendo B continuo

Si se verifica que una de las dos distribuciones de posibilidad es continua, se tiene:

$$\Pi(A \leq B) = \sup_{s \leq t} \min(\pi_A(s), \pi_B(t)) = \alpha_1 = \sup_{s < t} \min(\pi_A(s), \pi_B(t)) = \Pi(A < B)$$

### B.3.2. Conjuntos no convexos

En este apartado se analiza la comparación entre dos conjuntos borrosos no convexos y posibles formas de calcular las medidas de posibilidad y de necesidad. Se concluye que los valores de posibilidad y de necesidad de la comparación de dos conjuntos borrosos no convexos son los mismos que se obtienen si en su lugar se comparan otros conjuntos borrosos con los mismos límites de los  $\alpha$ -cortes que los primeros.

Con la misma definición de  $]A, +\infty)$  y  $(-\infty, A]$  los conjuntos obtenidos dada una fecha borrosa no convexa  $A$  son:

$$\mu_{]A, +\infty)}(t) = \sup_{s \leq t} \pi_A(s)$$

$$\mu_{(-\infty, A]}(t) = \sup_{s \geq t} \pi_A(s)$$

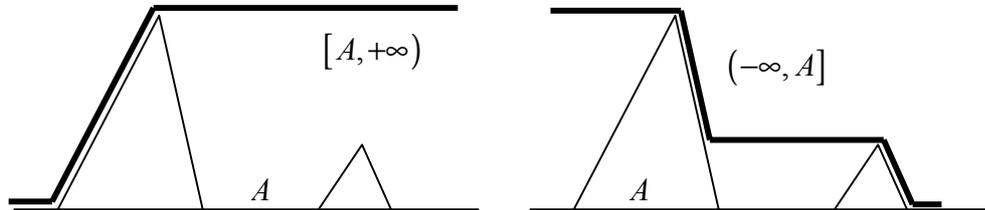


Figura B.8: Representación de  $[A, +\infty)$  y  $(-\infty, A]$  siendo  $A$  no convexo

Las funciones de pertenencia de los conjuntos borrosos  $]A, +\infty)$  y  $(-\infty, A[$  son:

$$\mu_{]A, +\infty)}(t) = \inf_{s \geq t} (1 - \pi_A(s))$$

$$\mu_{(-\infty, A[}(t) = \inf_{s \leq t} (1 - \pi_A(s))$$

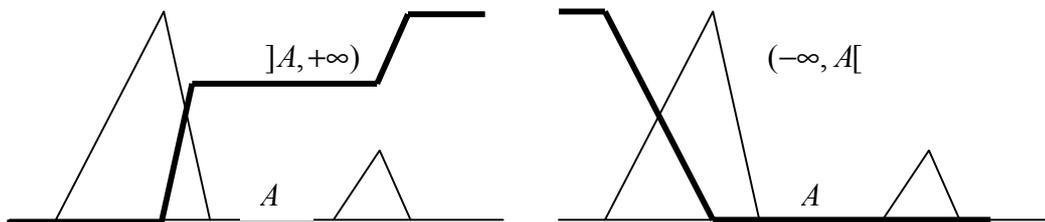


Figura B.9: Representación de  $]A, +\infty)$  y  $(-\infty, A[$  siendo  $A$  no convexo

Se observa por tanto que los conjuntos borrosos  $[A, +\infty)$ ,  $(-\infty, A]$ ,  $]A, +\infty)$  y  $(-\infty, A[$  asociados al conjunto no convexo  $A$  coinciden con los conjuntos  $[A^*, +\infty)$ ,  $(-\infty, A^*]$ ,  $]A^*, +\infty)$  y  $(-\infty, A^*]$  asociados al conjunto convexo  $A^*$  definido como la aproximación válida convexa de  $A$  (ver Figura B.10)

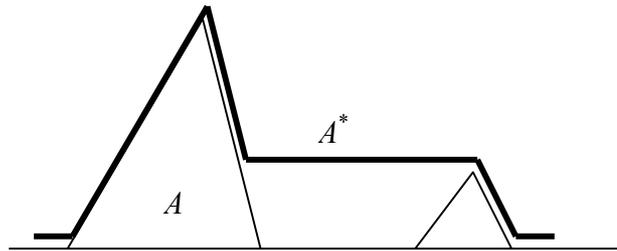


Figura B.10: Conjunto borroso no convexo  $A$  y su aproximación convexa  $A^*$

Por lo tanto se tiene:

$$\Pi(A \leq B) = \text{altura}([A, +\infty) \cap B) = \text{altura}([A^*, +\infty) \cap B) = \Pi(A^* \leq B)$$

La misma conclusión se puede obtener para  $\Pi(A \geq B)$ , con lo que el resultado de comparar un conjunto borroso no convexo  $A$  con otro número cualquiera coincide en sus valores de posibilidad y necesidad de estar antes y después con los valores obtenidos al compararlo con su aproximación convexa  $A^*$  (ver Figura B.10) con el mismo número.

Nótese que cualquier otro conjunto borroso  $A^{**}$  (y no sólo la aproximación convexa) que tenga los mismos límites de los  $\alpha$ -cortes que  $A$  proporciona los mismos resultados al compararlo con otro número (ver Figura B.11, aproximación no convexa  $A^{**}$ ).

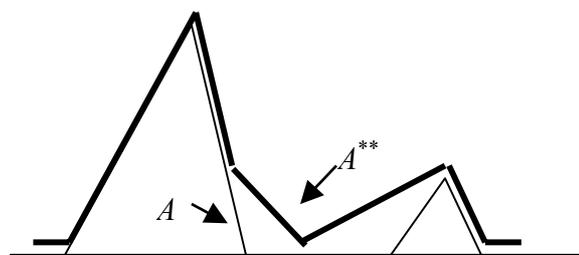


Figura B.11: Conjunto borroso no convexo  $A$  y aproximación no convexa  $A^{**}$  con los mismos límites de los  $\alpha$ -cortes

Esta conclusión es muy importante para los objetivos de esta tesis porque permite afirmar que cuando se compara el tiempo de finalización borroso de una tarea con su plazo borroso, si el tiempo de finalización borroso obtenido mediante el principio de extensión es un conjunto borroso no convexo, se pueden aplicar otros métodos de cálculo del tiempo de finalización más rápidos, siempre que la aproximación borrosa

obtenida conserve los límites de los  $\alpha$ -cortes. Por otro lado, se tiene que la comparación entre el tiempo de finalización borroso y el plazo modelado como un conjunto borroso no convexo proporciona los mismos resultados la comparación con una aproximación del plazo con los mismos límites de los  $\alpha$ -cortes.



## **C. ANEXO: Probabilidad y Posibilidad**

Uno de los puntos más controvertidos en el modelado de la incertidumbre es la relación entre la teoría de la probabilidad y las distribuciones de posibilidad asociadas a conjuntos borrosos [Dubois, Prade, 1993b]. Aunque los conjuntos borrosos y las medidas de probabilidad son diferentes, existen puntos de unión que los relacionan. De forma especial, la teoría de la posibilidad sirve de enlace entre ambas. Es posible encontrar interpretaciones probabilistas de los conjuntos borrosos y otras interpretaciones no probabilistas. En la literatura se pueden encontrar diferentes transformaciones entre probabilidad y posibilidad, tal y como se verá más tarde [Dubois et al., 1993], [Lasserre et al., 1997].

A continuación se va a analizar la relación entre probabilidad y conjuntos borrosos, a presentar las interpretaciones probabilistas de las funciones de pertenencia, las interpretaciones no probabilistas, y las transformaciones entre probabilidad y posibilidad.

## C.1. Probabilidad y posibilidad

### *Función de pertenencia y medida de probabilidad*

Un conjunto borroso  $F$  en un universo  $U$  se define mediante una función de pertenencia  $\mu_F : U \rightarrow [0,1]$  [Zadeh, 1965] donde  $\mu_F(u)$  es el grado de pertenencia de un elemento  $u$  en  $F$ . Por sencillez se hablará sólo de universos finitos aunque se puede hacer extensivo a universos no finitos.

Por otro lado, una medida de probabilidad  $P$  es una función  $2^U \rightarrow [0,1]$  que asigna un número  $P(A)$  a cada subconjunto de  $U$  y que satisface los axiomas de Kolmogorov:

$$P(U)=1$$

$$P(\emptyset)=0$$

$$\text{Si } A \cap B = \emptyset, P(A \cup B) = P(A) + P(B)$$

$P(A)$  es la probabilidad de que una variable  $x$  que toma un solo valor conocido de forma incierta esté en un conjunto fijo y bien conocido  $A$ . Un posible error consiste en confundir  $P(A)$  con un grado de pertenencia. Pero hay que tener claro que  $\mu_F$  es una función de elementos y  $P(A)$  es una función de conjuntos. En  $\mu_F(u)$ , el elemento  $u$  es fijo y conocido y el conjunto se conoce de forma incierta mientras que en  $P(A)$ , el conjunto  $A$  está bien definido, pero el valor de la variable subyacente  $x$  es

desconocida y varía.  $\mu_F(u)$  tiene más relación con  $P(\{u\})=p(u)$ , aunque  $p(u)$  cumple además la condición  $\sum_{u \in U} p(u) = 1$ .

Desde el punto de vista matemático, el dominio de la aplicación  $P$  es el álgebra de Boole  $2^U$  mientras que el conjunto de conjuntos borrosos es  $[0,1]^U$  y nunca puede ser un álgebra de Boole. Este tiene necesariamente una estructura más débil, dependiendo de la elección de operaciones entre conjuntos borrosos.

Otra posible fuente de confusión está en las definiciones de unión de conjuntos borrosos y de la propiedad de descomponibilidad de una medida de probabilidad. La unión de conjuntos borrosos se define como

$$(C.1) \mu_{F \cup G}(u) = S(\mu_F(u), \mu_G(u))$$

y la propiedad de descomponibilidad de la medida de probabilidad respecto de dos eventos dados  $A$  y  $B$  se define como

$$(C.2) P(A \cup B) = S'(P(A), P(B))$$

En primer lugar (C.2) no se cumple para todo  $A$  y  $B$ , mientras que (C.1) se cumple para todo  $u \in U$ . La confusión puede venir de que tanto  $S$  como  $S'$  se pueden elegir de entre las co-normas triangulares, tales como  $\max(a,b)$ ,  $a+b-ab$  o  $\min(1,a+b)$ . Pero el hecho de utilizar  $\max$  o  $\min$  no quiere decir que se esté trabajando con conjuntos borrosos. De forma similar, el hecho de que a veces  $P(A \cup B) = \max(P(A), P(B))$  (por ejemplo, si  $A \subseteq B$ ) no quiere decir que los conjuntos borrosos sean probabilidades.

Otro punto de vista es comenzar con un conjunto de sentencias  $\mathcal{S}$  y considerar diversas formas de asignar un número a cada sentencia  $\alpha$  de  $\mathcal{S}$ , con  $v(\alpha) \in [0,1]$ . La cuestión es entonces analizar la estructura de  $\mathcal{S}$  con las conectivas AND, OR, NOT, etc. Gaines demostró que se pueden tener dos representaciones de  $\mathcal{S}$ : como  $2^U$  y  $v(\alpha) = P([\alpha])$  donde  $[\alpha]$  es el subconjunto de mundos en  $U$  donde  $\alpha$  es cierto; y como  $\mathcal{S} = [0,1]^U$  y  $v(\alpha) = \mu_{[\alpha]}(u)$  donde  $u$  es fijo en  $U$  y  $\alpha$  es una sentencia borrosa. Ambas representaciones se pueden situar en el grupo de lógicas con pesos. Sin embargo, esto no debe llevar a confundir los grados de probabilidad con los grados de pertenencia [Dubois, Prade, 1993b].

### *Teoría de la posibilidad*

Relacionada con los conjuntos borrosos se encuentra la teoría de la posibilidad que tiene más conexión con la teoría de la probabilidad que los conjuntos borrosos debido a que propone una función de conjuntos que cuantifica la incertidumbre de eventos. Una medida de posibilidad en un conjunto finito  $U$  es una aplicación de  $2^U$  a  $[0,1]$ , tal que  $\Pi(\emptyset) = 0$

$$(C.3) \Pi(A \cup B) = \max(\Pi(A), \Pi(B))$$

Se añade la condición  $\Pi(U)=1$  para tener medidas de posibilidad normalizadas. Estas medidas vienen caracterizadas completamente por una distribución de posibilidad  $\pi: U \rightarrow [0,1]$  (tal que  $\pi(u)=1$  para algún  $u \in U$ ), ya que  $\Pi(A) = \max\{\pi(u), u \in A\}$ . Zadeh [Zadeh, 1978] afirmó que la distribución de posibilidad  $\pi$  viene determinada por la función de pertenencia  $\mu_F$  de un conjunto borroso  $F$ . Sin embargo esto no quiere decir que ambos conceptos sean lo mismo. La ecuación de Zadeh  $\pi_X(u) = \mu_F(u)$  es en realidad  $\pi(x=u | F) = \mu_F(u)$ .  $\pi_X(u)$  estima la posibilidad de que la variable  $x$  sea igual a  $u$ , teniendo un conocimiento incompleto del tipo “ $x$  es  $F$ ”.  $\mu_F(u)$  estima el grado de compatibilidad de la información precisa  $x=u$  con la afirmación a evaluar “ $x$  es  $F$ ”.

A pesar de (C.3) la teoría de la posibilidad no es composicional. Las medidas de posibilidad se pueden descomponer únicamente respecto a la unión, mientras que las medidas de necesidad, definidas como  $N(A)=1-\Pi(\bar{A})$ , se pueden descomponer únicamente respecto a la intersección  $N(A \cap B) = \min(N(A), N(B))$ . De hecho, la teoría de la posibilidad es capaz de modelar la total ignorancia acerca de un evento  $A$  teniendo  $\Pi(A) = \Pi(\bar{A}) = 1$  (y  $N(A) = N(\bar{A}) = 0$ ), con  $\Pi(A \cap \bar{A})=0$  y  $N(A \cup \bar{A})=1$ . En cambio, la teoría de la probabilidad no puede modelar la ignorancia. Por esta razón ambas teorías, que modelan distintos estados del conocimiento, son complementarias.

## C.2. Interpretaciones probabilistas de las distribuciones de posibilidad

Veamos ahora la relación entre conjuntos borrosos vistos como distribuciones de posibilidad y las medidas de probabilidad.

### C.2.1. Cotas de probabilidad

El primer punto de vista es interpretar una medida de posibilidad como una envolvente superior de la probabilidad. Supongamos que un determinado estado de incertidumbre debería expresarse mediante una medida de probabilidad, y que no es posible describir de forma precisa esa probabilidad sino únicamente un subconjunto  $\mathcal{P}$  de medidas de probabilidad que contiene la probabilidad real. Entonces sólo se dispone de cotas inferiores y superiores de probabilidad:

$$(C.4) P_*(A) = \inf\{ P(A) \mid P \in \mathcal{P} \}$$

$$(C.5) P^*(A) = \sup\{ P(A) \mid P \in \mathcal{P} \}$$

Las cotas inferiores (resp. superiores) de probabilidad se pueden interpretar como los precios aceptables para apostar a favor (resp. en contra) de  $A$  ([Walley, 1996]). Supongamos que hay que decidir si apostar a favor o en contra de  $A$ , a un precio de apuesta  $x$ , con lo que la ventaja de  $A$  de  $x$  a  $1-x$ . La apuesta será a favor de  $A$  si  $x$  es menor que  $P_*(A)$ , en contra de  $A$  si  $x$  es mayor que  $P^*(A)$ , y estará sin determinar si  $x$  se encuentra entre  $P_*(A)$  y  $P^*(A)$ , donde  $A$  es un evento o un estado del espacio de posibles estados  $\Omega$ .

Las cotas inferiores y superiores de probabilidad tienen las siguientes propiedades básicas:

$$P_*(\Omega) = P^*(\Omega) = 1$$

$$P^*(A) = 1 - P_*(\bar{A})$$

$$P_*(A) + P_*(B) \leq P_*(A \cup B) \leq P_*(A) + P^*(B) \leq P^*(A \cup B) \leq P^*(A) + P^*(B)$$

siendo  $A$  y  $B$  disjuntos.

Las ecuaciones (C.4) y (C.5) definen funciones de conjuntos más generales que las medidas de credibilidad y plausibilidad. Las medidas de credibilidad se pueden interpretar como cotas inferiores de probabilidad, esto es  $P_*(A)$  sería el supremo de los precios de apuesta a favor de  $A$  que serían razonables. Por tanto, las medidas de posibilidad son un caso especial de cotas superiores de probabilidad y, debido a que  $P_*(\bar{A}) = 1 - P^*(A)$ , las medidas de necesidad son un caso especial de cotas inferiores de probabilidad. De hecho, la teoría de la posibilidad propone el sistema más sencillo de cotas superiores e inferiores de probabilidad ([Dubois, Prade, 1989a], [Dubois, Prade, 1996]).

Es posible obtener las cotas superior e inferior del valor esperado (o previsión) a partir de las cotas superior e inferior de probabilidad, aplicando la “extensión natural” descrita en [Walley, 1996]. La cota del valor esperado es una función de  $f$  a los números reales, donde  $f$  se puede interpretar como una jugada con cierto riesgo o con una recompensa incierta y es una función del espacio de estados posibles  $\Omega$  a los números reales (valor de utilidad). La cota inferior del valor esperado o previsión se puede interpretar como el máximo precio que se está dispuesto a pagar por una jugada con recompensa incierta.

Por tanto, al igual que con las cotas de probabilidad, se puede utilizar el enlace entre teoría de la posibilidad y teoría de la probabilidad para extender el concepto de valor

esperado en teoría de la posibilidad, a cotas inferiores y superiores del valor esperado derivadas de (C.4) y (C.5) para obtener un intervalo  $[E_*, E^*]$ . Sea  $f$  una función de  $\Omega$  a  $\mathbb{R}$ ,  $F$  un conjunto borroso en  $\Omega$  y  $F_\alpha$  un  $\alpha$ -corte de  $F$ . Las cotas del valor esperado de  $f$  obtenidas a partir de la distribución de posibilidad  $\mu_F$  son:

$$E_*(f) = \int_0^1 \inf \{ f(\omega) \mid \omega \in F_\alpha \} d\alpha$$

$$E^*(f) = \int_0^1 \sup \{ f(\omega) \mid \omega \in F_\alpha \} d\alpha$$

Particularmente, cuando  $F$  es un intervalo borroso (conjunto borroso en la línea real, unimodal y semicontinuo hacia arriba), entonces  $E_*$  y  $E^*$  definen el valor medio de dicho intervalo borroso.

$$E_* = \int_0^1 \inf F_\alpha d\alpha$$

$$E^* = \int_0^1 \sup F_\alpha d\alpha$$

El valor esperado resume un número aleatorio mediante un número preciso, eliminando la incertidumbre. Un intervalo borroso combina imprecisión e incertidumbre. Cuando se elimina la incertidumbre, queda la imprecisión ([Dubois, Prade, 1989a]).

### C.2.2. Conjuntos aleatorios y Teoría de la Evidencia

Esta visión de las medidas de posibilidad como cotas de las medidas de probabilidad se ha aplicado al campo de los conjuntos aleatorios y a la teoría de la evidencia.

Dado un conjunto aleatorio  $\mathcal{R}$  en  $U$ , es decir, un conjunto de subconjuntos  $A_1, A_2, \dots$

$A_n \subseteq U$ , y una asignación de probabilidad  $m$  tal que  $m(A_i) = m_i \in (0, 1]$  y que cumple

$\sum_{i=1}^n m_i = 1$ , entonces se puede definir una función de pertenencia a partir de  $\mathcal{R}$  de la siguiente forma:

$$(C.6) \mu_F(u) = \sum_{u \in A_i} m_i$$

Definiendo  $\{ (A_i, m_i), i=1, n \}$  como un conjunto aleatorio  $F$  con realizaciones  $A_i$ , el grado de pertenencia  $\mu_F(u)$  se interpreta como la probabilidad de  $u \in F$ . Si se interpretan los subconjuntos  $A_i$  como observaciones imprecisas, se pueden definir las siguientes funciones de conjuntos clásicos dentro de la teoría de la evidencia:

$$(C.7) \text{Bel}(A) = \sum_{A_i \subseteq A} m_i$$

$$(C.8) \text{Pl}(A) = \sum_{A_i \cap A \neq \emptyset} m_i$$

denominadas funciones de credibilidad y de plausibilidad respectivamente.

Tal y como se ha dicho antes, estas funciones son cotas inferiores y superiores de probabilidad. La función de plausibilidad es una medida de posibilidad y la función de credibilidad es una medida de necesidad si los subconjuntos  $A_i$  están anidados. Por tanto, las medidas de posibilidad y necesidad son un caso especial de funciones de plausibilidad y credibilidad, que a su vez son casos particulares de cotas superiores e inferiores de probabilidad ([Klir, Folger, 1988], [Dubois, Prade, 1986a], [Dubois, Prade, 1989a], [Dubois, Prade, 1993b]).

Se puede establecer entonces la relación entre el conjunto aleatorio y la función de pertenencia de un conjunto borroso de la siguiente forma. La función de pertenencia de un conjunto borroso se puede definir a partir de sus  $\alpha$ -cortes  $F_\alpha$  como

$$\mu_F(x) = \sup \{ \alpha \in (0,1] \mid x \in F_\alpha \}$$

con  $F_\alpha = \{ x \mid \mu_F(x) \geq \alpha, \alpha \in (0,1] \}$  (notar que son conjuntos clásicos)

Esta función de pertenencia está asociada a una distribución de posibilidad  $\pi_F(x)$  y se corresponde a la distribución de plausibilidad (C.8) cuando los subconjuntos  $A_i$  están anidados ([Goodman, 1980], [Goodman, 1984]):

$$\mu_F(x) = \pi_F(x) = \text{Pl}(\{x\}) = \sum_{A_i \cap \{x\} \neq \emptyset} m_i$$

siendo en este caso los  $\alpha$ -cortes dichos subconjuntos  $A_i$  del conjunto aleatorio asociado.

La función de pertenencia se puede expresar también en forma de integral como ([Dubois, Prade, 1989a]):

$$(C.9) \mu_F(x) = \int_0^1 \mu_{F_\alpha}(x) d\alpha$$

donde  $\mu_{F_\alpha}(x) = 1$  si  $x \in F_\alpha$  y 0 si no. Por tanto, la función de pertenencia se puede ver como un conjunto aleatorio uniformemente distribuido, que consiste en una medida de Lebesgue en  $[0,1]$  y una aplicación  $\alpha \rightarrow F_\alpha$ . Si  $F_\alpha$  es finito, entonces el conjunto borroso  $F$  es finito y discreto y se pueden definir los valores de pertenencia positivos de forma ordenada de manera que  $\alpha_1 > \alpha_2 > \dots > \alpha_n$ ,  $\alpha_{n+1} = 0$ , y  $m_i = \alpha_i - \alpha_{i+1}$ . Entonces (C.9) se expresa como

$$\mu_F(x) = \sum_{x \in F_{\alpha_i}} m_i$$

Se tiene que  $\sum_{i=1}^n m_i = \alpha_1 = 1$  ya que  $\exists x \mid \mu_F(x)=1$ . Así, los conjuntos anidados  $F_{\alpha_i}$  junto con los pesos  $m_i$  que suman 1, caracterizan un conjunto borroso, discreto, finito, consonante y normalizado, y viceversa. Sin embargo, si los subconjuntos  $A_i$  no están anidados, la información de la función de pertenencia no es suficiente para reconstruir el conjunto aleatorio no consonante correspondiente.  $m_i$  se puede interpretar como la probabilidad de que  $F_{\alpha_i}$  sea un representante clásico de  $F$ .

Al igual que se ha descrito  $\{ (A_i, m_i), i=1, n \}$  como un conjunto aleatorio, se puede definir también como un cuerpo de evidencia donde  $A_i$  son los elementos focales y  $m_i$  es la asignación básica de probabilidad. De esta forma se enlaza la teoría de la posibilidad con la teoría de la evidencia ([Klir, Folger, 1988]).

### C.2.3. Función de verosimilitud

Hisdal ([Hisdal, 1988]) analiza el concepto de grado de pertenencia desde el punto de vista de la verosimilitud. La autora crea una teoría descriptiva (o interpretativa como ella la denomina) de conjuntos borrosos. El modelo de Hisdal TEE (“Threshold, Error, assumption of Equivalence”) para las funciones de pertenencia considera varias fuentes de información borrosa: errores de medida (de naturaleza esencialmente estadística), información incompleta, y contradicciones entre personas.

La autora afirma que para un solo individuo con información perfecta (como por ejemplo, la altura de Juan) no hay fuentes de información borrosa. En ese caso se dirá que Juan es “alto” si su altura supera un umbral bien definido.

Hisdal considera tres formas de construir la función de pertenencia:

- El experimento de etiqueta: Juan es alto, donde  $\text{alto} \in \Lambda = \{\text{muy alto, alto, bajo}\}$
- El experimento sí-no: ¿Es alto Juan?
- El experimento de grado de pertenencia: ¿En qué grado pertenece Juan al conjunto de gente alta?

Cuando se le pregunta alguien en qué grado es alto Juan, el entrevistado, que tiene una medida imprecisa de la altura de Juan (por ejemplo  $180 \pm 5$  cm), construye una curva de error para cuantificar el posible error entorno a 180 cm. Luego afirma que en el 75% de los casos la altura de Juan cae en el intervalo de cuantización del entrevistado para el concepto “alto”. Por tanto

$$\mu_{\text{alto}}(180\text{cm}) = P(\text{alto} | x=180\text{cm}) = 0.75$$

donde  $P(\text{alto} | x=180\text{cm})$  es la verosimilitud de que se asigne la etiqueta “alto” a Juan, dado que su altura es 180cm.

Se puede expresar también de la siguiente forma. Dada una población de individuos y un concepto borroso  $F$ , se le pregunta a cada individuo si un elemento dado puede ser denominado  $F$  o no. Se obtiene entonces la función de verosimilitud  $P('F'|u)$  que representa la proporción de individuos que respondieron “sí” a la pregunta. ‘ $F$ ’ es la etiqueta correspondiente. Se puede decir que:

$$(C.10) \mu_F(u) = P('F' | u), \forall u \in U$$

donde  $u$  es un valor exacto del atributo (por ejemplo  $u=180\text{cm}$ ). El grado de pertenencia es entonces la modificación de sus respuestas numéricas o clásicas a los experimentos de etiqueta o si-no modificados por su estimación de la curva de error. Esta interpretación es aplicable a la primera fuente de información borrosa: los errores de medida. [Bilgic, Turksen, 1997]

Aunque  $P('F' | u)$  expresa una probabilidad, no es una distribución de probabilidad en  $u$ , sino en el conjunto  $\Lambda$  de etiquetas.  $P('F' | u)$  está condicionado al valor de  $u$ , con lo que en estadística se le denomina distribución de verosimilitud en  $u$ . A diferencia de la probabilidad, la verosimilitud no tiene por qué sumar 1.

La clave fundamental en (C.10) es el hecho de ver la verosimilitud como una medida de incertidumbre condicional, en este caso una probabilidad. Se podrían utilizar también otras medidas de incertidumbre, como por ejemplo la medida de posibilidad ([Dubois, Prade, 1993b]), con lo que:

$$\mu_F(u) = \Pi('F' | u), \forall u \in U$$

donde se iguala la función de pertenencia de  $F$  vista como función de verosimilitud, con la posibilidad de que un elemento  $u$  se clasifique en  $F$ . Esto se justifica con la versión posibilista del teorema de Bayes:

$$\min(\pi(u|'F'), \Pi('F')) = \min(\Pi('F'|u), \pi(u))$$

supuesto que no hay información disponible a priori, esto es,  $\forall u, \pi(u)=1$ , con lo que  $\pi(u|'F') = \Pi('F'|u)$  ([Hisdal, 1988], [Dubois, Prade, 1993b]).

### C.2.4. Interpretación probabilista de las operaciones con conjuntos borrosos

La relación entre función de pertenencia y función de verosimilitud lleva a interpretar las conectivas de conjuntos borrosos dentro de la teoría de la probabilidad. Por ejemplo, la regla del mínimo de conjuntos borrosos se puede ver como:

$$(C.11) P('F' \cap 'G' | u) = \min( P('F'|u), P('G'|u) )$$

Donde 'F' y 'G' son eventos clásicos que consisten en asignar etiquetas F o G a u. Hay que tener en cuenta que los eventos clásicos 'F' y 'G' (por ejemplo, las preguntas realizadas en el experimento) están relacionados con los conjuntos borrosos F y G sólo a través del experimento aleatorio utilizado para construir las funciones de pertenencia. Por tanto hay dependencia entre los eventos clásicos, no entre los conjuntos borrosos.

Además, teniendo en cuenta el papel de las distribuciones de posibilidad como cotas superiores de probabilidad, la igualdad en (C.11) se transforma en desigualdad, donde el término de la derecha es la cota superior del término de la izquierda.

$$P('F' \cap 'G' | u) \leq \min( P('F'|u), P('G'|u) )$$

De esta forma, la regla del mínimo expresa falta de conocimiento acerca de la dependencia, más que suponer una dependencia fuerte. Este es el significado de la premisa de no interactividad en teoría de la posibilidad. En cualquier caso, la interpretación de funciones de pertenencia como verosimilitudes no puede explicar todas las conectivas de conjuntos borrosos. Sólo algunas conectivas que se encuentran por debajo del mínimo y por encima del máximo (como son las t-normas y las t-conormas) son compatibles con la interpretación como verosimilitud, debido a la monotonicidad de la inclusión  $P(A \cap B|x) \leq \min( P(A|x), P(B|x) )$  y  $P(A \cup B|x) \geq \max( P(A|x), P(B|x) )$ . Por tanto, las conectivas borrosas tipo media y sumas simétricas no se pueden expresar de esta forma.

La interpretación de funciones de pertenencia como conjuntos aleatorios consonantes proporcionan significado a las principales definiciones de uniones e intersecciones borrosas ([Dubois, Prade, 1986b], [Dubois, Prade, 1989a]). Las uniones e intersecciones borrosas se corresponden a uniones e intersecciones de conjuntos aleatorios, pero además otras conectivas como medias tienen sentido también en el campo de los conjuntos aleatorios ([Dubois, Prade, 1985]). Por tanto los conjuntos

aleatorios son más flexibles en la interpretación de conectivas borrosas que las funciones de verosimilitud.

Por último, las conectivas de conjuntos borrosos se pueden interpretar también dentro del campo de las cotas superiores e inferiores de probabilidad. De nuevo, la regla del mínimo no debería entenderse como una premisa de fuerte dependencia sino como regla de no interactividad cuando no se sabe nada acerca de la dependencia ([Dubois, Prade, 1993b]).

### C.3. Interpretaciones no probabilistas de las distribuciones de posibilidad

En este apartado se presentan dos interpretaciones no probabilistas de los conjuntos borrosos y distribuciones de posibilidad: la posibilidad como preferencia y como similitud.

#### C.3.1. Posibilidad como preferencia

No siempre tiene sentido relacionar incertidumbre con frecuencia. Algunos eventos pueden ser raros, no repetibles, o simplemente puede no haber datos estadísticos disponibles. Sin embargo esto no evita que un individuo considere unos eventos más posibles, probables o ciertos que otros. La forma más primitiva de comparar eventos es definir un preorden completo en  $2^U$  que exprese comparación de posibilidades, probabilidades o credibilidades.

En posibilidad, el preorden completo  $\geq_{\Pi}$  tal que  $A \geq_{\Pi} B$  significa que A es al menos tan posible como B, debe cumplir:

$$\forall C, A \geq_{\Pi} B \Rightarrow A \cup C \geq_{\Pi} B \cup C.$$

Y la relación dual de certidumbre  $\geq_C$  cumple que  $A \geq_C B \Leftrightarrow \bar{B} \geq_{\Pi} \bar{A}$

Una relación de comparación de posibilidades en  $2^U$  describiendo la posición de una variable  $x$  induce un preorden completo en  $U$  que se puede ver como una relación de preferencia en los valores posibles de  $x$  ([Dubois, Prade, 1993b]).

### C.3.2. Posibilidad como similitud

Existe una corriente dentro de la teoría de conjuntos borrosos según la cual el grado de pertenencia  $\mu_F(u)$  refleja la similitud entre  $u$  y un prototipo ideal  $u_F$  de  $F$  (para el cual  $\mu_F(u_F)=1$ ). Esta interpretación de pertenencia parcial está relacionada con la relación de distancia y no con la probabilidad. Si a una variable  $x$  se le asigna una distribución de posibilidad  $\pi=\mu_F$ ,  $x=u$  es más posible cuanto más se parezca  $u$  a  $u_F$ , o cuanto más cerca esté de  $u_F$  ([Dubois, Prade, 1993b], [Bilgic, Turksen, 1997]).

Dentro de esta interpretación, Zimmermann propone un modelo de función de pertenencia definido en la línea real, que es una función de la distancia entre un objeto y un objeto ideal.

Pero cuando se habla de distancia se supone que existe un espacio métrico, con lo que es necesario las propiedades del mismo y su correspondencia con las leyes psicofísicas. Zwick, Carlstein y Budescu analizan diferentes medidas de similitud entre conceptos borrosos. Estudian, tanto medidas de similitud definidas en espacios métricos como medidas definidas desde el punto de vista de teoría de conjuntos y las comparan llegando a la conclusión de que aunque todas funcionan bien para expresar los conceptos de similitud y no similitud, las de espacios métricos distinguen mejor entre grados de similitud ([Bilgic, Turksen, 1997]). Esto es lo esperado ya que si lo que se busca es el grado en que dos conceptos son similares hacen falta medidas métricas, que contienen más información.

Ruspini también analiza la semántica de la similitud en teoría de conjuntos borrosos, construyendo el concepto de similitud  $R$  dentro de una lógica modal. Dadas dos proposiciones clásicas  $\alpha$  y  $\beta$  el grado en que  $\alpha$  implica  $\beta$  es la mínima cantidad de “estiramiento” que hace falta para incluir el conjunto de mundos en los que  $\alpha$  es cierto en el entorno de los mundos en los que  $\beta$  es cierto, donde la verdad o falsedad de las proposiciones  $\alpha, \beta...$  se refiere a  $U$  como un conjunto de mundos posibles. De manera más formal, sea  $U$  el conjunto de mundos posibles, si  $u \models \alpha$  significa que  $\alpha$  es cierto en  $u \in U$ , entonces

$$I(\beta|\alpha) = \inf_{u \models \alpha} \sup_{u' \models \beta} \mu_R(u, u')$$

Donde  $R$  es una relación borrosa de proximidad, que expresa indiscernibilidad en  $U$ .

## C.4. Transformaciones Posibilidad-Probabilidad

La transformación de medidas de posibilidad en medidas de probabilidad o al contrario puede ser de utilidad en problemas donde se manejan datos imprecisos e

inciertos heterogéneos, tales como datos subjetivos, lingüísticos y estadísticos ([Dubois et al., 1993]).

La pregunta básica es si la relación entre las representaciones posibilista y probabilista de la incertidumbre es una simple traducción entre lenguas, de las cuales ninguna es más fuerte o débil que la otra. En este supuesto se basan las transformaciones que respetan el principio de invarianza de la información y la incertidumbre: la incertidumbre en la distribución de probabilidad ha de ser igual a la incertidumbre en la distribución de posibilidad ([Geer, Klir, 1992]).

El otro punto de vista a la hora de realizar transformaciones entre probabilidad y posibilidad es considerar que ambas no tienen el mismo poder descriptivo. Aparte del hecho de que hay estados de información que la probabilidad puede describir pero la posibilidad no (como la aleatoriedad total) y al contrario (una distribución de probabilidad no puede expresar ignorancia), se puede decir que la representación posibilista es más débil, puede representar estados de información más débiles, porque maneja de forma explícita la imprecisión (esto es, conocimiento incompleto) y porque las medidas de posibilidad se basan en una estructura de ordenación más que en una estructura aditiva. Puede resultar útil transformar una medida de probabilidad en una medida de posibilidad cuando hay otras fuentes de información más débiles o cuando el cálculo con posibilidades es más sencillo que el cálculo con probabilidades. Puede resultar útil transformar una medida de posibilidad en una medida de probabilidad en el campo de la toma de decisiones. Sin embargo, no hay que olvidar que las representaciones probabilistas y posibilistas no son dos representaciones equivalentes de incertidumbre, por lo que no tiene por qué haber simetría entre dos procedimientos de transformación mutua.

Dubois, Prade et al. ([Dubois et al., 1993]) sugieren que las transformaciones deberían estar guiadas por dos principios de información diferentes: el principio de razón insuficiente o “insufficient reason” de posibilidad a probabilidad y el principio de máxima especificidad de probabilidad a posibilidad. El primer principio busca una medida de probabilidad que preserve la incertidumbre en la elección de las salidas. El segundo principio busca la distribución de posibilidad más informativa, bajo las restricciones dictadas por el principio de consistencia entre posibilidad y probabilidad.

#### **C.4.1. Transformaciones basadas en los principios de máxima especificidad y razón insuficiente**

##### *Principios básicos de las transformaciones*

En primer lugar hay que tener en cuenta las diferencias estructurales entre las medidas de posibilidad y las de probabilidad. La característica principal de las

representaciones probabilistas de incertidumbre es la aditividad. Las medidas de probabilidad aprovechan la estructura algebraica del intervalo unidad. Sin embargo, las distribuciones de probabilidad uniformes de conjuntos finitos capturan únicamente la idea de falta de decisión en el número de dichas salidas; y la probabilidad de una salida depende del número de salidas.

Por el contrario, las medidas de posibilidad sólo utilizan el hecho de que el intervalo unidad es una ordenación total. Esto lleva a un modelo cuasi-cualitativo de incertidumbre que es menos expresivo que la probabilidad, pero también requiere menos información. Es capaz de capturar perfectamente la ignorancia, haciendo que la posibilidad de cualquier evento sea 1, excepto para los valores que siempre son imposibles; ninguna medida de probabilidad es capaz de capturar la ignorancia. Por tanto la teoría de la probabilidad ofrece un buen modelo cuantitativo para expresar aleatoriedad e indecisión, mientras que la teoría de la posibilidad ofrece un buen modelo cualitativo de ignorancia parcial. Posibilidad y probabilidad no modelan las mismas facetas de la incertidumbre.

Como ya se ha dicho anteriormente, una medida de posibilidad se puede ver como una función de probabilidad superior (“upper probability”) ([Dubois, Prade, 1993b]) y este hecho puede permitir transformar una distribución de posibilidad en una distribución de probabilidad y al contrario. Se deduce que se pierde información en el paso de una representación probabilista a una representación posibilista, debido a que se pasa de probabilidades de puntos a probabilidades de intervalos. La transformación inversa añade información al conocimiento incompleto posibilista; esta información adicional es arbitraria.

Formalmente, una medida de posibilidad  $\Pi$  en un conjunto  $X$  es equivalente a la familia  $\mathcal{P}(\Pi)$  de medidas de probabilidad tales que  $\mathcal{P}(\Pi) = \{P, \forall A \subseteq X, P(A) \leq \Pi(A)\}$ . Parece natural tomar de este conjunto  $\mathcal{P}(\Pi)$  el resultado de transformar la medida de posibilidad en una de probabilidad. Este es el principio de consistencia entre probabilidad y posibilidad. El problema consiste en encontrar la medida de probabilidad menos informativa posible, ya que en cualquier caso hay que añadir información. Una medida de posibilidad  $\Pi$  en un conjunto finito se caracteriza por una distribución de posibilidad  $\pi: X \rightarrow [0, 1]$  tal que  $\forall A \subseteq X, \Pi(A) = \max\{\pi(x), x \in A\}$ . La característica principal de una distribución de posibilidad es la ordenación por preferencia que induce en  $X$ . Esto es,  $\pi$  describe lo que se conoce del valor de una variable  $\mathcal{V}$ , y  $\pi(x) > \pi(x')$  significa que  $\mathcal{V}=x$  se prefiere a  $\mathcal{V}=x'$ . Por tanto, la distribución de probabilidad  $p$  obtenida a partir de  $\pi$  debería satisfacer la restricción:  $\pi(x) > \pi(x') \Leftrightarrow p(x) > p(x')$  (preservar la preferencia).

La otra transformación (probabilidad a posibilidad) busca las cotas  $[N(A), \Pi(A)]$  de  $P(A)$  para todo  $A \subseteq X$ , donde  $N(A) = 1 - \Pi(\bar{A})$  es un grado de necesidad de  $A$ , y  $\bar{A}$  es el complemento de  $A$ . Cuando  $N(A)$  sirve de cota inferior y  $\Pi(A)$  de cota superior de  $P(A)$ ,  $\pi$  se dice que domina a  $p$ . Como  $N(A) > 0 \Rightarrow \Pi(A) = 1$ , las cotas son de la forma

$[\alpha, 1]$  o  $[0, \beta]$ . Para conservar tanta información como sea posible, hay que conseguir intervalos lo más ajustado posibles. Esto quiere decir que un conjunto borroso con función de pertenencia  $\pi$  debería ser mínimo en el sentido de inclusión, de forma que  $\pi$  es máximamente específico. Un refinamiento de este requisito consiste en pedir que el conjunto borroso tenga cardinalidad mínima, esto es, que  $\sum_{x \in X} \pi(x)$  (en el caso finito) sea mínimo. Por supuesto, se debe respetar también la restricción de preservar la preferencia.

Estos son los principios básicos presentados por Dubois y Prade para las transformaciones entre posibilidad y probabilidad, aunque no son los únicos. Geer y Klir ([Geer, Klir, 1992]) sugieren transformaciones basadas en teoría de medidas, donde los principios básicos son el escalado y la invarianza en la incertidumbre.

Las siguientes transformaciones se basan en los principios presentados por Dubois y Prade.

*De posibilidad a probabilidad.*

Dada una distribución de posibilidad  $\pi$ , encontrar una distribución de probabilidad  $p$  tal que

$\forall A, P(A) \leq \Pi(A)$  (consistencia entre probabilidad y posibilidad)

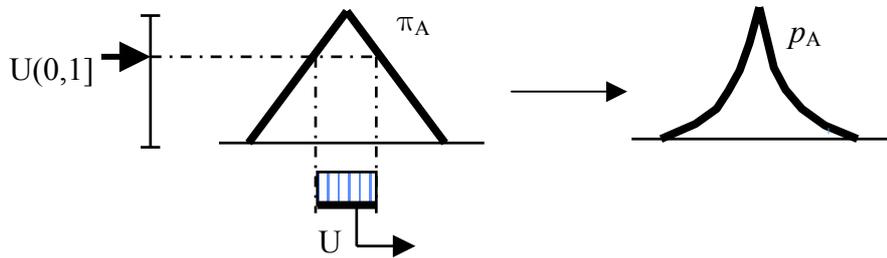
$\pi(x) > \pi(x') \Leftrightarrow p(x) > p(x')$  (preservar la preferencia)

$p$  contiene tanta incertidumbre como sea posible

Una forma natural de expresar el último requisito es utilizar el principio de “insufficient reason” o razón insuficiente. Este principio afirma que si lo único que sabemos de  $x$  es que  $x$  está en un conjunto  $A$ , entonces hay que asumir que la máxima incertidumbre acerca de  $x$  se puede describir mediante una distribución de probabilidad uniforme en  $A$ . Dada una distribución de posibilidad  $\pi$ , se aplica este principio dos veces:

- En el intervalo unidad: se selecciona un valor de  $\alpha$  de forma aleatoria (distribución uniforme) en  $(0, 1]$  y se considera  $A_\alpha = \{x \mid \pi(x) \geq \alpha\}$
- En el  $\alpha$ -corte seleccionado  $A_\alpha$ : se obtiene un valor de  $x$  de forma aleatoria (distribución uniforme) en el intervalo  $A_\alpha$ .

Este procedimiento de obtención de la distribución de probabilidad a partir de la de posibilidad se ilustra en la Figura C.1



*Figura C.1: Transformación de posibilidad a probabilidad basada en máxima especificidad y razón insuficiente.*

Este procedimiento fue sugerido por Yager ([Yager, 1982]) y analizado en más profundidad por Chanas et al ([Chanas, Nowakowski, 1988], [Chanas, Heilpern, 1989]). Si se define  $\pi$  mediante un conjunto finito de  $\alpha$ -cortes  $A_1, \dots, A_n$  correspondientes a  $\pi_1=1 > \pi_2 > \dots > \pi_n > \pi_{n+1}=0$ , la densidad de probabilidad viene dada por

$$(C.12) \quad p(x) = \sum_{i=1}^n \frac{\pi_i - \pi_{i+1}}{|A_i|} \mu_{A_i}(x), \quad \forall x$$

Esta transformación es la misma presentada en el campo de funciones de credibilidad ([Dubois, Prade, 1989a]).

En el caso continuo, sea  $X=[a,b] \subseteq \mathbb{R}$ . Supongamos que  $\pi$  es la función de pertenencia de un número borroso semicontinuo hacia arriba, unimodal y con soporte acotado; entonces la transformación se puede expresar como:

$$\forall x \in [a,b], \quad p(x) = \int_0^{\pi(x)} \frac{d\alpha}{|A_\alpha|}$$

donde  $|A_\alpha|$  es el ancho del  $\alpha$ -corte y  $p$  es la densidad de probabilidad.

*De probabilidad a posibilidad*

Se trata de encontrar una distribución de posibilidad que domine a  $p$ , que satisfaga la restricción de preservar la precedencia y la de máxima especificidad. En el caso finito no hay una solución única, puede haber tantas como el número de permutaciones de elementos de  $X$ . Este problema ha sido resuelto minimizando  $\sum_{x \in X} \pi(x)$ . Si  $X=\{x_1, x_2, \dots, x_n\}$ ,  $p_i = P(\{x_i\})$ , con las  $x_i$  ordenadas según sus probabilidades  $p_1 \geq p_2 \geq \dots \geq p_n$  y  $\pi_i = \Pi(\{x_i\})$  y, entonces la solución óptima es

$$(C.13) \quad \forall i=1, n \quad \pi_i = \sum_{j=i}^n p_j$$

A continuación se da la solución en el caso continuo para funciones de densidad de probabilidad unimodales continuas  $p$ , con soporte acotado  $[a,b]$ , tal que  $p$  es creciente en  $[a,x_0]$  y decreciente en  $[x_0,b]$ , siendo  $x_0$  el valor modal de  $p$ .

Se define una función  $f: [a,x_0] \rightarrow [x_0,b]$  mediante  $f(x)=\max\{y \mid p(y)\geq p(x)\}$ . Entonces, la distribución de posibilidad más específica  $\pi$  que domina a  $p$  viene definida por:

$$\pi(x) = \pi(f(x)) = \int_{-\infty}^x p(y) dy + \int_{f(x)}^{\infty} p(y) dy$$

En la Figura C.2 se representa dicho cálculo de  $\pi(x)$  a partir de  $p(x)$  como suma de las dos áreas sombreadas. Esta transformación establece un conjunto de intervalos de confianza anidados donde los grados de posibilidad  $\pi(x)$  representan la probabilidad de perder el  $\pi(x)$ -corte.

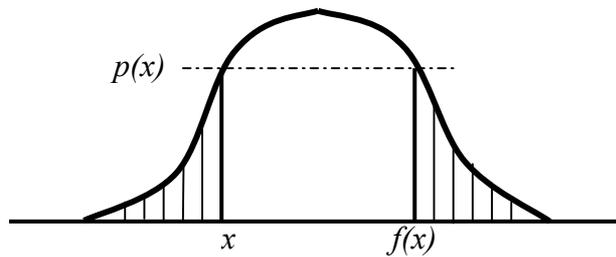


Figura C.2: Representación del cálculo de  $\pi(x)$  a partir de  $p(x)$

### C.4.2. Otras transformaciones

Las transformaciones citadas antes no son una la inversa de la otra. De hecho, no están basadas en los mismos principios. La transformación inversa de (C.12) se puede expresar de la siguiente forma:

$$(C.14) \quad \pi(x_i) = \sum_{j=1}^n \min(p(x_i), p(x_j))$$

Esta transformación (C.14) no cumple el criterio de máxima especificidad. La solución continua es:

$$\pi(x) = \int_{-\infty}^{\infty} \min(p(x), p(t)) dt$$

Este cálculo de  $\pi(x)$  a partir de  $p(x)$  aparece representado en la Figura C.3 como el área sombreada.

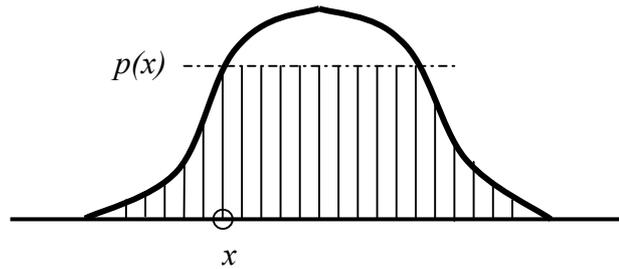


Figura C.3: Representación del cálculo de  $\pi(x)$  a partir de  $p(x)$

Veamos ahora la transformación inversa de (C.13). Suponiendo que  $\pi_1 \geq \pi_2 \geq \dots \geq \pi_n \geq \pi_{n+1} = 0$ :

$$(C.15) \quad p_i = \pi_i - \pi_{i+1} \quad i=1, n$$

El problema de la transformación (C.15) es que no respeta el orden de preferencia, esto es,  $\pi_i > \pi_{i+1}$  no implica  $p_i > p_{i+1}$ .

En el caso continuo se puede expresar como:

$$p(x) = \frac{1}{\frac{1}{\pi'(x)} - \frac{1}{\pi'(y)}}$$

donde  $[x, y]$  es el  $\pi(x)$ -corte de  $\pi$  y  $\pi'$  es la derivada de  $\pi$ .

Veamos otras transformaciones presentadas en [Lasserre et al., 1997].

*La transformación normalizada o escalada* ([Dubois, Prade, 1980]):

$$(C.16) \pi_i = p_i / p_1$$

que en el caso continuo se expresa como

$$\pi(x) = p(x) / p_m, \text{ con } p_m = \sup_x p(x)$$

Esta transformación puede contradecir el principio de consistencia entre probabilidad y posibilidad.

*La transformación que preserva la información:*

Geer y Klir ([Geer, Klir, 1992]) proponen los siguientes principios básicos:

- el escalado:  $\pi_i$  es una función de  $p_i / p_1$  donde  $p_1 \geq p_2 \geq \dots \geq p_n$
- la invarianza en la incertidumbre: La incertidumbre de  $p$ , medida como la entropía de Shannon  $H(p)$  debe ser igual a la incertidumbre de  $\pi$ ,  $E(\pi)$ , donde  $\pi$  es la transformada de  $p$ .  $E(\pi)$  está formado por dos sumandos: la no especificidad  $N(\pi)$  y la discordancia  $D(\pi)$  ([Klir, Folger, 1988]).

Por tanto, la transformación es de la forma

$$\pi_i = (p_i / p_1)^\alpha, \text{ donde } \alpha \text{ viene determinada por } H(p) = N(\pi) + D(\pi)$$

Esta transformación puede contradecir el principio de consistencia entre probabilidad y posibilidad.

*La transformación de Civanlar y Trussell* ([Lasserre et al., 1997], [Dubois et al., 1993]):

Esta transformación de probabilidad a posibilidad, busca la distribución  $\pi$  que se encuentra por encima de un umbral dado y que minimiza  $\sum (\pi_i)^2$ .

El primer requisito se expresa como

$$\sum_{i=1}^n \pi_i \cdot p_i \geq c, \text{ que es la condición de consistencia propuesta por Zadeh.}$$

Esta transformación no satisface siempre el principio de consistencia de probabilidad y posibilidad.

*La transformación triangular truncada* ([Lasserre et al., 1997]):

Los autores proponen una transformación de probabilidad a posibilidad apropiada para la representación de errores de medida en sensores. La distribución de posibilidad tiene forma triangular truncada y se selecciona teniendo en cuenta el principio de máxima especificidad. Para más información ver [Lasserre et al., 1997].

## **Bibliografía**

- Aldarmi, S. A., Burns, A. (1999a). *Dynamic CPU scheduling with imprecise knowledge of computation time*. Paper presented at the The 24th IFAC/IFIP Workshop on Real Time Programming and 3rd International Workshop on Active and Real-Time Database.
- Aldarmi, S. A., Burns, A. (1999b). *Dynamic value-density for scheduling real-time systems*. Paper presented at the 11th Euromicro Conference on Real-Time Systems.
- Alsina, C., Trillas, E. (1984). Additive homogeneity of logical connectives for membership functions. In Bezdek (Ed.), *Analysis of fuzzy information* (Vol. 1, pp. 179-184).
- Alsina, C., Trillas, E., Valverde, L. (1983). On some logical connectives for fuzzy sets theory. *Journal of Mathematical Analysis and Applications*, 93(1).
- Alvarez, E., Castillo, E. (1994). Uncertainty measures in expert systems. *Microcomputers in Civil Engineering*, 9, 359-366.
- Audsley, N. (1990). *Deadline Monotonic Scheduling* (YCS146). York: University of York. Department of Computer Science.
- Audsley, N., Burns, A., Richardson, M., Tindell, K., Wellings, A. J. (1991). *Hard real-time scheduling: the deadline monotonic approach*. Paper presented at the IEEE Workshop on Real-Time Operating Systems and Software, Atlanta, USA.
- Audsley, N., Burns, A., Richardson, M., Tindell, K., Wellings, A. J. (1993a). Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5), 284-292.
- Audsley, N., Tindell, K., Burns, A. (1993b). *The end of the line for static cyclic scheduling?* Paper presented at the 5th Euromicro Workshop on Real-Time Systems, Oulu, Finland.
- Baas, S. M., Kwakernaak, H. (1977). Rating and ranking of multiple-aspect alternatives using fuzzy sets. *Automatica*, 13, 47-58.
- Bailey, C. M., Fyfe, E., Vardanega, T., Wellings, A. J. (1993, December 1993). *The use of preemptive priority-based scheduling for space applications*. Paper presented at the IEEE Real-Time Systems Symposium, Durham, SC, USA.

- Baker, T. P. (1993). Stack-based scheduling of realtime processes, *Advances in Real Time Systems* (pp. 64-96): IEEE Computer Society Press.
- Bate, I. J. (1998). *Scheduling and timing analysis for safety critical real-time systems.* , University of York, York.
- Bernat, G., Burns, A. (1999). *New results on fixed priority aperiodic servers.* Paper presented at the IEEE Real-Time Systems Symposium.
- Bernat, G., Burns, A., Llamosí, A. (1999). *Weakly hard real-time systems* (YCS-99-320). York: University of York.
- Bilgic, T., Turksen, I. B. (1997). Measurement of membership functions: theoretical and empirical work. In D. Dubois & H. Prade (Eds.), *Handbook of fuzzy sets and systems* (Vol. 1, ).
- Borger, M. W., Rajkumar, R. (1989). *Implementing priority inheritance algorithms in an Ada runtime system* (CMU/SEI-89-TR-15). Pittsburgh: Carnegie-Mellon University, Software Engineering Institute.
- Bortolan, G., Degani, R. (1985). A review of some methods for ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15(1), xx-19.
- Buckley, J. J. (1985). Ranking alternatives using fuzzy numbers. *Fuzzy Sets and Systems*, 15, 21-31.
- Burns, A., Wellings, A. (1989). *Real-time systems and programming languages.*
- Burns, A., Wellings, A. (1996). Advanced fixed priority scheduling. In M. Joseph (Ed.), *Real-time systems specification, verification and analysis* : Prentice Hall.
- Buttazzo, G. C. (1997). *Hard real-time computing systems. Predictable scheduling algorithms and applications.* (Vol. 416): Kluwer Academic Publishers.
- Buttazzo, G. C., Sensini, F. (1997). *Optimal deadline assignment for scheduling soft aperiodic tasks in hard real-time environments.* Paper presented at the IEEE International Conference on Engineering of Complex Computer Systems, Como, Italy.
- del Castillo, F. (1980). *Análisis matemático II.* Alhambra Universidad.
- Chanas, S., Heilpern, S. (1989). Single value simulation of fuzzy variable- Some further results. *Fuzzy Sets and Systems*, 33, 29-36.

- Chanas, S., Nowakowski, M. (1988). Single value simulation of fuzzy variable. *Fuzzy Sets and Systems*, 25, 43-57.
- Chetto, H., Chetto, M. (1989). Some results of the earliest deadline scheduling algorithm. *IEEE Transactions on Software Engineering*, 15(10), 1261-1269.
- Chetto, H., Silly, M., Bouchentouf, T. (1990). Dynamic scheduling of real-time tasks under precedence constraints. *The Journal of Real-Time Systems*, 2, 181-194.
- Cucala, A., Villar, J. (2000). *Graphical Fuzzy Algebra Applied to Schedulability Analysis of Real Time Systems*. Paper presented at the international conference IPMU2000, 3, 1612-1619.
- Cucala, A., Villar, J. (2001). *Fuzzy Schedulability Analysis of Real Time Systems*. Paper presented at the WSES international conference on Fuzzy Sets and Fuzzy Systems 2001, 76-81.
- Cucala, A., Villar, J. (2002). *Schedulability Analysis of Real Time Systems under Uncertainty: Fuzzy Approach*. Paper presented at the international conference IPMU 2002, 3, 1509-1516.
- Davis, R. I., Tindell, K. W., Burns, A. (1993). *Scheduling slack time in fixed priority preemptive systems*. Paper presented at the Real-Time Systems Symposium.
- De Baets, B., Markova-Stupnanova, A. (1997). Analytical expressions for the addition of fuzzy intervals. *Fuzzy Sets and Systems*, 91, 203-213.
- Dubois, D., Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*. (Vol. 144): Academic Press, Inc.
- Dubois, D., Prade, H. (1983). Ranking Fuzzy Numbers in the Settings of Possibility Theory. *Information Sciences*, 30, 183-224.
- Dubois, D., Prade, H. (1985). A review of fuzzy set aggregation connectives. *Information Sciences*, 36, 85-121.
- Dubois, D., Prade, H. (1986a). Fuzzy Sets and Statistical Data. *European Journal of Operational Research*, 25, 345-356.
- Dubois, D., Prade, H. (1986b). A Set -Theoretic View of Belief Functions. *International Journal of General Systems*, 12, 193-226.

- Dubois, D., Prade, H. (1988a). An introduction to possibilistic and fuzzy logics. In P. Smets, E. H. Mamdani, D. Dubois, & H. Prade (Eds.), *Non-standard logics for automated reasoning* (pp. 287-326): Academic Press.
- Dubois, D., Prade, H. (1988b). Representation and Combination of Uncertainty with Belief Functions and Possibility Measures. *Computational Intelligence*, 4, 244-264.
- Dubois, D., Prade, H. (1989a). Fuzzy Sets, Probability and Measurement. *European Journal of Operational Research*, 40, 135-154.
- Dubois, D., Prade, H. (1989b). Processing Fuzzy Temporal Knowledge. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4), 729-744.
- Dubois, D., Prade, H. (1990). Consonant Approximations of Belief Functions. *International Journal of Approximate Reasoning*, 4, 419-449.
- Dubois, D., Prade, H. (1991a). Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions. *Fuzzy Sets and Systems*, 40, 143-202.
- Dubois, D., Prade, H. (1991b). Random Sets and Fuzzy Interval Analysis. *Fuzzy Sets and Systems*, 42, 87-101.
- Dubois, D., Prade, H. (1992). Fuzzy rules in knowledge-based systems. In R. R. Yager & L. A. Zadeh (Eds.), *An introduction to fuzzy logic applications in intelligent systems* : Kluwer Academic Publishers.
- Dubois, D., Prade, H. (1993a). Fuzzy rules in knowledge-based systems - Modelling gradeness, uncertainty and preference-. In R. R. Yager & L. A. Zadeh (Eds.), *An introduction to fuzzy logic applications in intelligent systems* (pp. 45-68): Kluwer Academic Publishers.
- Dubois, D., Prade, H. (1993b, March, 1993). *Fuzzy sets and probability: misunderstandings, bridges and gaps*. Paper presented at the 2° IEEE International Conference on Fuzzy Systems, San Francisco. California.
- Dubois, D., Prade, H. (1996). Focusing vs. revision in Possibility Theory. *IEEE*, 1700-1705.
- Dubois, D., Prade, H. (2000, 10-13 julio 2000). *Possibility theory in information fusion*. Paper presented at the Information Fusion, Paris.
- Dubois, D., Prade, H., Sandri, S. (1993). On possibility/probability transformations. In R. L. a. M. Roubens (Ed.), *Fuzzy Logic* (pp. 103-112): Kluwer Academic Publishers.

- Geer, J. F., Klir, G. J. (1992). A mathematical analysis of information-preserving transformations between probabilistic and possibilistic formulations of uncertainty. *International Journal of General Systems*, 20(2), 143-176.
- Ghazalie, T. M., Baker, T. P. (1995). Aperiodic servers in a deadline scheduling environment. *The Journal of Real-Time Systems*, 9(1), 31-68.
- Gilles, D. W. (1993). *Algorithms to schedule tasks with AND/OR precedence constraints*. , PhD. Thesis, University of Illinois at Urbana-Champaign.
- González Arechavala, Y., (2003). *Desarrollo de Sistemas Digitales de Seguridad: Enfoque basado en un Modelo Unificado de Simulación*. Tesis Doctoral, Universidad Pontificia Comillas de Madrid, Abril 2003.
- González Harbour, M., Medina, J.L., Gutiérrez, J.J., Palencia, J.C., Drake, J.M. (2002). *MAST: an open environment for modelling, analysis, and design of real-time systems*. 1<sup>st</sup> CARTS Workshop. Aranjuez, Spain. October 2002.
- González Harbour, M., Klein, M.H., Lehoczky, J.P. (1994). *Timing analysis for fixed priority scheduling of hard real-time systems*. IEEE Transactions on Software Engineering, 20(1), 13-28, January 1994.
- Goodenough, J. B., Sha, L. (1988). *The priority ceiling protocol: a method for minimizing the blocking of high-priority Ada tasks* (CMU/SEI-88-SR-4). Pittsburgh: Carnegie-Mellon University, Software Engineering Institute.
- Goodman, I. R. (1980). *Identification of fuzzy sets with a class of canonically induced random sets*. Paper presented at the 19th Conference on Decision and Control.
- Goodman, I. R. (1984). Some new results concerning random sets and fuzzy sets. *Information Sciences*, 34, 93-113.
- Grigg, A., Audsley, N. C. (1999a). *A method for design and analysis of next generation aircraft computer systems*. Paper presented at the 9th Symposium of the Int. Council on Systems Engineering.
- Grigg, A., Audsley, N. C. (1999b). Towards a scheduling and timing analysis solution for integrated modular avionic systems. *Microprocessors and Microsystems Journal*.
- Heilpern, S. (1997). Representation and application of fuzzy numbers. *Fuzzy Sets and Systems*, 91, 259-268.

- Hisdal, E. (1988). Are grades of membership probabilities? *Fuzzy Sets and Systems*, 25, 325-348.
- Horn, W. A. (1974). Some simple scheduling algorithms. *Naval Research Logistics Quarterly*, 21.
- Joseph, M. (1996). Time and real-time. In M. Joseph (Ed.), *Real-time systems. Specification, verification and analysis* : Prentice Hall.
- Joseph, M., Padya, P. (1986). Finding response times in a real-time system. *The Computer Journal (British Computing Society)*, 29(5), 390-395.
- Klein, M. H., Ralya, T., Pollak, B., Obenza, R., González Harbour, M. (1993). *A practitioner's handbook for real-time analysis: guide to rate monotonic analysis for real-time systems*. (5 ed.): Kluwer Academic Publishers.
- Klir, G. J. (1997). Fuzzy arithmetic with requisite constraints. *Fuzzy Sets and Systems*, 91, 165-175.
- Klir, G. J., Folger, T. A. (1988). *Fuzzy Sets, Uncertainty, and Information*.
- Klir, G. J., Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic. Theory and Applications*: Prentice Hall.
- Kosko, B. (1992). *Neural networks and fuzzy systems. A dynamical approach to machine intelligence*.
- Kruse, R., Gebhardt, J., Klawonn, F. (1994). *Foundations of fuzzy systems*: John Wiley & Sons Ltd.
- Lasserre, V., Mauris, G., Foulloy, L. (1997). *A simple probability-possibility transformation for measurement error representation: a truncated triangular transformation*. Paper presented at the 7th IFSA World Congress, Prague.
- Lehoczky, J. P. (1990). *Fixed priority scheduling of periodic task sets with arbitrary deadlines*. Paper presented at the IEEE Real-Time Systems Symposium, Los Alamitos, CA.
- Lehoczky, J. P., Ramos-Thuel, S. (1992). *An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems*. Paper presented at the Real-Time Systems Symposium.
- Lehoczky, J. P., Sha, L., Ding, Y. (1989). *The rate monotonic scheduling algorithm: exact characterization and average case behaviour*. Paper presented at the IEEE Real-Time Systems Symposium, Los Alamitos, CA.

- Lehoczky, J. P., Sha, L., Strosnider, J. K. (1987). *Enhanced aperiodic responsiveness in hard real-time environments*. Paper presented at the Real-Time Systems Symposium.
- Lehoczky, J. P., Sha, L., Strosnider, J. K., Tokuda, H. (1991). Fixed priority scheduling theory for hard real-time systems. In A. Tilbury & G. M. Koob (Eds.), *Foundations of real-time computing: scheduling and resource management* (pp. 1-30). Boston: Kluwer Academic Publishers.
- Leung, J. L. T., Whitehead, J. (1982). On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance Evaluation (Netherlands)*, 2(4), 237-250.
- Liou, T.-S., Wang, M.-J. J. (1992). Ranking fuzzy numbers with integral value. *Fuzzy Sets and Systems*, 50, 247-255.
- Lipari, G., Buttazzo, G. C. (2000). Schedulability analysis of periodic and aperiodic tasks with resource constraints. *Journal of Systems Architecture*, 46(4), 327-338.
- Litoiu, M., Tadei, R. (1997). Real Time Task Scheduling Allowing Fuzzy Due Dates. *European Journal of Operational Research*, 100(3), 475-481.
- Litoiu, M., Tadei, R. (2001a). Fuzzy scheduling with application to real-time systems. *Fuzzy Sets and Systems*, 121, 523-535.
- Litoiu, M., Tadei, R. (2001b). Real-time task scheduling with fuzzy deadlines and processing times. *Fuzzy Sets and Systems*, 117, 35-45.
- Liu, C., Layland, J. (1973). Scheduling algorithms for multiprogramming in a hard real-time environment. *JACM*, 20(1), 46-61.
- Liu, J. W. S., Lin, K.-J., Shih, W.-K., Yu, A. C.-s., Chung, J.-Y., Zhao, W. (1991). Algorithms for Scheduling Imprecise Computations. *Computer*(May), 58-68.
- Locke, C. D., Vogel, D. R., Mesler, T. J. (1991). *Building a predictable Avionics platform in Ada: a case study*. Paper presented at the IEEE 12th Real Time Systems Symposium.
- Lootsma, F. A. (1997). *Fuzzy logic for planning and decision making*. (Vol. 8): Kluwer Academic Publishers.
- Markova-Stupnanova, A. (1997). A note to the addition of fuzzy numbers based on a continuous Archimedean t-norm. *Fuzzy Sets and Systems*, 91, 253-258.

- Mesiar, R. (1997). Triangular-norm-based addition of fuzzy intervals. *Fuzzy Sets and Systems*, 91, 231-237.
- Nakamura, Y. (1993). Some Basic Operations on Fuzzy Sets and a Generalization of t-Norm Operations on Fuzzy Numbers by using Interval Representation, Part 1: Some Basic Fuzzy Set Operations and Max-Min Fuzzy Number Operations. *Japanese Journal of Fuzzy Theory and Systems*, 5(5), 689-703.
- Oberguggenberger, M., Fellin, W. (2002). *From probability to fuzzy sets: the struggle for meaning in geotechnical risk assesment*. Proceedings of Probabilistics in Geotechnics: Technical and Economic Risk Estimation, Graz, Austria, Verlag Glückauf Essen 2002.
- OMG. (2001). *Response to the OMG RFP for schedulability, performance and time* (ad/2001-06-14).
- OMG. (1999), A. R.-s. . *UML Profile for scheduling. Performance and time request for proposal*. (ad/99-03-13).
- Palencia, J.C. (1999). *Análisis de planificabilidad de sistemas distribuidos de tiempo real basados en prioridades fijas*. Tesis Doctoral. Universidad de Cantabria, Julio 1999.
- Ramamritham, K. (1996). Dynamic priority scheduling. In Joseph (Ed.), *Real-time systems. Specification, verification and analysis* (pp. 66-96): Prentice-Hall.
- Sha, L., Goodenough, J. B. (1989). *Real-time scheduling theory and Ada* (CMU/SEI-89-TR-14). Pittsburgh: Carnegie-Mellon University, Software Engineering Institute.
- Sha, L., Klein, M. H., Goodenough, J. B. (1991). *Rate monotonic analysis for real-time systems* (CMU/SEI-91-TR-6). Pittsburgh: Carnegie-Mellon University, Software Engineering Institute.
- Sha, L., Rajkumar, R., Lehoczky, J. P. (1990). Priority inheritance protocols: an approach to real-time synchronization. *IEEE Transactions on Computers*, 39(9), 1175-1185.
- Sha, L., Rajkumar, R., Lehoczky, J. P., Ramamritham, K. (1989). Mode change protocols for priority-driven preemptive scheduling. *The Journal of Real-Time Systems*, 1, 243-264.
- Shafer, G. (1987). Belief functions and possibility measures. In J. C. Bezdek (Ed.), *Analysis fuzzy information* (Vol. 1, pp. 51-84).

- Shih, W. K., Liu, J. W. S. (1995). Algorithms for scheduling imprecise computations with time constraints to minimize maximum error. *IEEE Transactions on Computers*, 44(3), 466-471.
- Shih, W. K., Liu, J. W. S., Liu, C. L. (1993). Modified Rate-Monotonic Algorithm for Scheduling Periodic Jobs with Deferred Deadlines. *IEEE Transactions on Software Engineering*, 19(12), 1171-1179.
- Shin, K. G., Ramanathan, P. (1994). Real time computing: a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1), 6-24.
- Smets, P. (1988). Belief functions. In P. Smets, E. H. Mamdani, D. Dubois, & H. Prade (Eds.), *Non-standard logics for automated reasoning* : Academic Press.
- Sprunt, B., Sha, L., Lehoczky, J. P. (1989). Aperiodic task scheduling for hard real-time systems. *The Journal of Real-Time Systems*, 1, 27-60.
- Spuri, M., Buttazzo, G., Sensini, F. (1995). *Robust Aperiodic Scheduling under Dynamic Priority*. Paper presented at the IEEE Real-Time Systems Symposium, Pisa, Italy.
- Spuri, M., Buttazzo, G. C. (1994). *Efficient aperiodic service under earliest deadline scheduling*. Paper presented at the IEEE Real-Time Systems Symposium, San Juan, Portorico.
- Spuri, M., Buttazzo, G. C. (1996). Scheduling aperiodic tasks in dynamic priority systems. *The Journal of Real-Time Systems*, 10(2), 179-210.
- Strosnider, J. K., Lehoczky, J. P., Sha, L. (1995). The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments. *IEEE Transactions on Computers*, 44(1).
- Sun, J., Gardner, M. K., Liu, J. W. S. (1997). Bounding Completion Times of Jobs with Arbitrary Release Times, Variable Execution Times, and Resource Sharing. *IEEE Transactions on Software Engineering*, 23(10), 603-615.
- Sun, J., Liu, J. W. S. (1996). *Bounding Completion Times of Jobs with Arbitrary Release Times, Variable Execution Times*. Paper presented at the 17th IEEE Real-Time Systems Symposium, Washington, DC.
- Tan, S. K., Wang, P. Z., Zhang, X. Z. (1993). Fuzzy Inference Relation based on the theory of Falling Shadow. *Fuzzy Sets and Systems*, 53, 179-188.

- Terrier, F., Chen, Z. (1994, Jun 1994). *Fuzzy Calculus Applied to Real-Time Scheduling*. Paper presented at the Third IEEE International Conference on Fuzzy Systems, Orlando.
- Terrier, F., Rioux, L., Chen, Z. (1995, 20-24 March 1995). *Real Time Scheduling Under Uncertainty*. Paper presented at the Proceedings of 1995 IEEE International Conference on Fuzzy Systems., Yokohama.
- Tia, T. S. (1995). *Utilizing slack time for aperiodic and sporadic requests scheduling in real-time systems.*, University of Illinois at Urbana Champaign.
- Tia, T. S., Deng, Z., Shankar, M., Storch, M., Sun, J., Wu, L. C., Liu, J. W. S. (1995, May 1995). *Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times*. Paper presented at the IEEE Real-Time Technology and Applications Symposium, Chicago, Illinois.
- Tia, T. S., Liu, J. W. S., Shankar, M. (1996). Algorithms and optimality of scheduling soft aperiodic requests in fixed-priority preemptive systems. *The Journal of Real-Time Systems*, 10(1), 23-43.
- Tindell, K. (1992). *An extendible approach for analysing fixed priority hard real-time tasks* (YCS189). York: University of York.
- Tindell, K., Burns, A., Wellings, A. (1992). *Mode Changes in Priority Preemptively Scheduled Systems*. Paper presented at the Real Time Systems Symposium, Phoenix Arizona, U.S.A.
- Tindell, K., Burns, A., Wellings, A. (1994). An extendible approach for analysing fixed priority hard real-time tasks. *The Journal of Real-Time Systems*, 6(2), 133-151.
- Villar, J. (1997). *Aplicación de la teoría de conjuntos borrosos al diagnóstico de procesos industriales.*, Universidad Pontificia Comillas de Madrid.
- Walley, P. (1996). Measures of uncertainty in expert systems. *Artificial Intelligence*, 83, 1-58.
- Wang, P. Z., Zhang, H. M., Ma, X. W., Xu, W. (1991). *Fuzzy Set Operations Represented by Falling Shadow Theory*. Paper presented at the Fuzzy Engineering toward Human Friendly Systems. IFES'91.
- Watson, S. R., Weiss, J. J., Donnell, M. L. (1979). Fuzzy decision analysis. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(1), 1-9.

Yager, R. R. (1982). Level sets for membership evaluation of fuzzy subsets. In R. R. Yager (Ed.), *Fuzzy Sets and Possibility Theory: recent developments* (Vol. 1, pp. 90-97). Oxford: Pergamon Press.

Zadeh, L. A. (1965). Fuzzy Sets. *Inform. Control*, 8(3), 338-353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1, 3-28.