



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA
AVANZADA

OPTIMIZACIÓN DEL LAYOUT DE WAREHOUSE APLICADO A MÉTODOS CONSTRUCTIVOS Y DE MEJORA ESPECÍFICOS EN PYTHON

Autor: José Luis Fernández-Figueroa Díaz-Trehuelo

Director: Guillermo Bonmati Arias

Madrid

Abril 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Optimización del layout de Warehouse aplicado a métodos constructivos y de mejora
específicos en Python.

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: José Luis Fernández-Figueroa Díaz-Trechuelo

Fecha: 29/ 04/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Guillermo Bonmati Arias

Fecha: 29/ 04/ 2025

Vº Bº del Coordinador de Proyectos

Fdo.: Carlos Morrás Ruiz-Falcó

Fecha: .30.../ .04.../ .2025

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. José Luis Fernández-Figueroa Díaz-Trechuelo

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Optimización del layout de Warehouse aplicado a métodos constructivos y de mejora específicos en Python que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

- El autor se compromete a:
 - a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
 - b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
 - c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
 - d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 29 de Abril de 2025

ACEPTA

Fdo: José Luis Fernández Figueroa Díaz-Trechuelo

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA
AVANZADA

OPTIMIZACIÓN DEL LAYOUT DE WAREHOUSE APLICADO A MÉTODOS CONSTRUCTIVOS Y DE MEJORA ESPECÍFICOS EN PYTHON

Autor: José Luis Fernández-Figueroa Díaz-Trechuelo

Director: Guillermo Bonmati Arias

Madrid

Abril 2025

Agradecimientos

Los agradecimientos de este trabajo de fin de máster van dirigidos primordialmente a mi familia. La cuál me ha hecho posible cumplir mis objetivos académicos y ha sido un pilar fundamental a lo largo de mi vida para cumplir mis objetivos, gracias por los valores y consejos que me han hecho convertirme en la persona que soy hoy día.

También me gustaría agradecer a la Universidad Pontificia de Comillas por su dedicación y compromiso con mi aprendizaje. Los conocimientos y experiencias adquiridos durante el curso de este máster han sido cruciales para mi desarrollo profesional y personal.

Finalmente quería agradecer a la empresa Accenture, la cual me ha dado la oportunidad de trabajar con su increíble grupo humano, el cuál es distinguido por su nivel profesional. Desde el primer momento me sentí muy involucrado con la empresa y con el proyecto que le da sentido a este trabajo de fin de máster.

Índice de la memoria

Capítulo 1. Introducción	8
1.1 Motivación del proyecto.....	8
1.2 Descripción de las tecnologías	8
Capítulo 2. Estado del arte	11
2.1 Optimización	11
2.2 Simulación.....	12
2.3 Almacenes	15
2.4 Generación de layout por computador.	16
Capítulo 3. Definición del Trabajo	18
3.1 Objetivos	18
3.2 Requisitos	18
3.3 Metodología.....	19
3.4 Planificación y Estimación Económica	20
3.4.1 Planificación.....	20
3.4.2 Estimación Económica	21
Capítulo 4. Sistema/Modelo Desarrollado	24
4.1 Análisis del Sistema	24
4.1.1 Departamentos	24
4.1.2 Inputs del caso de uso.....	29
4.2 Diseño.....	31
4.3 Implementación	33
4.3.1 Modelos a desarrollar	33
4.3.1 Características del modelo.....	34
4.3.2 Función objetivo.....	38
4.3.3 Ingesta de datos.....	39
4.3.4 Métodos constructivos	49
4.3.5 Métodos de mejora	65
4.3.6 Implementación de Simio	67
Capítulo 5. Análisis de Resultados.....	89

5.1 Python.....	89
5.2 SIMIO.....	94
5.2.1 Distribución inicial.....	95
5.2.2 Distribución optimizada	97
Capítulo 6. Conclusiones y Trabajos Futuros	99
6.1 Conclusiones	99
6.1.1 Generales.....	99
6.1.2 Caso de uso.....	100
6.2 Trabajos futuros.....	100
Capítulo 7. Bibliografía.....	102
ANEXO A	103

Índice de ilustraciones

Ilustración 1. Gantt del proyecto	21
Ilustración 2. Muelles de salida y de entrada.....	25
Ilustración 3. Almacén tradicional	25
Ilustración 4. ASRS	26
Ilustración 5. Planta de producción	27
Ilustración 6. Área de carriles de circulación	27
Ilustración 7. Centradora y envolvedora.....	28
Ilustración 8. Oficina	28
Ilustración 9. Distribución inicial del almacén.....	30
Ilustración 10. Diseño del proyecto	32
Ilustración 11. Valle de soluciones.....	34
Ilustración 12. Ejemplo de departamentos fijos	35
Ilustración 13. Ejemplo de pasillos	36
Ilustración 14. Ejemplo de puntos de entradas y salidas.....	37
Ilustración 15. Ejemplo de rotación.....	37
Ilustración 16. Ejemplo de limitaciones	38
Ilustración 17. Distancias de la función objetivo	39
Ilustración 18. Valores cualitativos del TCR.....	50
Ilustración 19. Matriz total de flujo	52
Ilustración 20. Creación de la secuencia	52
Ilustración 21. Explicación de CORELAP, datos de ejemplo.....	56
Ilustración 22. Explicación de CORELAP, posibles posiciones	57
Ilustración 23. Explicación de CORELAP, posiciones que minimizan la función objetivo.....	57
Ilustración 24. Explicación de CORELAP, localización del siguiente departamento.....	58
Ilustración 25. Ejemplo de CORELAP del ciclo completo con el siguiente departamento	58
Ilustración 26. Ejemplo de secuencias factibles e infactibles.....	59

Ilustración 27. Ejemplo de relocalización del primer departamento	61
Ilustración 28. Anillo interior de la búsqueda por anillos	63
Ilustración 29 Anillo exterior de la búsqueda por anillos.....	64
Ilustración 30. Diagrama de flujo del CRAFT	66
Ilustración 31. Ejemplo de resultados del CRAFT.....	67
Ilustración 32. Diseño de la planta inicial en SIMIO	69
Ilustración 33. Modelo del almacén tradicional en SIMIO	70
Ilustración 34. Modelo de los muelles de entrada en SIMIO	71
Ilustración 35. Modelo de los muelles de salida en SIMIO.....	71
Ilustración 36. Modelo de la planta de producción en SIMIO	72
Ilustración 37. Modelo del ASRS en SIMIO.....	73
Ilustración 38. Modelo de oficina en SIMIO.....	74
Ilustración 39. Modelo de líneas de flujo en SIMIO	75
Ilustración 40. Modelo de centradora y envolvedora en SIMIO	75
Ilustración 41. Procesos de rutas 1 en SIMIO	77
Ilustración 42. Procesos de rutas 2 en SIMIO	78
Ilustración 43. Procesos de rutas 3 en SIMIO	79
Ilustración 44. Procesos de control de entradas al ASRS en SIMIO.....	80
Ilustración 45. Procesos de control de salidas del ASRS en SIMIO	81
Ilustración 46. Procesos de control de entradas a las líneas de flujo en SIMIO.....	82
Ilustración 47. Procesos de control de salida de las líneas de flujo en SIMIO.....	83
Ilustración 48. Check de solape en SIMIO.....	84
Ilustración 49. Check de fuera de rango en SIMIO.....	84
Ilustración 50. CRAFT en SIMIO 1	85
Ilustración 51. CRAFT en SIMIO 2	85
Ilustración 52. Procesos de KPIS en SIMIO	86
Ilustración 53. Proceso de inicio en SIMIO	86
Ilustración 54. KPIS en SIMIO	88
Ilustración 55. Interacción con la App de Streamlit 1	90
Ilustración 56. Interacción con la App de Streamlit 2	90

Ilustración 57. Interacción con la App de Streamlit 3	91
Ilustración 58. Interacción con la App de Streamlit 4	91
Ilustración 59. Interacción con la App de Streamlit 5	92
Ilustración 60. Interacción con la App de Streamlit 6	92
Ilustración 61. Solución de CORELAP.....	93
Ilustración 62. Solución de CRAFT	94
Ilustración 63. Resultados de la simulación de la distribución inicial	96
Ilustración 64. Resultados de la simulación de la distribución optimizada.....	98

Índice de tablas

Tabla 1. Métodos Heurísticos.....	17
Tabla 2. Costes humanos del proyecto	22
Tabla 3. Coste de licencias del proyecto	22
Tabla 4. Coste de materiales del proyecto.....	23
Tabla 5. Costes totales del proyecto	23
Tabla 6. Dimensiones del almacén	29
Tabla 7. Dimensiones de departamentos	29
Tabla 8. Atriz de flujo del almacén	30
Tabla 9. Matriz de pasillos del almacén	31
Tabla 10. Departamentos fijos del almacén.....	31
Tabla 11. Puntos de entradas y salidas de los departamentos del almacén	31
Tabla 12. Ejemplo de inputs dimensiones del almacén.....	40
Tabla 13. Ejemplo de inputs de elementos del sistema	41
Tabla 14. Ejemplo de input de la matriz de flujo	42
Tabla 15. Ejemplo de input de la matriz de coste.....	43
Tabla 16. Ejemplo de input de la matriz de pasillos.....	44
Tabla 17. Ejemplo de cómo introducir un departamento fijo.....	45
Tabla 18. Ejemplo de input de los departamentos fijos.....	46
Tabla 19. Ejemplo de cómo introducir los puntos de entrada y salida	46
Tabla 20. Ejemplo de input de los puntos de entrada y salida	47
Tabla 21. Ejemplo de cómo introducir las limitaciones	48
Tabla 22. Ejemplo de input de limitaciones	49
Tabla 23. Tabla de cálculo cualitativo del TCR	51
Tabla 24. Test de eficiencia.....	64
Tabla 25. Ejemplo de inputs de SIMIO, hoja 1	68
Tabla 26. Ejemplo de input de SIMIO, hoja 2	68

Tabla 27. Ahorro total tras la optimización 100

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

Este proyecto nació de una necesidad por parte de la empresa Accenture. Accenture es una empresa especializada en consultoría tecnológica, por lo que la mayoría de los proyectos a los que se enfrenta son del ámbito industrial.

Una de las necesidades que tenía Accenture al comienzo de este trabajo de fin de grado era desarrollar un código en Python que optimizara la distribución en planta de los departamentos de un almacén, para minimizar el flujo de entidades entre ellos y de esta manera ahorrar coste y tiempo.

Un requisito importante de este proyecto era que la optimización estuviera basada en métodos heurísticos y no en modelos matemáticos, con el objetivo de tener una buena solución en un tiempo considerable de ejecución.

1.2 DESCRIPCIÓN DE LAS TECNOLOGÍAS

Para el desarrollo de este proyecto se han utilizado dos tecnologías que han ayudado a conseguir los objetivos de este, estas tecnologías son las siguientes:

- **Python:** Es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y su primera versión se lanzó en 1991. Es conocido por su sintaxis clara y legible, lo que facilita su aprendizaje y uso, especialmente para los principiantes. Python admite múltiples paradigmas de programación, incluyendo la programación orientada a objetos, la programación imperativa y la programación funcional.

Características principales de Python:

- **Sintaxis simple y legible:** Python utiliza una sintaxis que es fácil de leer y escribir, lo que ayuda a los desarrolladores a concentrarse en resolver problemas en lugar de preocuparse por la sintaxis del lenguaje.
 - **Bibliotecas extensas:** Python cuenta con una amplia gama de bibliotecas estándar y de terceros que facilitan el desarrollo de aplicaciones en diversos dominios, como ciencia de datos, inteligencia artificial, desarrollo web, automatización, y más.
 - **Interpretable:** Python es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea, facilitando la depuración y el desarrollo interactivo.
 - **Comunidad activa:** La comunidad de Python es muy activa y contribuye continuamente al desarrollo de nuevas herramientas, bibliotecas y recursos para los desarrolladores.
 - **Multiparadigma:** Python soporta diferentes paradigmas de programación, lo que permite a los desarrolladores elegir el estilo de programación que mejor se adapte a sus necesidades.
-
- **Simio:** Es una plataforma de software de simulación y modelado de eventos discretos y continuos. Fue desarrollada por Simio LLC y se utiliza para diseñar, analizar y mejorar sistemas complejos en diferentes industrias, como manufactura, logística, atención médica y servicios públicos.

Características principales de Simio:

- **Modelado visual:** Simio proporciona una interfaz gráfica de usuario intuitiva que permite a los usuarios construir modelos de simulación visualmente, utilizando un entorno de arrastrar y soltar.
- **Flexibilidad y personalización:** Simio permite a los usuarios personalizar sus modelos con lógica compleja utilizando su lenguaje de scripting basado en objetos.
- **Integración:** Simio puede integrarse con otras herramientas y sistemas empresariales, como bases de datos y sistemas de planificación de recursos empresariales (ERP), facilitando el flujo de datos y la sincronización.

- **Análisis y optimización:** Simio ofrece potentes herramientas de análisis y optimización que permiten a los usuarios evaluar diferentes escenarios y encontrar soluciones óptimas para mejorar la eficiencia y el rendimiento de los sistemas.
- **Simulación en tiempo real:** Simio puede simular tanto eventos discretos como procesos continuos, y también permite la simulación en tiempo real, lo que es útil para aplicaciones como la programación y control en tiempo real.

Capítulo 2. ESTADO DEL ARTE

2.1 OPTIMIZACIÓN

La optimización es un proceso matemático y computacional que consiste en encontrar la mejor solución posible (o soluciones) para un problema determinado, dentro de un conjunto de posibles soluciones, bajo ciertas restricciones y criterios definidos. El objetivo principal de la optimización es maximizar o minimizar una función objetivo, que representa la cantidad o calidad que se desea optimizar.

1. Tipos de Optimización

1. **Optimización Lineal:** Se enfoca en maximizar o minimizar una función lineal sujeta a restricciones lineales.
 - Ejemplo: Asignación de recursos en una empresa para maximizar ganancias.
2. **Optimización No Lineal:** Trata con funciones objetivo y/o restricciones no lineales.
 - Ejemplo: Diseño de productos con materiales y formas que minimicen el costo y maximicen la resistencia.
3. **Optimización Entera:** Involucra variables que deben tomar valores enteros.
 - Ejemplo: Planificación de rutas de vehículos donde los vehículos deben visitar un número entero de destinos.
4. **Optimización Combinatoria:** Se enfoca en encontrar la mejor solución de un conjunto finito pero grande de soluciones posibles.
 - Ejemplo: Problema del viajante (Traveling Salesman Problem).

2. Aplicaciones de la Optimización

3. **Industria y Manufactura:** Optimización de procesos de producción para minimizar costos y maximizar eficiencia.
4. **Logística y Transporte:** Planificación de rutas óptimas para la distribución de productos.
5. **Finanzas:** Maximización de la rentabilidad de una cartera de inversiones.
6. **Ingeniería:** Diseño de estructuras y sistemas que optimicen ciertos parámetros como peso, costo y durabilidad.
7. **Inteligencia Artificial:** Entrenamiento de modelos de aprendizaje automático para mejorar su precisión y eficiencia.
8. **Métodos de Optimización**
 1. **Métodos Analíticos:** Utilizan fórmulas y derivadas para encontrar soluciones óptimas de manera exacta.
 2. **Métodos Numéricos:** Utilizan algoritmos iterativos para aproximar soluciones óptimas cuando las soluciones exactas son difíciles de obtener.
 3. **Algoritmos Heurísticos:** Estrategias aproximadas que buscan soluciones buenas (aunque no necesariamente óptimas) en un tiempo razonable. Ejemplos incluyen algoritmos genéticos y búsqueda por enjambre de partículas.

La optimización es fundamental en muchas áreas debido a su capacidad para mejorar la eficiencia, reducir costos y maximizar resultados.

2.2 SIMULACIÓN

La simulación es una técnica utilizada para imitar el comportamiento de un sistema real mediante un modelo computacional, matemático o físico. Este modelo permite estudiar y

analizar el sistema bajo diversas condiciones sin necesidad de interactuar directamente con el sistema real. La simulación se utiliza en diversas disciplinas, incluidas la ingeniería, las ciencias, la economía, la medicina y la investigación operativa, para prever y entender el comportamiento de sistemas complejos.

Tipos de Simulación

1. Simulación Discreta

- **Definición:** Modela sistemas en los que los cambios de estado ocurren en momentos específicos.
- **Aplicaciones:** Modelado de líneas de espera, redes de comunicación, producción industrial.
- **Ejemplo:** Simulación del flujo de clientes en un supermercado para optimizar el número de cajas registradoras abiertas.

2. Simulación Continua

- **Definición:** Modela sistemas donde los cambios de estado ocurren de manera continua a lo largo del tiempo.
- **Aplicaciones:** Modelado de fenómenos físicos como la dinámica de fluidos, el clima, la ecología.
- **Ejemplo:** Simulación de la dispersión de contaminantes en un río.

3. Simulación Híbrida

- **Definición:** Combina elementos de simulación discreta y continua.
- **Aplicaciones:** Sistemas donde hay interacción entre eventos discretos y procesos continuos.

- **Ejemplo:** Modelado de un sistema de producción donde las máquinas (procesos continuos) se detienen y arrancan en momentos específicos (eventos discretos).

Aplicaciones de la Simulación

1. Industria y Manufactura

- Optimización de líneas de producción, gestión de inventarios, planificación de recursos.

2. Salud

- Modelado de la propagación de enfermedades, planificación de recursos hospitalarios, diseño de tratamientos.

3. Finanzas

- Análisis de riesgo, valoración de derivados, planificación financiera.

4. Transporte y Logística

- Planificación de rutas, optimización de redes de transporte, gestión de almacenes.

5. Investigación Científica

- Modelado de fenómenos naturales, experimentación virtual, predicción de cambios climáticos.

La simulación es una herramienta poderosa para entender sistemas complejos, predecir resultados y mejorar la toma de decisiones en diversas áreas.

2.3 ALMACENES

En el contexto empresarial y logístico, un almacén es una instalación donde se almacenan bienes, productos y materiales de manera organizada y segura hasta que sean distribuidos o utilizados. Los almacenes son cruciales para la gestión de inventarios y la cadena de suministro, proporcionando un lugar centralizado para la recepción, almacenamiento, manejo y distribución de mercancías.

Los almacenes suelen variar según el tipo de producto que se almacena en ellos, aunque sí que suelen mantener una arquitectura común entre todos ellos. Los departamentos más comunes de un almacén son los siguientes:

Recepción

- **Función:** Inspeccionar y registrar la entrada de mercancías en el almacén.
- **Actividades:** Descarga de camiones, verificación de cantidades y calidad, registro en el sistema de inventario.

Almacenamiento

- **Función:** Colocar las mercancías en ubicaciones designadas dentro del almacén.
- **Actividades:** Organización del espacio, uso de estanterías y sistemas de almacenamiento, mantenimiento del orden y limpieza.

Empaque

- **Función:** Empacar los productos recogidos para su envío.
- **Actividades:** Empaque seguro y adecuado, etiquetado, uso de materiales de embalaje.

Despacho y Envío

- **Función:** Preparar y enviar los pedidos a los clientes o a otros destinos.

- **Actividades:** Carga de camiones, coordinación con transportistas, generación de documentación de envío.

Cada uno de estos departamentos juega un papel esencial en el funcionamiento eficiente de un almacén. La coordinación y la gestión efectiva entre estos departamentos son cruciales para asegurar una operación fluida, minimizar errores y maximizar la eficiencia en la cadena de suministro.

2.4 GENERACIÓN DE LAYOUT POR COMPUTADOR.

Se entiende por Métodos Analíticos de Generación de Layouts (MGL) el conjunto de técnicas que permiten obtener un abanico de alternativas que ayuden al proyectista en la búsqueda de una solución para el diseño general de una implantación (Layout es diseños de plantas).

Los MGL consiguen la ordenación topológica de los elementos que intervienen en un problema de implantación, pero es difícil conseguir un control geométrico aceptable para su aplicación en un planteamiento de detalle.

En general habrá que hacer pequeños retoques finales. Estos métodos pueden clasificarse, atendiendo a los algoritmos utilizados en su desarrollo, en métodos de construcción de Layouts (MCL) y en métodos de mejora de Layouts (MML). Los métodos de construcción generan soluciones por síntesis de la información de acuerdo con la heurística propia del método; los de mejora, partiendo de una solución inicial, tratan de superarla siguiendo criterios generalmente de índole económica y relacionados con la manutención y el transporte.

Los MCL parten de la Tabla Relacional de Actividades entre elementos del sistema y tienen su mayor aplicación en aquellos casos en los que las relaciones dadas para los servicios anexos (medios auxiliares de producción) constituyen un factor de peso en la implantación, pudiendo o no tener importancia el recorrido de los productos. Los MML se basan en el tráfico de materiales y en el recorrido de los productos, tratando de optimizar una función

asociada a los costes de transporte y manutención de los mismos. Los servicios anexos, en estos casos, no constituyen un factor básico en la implantación. [1]

Dado que la tabla relacional de actividades tiene un carácter cualitativo y ya que el tráfico de materiales puede ser cuantificado, a los MCL también se les denomina métodos cualitativos o de construcción de Layouts, y a los MML métodos cuantitativos o de mejora de Layouts. Se ha de observar que la mayoría de los casos reales son intermedios entre las dos situaciones límites antes descritas y por tanto no se ajustan a las hipótesis de partida asumidas por los procedimientos descritos. Pero también es cierto que los métodos analíticos de generación de layouts constituyen una herramienta al servicio del proyectista que le permitirá analizar un mismo problema desde ópticas distintas y confrontar resultados en el proceso de búsqueda de una solución. [2]

Construcción	Mejora
Método gráfico CORELAP PLANET ALDEP	Método de intercambio por parejas CRAFT MCCRAFT MULTIPLE
BLOCPAN LOGIC Programación entera mixta	

Tabla 1. Métodos Heurísticos

Capítulo 3. DEFINICIÓN DEL TRABAJO

3.1 OBJETIVOS

El objetivo de este proyecto es desarrollar un código en Python que sea capaz de optimizar la distribución de departamentos en un almacén en un tiempo breve de ejecución y consiguiendo una solución de óptimo local considerantemente buena, nunca buscando el óptimo global del problema.

Una vez el código esté desarrollado también se tiene como objetivo desarrollar un modelo de Simio donde se pueda simular la solución propuesta por el código tras realizar la optimización, con el objetivo de ver cómo se comporta el sistema en términos de eficiencia y capacidades.

3.2 REQUISITOS

Los requisitos exigidos para el siguiente proyecto son los siguientes:

1. Realizar el código en Python.
2. Desarrollar la optimización basado en heurísticas.
3. Realizar un código robusto y flexible.
4. Desarrollar Backend para interactuar con el modelo.
5. Desarrollar modelo de simulación en Simio.
6. Medir resultados en KPIS.

3.3 METODOLOGÍA

En este proyecto al tratarse de un proyecto de investigación y desarrollo no ha seguido el curso normal de un proyecto de Bigdata en consultoría, los cuales suelen seguir una metodología de tipo ágil.

La metodología ágil es un enfoque para la gestión de proyectos y desarrollo de software que enfatiza la entrega incremental, la colaboración constante con los clientes y la capacidad de adaptación a los cambios.

Para conseguir los objetivos de este proyecto en específico se han seguido los siguientes pasos dentro de una serie de etapas claramente diferenciadas:

- **Etapas de investigación.**

Durante esta etapa se enfocó el proyecto en investigar y recopilar toda la información posible para conocer los métodos existentes para abordar este tipo de problemas, con el objetivo de obtener el conocimiento necesario para posteriormente tomar la decisión de que métodos heurísticos se desarrollarán en el proyecto para dar solución al problema que este plantea.

- **Etapas de desarrollo de heurísticas constructivas.**

En esta etapa se comenzó a desarrollar el código de la heurística constructiva, en la cual hubo que diseñar la estructura del código, desarrollar el código en sí, realizar test funcionales y por último modificar y aplicar las correcciones pertinentes tras los resultados de los test.

- **Etapas de desarrollo de heurísticas de mejora.**

En esta etapa se comenzó a desarrollar el código de la heurística de mejora, en la cual hubo que diseñar la estructura del código, desarrollar el código en sí, realizar test funcionales y por último modificar y aplicar las correcciones pertinentes tras los resultados de los test.

- **Etapas de desarrollo de modelo de Simio.**

Esta etapa fue desarrollada en paralelo a las 2 anteriores. Esta etapa es la mas larga ya que esta aplicaba un 25% del tiempo de desarrollo mientras que las 2 anteriores aplicaban un 75% del tiempo.

Antes de que se empezara el desarrollo del modelo de simulación, se tuvo que realizar una formación previa en la herramienta de simulación.

- **Etapa de desarrollo del Backend.**

Durante esta etapa se desarrolló una App en Streamlit en la que de manera visual el cliente pueda interactuar con el modelo de optimización y hacer uso de él. Esta app aparte de ser desarrollada también fue sometido a test funcionales para comprobar que se integraba bien con el modelo de optimización y se hicieron correcciones teniendo en cuenta los resultados de los test.

3.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

3.4.1 PLANIFICACIÓN

Tras definir previamente la metodología y los pasos a seguir en el proyecto se realizó una planificación de las tareas necesarias para alcanzar los objetivos del proyecto y llegar al éxito del mismo.

Estas tareas son las representadas en la *Ilustración 1* y su estimación de tiempo fueron calculadas utilizando técnicas como la de juicio de expertos y estimaciones análogas los cuales son métodos recomendados por el Project Management Institute (PMI) en su guía de gestión de proyectos Project Management Body of Knowledge (PMBOK).

- Juicio de expertos: Consiste en consultar a personas con experiencia, conocimientos y habilidades especializadas en una materia particular para tomar decisiones informadas.
- Estimaciones análogas: Se basa en la premisa de que, si un proyecto anterior es similar al actual, las métricas de tiempo, costo y esfuerzo serán comparables.

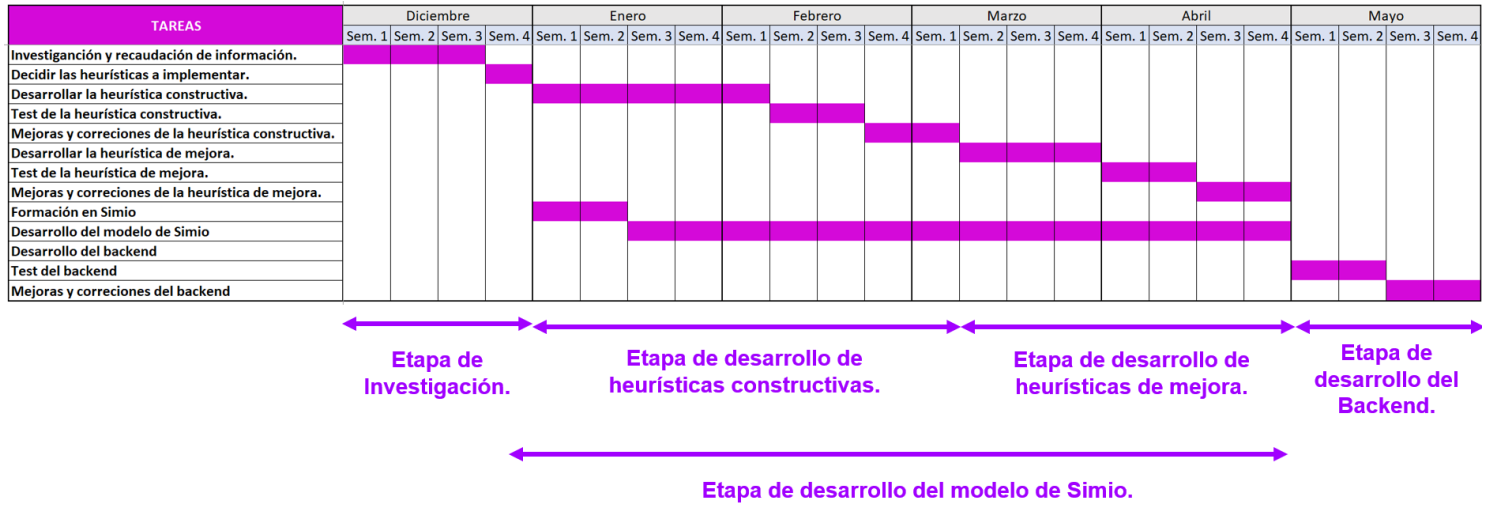


Ilustración 1. Gantt del proyecto

3.4.2 ESTIMACIÓN ECONÓMICA

La estimación de costes de este proyecto ha sido dividida en tres tipos de costes, los humanos, los de licencia y los materiales.

Todos los costes han sido calculados para los meses de duración de este proyecto.

3.4.2.1 Costes Humanos

Estos costes han sido estimados con datos aproximados al valor real de salarios brutos de los empleados de la empresa Accenture.

Para el desarrollo de este proyecto han sido necesarios un empleado con puesto de becario, otro con puesto de Consultor y otro con puesto de Manager. Pero no todos ellos tendrán la misma implicación en cuestión de tiempo en el desarrollo del proyecto.

Costes humanos					
Empleados	Cantidad	Meses de duración	Ocupación	Coste al mes para la empresa	Coste en el proyecto
Intern	1	6	100%	1.027 €	6.162 €
Consultant	1	6	10%	4.875 €	2.925 €
Manager	1	6	5%	7.042 €	2.113 €
Coste total					11.200 €

Tabla 2. Costes humanos del proyecto

- Ocupación: Es el porcentaje de tiempo que cada persona involucrada en el proyecto le ha dedicado a este.
- Coste al mes para la empresa: Este coste es calculado como el salario bruto mensual de cada empleado (dato aproximado) más el 30% de seguridad y prestaciones sociales. [4]
- Coste en el proyecto: Se calcula como la cantidad de empleados necesarios por los meses de duración por la ocupación por el coste al mes para la empresa.

3.4.2.2 Costes de Licencia

Las licencias que han sido necesarias para el desarrollo del proyecto han sido únicamente las de las licencias de Simio ya que Python es un lenguaje de programación de código abierto la cuál es gratuita.

El dato de coste al año de una licencia de Simio para un estudiante es un dato real sacado de la página oficial de Simio. [5]

Costes de licencias				
Licencia	Cantidad	Meses de duración	Coste al año	Coste en el proyecto
Simio	1	6	25 €	13 €
Coste total				13 €

Tabla 3. Coste de licencias del proyecto

- Coste en el proyecto: Se calcula como la cantidad de ordenadores necesarios por los meses de duración por el coste al año entre doce meses.

3.4.2.3 Coste Materiales

Dentro de los costes materiales somos hemos considerado un único ordenador, el cuál hemos buscado su precio actual en páginas oficiales de venta. [6]

Costes materiales						
Objeto	Cantidad	Meses de duración	Coste	% de amortización al año	Coste de amortización del año 1	Coste en el proyecto
Ordenador	1	6	749 €	25%	187 €	94 €
Coste total						94 €

Tabla 4. Coste de materiales del proyecto

- % de amortización: Este porcentaje indica cuanto es el coste anual de este activo en un año fiscal para una empresa. Tras investigar cómo se amortizan los equipos informáticos en las empresas, un 25% de su coste de adquisición es el dato que aplicaremos para realizar nuestra estimación. [7]

3.4.2.4 Costes Totales

Finalmente, los costes totales será un sumatorio de los tres costes vistos previamente.

Coste total del proyecto	
Coste	Cantidad
Costes humanos	11.200 €
Costes de licencias	13 €
Costes materiales	94 €
Coste total	11.306 €

Tabla 5. Costes totales del proyecto

Tras realizar dicha operación podemos concluir que el precio estimado del proyecto será de 11.306 euros.

Capítulo 4. SISTEMA/MODELO DESARROLLADO

En este capítulo se expondrá el caso de uso en el que se basará este trabajo de fin de máster, el diseño de cómo se abordará dicho caso de uso y finalmente se expondrá como se ha implementado la solución.

4.1 ANÁLISIS DEL SISTEMA

El caso de uso que se utilizará se basa en la optimización del almacén de un cliente de Accenture. En este apartado se explicarán los inputs del sistema y los departamentos que tenía este almacén los cuales son los que se tratarán de posicionar dentro del almacén de manera óptima.

4.1.1 DEPARTAMENTOS

- **Muelles de llegada**

Este espacio suele estar situado cerca de la entrada principal del almacén. Puede ocupar aproximadamente entre el 5% y el 10% de la superficie total del almacén, en función del volumen de mercancías que se reciban.

- **Muelles de salida**

Similares a los muelles de llegada, situados en el lado opuesto del almacén o en un lugar estratégico para facilitar la carga de los vehículos de transporte. También puede ocupar entre el 5% y el 10% de la superficie total.



Ilustración 2. Muelles de salida y de entrada

- **Estanterías (Almacén tradicional)**

Distribuidos por la mayor parte del almacén, excluyendo las zonas de muelles. Pueden ocupar entre el 40% y el 60% de la superficie total, en función de la densidad de almacenamiento y la altura de las estanterías.



Ilustración 3. Almacén tradicional

- **ASRS**

Sistema Automático de Almacenamiento y Recuperación. Consiste en sistemas automatizados que manejan, almacenan y recuperan bienes con un mínimo de intervención humana.

Los ASRS se utilizan en una variedad de industrias para mejorar la eficiencia y reducir costos. Puede ocupar entre el 15% y el 25% de la superficie total del almacén.



Ilustración 4. ASRS

- **Área de producción**

En el contexto de un almacén, el área de producción es una zona dedicada a la producción o ensamblaje de productos. Esta área está integrada dentro del almacén y puede beneficiarse significativamente del uso de sistemas automatizados como los ASRS.

Situado en una zona designada, separada de la zona de almacenamiento principal. La proporción dependerá de la relación entre almacenamiento y producción, pero podría ocupar entre el 10% y el 20% de la superficie total.



Ilustración 5. Planta de producción

- **Área de carriles de circulación**

Integrado en el diseño del almacén para permitir un flujo eficaz de productos entre los AGV (vehículos de guiado automático) y las carretillas elevadoras. It can occupy approximately 5% to 10% of the total area, depending on the number of AGVs and forklifts.



Ilustración 6. Área de carriles de circulación

- **Centradora y Envolvedora:** La centradora se utiliza para alinear los distintos productos de un palé y darle una forma rectangular.

Los equipos de envoltura se utilizan para aplicar una película protectora o envoltura alrededor de los productos en un palé para garantizar su estabilidad, seguridad y protección, este departamento no tendrá interacción con el resto de los departamentos. Puede ocupar aproximadamente entre el 5% y el 10% de la superficie total.

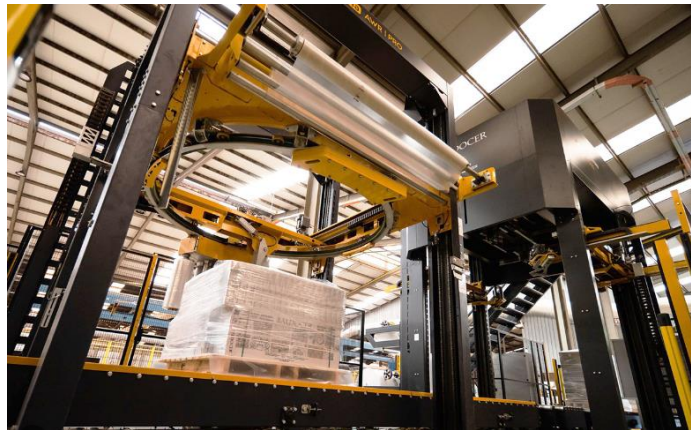


Ilustración 7. Centradora y envolvedora

- **Oficina**

Una zona de oficinas administrativas para gestionar las operaciones, la documentación y coordinar las actividades del almacén.



Ilustración 8. Oficina

4.1.2 INPUTS DEL CASO DE USO

Al tratarse de los datos de un cliente de Accenture los datos han sido tratados y anonimizados para desarrollar este proyecto.

- Dimensiones del almacén.

Facility Dimensions	
width	length
210	240

Tabla 6. Dimensiones del almacén

- Dimensiones de los departamentos.

element	name	surface	width	length
0	Arrival Docks	600	60	10
1	Departure Docks	600	60	10
2	Traditional WH	10200	60	170
3	ASRS	7700	110	70
4	ManufacturingArea	7700	110	70
5	Flow Lanes	130	10	13
6	Office Area	3500	70	50
7	PalletAdjustmentLineIArea	48	4	12
8	PalletAdjustmentLineIIArea	96	8	12

Tabla 7. Dimensiones de departamentos

La distribución que se tenía en este almacén es el que se muestra en la *Ilustración 9*.

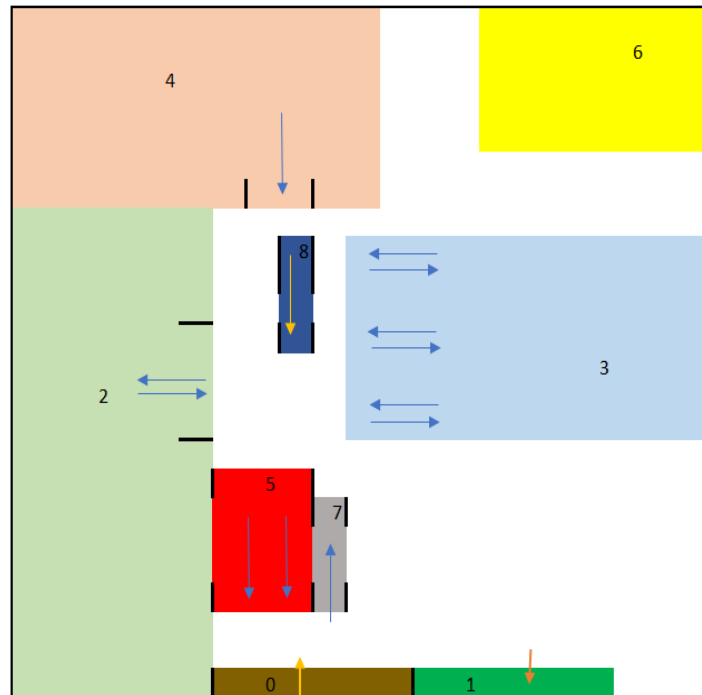


Ilustración 9. Distribución inicial del almacén.

Y por último tenemos las matrices de relación entre departamentos:

- La matriz de flujo hace referencia a las unidades de pallets por hora que van desde un departamento X a un departamento Y.

from \ to	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	25	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	10	0	10	0	0	0
3	0	0	20	0	0	45	0	0	0
4	0	0	0	0	0	6	0	0	45
5	0	61	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	5	20	0	0	0	0	0
8	0	0	10	35	0	0	0	0	0

Tabla 8. Atriz de flujo del almacén

- La matriz de pasillos indica la mínima distancia que debe haber entre un departamento X y un departamento Y.

from \ to	0	1	2	3	4	5	6	7	8
0	0	0	3	3	3	3	3	10	3
1	0	0	3	3	3	3	3	3	3
2	3	3	0	10	3	10	3	3	3
3	3	3	10	0	3	10	3	3	3
4	3	3	3	3	0	5	3	3	5
5	3	10	3	3	3	0	3	3	3
6	15	15	15	15	15	15	0	15	15
7	3	3	10	10	3	3	3	0	3
8	3	3	10	10	3	3	3	3	0

Tabla 9. Matriz de pasillos del almacén

- Los departamentos que deberán ir en una posición específica del almacén.

element	x_start	x_end	y_start	y_end
0	15	74	230	239
1	75	134	230	239

Tabla 10. Departamentos fijos del almacén

- Los puntos de entrada y salida de los departamentos.

element	type	x_en	y_en	x_ex	y_ex
0	EX			30	0
1	EN	30	0		
2	EE	59	60	59	90
3	EE	0	30	0	40
4	EX			60	69
5	EE	4	0	4	12
6	NO				
7	EE	1	11	1	0
8	EE	3	0	3	11

Tabla 11. Puntos de entradas y salidas de los departamentos del almacén

4.2 DISEÑO

El proyecto que se plantea seguirá un diseño como el de la *Ilustración 10*.

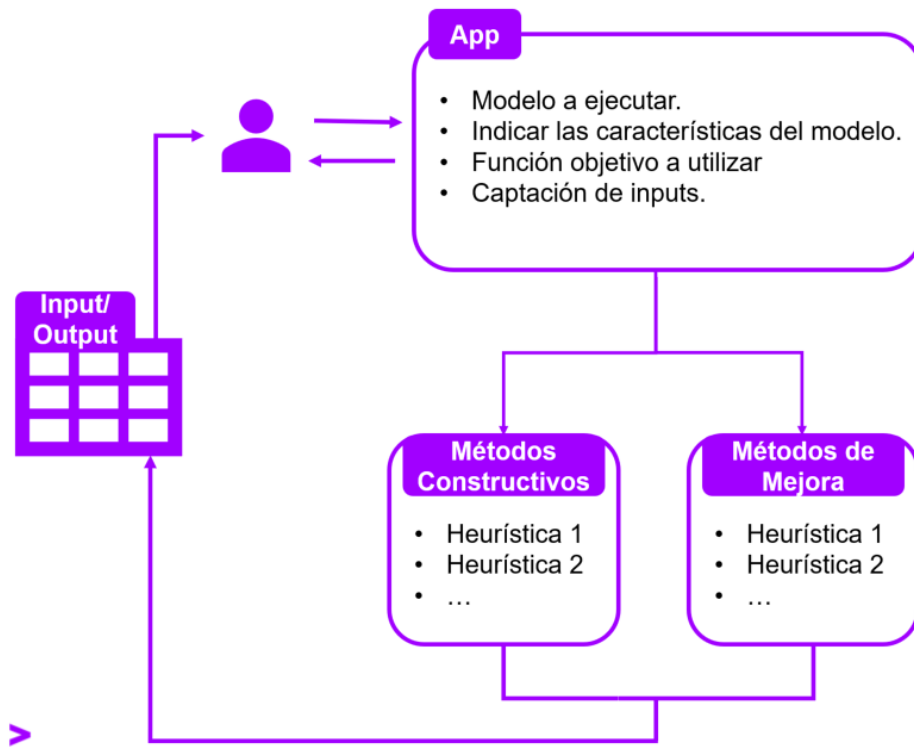


Ilustración 10. Diseño del proyecto

El proyecto deberá tener desplegada una app que sirve de interacción entre los modelos heurísticos y el cliente. Esta app deberá cumplir con los siguientes requisitos:

- Captación de inputs: Deberá ser capaz de ingerir archivos Excel con los inputs del modelo.
- Características del modelo: El cliente podrá elegir las características que quiere que se tengan en cuenta a la hora de realizar la optimización del modelo
- Función objetivo: El cliente deberá indicar la distancia que se quiere medir en la función objetivo.
- Modelo: El cliente debe elegir el modelo que se quiere implementar en la optimización.

Tras haber interactuado con la app el modelo debe ser ejecutado llamado a la clase correspondiente dependiendo si queremos ejecutar un modelo constructivo o un modelo de mejora.

Tras la ejecución se deberá mostrar el resultado de la optimización por la app y guardar los resultados de la optimización.

Una vez que se tiene claro el diseño que va a tener el proyecto procedemos a la implementación de este.

4.3 IMPLEMENTACIÓN

En este apartado se desarrollará como se ha implementado este trabajo de fin de grado, comenzado por la decisión de que modelos desarrollar, siguiendo por el desarrollo de estos y finalizando con la simulación de la solución propuesta tras la optimización.

4.3.1 MODELOS A DESARROLLAR

Tras realizar la investigación sobre las heurísticas existentes para resolver los problemas de diseño de plantas se eligieron dos de ellos para llevar a cabo su implementación, uno de tipo constructivo y otro de mejora.

Como heurística constructiva se eligió el método CORELAP y de mejora el método CRAFT.

- **CORELAP:**

A pesar de ser un algoritmo que teóricamente es cualitativo, puede ser fácilmente convertido a cuantitativo.

De las mayores virtudes de este algoritmo son las siguientes:

- **Flexibilidad:** Es un algoritmo que no tiene excesivas restricciones y da libertad para incorporar mejoras en él.
- **Simplicidad:** Fácil de implementar y de entender.

Otro de los motivos por el que se ha elegido este algoritmo es por comenzar a localizar los departamentos con más flujo sobre los demás en el centro de la superficie y en caso de un almacén encaja bien esa lógica para minimizar las distancias entre los departamentos y llegar a un buen óptimo local.

- CRAFT:

Este algoritmo de mejora es de tipo determinista, es decir que no mete aleatoriedad a la solución y siempre generará la misma solución para la misma entrada y este era un requisito que cumplir en este proyecto.

Una de las ventajas de este algoritmo son las siguientes:

- Flexibilidad: Da la oportunidad de introducir mejoras en él.
- Simplicidad: Fácil de entender e implementar.

Este algoritmo explora nuevas soluciones con el fin de encontrar una que mejore la solución inicial.

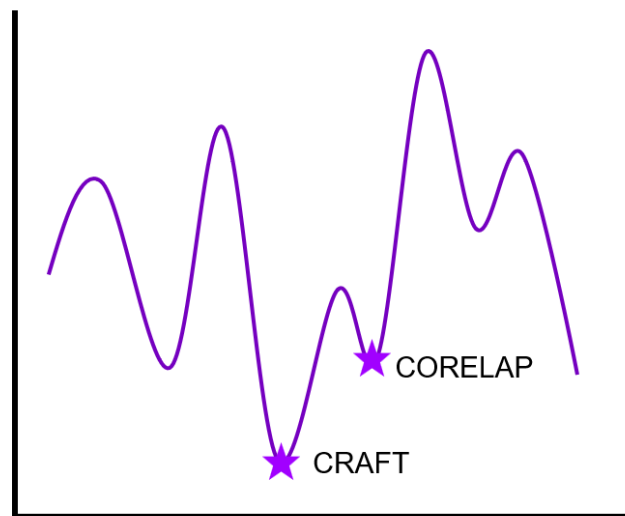


Ilustración 11. Valle de soluciones

4.3.1 CARACTERÍSTICAS DEL MODELO

Como hemos comentado en la sección anterior los modelos escogidos para su desarrollo admiten mejoras en ellos que pueden ayudar al modelo a generar una solución óptima lo más cercana al problema real que se plantea.

Para realizar la explicación de las características del modelo se han realizado los ejemplos con datos distintos a los del caso de uso.

Las características que se han desarrollado para incorporarla a los modelos desarrollados son las siguientes:

- Objetos fijos: Útiles para indicar ubicación específica a elementos en la instalación, y prohibir a la heurística que lo ubique en otro lugar.
- Coste: Penaliza las rutas desde un elemento(x) a un elemento(y) «ejemplo: El transporte es de una carretilla es más caro que el de un AGV Automotive Guided Vehicle)».

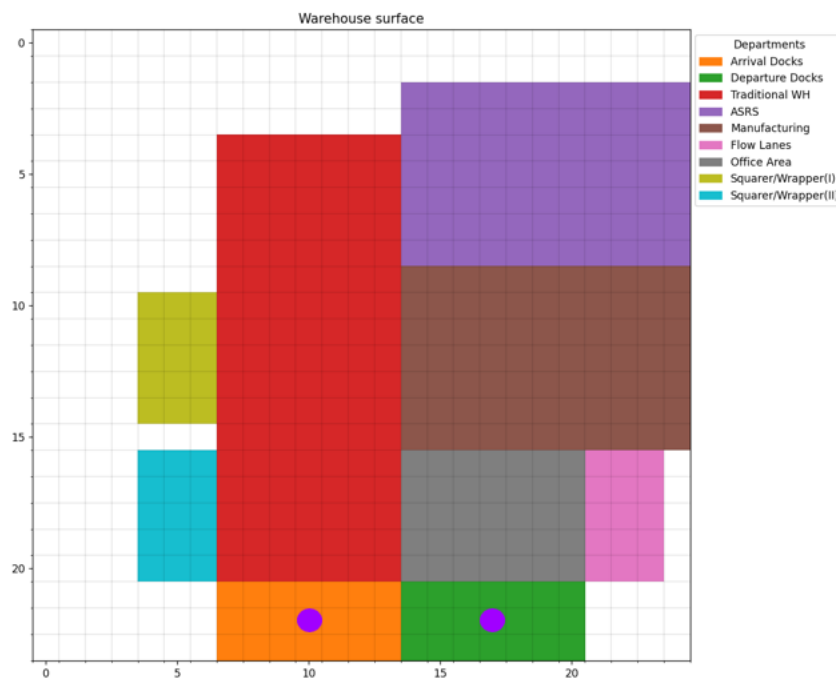


Ilustración 12. Ejemplo de departamentos fijos

- Pasillo: Útil para poner una restricción de distancia mínima entre elementos «ejemplo: La oficina debe estar a una distancia mínima de 10 metros del departamento de fabricación».

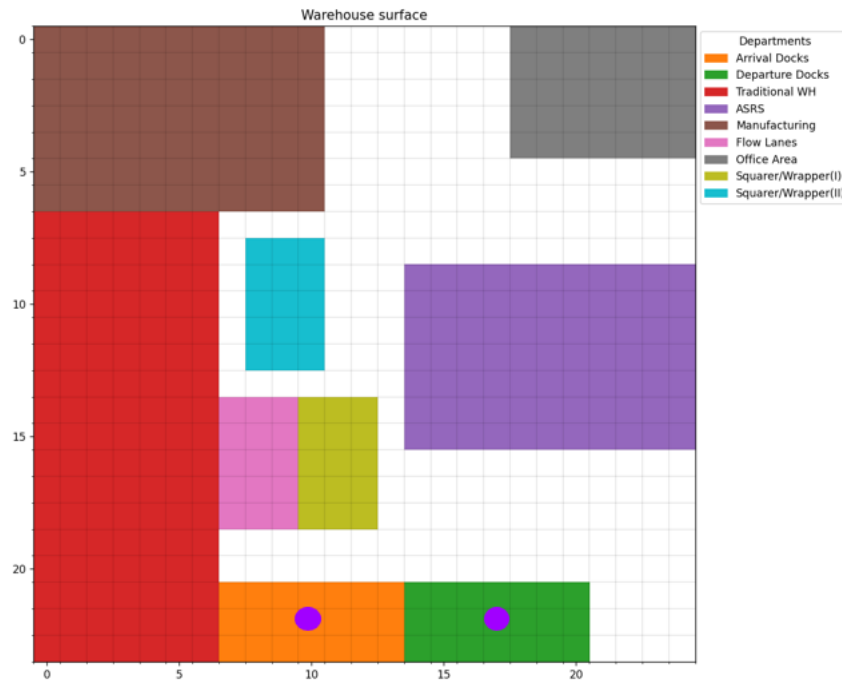


Ilustración 13. Ejemplo de pasillos

Se puede ver que al aplicar esta restricción el Layout varía.

- Puntos de entrada o salida: Esta función se utiliza cuando es necesario indicar puntos de entrada y salida a los elementos del sistema. Para esta función hay 4 escenarios posibles:
 - EE: Existen tanto puntos de entrada como de salida. "ejemplo: Almacén Tradicional".
 - ES: Sólo puntos de entrada. "ejemplo: Muelles de Salida".
 - EX: Sólo puntos de salida. "ejemplo: Muelles de Llegada".
 - None: No hay puntos de entrada ni de salida. "ejemplo: Oficinas".

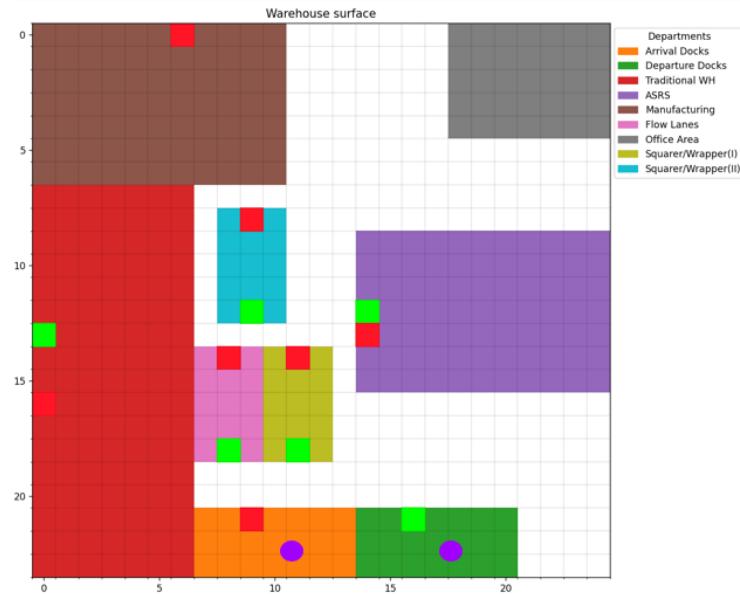


Ilustración 14. Ejemplo de puntos de entradas y salidas.

- Rotación: Permite que los elementos del sistema giren sobre sí mismos (De esta forma la heurística explora nuevas soluciones posibles). Los ángulos que se exploran son 0° , 90° , 180° y 270° .

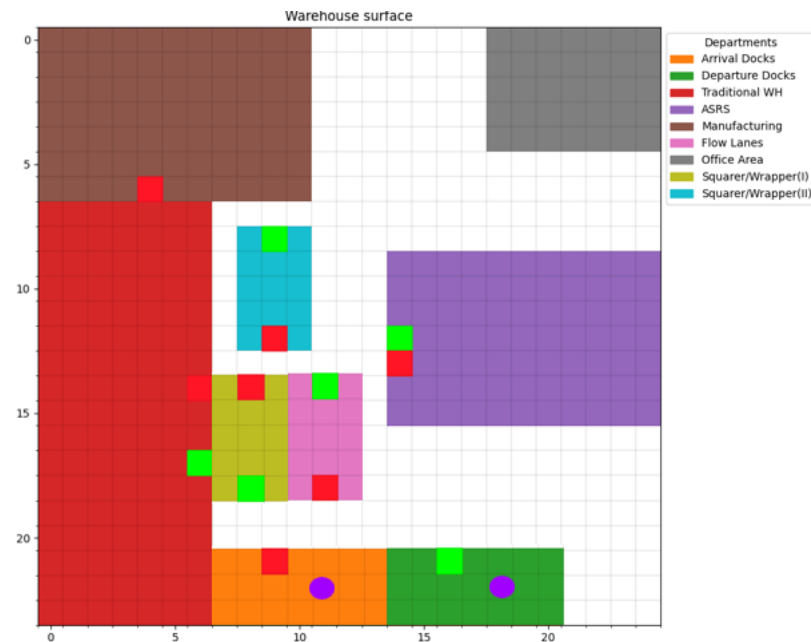


Ilustración 15. Ejemplo de rotación

- Limitaciones: Indicar las zonas de la instalación donde no se pueden localizar elementos. «ejemplo: columnas, escaleras mecánicas, pasillos en ASRS».

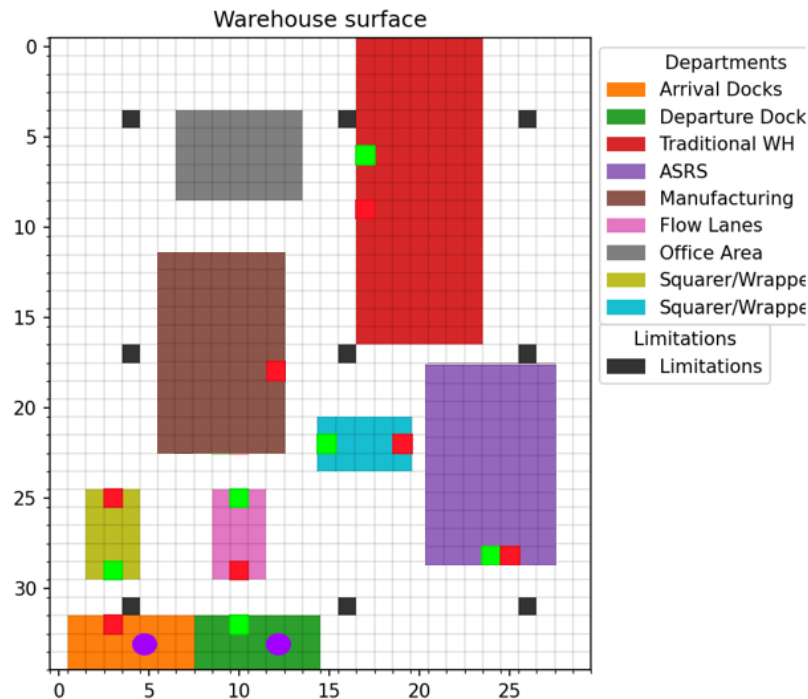


Ilustración 16. Ejemplo de limitaciones

4.3.2 FUNCIÓN OBJETIVO

Se han desarrollado dos tipos de funciones objetivo:

Euclídea: La distancia euclídea es una medida de la distancia en línea recta entre dos puntos en el espacio euclídeo. Se deriva del teorema de Pitágoras y es la métrica de distancia más utilizada.

Manhattan: La distancia Manhattan, también conocida como distancia L1 o distancia taxi, es una medida de distancia entre dos puntos en un sistema basado en cuadrículas. A diferencia de la distancia euclidiana, que mide el camino más corto (línea recta), la distancia Manhattan mide la suma de las diferencias absolutas de sus coordenadas.

Por favor, tenga en cuenta que si la característica Punto de Flujo no está seleccionada las distancias van a ser calculadas desde los centroides de los elementos con flujo entre ellos,

pero si está seleccionada la distancia va a ser calculada desde los puntos de entrada a los puntos de salida de los elementos con flujo entre ellos.

También si se selecciona la característica Coste la unidad de medida de las funciones objetivas es (dinero por tiempo), pero si no se selecciona la unidad de medida será (distancia por tiempo).

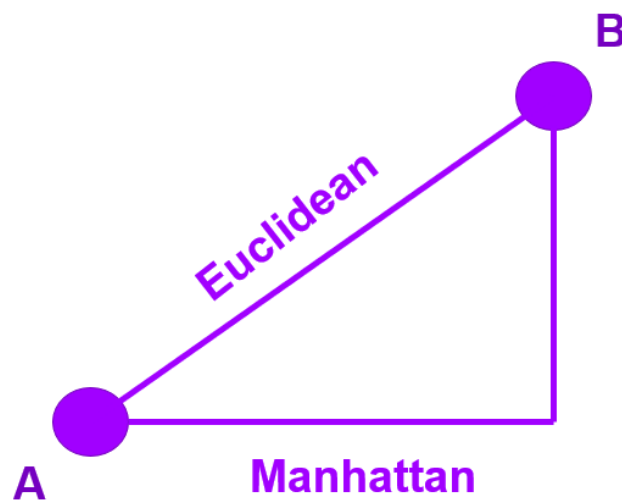


Ilustración 17. Distancias de la función objetivo

4.3.3 INGESTA DE DATOS

La ingesta de datos se realizará a través de Excel. Se realizará de esta manera ya que facilita el proceso humano de meter los inputs, siendo más clara y efectiva y evitar mala ingesta de estos.

Este archivo actuará como archivo de inputs y outputs a la vez como se verá a continuación.

Los inputs de ejemplo en las figuras de esta sección son diferentes al del caso de uso para una mayor comprensión de cómo se deben introducir de manera correcta.

Los inputs a introducir para crear una instancia son los siguientes:

- Dimensiones del Almacén: Estos deberán ser números enteros y en la misma unidad de medida en la que estarán las dimensiones de los departamentos.

Facility Dimensions	
width	length
30	35

Tabla 12. Ejemplo de inputs dimensiones del almacén

- Indicador del elemento: Se deberá indicar un indicador para cada departamento, estos indicadores deben ser números enteros consecutivos.
- Nombre del elemento: Será deberá indicar también el nombre de los departamentos.
- Áreas de los elementos: Las áreas de los departamentos deberán de indicarse (se calculan como $\text{área} = \text{largo} * \text{ancho}$).
- Ancho y Largo: Las dimensiones del departamento serán necesarias (se entiende que los departamentos son rectángulos).

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	element	name	surface	width	length	centroid_x	centroid_y	angles											
2	0	Arrival Docks	12	7	3	4	33	0											
3	1	Departure Docks	12	7	3	11	33	0											
4	2	Traditional WH	102	7	17	16	22	0											
5	3	ASRS	77	11	7	5	17	180											
6	4	Manufacturing	77	11	7	14	7	270											
7	5	Flow Lanes	15	3	5	8	28	90											
8	6	Office Area	35	7	5	26	22	0											
9	7	PalletAdjustmentLine1	4	3	5	3	27	0											
10	8	PalletAdjustmentLine1	8	3	5	7	8	180											
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			



Tabla 13. Ejemplo de inputs de elementos del sistema

Las columnas en morado sólo se completarán si se va a utilizar un algoritmo de mejora a partir de un layout existente.

- Cuando se optimice con un algoritmo de mejora una layout inicial, se guardará en estas columnas la nueva solución optimizada «sobrescribiendo los datos».
- Si no había datos en estas columnas porque se está utilizando un algoritmo constructivo, la solución se escribirá en estas columnas.

Outputs:

- Centroides: En estas columnas se almacenará la posición en el eje x y en el eje y del elemento una vez optimizada su posición.
- Ángulos: Esta columna indica el nuevo ángulo en el que se ha colocado el elemento tras la optimización. Si el elemento no ha sido rotado, el ángulo del elemento será 0°. Los ángulos posibles son los siguientes:
 - o 0°
 - o 90°
 - o 180° (Solo si puntos de flujo son especificados en los inputs)
 - o 270° (Solo si puntos de flujo son especificados en los inputs)

- Matriz de flujos: En esta matriz se indican las relaciones de flujo entre los elementos del sistema de la siguiente manera:
 - En la columna A y en la fila 1 deben indicarse los identificadores de los elementos.
 - En cada celda debe figurar el número de unidades que se transportan del elemento A al elemento B en una unidad de tiempo (ejemplo: Número de pallets que se trasladan de un Muelle de Salida a un ASRS en 1 hora).
 - La diagonal de la matriz debe ser siempre 0.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	from \ to	0	1	2	3	4	5	6	7	8								
2	0	0	0	0	0	0	0	0	25	0								
3	1	0	0	0	0	0	0	0	0	0								
4	2	0	0	0	10	0	10	0	0	0								
5	3	0	0	20	0	0	45	0	0	0								
6	4	0	0	0	0	0	6	0	0	45								
7	5	0	61	0	0	0	0	0	0	0								
8	6	0	0	0	0	0	0	0	0	0								
9	7	0	0	5	20	0	0	0	0	0								
10	8	0	0	10	35	0	0	0	0	0								
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		

Tabla 14. Ejemplo de input de la matriz de flujo

Hasta aquí todos los inputs que se han indicado son estrictamente necesarios para el modelo teniendo en cuenta como se ha indicado anteriormente que las columnas de outputs deberán ser completadas en caso de que se requiera hacer una optimización con una heurística de mejora sobre un layout inicial.

Los inputs que se van a mostrar ahora son opcionales, y estos dependen de las características que se le quiera meter al modelo. Estos son los inputs opcionales:

- Matriz de coste: En esta matriz se indicará la relación de costes entre los elementos del sistema de la siguiente manera:

- En la columna A y en la fila 1 deben indicarse los identificadores de los elementos.
- En cada celda debe figurar el coste de transportar una unidad del elemento transportable en el sistema (ejemplo: Palets) desde el elemento A al elemento B. (ejemplo: Desde un Muelle de Salida a un ASRS me cuesta 1 euro transportar un palet en un AGV, pero desde el departamento de Fabricación a una instalación tradicional me cuesta 5 euros transportar el palet en una Carretilla Elevadora).
- La diagonal de la matriz debe ser siempre 0.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	from \ to	0	1	2	3	4	5	6	7	8								
2	0	0	1	1	1	1	1	1	1	100	1							
3	1	1	0	1	1	1	1	1	1	1	1							
4	2	1	1	0	1	1	1	1	1	1	1							
5	3	1	1	1000	0	0	1	1	1	1	1							
6	4	1	1	1	1	0	1	1	1	1	1							
7	5	1	100	1	1	1	0	1	1	1	1							
8	6	1	1	1	1	1	1	0	1	1	1							
9	7	1	1	1	1	1	1	1	0	1	1							
10	8	1	1	1	1	1	1	1	1	0	1							
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		

Tabla 15. Ejemplo de input de la matriz de coste

- Matriz de pasillos: En esta matriz se indicará la relación de costes entre los elementos del sistema de la siguiente manera:
 - En la columna A y en la fila 1 deben indicarse los identificadores de los elementos.
 - En cada celda debe figurar el coste de transportar una unidad del elemento transportable en el sistema (ejemplo: Palets) desde el elemento A al elemento B. (ejemplo: Desde un Muelle de Salida a un ASRS me cuesta 1 euro transportar un palet en un AGV, pero desde el departamento de Fabricación

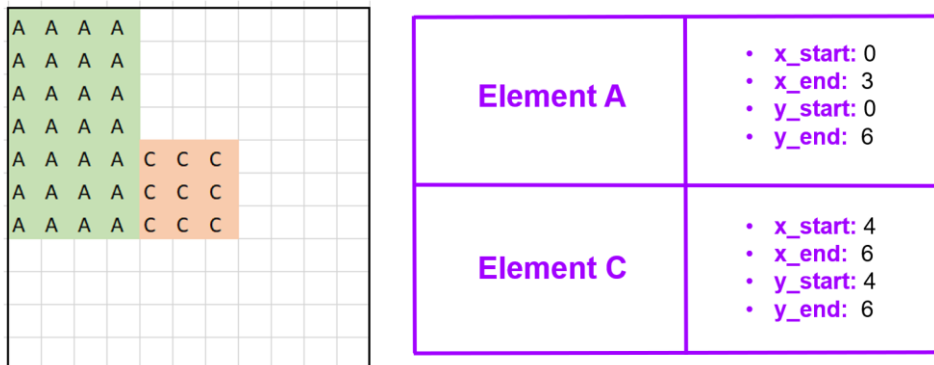
a una instalación tradicional me cuesta 5 euros transportar el palet en una Carretilla Elevadora).

- La diagonal de la matriz debe ser siempre 0.
- Los pasillos deben ser un número entero y deben estar en la misma unidad que las dimensiones de los elementos y de la superficie de la instalación. (Ejemplo: Entre el departamento de Oficina y la zona de fabricación debe haber una distancia mínima de 10 metros).

from \ to	0	1	2	3	4	5	6	7	8									
0	0	0	1	1	1	1	1	2	1									
1	0	0	1	1	1	1	1	1	1									
2	1	1	0	2	1	2	1	1	1									
3	1	1	2	0	1	2	1	1	1									
4	1	1	1	1	0	1	1	1	1									
5	1	2	1	1	1	0	1	1	1									
6	3	3	3	3	3	3	0	3	3									
7	1	1	2	2	1	1	1	0	1									
8	1	1	2	2	1	1	1	1	0									

Tabla 16. Ejemplo de input de la matriz de pasillos

- Elementos fijos: En esta hoja del Excel se indicarán las posiciones fijas de los departamentos si la característica de departamentos fijos es incluida en el modelo de optimización. De esta forma no permitiremos que el algoritmo de optimización lo ubique en otro lugar.



START COUNTING FROM THE TOP LEFT CORNER !

Tabla 17. Ejemplo de cómo introducir un departamento fijo

Las posiciones se empiezan a contar desde la esquina superior izquierda del layout.

- Elemento: Establece el indicador del elemento a fijar.
- x_start: Posición donde comienza el elemento en el eje x (La posición mínima que existe en el almacén es 0).
- x_end: Posición donde termina el elemento en el eje x (La posición máxima que existe en la instalación es el ancho del almacén - 1).
- y_start: Posición donde comienza el elemento en el eje y (La posición mínima que existe en el almacén es 0).
- y_end: Posición donde termina el elemento en el eje y (La posición máxima que existe en el almacén es la longitud de la instalación - 1).

Se debe ser coherente con la anchura y longitud especificadas en la página ELEMENTOS (ejemplo: anchura del elemento = 5, x_start = 0 y x_end = 3 NO ES VÁLIDO).

En el Excel se deberá indicar de la siguiente manera.

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
2	element	x_start	x_end	y_start	y_end													
3	0	1	7	32	34													
4	1	8	14	32	34													
5																		
6																		
7																		
8																		
9																		
10																		
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		

Tabla 18. Ejemplo de input de los departamentos fijos

- En caso de que la característica de puntos de flujo se quiera incluir en la optimización, estos puntos se deberán indicar de la siguiente manera:
 - Punto de entrada: Es el punto verde claro.
 - Punto de salida: Es el punto rojo claro.

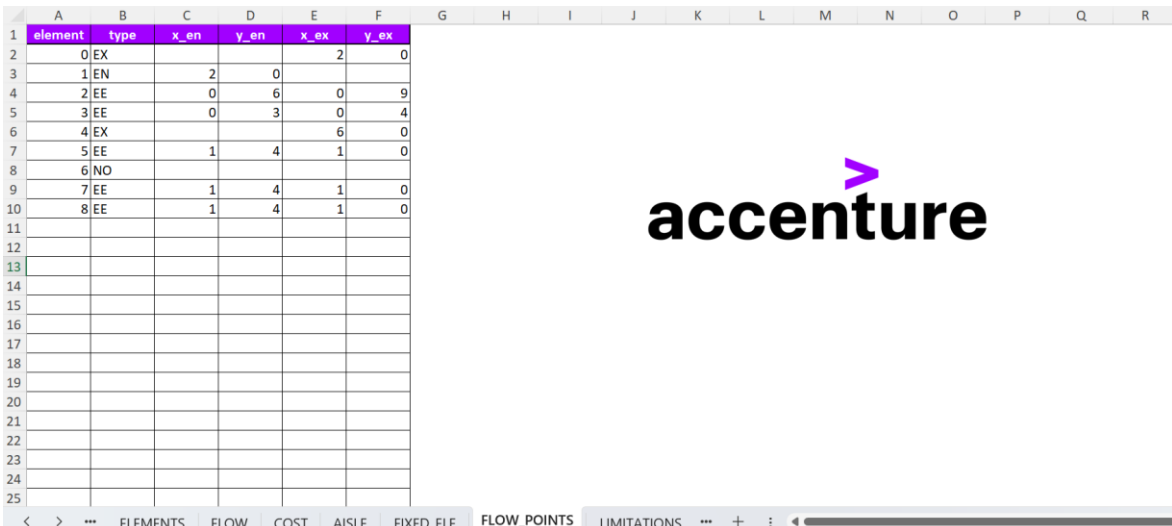
EE: Entry and Exit point.	EX: Exit point.	EN: Entry point.	None: No flow point.
<ul style="list-style-type: none"> • x_en: 3 • y_en: 10 • x_ex: 4 • y_ex: 10 • x_en: 0 • y_en: 1 • x_ex: 4 • y_ex: 1 	<ul style="list-style-type: none"> • x_en: • y_en: • x_ex: 6 • y_ex: 6 	<ul style="list-style-type: none"> • x_en: 2 • y_en: 0 • x_ex: • y_ex: 	<ul style="list-style-type: none"> • x_en: • y_en: • x_ex: • y_ex:

START COUNTING FROM THE TOP LEFT CORNER !

Tabla 19. Ejemplo de cómo introducir los puntos de entrada y salida

- x_es: Ubicación en el eje X del punto de entrada (La posición mínima que existe en el elemento es 0 y la posición máxima que existe en el elemento es el ancho del elemento - 1).

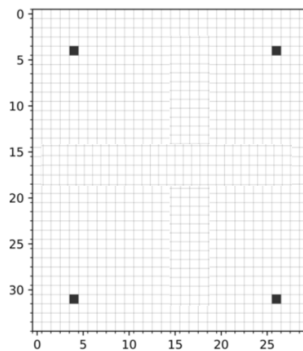
- y_{es} : Posición en el eje Y del punto de entrada (La posición mínima que existe en el elemento es 0 y la posición máxima que existe en el elemento es la longitud del elemento - 1).
- x_{ex} : Posición en el eje X del punto de salida (La posición mínima que existe en el elemento es 0 y la posición máxima que existe en el elemento es la anchura del elemento - 1).
- y_{ex} : Posición en el eje Y del punto de salida (La posición mínima que existe en el elemento es 0 y la posición máxima que existe en el elemento es la longitud del elemento - 1).



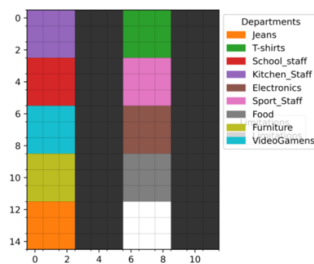
element	type	x_en	y_en	x_ex	y_ex
0	EX			2	0
1	EN	2	0		
2	EE	0	6	0	9
3	EE	0	3	0	4
4	EX			6	0
5	EE	1	4	1	0
6	NO				
7	EE	1	4	1	0
8	EE	1	4	1	0

Tabla 20 Ejemplo de input de los puntos de entrada y salida

- Limitaciones: En caso de que esta característica sea necesaria en el modelo a la hora de optimizar se deberán de indicar de la siguiente manera:



Col. Upper Left	<ul style="list-style-type: none"> • x_start: 4 x_end: 4 • y_start: 4 y_end: 4
Col. Upper Right	<ul style="list-style-type: none"> • x_start: 26 x_end: 26 • y_start: 4 y_end: 4
Col. Down Left	<ul style="list-style-type: none"> • x_start: 4 x_end: 4 • y_start: 31 y_end: 31
Col. Down Right	<ul style="list-style-type: none"> • x_start: 26 x_end: 26 • y_start: 31 y_end: 31



ASRS transit zone 1	<ul style="list-style-type: none"> • x_start: 3 x_end: 5 • y_start: 0 y_end: 29
ASRS transit zone 2	<ul style="list-style-type: none"> • x_start: 9 x_end: 11 • y_start: 0 y_end: 29

START COUNTING FROM THE TOP LEFT CORNER !

Tabla 21. Ejemplo de cómo introducir las limitaciones

Las posiciones del almacén se empiezan a contar desde la esquina superior izquierda.

En el Excel estas limitaciones se deberán indicar de la siguiente manera:

- Nombre: Nombre de la limitación (ejemplo: Columna, escalera mecánica, ascensor...)
- x_start: Posición donde comienza la limitación en el eje x (La posición mínima que existe en la instalación es 0).
- x_end: Posición donde termina la limitación en el eje x (La posición máxima que existe en la instalación es el ancho de la instalación - 1).
- y_start: Posición donde comienza la limitación en el eje y (La posición mínima que existe en la instalación es 0).
- y_end: Posición donde termina la limitación en el eje y (La posición máxima que existe en la instalación es la longitud de la instalación - 1).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	name	x_start	x_end	y_start	y_end													
2	Limitation 1	4	4	4	4													
3	Limitation 2	4	4	31	31													
4	Limitation 3	26	26	4	4													
5	Limitation 4	26	26	31	31													
6	Limitation 5	16	16	4	4													
7	Limitation 6	16	16	31	31													
8	Limitation 7	4	4	17	17													
9	Limitation 8	16	16	17	17													
10	Limitation 9	26	26	17	17													
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		



Tabla 22. Ejemplo de input de limitaciones

4.3.4 MÉTODOS CONSTRUCTIVOS

Como se ha comentado anteriormente en punto 4.3.1 el método constructivo a desarrollar será el CORELAP.

Los inputs que se necesitan para poder ejecutar la heurística son las siguientes:

- Dimensiones del Almacén.
- Nombre e indicadores de los departamentos a localizar.
- Dimensiones de los departamentos.
- Matriz de flujo.

4.3.4.1 Funcionamiento de la heurística.

El funcionamiento de la heurística de CORELAP resume su funcionamiento en los siguientes pasos:

1. Ordena los elementos del sistema en una secuencia donde el primero sea el que tenga mayor flujo sobre los demás elementos, y debe colocarse en el centro de la superficie de la instalación.
2. Completa la secuencia, el siguiente elemento de las secuencias será el que tenga mayor flujo con respecto a los elementos ya localizados en la superficie.

3. Busca en la superficie todas las ubicaciones posibles para ubicar el siguiente elemento y elige las localizaciones que minimicen la función objetivo.
4. Comprueba que se cumplan las restricciones de solape y de fuera de rango.
5. Localiza el departamento.
6. Repite el paso 3, 4 y 5 hasta que todos los elementos del sistema estén posicionados.

4.3.4.1.1 Formación de la secuencia

Como se ha comentado anteriormente esta heurística tiene un carácter cualitativo a la hora de construir la secuencia en la que se van a posicionar los departamentos en el almacén.

Este método propone construir la secuencia haciendo uso del TCR (Total Closeness Ratio), el cual es un índice que indica la relación de flujo de cada departamento sobre los demás siendo este más alto cuando la relación es mayor.

Las relaciones de importancia entre los departamentos se definen de la siguiente manera:

$$A=125, E=25, I=5, O=1, U=0, X=-125$$

Ilustración 18. Valores cualitativos del TCR

Esta relación de importancia establecida por los creadores del método CORELAP es subjetiva y lo solían definir los encargados de la planta los cuales tienen el mayor conocimiento de esta. Cada letra representa lo siguiente:

- A = Relación Absolutamente necesaria departamento X y otro departamento Y.
- E = Relación Especialmente importante entre un departamento X y un departamento Y.
- I = Relación Importante entre un departamento X y un departamento Y.
- O = Relación Ordinariamente cercana entre un departamento X y un departamento Y.
- U = Relación no importante (Unimportant) entre un departamento X y un departamento Y.

- X = Relación Indeseable entre un departamento X y un departamento Y.

Estas relaciones se recogen en una matriz donde el número de filas y de columnas lo determina el número de departamentos que tenga el almacén.

Y tras sumar por fila los valores de las relaciones que tengan entre todas se establece la secuencia en la que van a ser localizados en el almacén. Siendo el primero el de mayor relación con el resto de los departamentos (Ver *Tabla 23*, ejemplo con datos distinto al caso de uso).

Dept.	Department relationships									Summary						TCR	Placement Sequence
	1	2	3	4	5	6	7	8	9	A	E	I	O	U	X		
1	-	A	A	E	O	U	U	A	O	3	1	-	2	2	-	402	5
2	A	-	E	A	U	O	U	E	U	2	2	-	1	3	-	301	7
3	A	E	-	E	A	U	U	E	A	3	3	-	-	2	-	450	4
4	E	A	E	-	E	O	A	E	U	2	4	-	1	1	-	351	6
5	O	U	A	E	-	A	A	O	A	4	1	-	2	1	-	527	2
6	U	O	U	O	A	-	A	O	O	2	-	-	4	2	-	254	8
7	U	U	U	A	A	A	-	X	A	4	-	-	3	1	625	1	
8	A	E	E	E	O	O	X	-	X	1	3	-	2	-	2	452	9
9	O	U	A	U	A	O	A	X	-	3	-	-	2	2	1	502	3

The placement sequence: **7-5-9-3-1-4-2-6-8**

Tabla 23. Tabla de cálculo cualitativo del TCR

Este método cualitativo y basado en la opinión de un experto para establecer la relación de los departamentos ha sido modificado para que este sea cuantitativo y que la relación entre los departamentos sea establecida por un índice medible y objetivo.

Este nuevo TCR cuantitativo se calcula en base a la matriz de flujo de los departamentos de la siguiente manera. Los datos de este ejemplo son distintos al de caso de uso con motivo de facilitar la comprensión de la explicación.

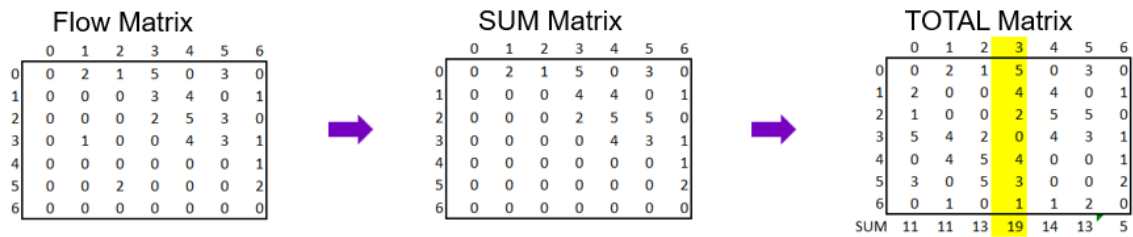


Ilustración 19. Matriz total de flujo

La matriz de flujo se sumará para tener el flujo total entre departamentos y posteriormente se sumarán por filas o por columnas y el departamento cuyo sumatorio sea el mayor este será el primer departamento que colocar ver *Ilustración 19*.

Para colocar el resto de los departamentos se utiliza la misma lógica, pero solo teniendo en cuenta el flujo entre los departamentos que ya han sido colocados. Para entender mejor ver *Ilustración 20*.

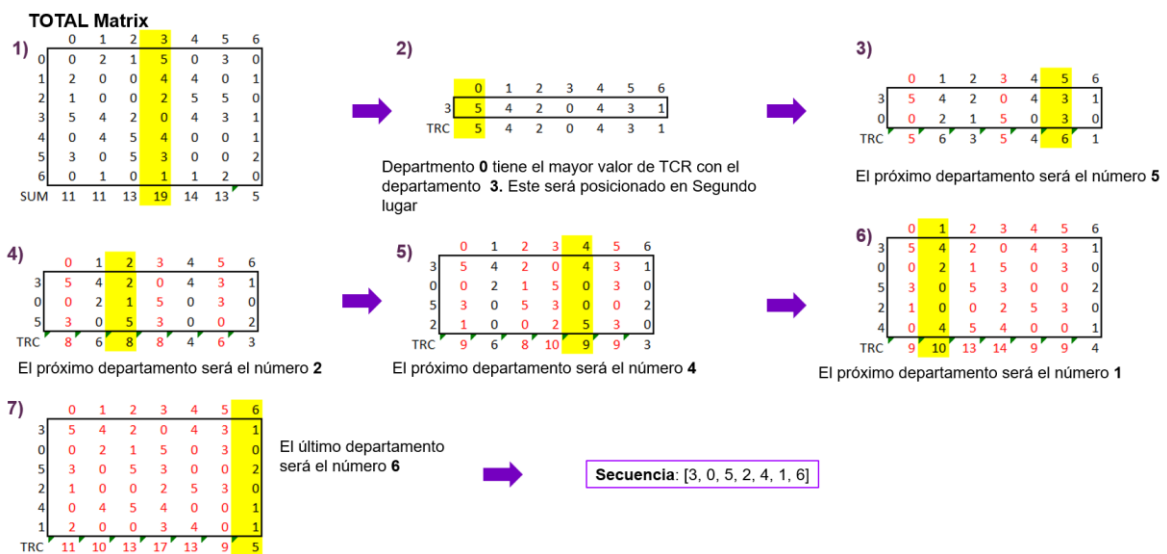


Ilustración 20. Creación de la secuencia

4.3.4.1.2 Restricciones de solape y la de fuera de rango

Tras tener construida la secuencia, antes de localizar a los departamentos en la superficie del almacén deberán cumplir con las restricciones de solape con otros departamentos y de fuera

de rango con los límites del almacén. Estas restricciones se traducen a la siguiente formulación.

- Restricciones de solape:

- Subíndice:

$$i = (0 \dots |\text{departamentos}| - 1)$$

dir = binario, 0 si la dirección es sobre el eje x y 1 si es sobre el eje y

- Variables:

$position_nextdep_{dir}$ = Es la posición donde se va a localizar el centroide del siguiente departamento de la secuencia en el layout en la dirección dir ($dir=0$ es la coordenada X y $dir=1$ es la coordenada Y del layout)

$locatedpos_{i\ dir}$ = Posición del centroide departamento i en el layout en la dirección dir ($dir=0$ es la coordenada X y $dir=1$ es la coordenada Y del layout)

$dim_nextdep_{dir}$ = Dimensión del siguiente departamento que se va a localizar el siguiente departamento de la secuencia en el layout en la dirección dir ($dir = 0$ es el ancho del departamento y $dir = 1$ es el largo del departamento)

$dim_{i\ dir}$ = Dimensión del departamento i en la dirección dir ($dir = 0$ es el ancho del departamento y $dir = 1$ es el largo del departamento)

- Restricción:

$$\forall_{i, dir} \ position_nextdep_{dir} + \frac{(dim_nextdep_{dir})}{2} < locatedpos_{i\ dir} - \frac{(dim_{i\ dir})}{2} \text{ OR } position_nextdep_{dir} - \frac{(dim_nextdep_{dir})}{2} > locatedpos_{i\ dir} + \frac{(dim_{i\ dir})}{2}$$

Si la característica de pasillos es activa la restricción pasa a ser la siguiente:

- Subíndice:

$$nextdep = (0 \dots |\text{departamentos}| - 1)$$

- Variable:

$pasillo_{i\ nextdep}$ = Distancia de separación entre el departamento i y el departamento $nextdep$.

- Restricción:

$$\forall_{i, \ dir} \ position_nextdep_{dir} + \frac{(dim_nextdep_{dir})}{2} + pasillo_{i\ nextdep} < locatedpos_{i\ dir} - \frac{(dim_{i\ dir})}{2} \ OR \ position_nextdep_{dir} - \frac{(dim_nextdep_{dir})}{2} > locatedpos_{i\ dir} + \frac{(dim_{i\ dir})}{2}$$

- Restricciones de fuera de rango:

- Variable:

wh_limit_{dir} = Es el límite de la superficie almacén en la dirección dir ($dir = 0$ el ancho del almacén y $dir = 1$ el largo el almacén)

$$\forall_{dir} \ 0 \leq position_nextdep_{dir} + \frac{(dim_nextdep_{dir})}{2} < wh_limit_{dir}$$

- Restricciones de límites

Estas restricciones se deberán cumplir si la característica de límites está activada en el modelo.

- Subíndice:

$$lim = (0 \dots |límites| - 1)$$

- Restricción:

$$\forall_{lim, \ dir} \ position_nextdep_{dir} + \frac{(dim_nextdep_{dir})}{2} < locatedpos_{lim\ dir} - \frac{(dim_{lim\ dir})}{2} \ OR \ position_nextdep_{dir} - \frac{(dim_nextdep_{dir})}{2} > locatedpos_{lim\ dir} + \frac{(dim_{lim\ dir})}{2}$$

4.3.4.1.3 Función objetivo

La función objetivo de este problema tendrá diferentes variantes dependiendo si la característica de coste se ha requerido en la optimización del problema y también si se quiere que las distancias se midan con la distancia Manhattan o como una distancia Euclídea.

Subíndices:

$$i = (0 \dots |\text{departamentos}| - 1)$$

$$j = (0 \dots |\text{departamentos}| - 1)$$

Variables:

$distancia_{x_{ij}}$ = Es la distancia sobre el eje X desde el departamento i al departamento j

$distancia_{y_{ij}}$ = Es la distancia sobre el eje Y desde el departamento i al departamento j

TCR_{ij} = Es el flujo total que existe entre el departamento i y el departamento j

$Coste_{ij}$ = Es el coste máximo de transporte que existe entre el departamento i y el departamento j

- Distancia Manhattan y coste no aplicado

$$\sum_{i=0}^m \sum_{j=0 | i \neq j}^m |distancia_{x_{ij}} + distancia_{y_{ij}}| \times TCR_{ij}$$

- Distancia Manhattan y coste aplicado

$$\sum_{i=0}^m \sum_{j=0 | i \neq j}^m |distancia_{x_{ij}} + distancia_{y_{ij}}| \times TCR_{ij} \times Coste_{ij}$$

- Distancia Euclídea y coste no aplicado

$$\sum_{i=0}^m \sum_{j=0 | i \neq j}^m \sqrt{(distancia_{x_{ij}})^2 + (distancia_{y_{ij}})^2} \times TCR_{ij}$$

- Distancia Euclídea y coste aplicado

$$\sum_{i=0}^m \sum_{j=0 | i \neq j}^m \sqrt{(distancia_{x_{ij}})^2 + (distancia_{y_{ij}})^2} \times TCR_{ij} \times Coste_{ij}$$

4.3.4.1.4 Posicionamiento de elementos del sistema.

Tras haber definido las restricciones que deberán cumplir los elementos del sistema y disponer de una secuencia de construcción se procederá al posicionamiento de los departamentos en el almacén en las posiciones donde minimicen la función objetivo.

Este procedimiento se ha realizado de la siguiente manera:

Tomaremos los siguientes datos para explicar el método de localización de elementos del sistema. Estos son distintos al del caso de uso con motivo de facilitar la comprensión de la explicación.

Datos Previos

Secuencia: [3, 0, 5, 2, 4, 1, 6]

$TCR_{3-5} = 3$	Dep 3: C
$TCR_{0-5} = 3$	Dep 0: A
	Dep 5: E

width_dep0 = 3
length_dep0 = 3

width_dep5 = 3
length_dep5 = 7

Ilustración 21. Explicación de CORELAP, datos de ejemplo

1. El primer departamento de la secuencia será colocado en el centro de la superficie del almacén.
2. Se ven cuales son las posibles localizaciones donde podremos posicionar el siguiente departamento respetando las restricciones calculando siempre desde el centroide del siguiente departamento.

	0	1	2	3	4	5	6	7
0								
1		x	x	x	x	x	x	
2		x						
3		x		C	C	C		
4		x		C	C	C		
5		x		C	C	C		
6		x						
7		x	x	x	x	x	x	
8								

Ilustración 22. Explicación de CORELAP, posibles posiciones

3. En las localizaciones donde puede ser localizada el siguiente departamento se le calcula la función objetivo teniendo en cuenta la distancia manhattan, y posicionaremos el siguiente departamento en la posición donde el valor de la función objetivo es mínimo.

	0	1	2	3	4	5	6	7
0								
1		24	21	18	15	18	21	
2		21						
3		18		C	C	C		
4		15		C	C	C		
5		18		C	C	C		
6		21						
7		24	21	18	15	18	21	
8								

Ilustración 23. Explicación de CORELAP, posiciones que minimizan la función objetivo

$$FO = |DistanciaX_{C-A} + DistanciaY_{C-A}| * TCR_{C-A}$$

4. Se localiza el siguiente departamento en una de las posiciones donde el valor de la función objetivo es mínimo.

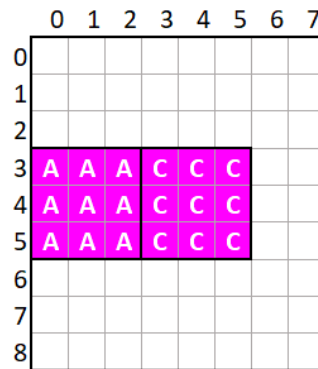


Ilustración 24. Explicación de CORELAP, localización del siguiente departamento

5. Repetir este proceso para el resto de los departamentos de la secuencia (ejemplo para el posicionamiento del departamento 5).

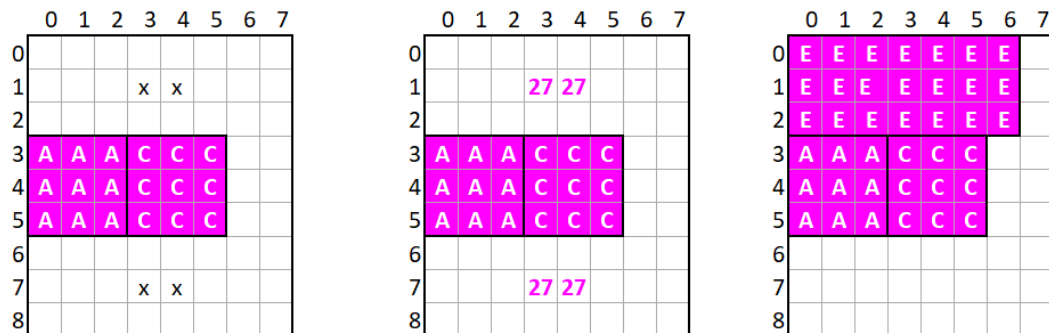


Ilustración 25. Ejemplo de CORELAP del ciclo completo con el siguiente departamento

$$FO = (|DistanciaX_{C-E} + DistanciaY_{C-E}| * TCR_{C-E}) + (|DistanciaX_{A-E} + DistanciaY_{A-E}| * TCR_{A-E})$$

4.3.4.2 Robustecimiento

4.3.4.2.1 Secuencias factibles.

Uno de los problemas que se detectaron fue que en algunos casos la secuencia generada, a la hora de posicionar los departamentos en la superficie, daba soluciones no factibles (ver *Ilustración 26*, el ejemplo de la figura es diferente al caso de uso con motivo de una mejor comprensión de la explicación).

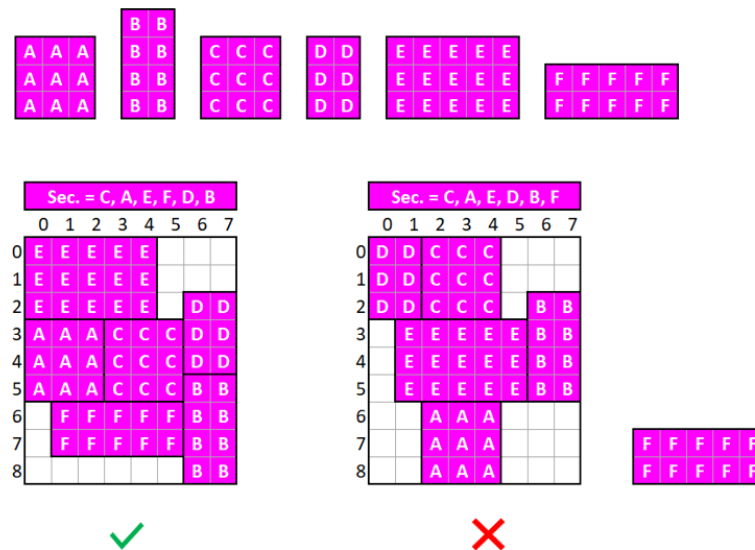


Ilustración 26. Ejemplo de secuencias factibles e infactibles

Para mitigar dicho problema y hacer el código más robusto en el aspecto de factibilidad se crearon nuevos constructores de secuencias con otras reglas diferentes al del TCR. De manera que el algoritmo probará con más secuencias para encontrar una solución factible.

Las reglas utilizadas para generar el resto de los constructores de secuencias son los siguientes:

- TCR y áreas de menor a mayor: Esta secuencia será la primera que se probará ya que es la que mejor solución da. Esta ordenar por relación de flujo y además ponderar positivamente los departamentos que tenga menos área para evitar que su propia área separe más los departamentos adyacentes a él.
- TCR y áreas de menor a mayor: Esta secuencia de media no suele dar tan buenas soluciones, pero sí que es más propensa a que la solución sea factible. Y las soluciones siguen siendo relativamente buenas ya que sigue teniendo en cuenta el flujo entre departamentos.
- Áreas de menor a mayor: Solo tiene en cuenta las áreas por lo que las soluciones son medianamente buenas y factibles ya que deja los departamentos más grandes para el final que es cuando hay menos huecos en el layout.
- Áreas de mayor a menor: Genera soluciones muy malas pero muy factibles.

El orden en el que se probarán las secuencias es de mejor solución y menos posible de que sea factible a peor solución y más posibilidad de que sea factible:

1. TCR y áreas de menor a mayor.
2. TCR.
3. TCR y áreas de mayor a menor.
4. Áreas de menor a mayor.
5. Áreas de mayor a menor.

4.3.4.2.2 Posicionamiento del primer elemento cuando hay límites

Otro de los problemas que surgió durante el desarrollo del proyecto fue que si la característica de limitaciones es considerada a la hora de ejecutar el modelo y una de las limitaciones está en el centro de la superficie del almacén el primer departamento no se podrá colocar.

Para ello se desarrolló un método para buscar la posición más cercana al centro del almacén y a su vez que respetara la limitación.

Dicho método consistía en realizar una búsqueda en espiral desde el centro de la superficie del almacén hasta encontrar una posición que respetara la limitación (ver *Ilustración 27*, el ejemplo de la figura es diferente al caso de uso con motivo de una mejor comprensión de la explicación).

puntos de entrada y salida el proceso tardaría cuatro veces más ya que lo tendría que hacer para (0°, 90°, 180°, 270°).

Para mitigar dicho problema se ha desarrollado un método de búsqueda por vecindades. Este método ha sido pensado y desarrollado en este trabajo de fin de máster, no se ha encontrado ninguno parecido por otras fuentes oficiales. Este método ha sido nombrado la Búsqueda por Anillos y consiste en lo siguiente.

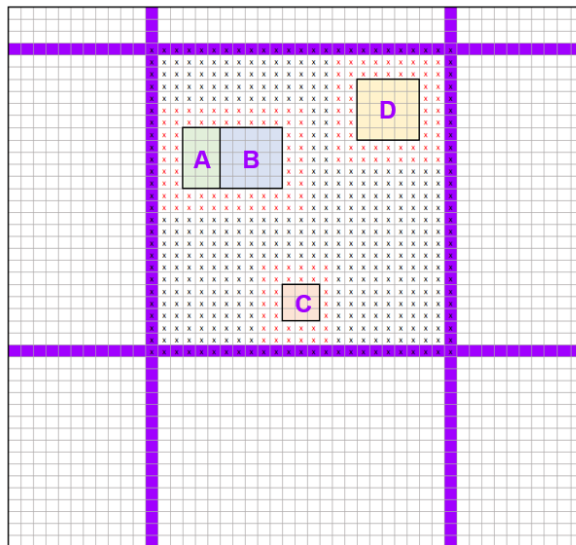
Tras realizar pruebas del espacio de soluciones, se llegó a la conclusión de que existe un subespacio donde siempre está la o las posiciones que tiene un valor mínimo de la función objetivo y este espacio siempre está comprendido entre los departamentos que estén situados más a la izquierda más a la derecha más arriba y más debajo del layout.

Si a los lados exteriores de estos departamentos es decir, el que este situado más a la derecha su lado derecho, el que esté situado más a la izquierda su lado izquierdo, el que este situado más arriba su lado superior y el de más abajo su lado inferior, se le suma la mitad de la dimensión del siguiente departamento esos serían los límites del espacio comentado en el párrafo anterior. (Ver *Ilustración 28* para comprender mejor)

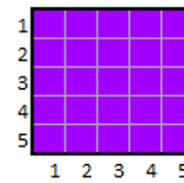
De esta manera en el anillo interior siempre quedará el espacio de soluciones donde está la posición que minimiza la función objetivo para localizar el siguiente departamento.

Los límites se pueden ver marcados en la *Ilustración 28* (El ejemplo de la figura es diferente al caso de uso con motivo de una mejor comprensión de la explicación).

Anillo Interior



Siguiente Departamento

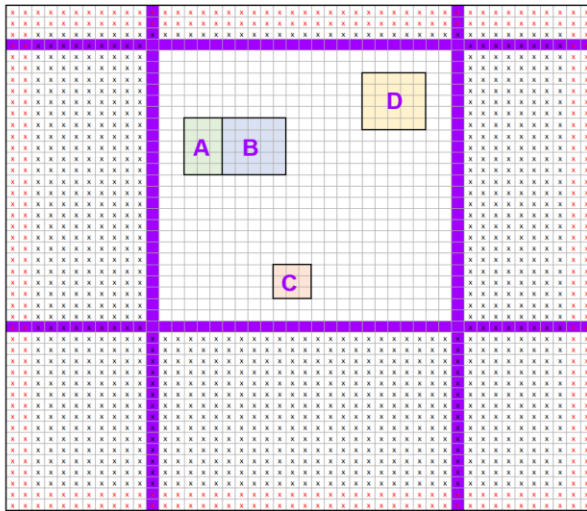


- Centroide de A = (15,12) **Min X**
- Centroide de B = (19,12)
- Centroide de C = (23,24) **Max X,Y**
- Centroide de D = (30,8) **Min Y**

Ilustración 28. Anillo interior de la búsqueda por anillos

En caso de que por restricciones el departamento no sea posible posicionarlo en las posiciones de dentro del anillo interior se buscará el valor de la posición que minimice la función objetivo en el anillo exterior para evitar soluciones no factibles (ver *Ilustración 29*, el ejemplo de la figura es diferente al caso de uso con motivo de una mejor comprensión de la explicación).

Anillo Exterior



Sin filtros de solape y fuera de rango

- Total de posibles centroides = 1996
- Posibles centroides en interior del anillo = 576
- Posibles centroides en el exterior del anillo = 1420

Con filtros de solape y fuera de rango

- Total de posibles centroides = 1491
- Posibles centroides en interior del anillo = 419
- Posibles centroides en el exterior del anillo = 1072

Ilustración 29 Anillo exterior de la búsqueda por anillos

Como se puede ver en la *Ilustración 29* la reducción del espacio de búsqueda es bastante notable y en el 90% solo se busca en el anillo interno y no hace falta buscar en el externo.

Tras desarrollar el algoritmo se hicieron unas pruebas para comprobar cuanto tiempo de ejecución reducía en escenarios grandes y este fue el resultado.

Modelo	TESTS					Sin búsqueda por anillos		Con búsqueda por anillos		REDUCCIÓN
	Superficie de almacén	Media de la superficie de los departamentos	Número de departamentos	Rotación	Puntos de entrada y salida	Tiempo de ejecución	% de la ejecución	Tiempo de ejecución	% de la ejecución	
Almacén y departamentos grandes	50400 metros ²	600 metros ²	8	Si	Si	1 horas	100%	8 minutos	100%	87%
Almacén grande y departamentos pequeños	37500 metros ²	16 metros ²	110	No	No	8 horas	45%	4 horas	100%	91%

Tabla 24. Test de eficiencia

Tras realizar las pruebas se pudo comprobar que la reducción de tiempo era entorno al 90%. Para el cálculo de la reducción del segundo modelo ya que nunca se llegó a terminar de ejecutar al completo la prueba sin la búsqueda por anillos, lo que se hizo es ver cuánto se tardaba en llegar al 45% de la ejecución con la técnica de búsqueda por anillos y esto tardaba 30 minutos.

El desarrollo de la técnica de la búsqueda por anillos fue fundamental para el éxito del proyecto.

4.3.5 MÉTODOS DE MEJORA

Como se comentó en la *sección 4.3.1* el algoritmo a desarrollar de mejora es el CRAFT

Los inputs que necesita este algoritmo para funcionar son los siguientes:

- Dimensiones del Almacén.
- Nombre e indicadores de los departamentos a localizar.
- Dimensiones de los departamentos.
- Matriz de flujo.
- Posición del centroide del elemento en X y en Y
- Ángulo en el que está con respecto a los inputs indicados.

4.3.5.1 Funcionamiento del algoritmo.

El funcionamiento de este algoritmo se basa en los siguientes pasos:

1. Inicializar la FO y la solución
2. Intercambio temporal de distintos departamentos
3. ¿Cumple con las restricciones y la nueva FO < es mejor que FO?
4. Cambia las posiciones de cada departamento y actualiza el mejor FO.
5. Reinicia el bucle.

Para entenderlo mejor ver *Ilustración 30* el cuál es un diagrama de flujo que representa el funcionamiento del algoritmo.

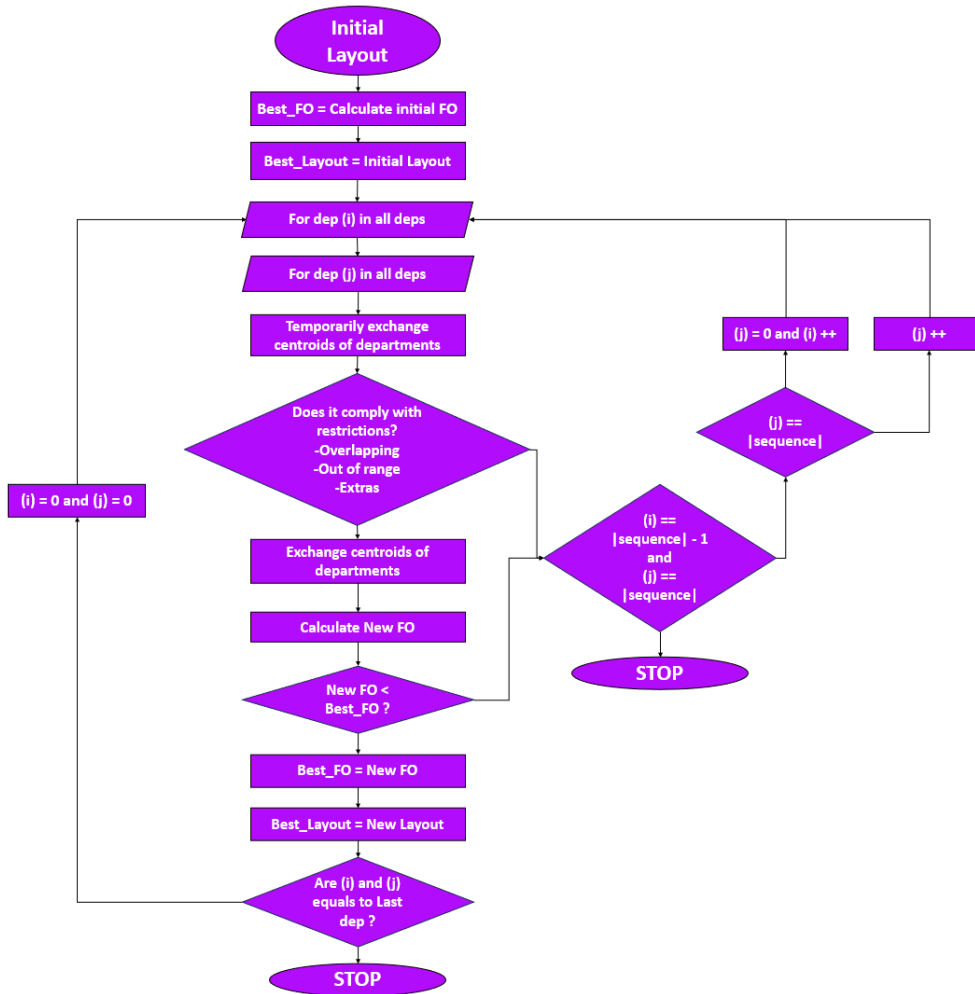


Ilustración 30. Diagrama de flujo del CRAFT

Este algoritmo cuenta en su implementación todas las características que se han comentado en la *sección 4.3.3*. De esta manera se aumenta el espacio de nuevas posibles soluciones y por tanto conseguir un óptimo local mejor.

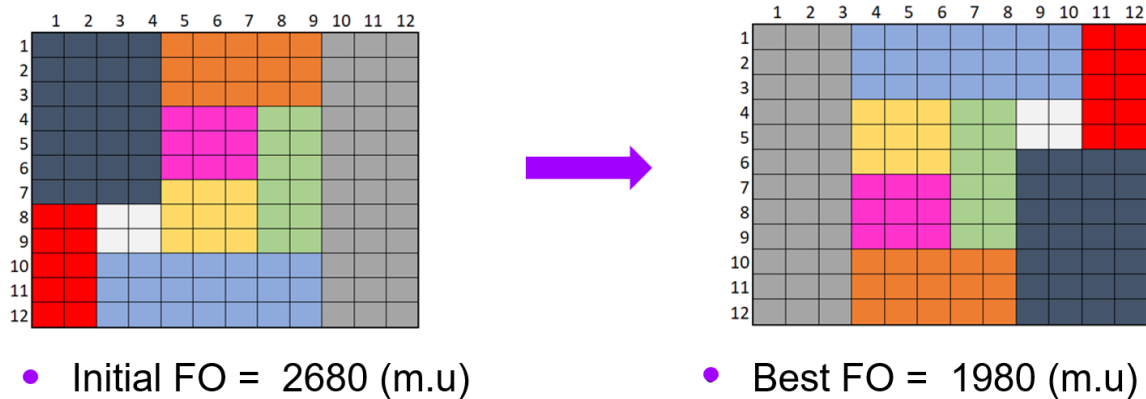


Ilustración 31. Ejemplo de resultados del CRAFT

4.3.5.2 Mejora estática

A este algoritmo le ha sido incluido una mejora que consiste en que en el caso de que la característica de rotación esta seleccionada. Los departamentos roten todos sobre si mismos para ver si existe una mejora realizando este movimiento antes de empezar a intercambiar las posiciones entre ellos y rotar entre ellos.

De esta manera ampliamos nuevamente el espacio de soluciones para encontrar un mejor óptimo local.

4.3.6 IMPLEMENTACIÓN DE SIMIO

En este capítulo se verá cómo se ha implementado el modelo en la herramienta Simio y el propósito de esta para comparar el comportamiento del Almacén inicialmente con los datos proporcionados por el cliente con el optimizado propuesto por el modelo heurístico creado.

4.3.6.1 Ingesta de datos (output del modelo)

Para que Simio sea capaz de leer los outputs del modelo de optimización ha sido necesario desarrollar un código que transforme los outputs en el formato de la ISA95 el cuál es el que es capaz de interpretar Simio para automatizar la carga de datos al modelo de simulación.

Estos inputs quedarán de la siguiente forma.

ResourceName	Description	Fixed	Department	DisplayCategory	ObjectType	XLocation	ZLocation	WorkSchedule	Capacity	CosteRate	ChangeSecondaryRe	ChangeoverEfficiency
Arrival_Docks	Department	TRUE			ArrivalDocks	44,5	-5,5	5DayWeek	1	0		1
Departure_Docks	Department	TRUE			DepartureDocks	104,5	-5,5	5DayWeek	1	0		1
CoreWH	Department	FALSE			TraditionalWH	44,5	-100,5	5DayWeek	1	0		1
ASRSArea	Department	FALSE			ASRS	139,5	-130,5	5DayWeek	1	0		1
ManufacturingArea	Department	FALSE			Manufacturing	134,5	-54,5	5DayWeek	1	0		1
Office	Department	FALSE			OfficeArea	124,5	-205,5	5DayWeek	1	0		1
MyFlowLanes	Department	FALSE			FlowLanes	149,5	-8	5DayWeek	1	0		1
PalletAdjustmentLinellArea	Department	FALSE			PalletAdjustmentLinell	163,5	-7,5	5DayWeek	1	0		1
PalletAdjustmentLinellArea	Department	FALSE			PalletAdjustmentLinell	1,5	-15,5	5DayWeek	1	0		1
Forklift1	Vehicle	FALSE			MyForklift	0	0	5DayWeek	1	0		1
AGV	Vehicle	FALSE			MyAGV	0	0	5DayWeek	1	0		1
Forklift2	Vehicle	FALSE			MyForklift	0	0	5DayWeek	1	0		1
ArrivalDocks_AuxNode	AuxNode	FALSE			BasicNode	0	0	5DayWeek	1	0		1
ASRS_AuxNode	AuxNode	FALSE			BasicNode	0	0	5DayWeek	1	0		1
CoreWH_AuxNode	AuxNode	FALSE			BasicNode	0	0	5DayWeek	1	0		1

Tabla 25. Ejemplo de inputs de SIMIO, hoja 1

Los Forklift1, Forklift2 y AGV son elementos de transporte propias de cada modelo en específico y serán añadidos de manera manual al Excel de inputs de Simio al igual de los ‘AUXNODE’ que son nodos que ayudan a definir la ruta de los vehículos y evitar a circulen por medio de los departamentos.

dep_from	dep_to	flow	cost	aisle
Arrival Docks	Arrival Docks	0	0	0
Arrival Docks	Departure Docks	0	1	0
Arrival Docks	Traditional WH	0	1	5
Arrival Docks	ASRS	0	1	5
Arrival Docks	ManufacturingArea	0	1	5
Arrival Docks	Flow Lanes	0	1	5
Arrival Docks	Office Area	0	1	5
Arrival Docks	PalletAdjustmentLinellArea	25	1	10
Arrival Docks	PalletAdjustmentLinellArea	0	1	5
Departure Docks	Arrival Docks	0	1	0
Departure Docks	Departure Docks	0	0	0
Departure Docks	Traditional WH	0	1	5
Departure Docks	ASRS	0	1	5
Departure Docks	ManufacturingArea	0	1	5
Departure Docks	Flow Lanes	0	1	5
Departure Docks	Office Area	0	1	5
Departure Docks	PalletAdjustmentLinellArea	0	1	5
Departure Docks	PalletAdjustmentLinellArea	0	1	5
Traditional WH	Arrival Docks	0	1	5
Traditional WH	Departure Docks	0	1	5
Traditional WH	Traditional WH	0	0	0
Traditional WH	ASRS	10	1	10
Traditional WH	ManufacturingArea	0	1	5
Traditional WH	Flow Lanes	10	1	10

Tabla 26. Ejemplo de input de SIMIO, hoja 2

En la otra hoja del Excel de entrada se indicarán las matrices de flujo de coste y la de pasillos.

4.3.6.2 Submodelos

El modelo principal consta de un conjunto de submodelos.

Todos los submodelos de esta sección son acordes con el caso de uso, respetando las medidas y el flujo que se indicaron en los inputs del mismo.

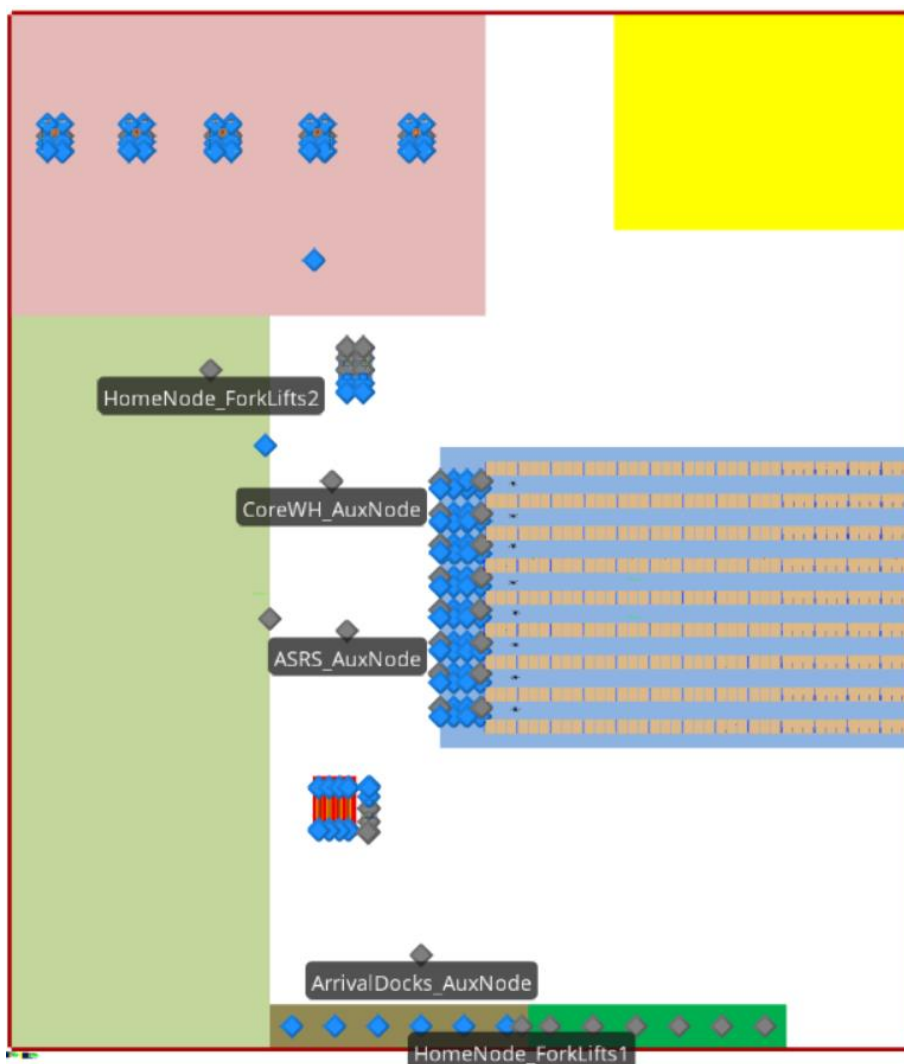


Ilustración 32. Diseño de la planta inicial en SIMIO

Cada submodelo es un departamento del modelo principal y cada uno de ellos tiene una serie de características que se explicarán a continuación:

- Almacén Tradicional:

El almacén tradicional se modela como una superficie con las dimensiones especificadas por el cliente. Este submodelo en su interior se comporta como un creador y un destructor de entidades ('source', crea entidades y 'sink' las destruye).

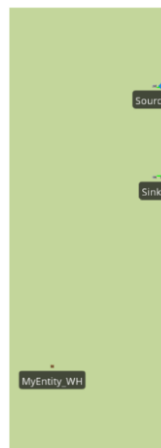


Ilustración 33. Modelo del almacén tradicional en SIMIO

De esta manera dependiendo del flujo que se haya indicado para este departamento el 'source' generará entidades cada x entidades en una cierta unidad de tiempo, en el caso de uso en el que nos encontramos se generarán 20 pallets a la hora (ver inputs en la *Sección 4.1.2* si no se ha comprendido porque es esa cantidad). El 'sink' destruye todas las entidades que llega para no acumular entidades en el sistema.

- Muelles de entrada:

Los muelles de entrada se modelan como una superficie con las dimensiones especificadas por el cliente. Este submodelo en su interior se comporta como un creador de entidades ('source', crea entidades).

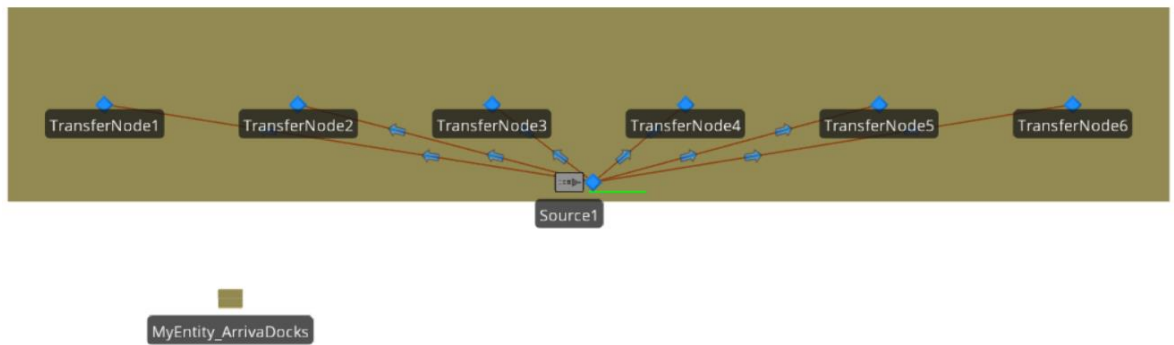


Ilustración 34. Modelo de los muelles de entrada en SIMIO

A este submodelo se le incorporan nodos para crear varios puntos de salidas de entidades y poder simular varios muelles.

El 'source' crea 27 entidades a la hora (ver inputs en la *Sección 4.1.2* si no se ha comprendido por qué es esa cantidad).

- Muelles de salida:

Los muelles de salida se modelan como una superficie con las dimensiones especificadas por el cliente. Este submodelo en su interior se comporta como un destructor de entidades ('sink', destruye entidades).

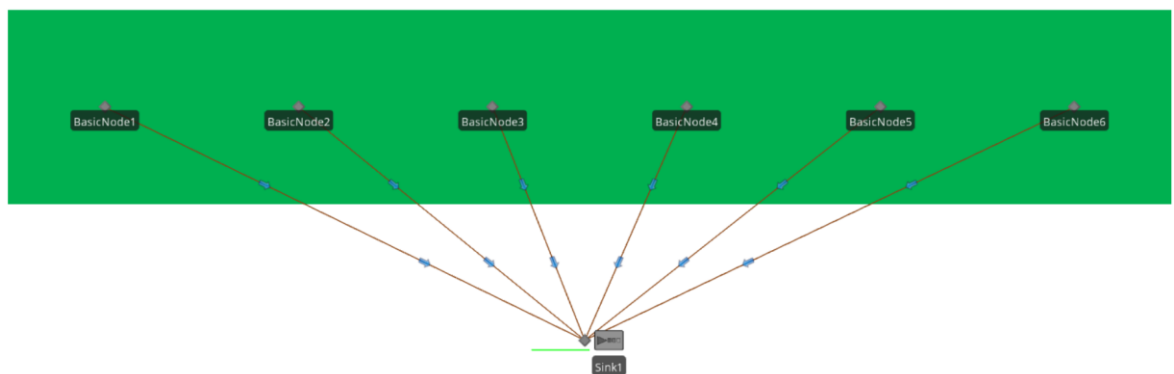


Ilustración 35. Modelo de los muelles de salida en SIMIO

A este submodelo se le incorporan nodos para crear varios puntos de entrada de entidades y poder simular varios muelles.

El 'sink' destruye las entidades para no acumularlas en el sistema.

- Planta de producción:

La planta de producción se modela como una superficie con las dimensiones especificadas por el cliente. Este submodelo en su interior se comporta como un creador de entidades ('source', crea entidades).

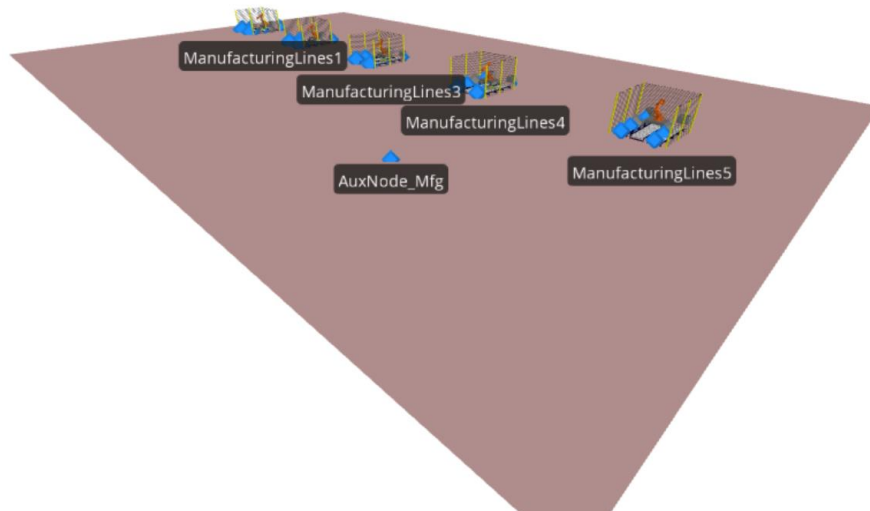


Ilustración 36. Modelo de la planta de producción en SIMIO

Dentro de este submodelo tenemos otros submodelos llamados 'Manufacturing Lines' los cuales son las máquinas de producción que actuarán como 'sources'.

En el caso de nuestro cliente existían cinco 'Manufacturing Lines' y cada una de las genera 10,2 entidades a la hora (ver inputs en la *Sección 4.1.2* si no se ha comprendido porque es esa cantidad).

Para salir de la planta de producción estas entidades deberán pasar primero por el Nodo auxiliar (el cuál simula la puerta de salida del almacén donde lo recogerá el forklift).

- ASRS:

El ASRS se modela como una superficie con las dimensiones especificadas por el cliente. Este submodelo en su interior se comporta muy parecida a un almacén tradicional con un creador y un destructor de entidades ('source', crea entidades y 'sink' las destruye).

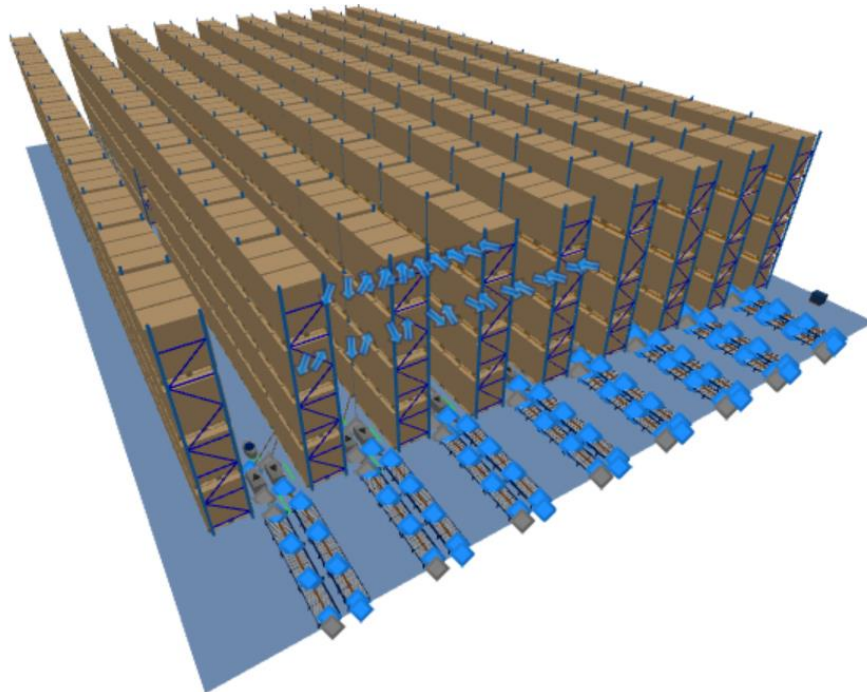


Ilustración 37. Modelo del ASRS en SIMIO

Las diferencias con un almacén tradicional son la siguientes:

- Tiene conveyors de entra y salida para las entidades (Un conveyor es un elemento de transporte que actúa como cinta transportadora). Los conveyors tienen una capacidad de ocho entidades en ellas.
- Tienen servers los cuales son elementos del sistema que simulan una tarea y en este caso es el tiempo de almacenamiento de un pallet en el almacén
- Cada pasillo del ASRS tiene un recurso el cuál será ocupado por uno de los dos servers a la hora de realizar una tarea.

El source generará 65 entidades a la hora y el sink destruirá todas las entidades que lleguen para no acumularlas en el sistema (ver inputs en la *Sección 4.1.2* si no se ha comprendido porque es esa cantidad).

- Oficina:

La oficina al no tener ningún flujo con ninguno de los elementos del sistema se modelará como una superficie con las dimensiones determinadas por el cliente en los inputs.

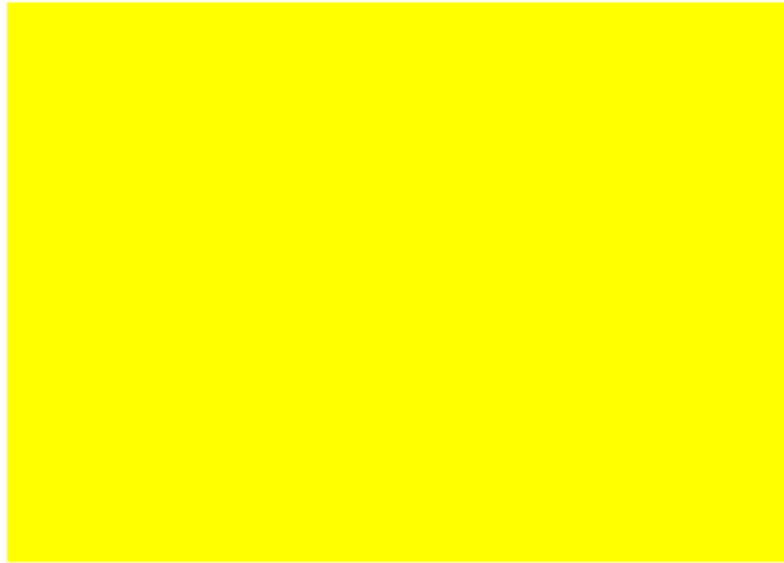


Ilustración 38. Modelo de oficina en SIMIO

- Líneas de flujo:

Las líneas de flujo en el caso de uso de este proyecto hay cuatro de ellas. Cada una de ellas tiene una capacidad de ocho entidades.

Estas líneas de flujo estarán sobre una superficie con las dimensiones indicadas por el cliente en los inputs del modelo.

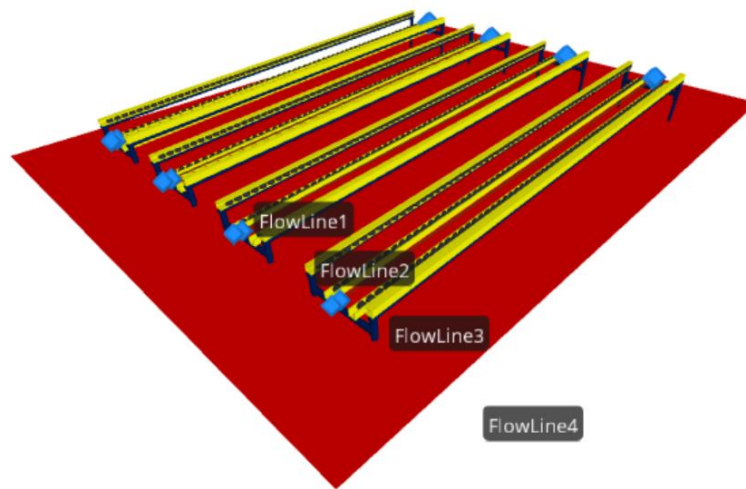


Ilustración 39. Modelo de líneas de flujo en SIMIO

- Centradora y envolvedora I y II:

En el caso de uso de este proyecto existen dos tipos de centradoras y envolvedoras, la simple y la doble.

Estos submodelos están comprendidos de otros submodelos con son los 'Pallet Adjustment Line', estos están dentro de una superficie indicada por el cliente en los inputs del modelo.

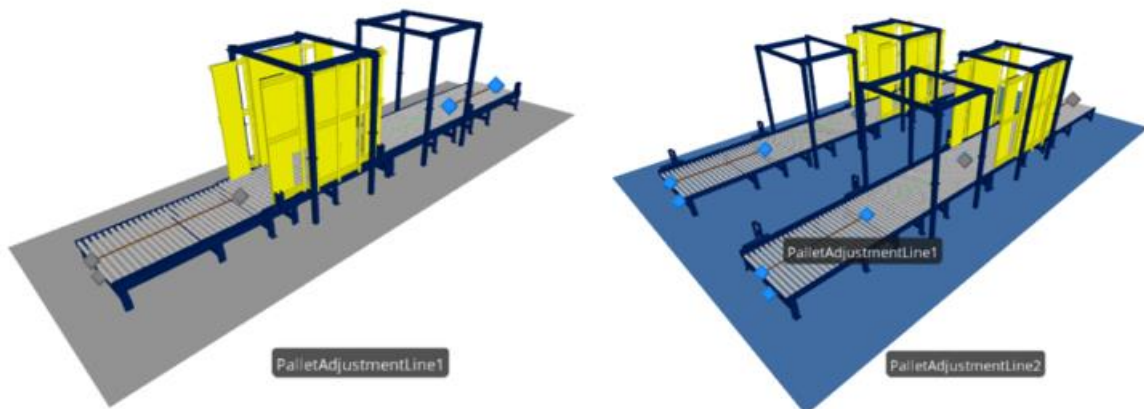


Ilustración 40. Modelo de centradora y envolvedora en SIMIO

4.3.6.3 Procesos

Para conectar cada uno de los departamentos se han tenido que crear una serie de procesos para conectar los flujos de entidades entre ellos y conseguir que el modelo de simulación refleje con la mayor precisión posible la realidad del caso de uso.

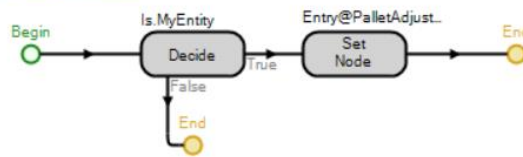
Para este cometido varios tipos de procesos han sido creados.

- **Procesos de rutas:**

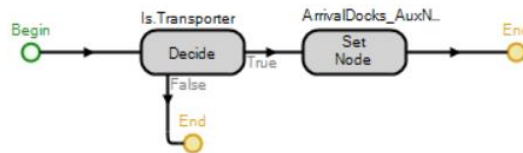
Estos procesos indican el direccionamiento de los flujos cuando salen de un elemento del sistema. Para construir estos procesos se harán uso de los siguientes procesos:

- **Decide:** El proceso de toma de decisiones involucra la lógica y las reglas para determinar el flujo y el comportamiento de las entidades dentro del sistema.
- **Set Node:** Implica definir puntos en el sistema donde ocurren acciones o procesos específicos.
- **Route:** Implica definir los caminos y la lógica para mover entidades a través del sistema.

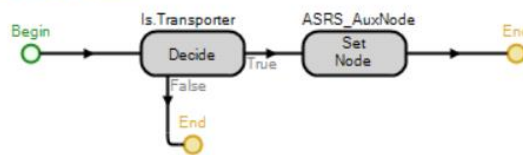
Arriva_Docks_Destination



ArrivaDocks_AuxRouting_Exited



ASRS_AuxRouting_Exited



ASRS_Destinations

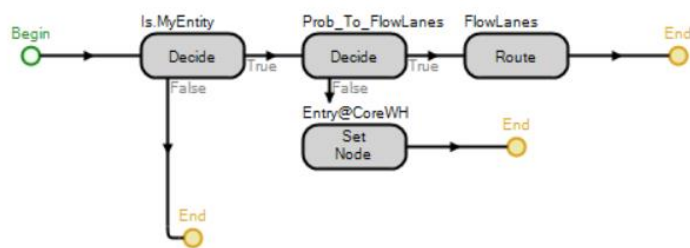
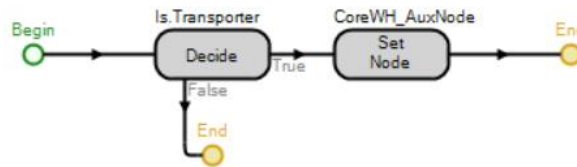
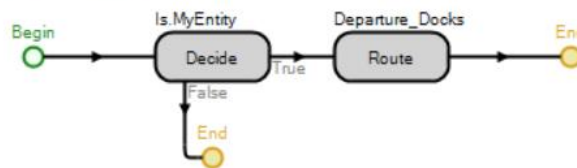


Ilustración 41. Procesos de rutas 1 en SIMIO

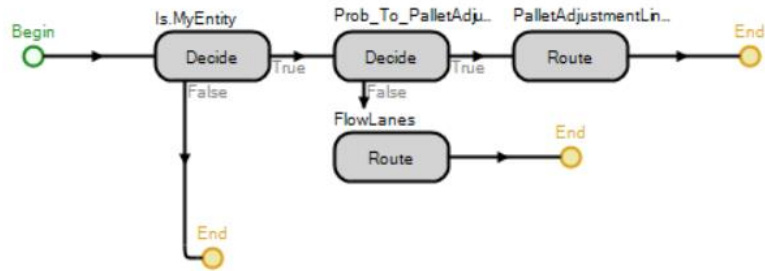
CoreWH_AuxRouting_Exitd



FlowLines_Destinations



Manufacturing_Destination



PalletAdjustmentLine1_Destination

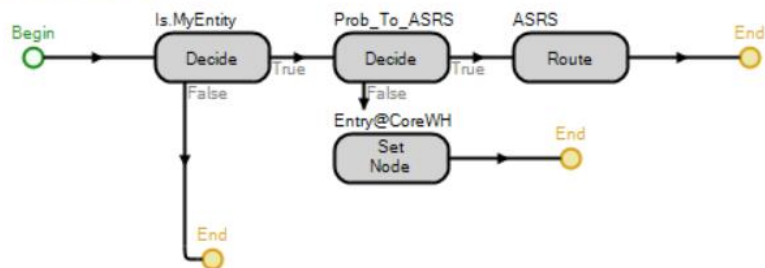


Ilustración 42. Procesos de rutas 2 en SIMIO

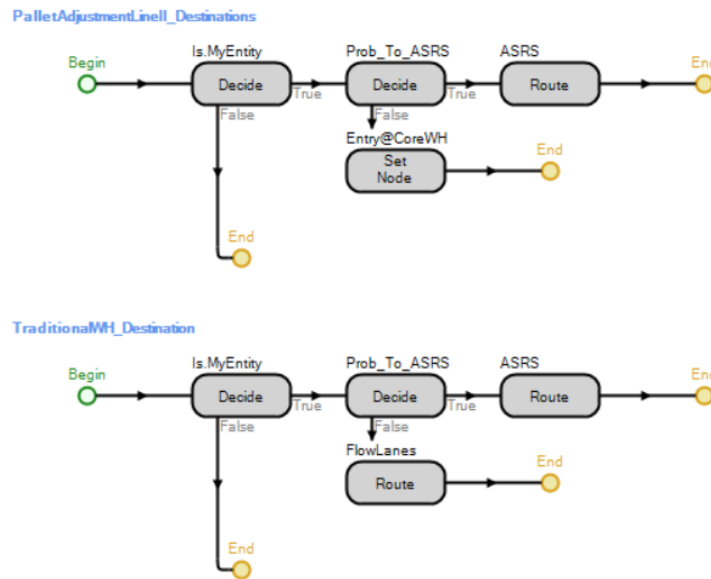


Ilustración 43. Procesos de rutas 3 en SIMIO

- Procesos de control de entradas y salidas de elementos del sistema:
 - ASRS:

Estos procesos son necesarios para que eviten que entidades sean asignadas a los ‘conveyors’ de cada uno de los pasillos del ASRS si estos ya tienen su capacidad completa y generen acumulaciones de inventario indeseadas en ellas.

Para ello usaremos procesos de ‘assign’ que actualizan variables globales indicando si en los ‘conveyors’ existen espacios para que las entidades puedan ser asignadas a estos como destino.

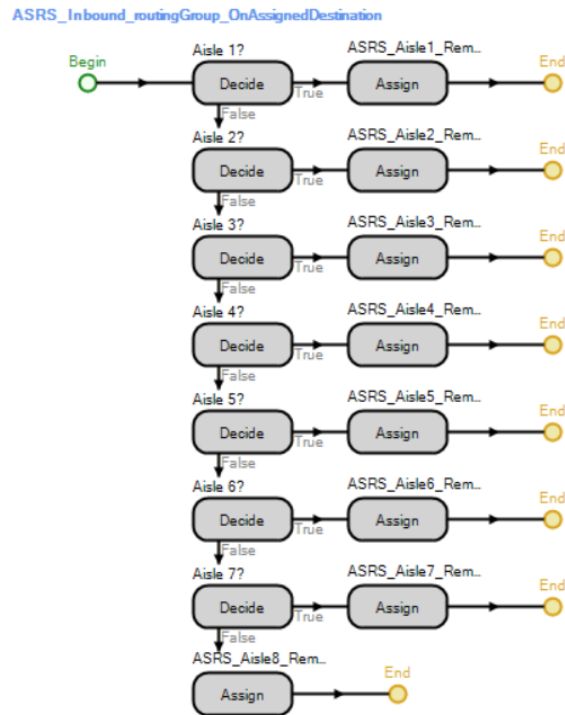


Ilustración 44. Procesos de control de entradas al ASRS en SIMIO

Lo mismo para los ‘conveyors’ de salida del ASRS, se actualizarán variables globales de los posibles destinos del ASRS en este caso de uso los posibles destinos de una entidad que sale del ASRS son: El almacén tradicional y las líneas de flujo.

Cuando en estos elementos del sistema haya espacio para que pueda llegar otra entidad a ellos los ‘conveyors’ de salida emitirán la señal al modelo para que las entidades que estén en estos sean recogidas por un ‘forklift’ y sea trasladado a su destino siguiendo el flujo entre elementos del modelo definido en la matriz de flujos de los inputs.

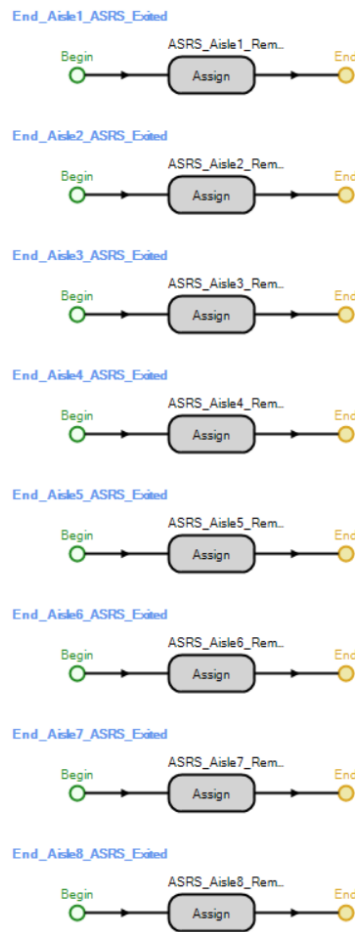


Ilustración 45. Procesos de control de salidas del ASRS en SIMIO

- Líneas de Flujo:

De la misma manera que el ASRS se crearán los procesos correspondientes para las líneas de flujo ya que este elemento del sistema tiene capacidades y pueden llegar a tener el mismo problema.

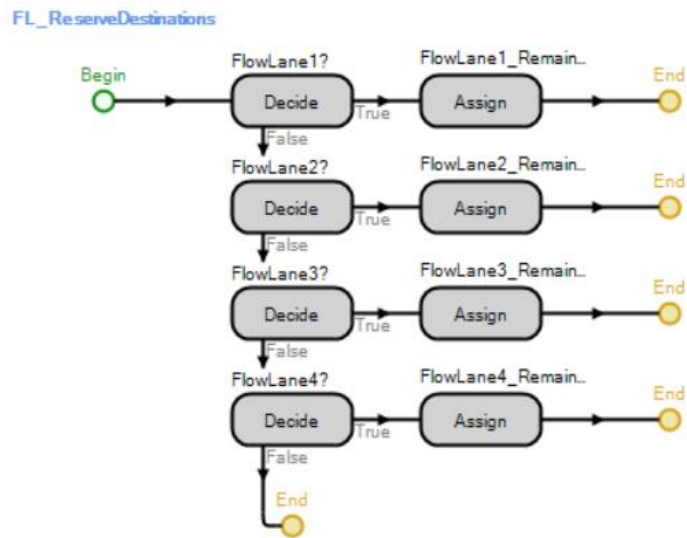


Ilustración 46. Procesos de control de entradas a las líneas de flujo en SIMIO

En este caso para la salida de las líneas de flujo los muelles de salida.

- **Procesos de Flujos:**

Estos procesos se utilizarán para calcular los flujos totales de la tabla de flujos y las guardarán en variables globales del modelo. Estas variables serán útiles para indicarle a los ‘sources’ cuantas unidades por hora se ha de crear en cada uno de ellos y también para calcular el porcentaje de entidades se deberán de ir por una ruta u otra en los ‘decide’ de los procesos de rutas.

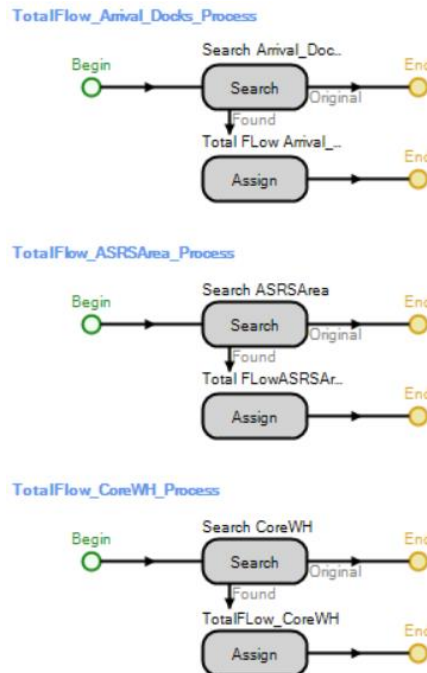


Ilustración 47. Procesos de control de salida de las líneas de flujo en SIMIO

El proceso de ‘search’ realiza una búsqueda bajo una condición a un elemento iterable, en este caso la búsqueda se realizará sobre la tabla de flujos que introducimos como input, con la condición de que busque el departamento al cual le queremos calcular su flujo total.

Cuando encuentre ese departamento cogerá su flujo con el ‘assign’ y actualizará una variable donde iremos acumulando todo el flujo que vaya saliendo de dicho departamento, esta variable ha tenido que ser creada previamente como variable global del modelo.

- Procesos de Checks:

Los procesos de check hacen la función de comprobar con las restricciones del modelo de optimización se cumplan de la misma manera en el modelo de Simio:

- Check de la restricción de solapamiento:

Verifica que ningún elemento del sistema se superponga con otro.

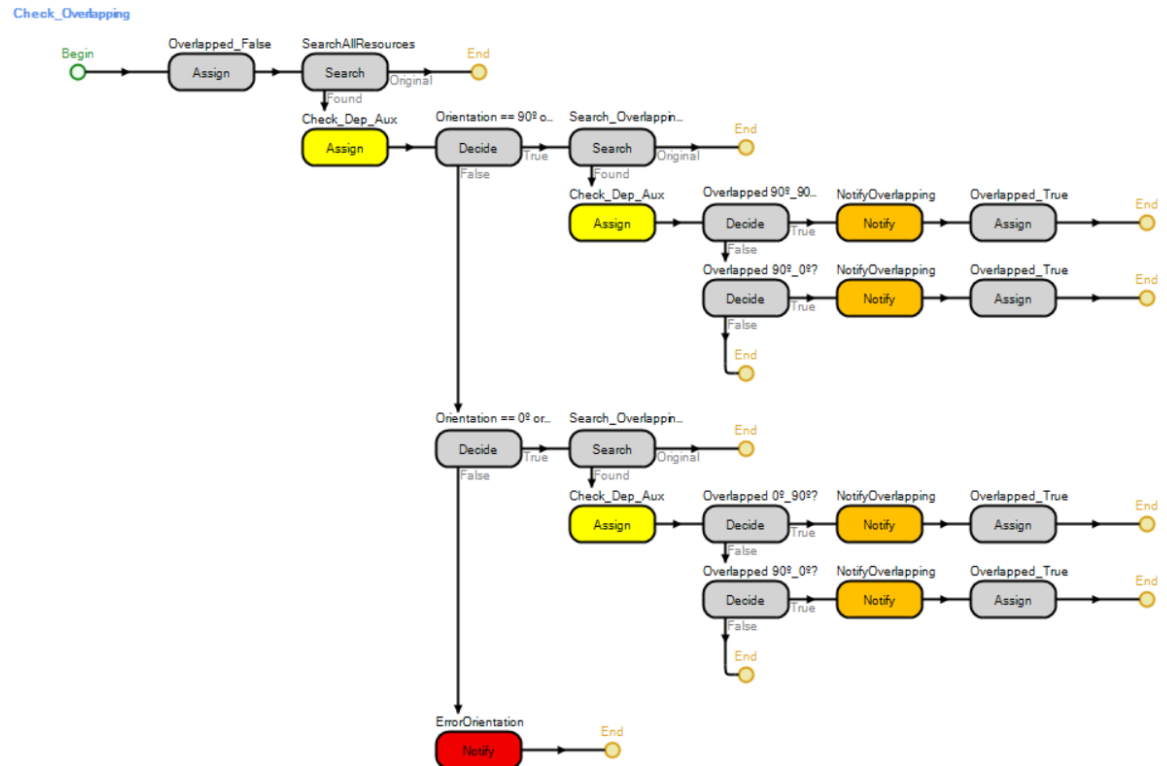


Ilustración 48. Check de solape en SIMIO

- Check de la restricción de fuera de rango:
- Verifica que ningún elemento del sistema salga de las dimensiones indicadas por el cliente en los inputs del modelo.

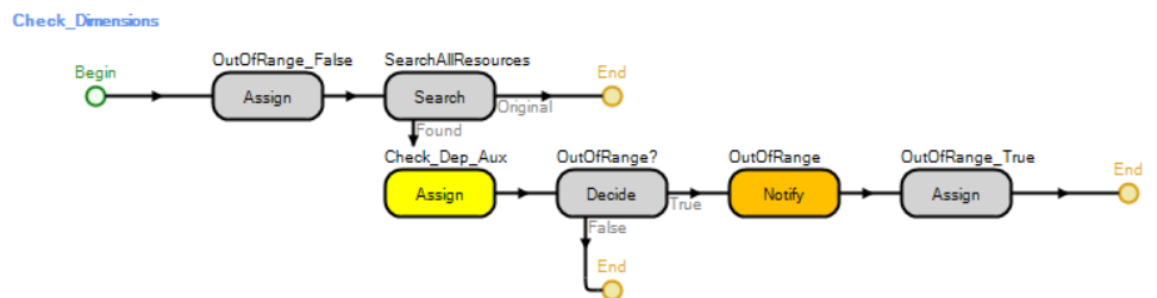


Ilustración 49. Check de fuera de rango en SIMIO

- Proceso del método CRAFT:
Este proceso realiza el algoritmo de CRAFT con procesos de SIMIO, este proceso no era estrictamente necesario para este proyecto, pero se decidió hacer para aportar valor extra.

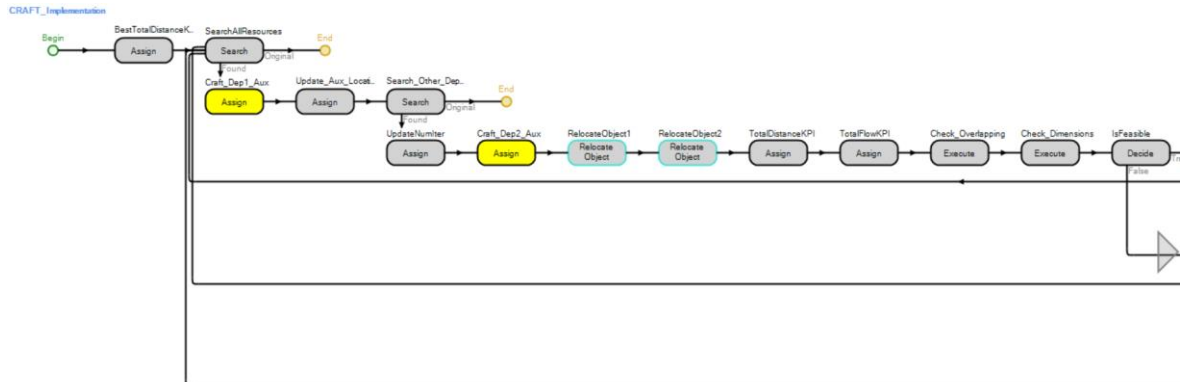


Ilustración 50. CRAFT en SIMIO 1

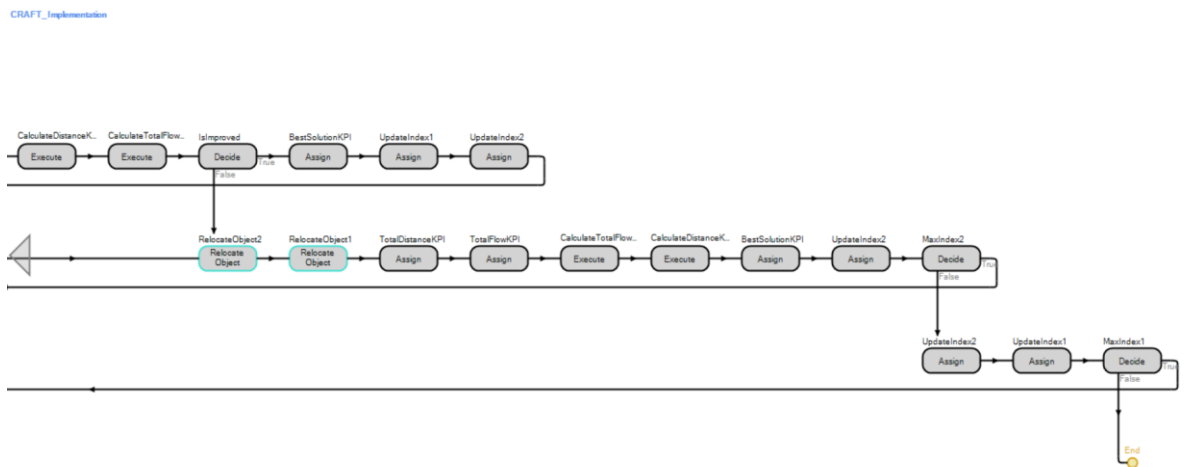


Ilustración 51. CRAFT en SIMIO 2

- Procesos de KPIS:

Estos procesos se encargarán de calcular y actualizar las variables durante la simulación que ayudarán a calcular alguno de los KPIS que se medirán en el modelo de SIMIO.

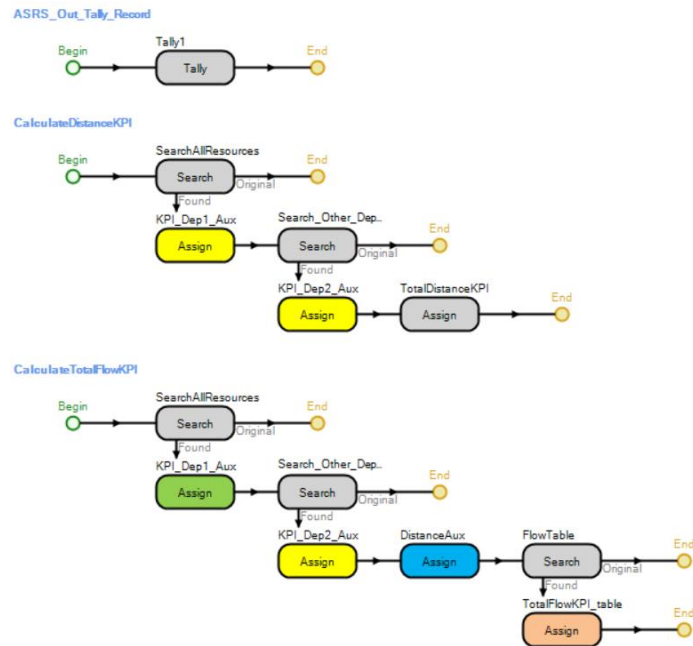


Ilustración 52. Procesos de KPIS en SIMIO

- Procesos de inicio:

Estos procesos se ejecutan antes de comenzar la simulación y es la encargada de ejecutar los procesos de check, cálculo de algunos KPIS, el algoritmo CRAFT si se precisa y los procesos de cálculo de flujos totales de los elementos del sistema los cuales tengas 'sources' los cuales se encargan de generar las entidades del sistema según la tabla de flujos indicada por el cliente como input.

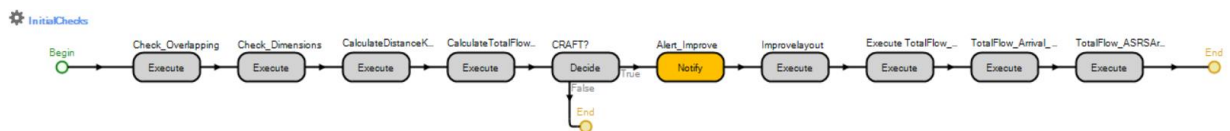


Ilustración 53. Proceso de inicio en SIMIO

4.3.6.4 KPIS

Los KPIS que se querrán medir durante la simulación de SIMIO son los siguientes:

4.3.6.4.1 KPIS de Entidades

- Número total de entidades que salen de cada elemento del sistema que tenga un ‘source’.
- Número de entidades a la hora que salen de cada elemento del sistema que tenga un ‘source’.
- Número total de entidades que entran a cada elemento del sistema que tenga un ‘sink’.
- Número de entidades a la hora que entran a cada elemento del sistema que tenga un ‘sink’.

4.3.6.4.2 KPIS de Vehículos

- % De utilización de cada vehículo.
- Distancia total recorrida de cada vehículo.

4.3.6.4.3 KPIS de Acumulaciones

- Acumulación media de entidades en los conveyors.
- Acumulación máxima de entidades en los conveyors.

4.3.6.4.4 KPIS relacionados con la función objetivo

Estos KPIS son estáticos, es decir no varían a lo largo de la simulación y son propios del diseño de la planta.

Estos son calculados en los procesos de inicio.

- Distancia total entre centroides de los departamentos.
- Flujo total entre departamentos, haciendo uso de las distancias entre centroides de los departamentos.
- Mejor solución tras implementar el CRAFT en SIMIO (Este último solo se calcula si se utiliza el proceso de CRAFT en SIMIO).

Estos KPIS serán los mismos que saldrán en el modelo de optimización si las características de puntos de entradas y salidas no está aplicado, ya que de esa manera las distancias serán calculadas desde los centroides de los departamentos.

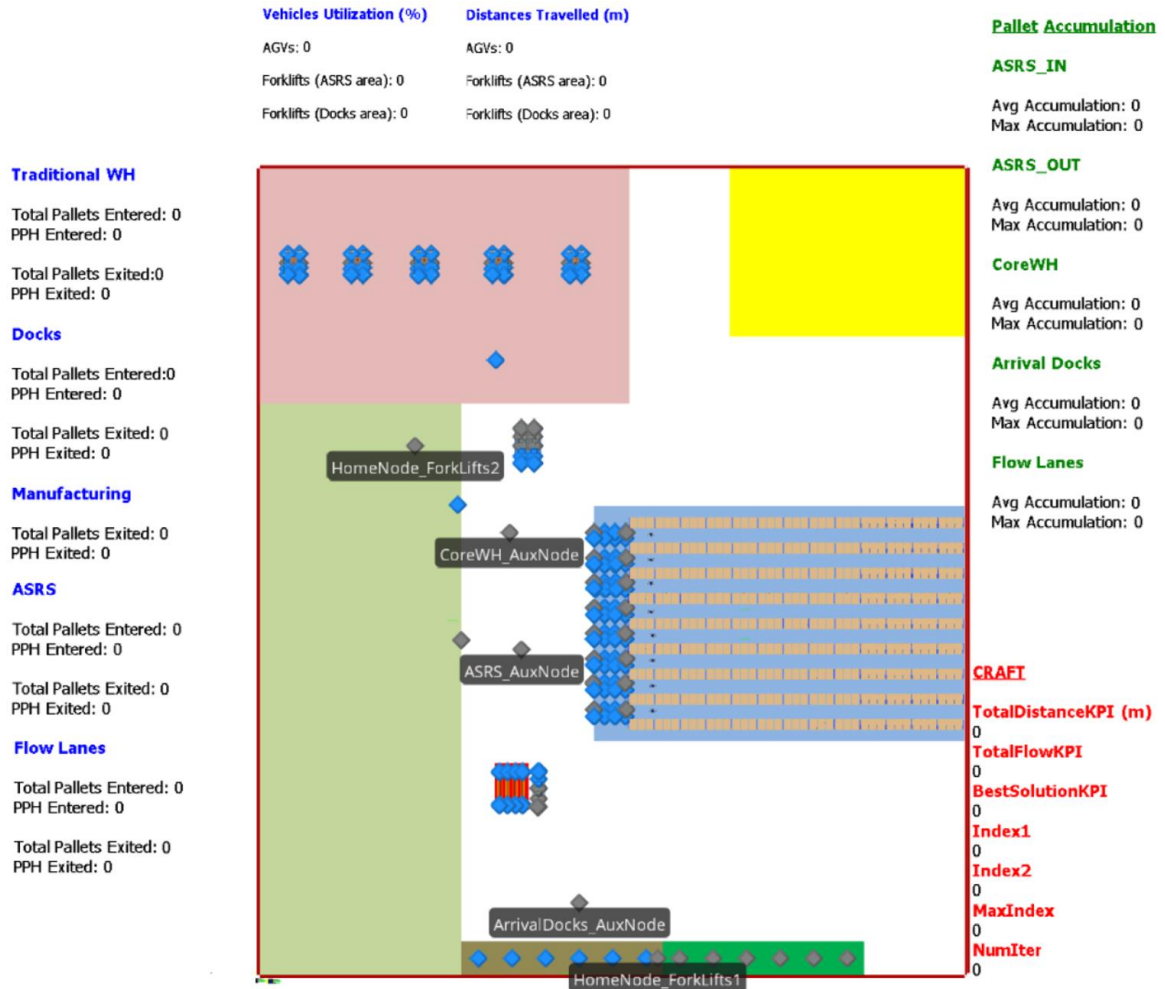


Ilustración 54. KPIS en SIMIO

Capítulo 5. ANÁLISIS DE RESULTADOS

5.1 PYTHON

La optimización se realiza sobre la app desarrollada con Streamlit la cuál facilita la interacción con el modelo de optimización desarrollado.

En la app hay una página principal en la que se da a conocer todas las posibles interacciones con el modelo las cuales se han ido comentando en los capítulos anteriores de este trabajo de fin de máster.

En esta app existen 3 páginas de interacción con el modelo dependiendo de lo que se quiera hacer:

- Construct page: En esta página solo se interacciona con el modelo si se quiere realizar una heurística constructiva cuando no se dispone de un layout inicial.
- Improve page: En esta página solo se ejecutan heurísticas de mejora partiendo de un layout inicial.
- Construct&Improve page: Combina una heurística constructiva y otra de mejora

Para este caso de uso a pesar de que disponíamos de un layout inicial realizaremos una heurística constructiva y otra de mejora ya que el objetivo de este proyecto estaba mas enfocado a proponerle al cliente como debería de construir el próximo almacén que tenía pensado llevar a cabo por necesidades de su red de distribución.

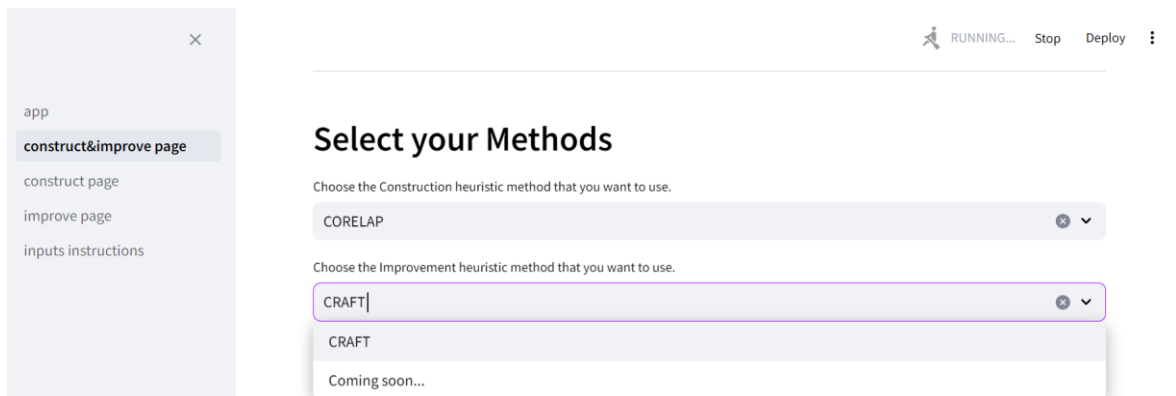


Ilustración 55. Interacción con la App de Streamlit 1

En esta página indicaremos los métodos heurísticos que se querrán implementar en la optimización. En este caso se seleccionarán el método CORELAP como algoritmo constructivo y el método CRAFT como de mejora ya que son los que se han implementado en este proyecto.

Set up your Methods

Select the features that you will need in your layout

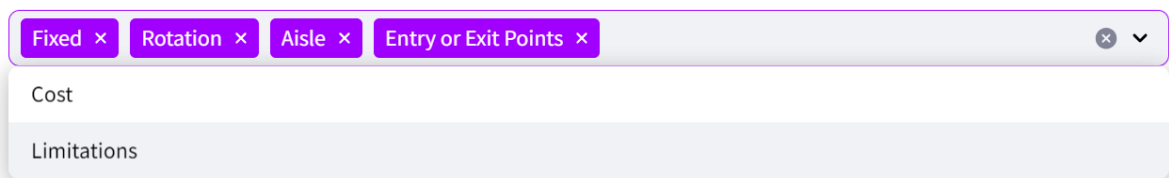
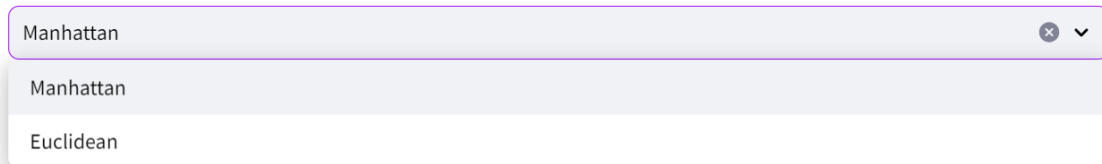


Ilustración 56. Interacción con la App de Streamlit 2

A continuación, se deberán de indicar también las características que se quieren que se tengan en cuenta a la hora de realizar la optimización que en este caso serán las de los departamentos fijos, la de los pasillos, rotación y puntos de entrada y salida de los elementos del sistema.

Select the Objetivo Function

Select the type of distance that you want to measure in your objective function



A screenshot of a Streamlit application showing a dropdown menu. The menu is open, displaying two options: 'Manhattan' and 'Euclidean'. The 'Manhattan' option is currently selected and highlighted in light blue. The dropdown has a close button (an 'x' in a circle) and a downward arrow on the right side.

Ilustración 57. Interacción con la App de Streamlit 3

El tipo de distancia que se quiere emplear para calcular la función objetivo del modelo de optimización también se podrá elegir.

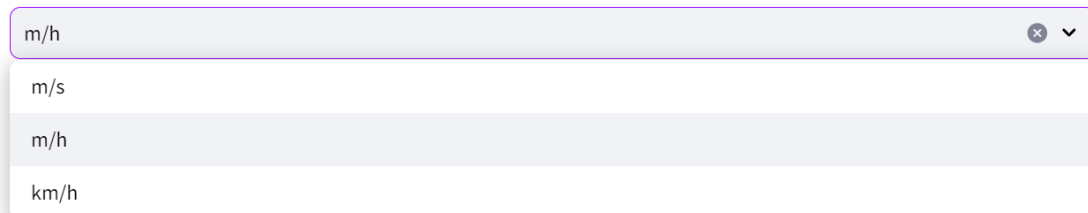
Select the Objetivo Function

Select the type of distance that you want to measure in your objective function



A screenshot of a Streamlit application showing a dropdown menu. The menu is closed, and the selected option 'Manhattan' is visible. The dropdown has a close button (an 'x' in a circle) and a downward arrow on the right side.

Select the Unit of Measure (Please be consistent with the units of the inputs):



A screenshot of a Streamlit application showing a dropdown menu. The menu is open, displaying four options: 'm/h', 'm/s', 'm/h', and 'km/h'. The 'm/h' option is currently selected and highlighted in light blue. The dropdown has a close button (an 'x' in a circle) and a downward arrow on the right side.


Ilustración 58. Interacción con la App de Streamlit 4

También se indicarán las unidades de medida de la función objetivo en relación con las dimensiones de los inputs que se han introducido al modelo.

Optimise your Facility !!

Please upload an excel with the inputs to optimise the Facility.

Upload file

 Drag and drop file here
Limit 200MB per file • XLSX Browse files

 inputs_bigsurf_bigdeps.xlsx 67.2KB ×

The uploaded file is: inputs_bigsurf_bigdeps.xlsx

Ilustración 59. Interacción con la App de Streamlit 5

El Excel de inputs se podrá introducir mediante un ‘Drag and Drop’ o directamente buscando por los archivos locales del ordenador con el que se esté interactuando con la app.

Save the result in a ISA95 structure.

Do you want to save a file in a ISA95 structure ? (useful for Simio input table)

Yes ✕ ▼

Introduce the name of the excel for the construct solution:

Output_Corelap.xlsx

Introduce the name of the excel for the improve solution:

Output_Craft.xlsx

Execute optimizer

Ilustración 60. Interacción con la App de Streamlit 6

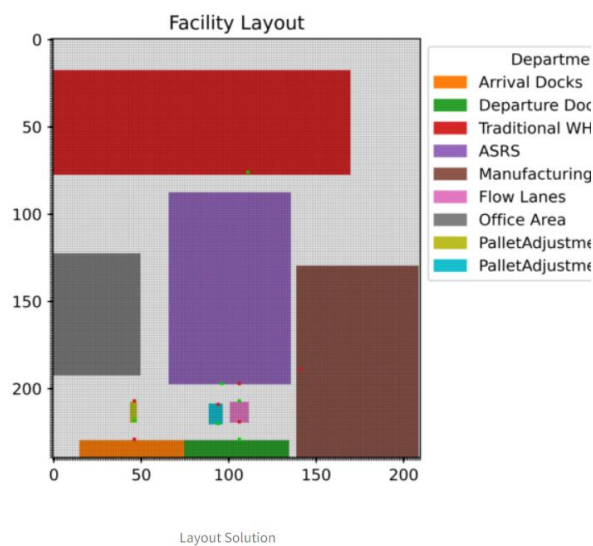
Y por último si se quiere guardar los outputs de modelo en formato de tabla de inptus para el modelo de SIMIO también se podrá realizar indicando el nombre de cómo se quiere que se guarden dichos archivos (Estos se guardarán en la ruta relativa al proyecto data/ISA95_outputs).

Una vez indicado todo lo anterior se deberá de pinchar en el botón de ejecutar y comenzará la optimización.

Los resultados una vez finalizada la optimización se guardarán en el archivo de inputs en la hoja de ELEMENTS en las columnas de centroides y de ángulos como se indicó en la Sección 4.3.3 y también se podrán ver de manera visual sobre la aplicación como se observa en la *Ilustración 61*.

Solución de CORELAP

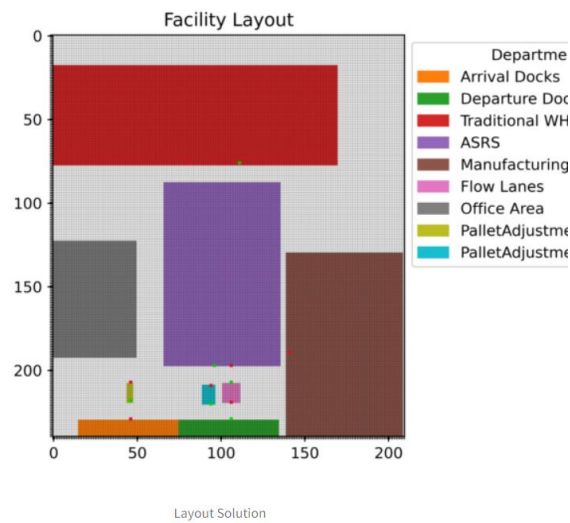
Solution ⇄



The objective function is: 14.773 m/h

Ilustración 61. Solución de CORELAP

Solución de CRAFT



The objective function is: 14.773 m/h

Attention!

The centroids and the rotation angles of the departments in the solution are saved in the input file.

Ilustración 62. Solución de CRAFT

En este caso la heurística de mejora no ha conseguido mejorar la solución encontrada por el algoritmo constructivo, esto es debido a que al realizar los intercambios de posiciones entre todos los departamentos no ha conseguido encontrar ningún cambio que haga mejorar la función objetivo lograda por el algoritmo constructivo respetando las restricciones del modelo descritas en la *Sección 4.3.4.1.2*.

5.2 SIMIO

Para realizar el análisis de resultados se comparará el layout inicial el cual tiene el cliente actualmente y el layout propuesto tras la optimización.

En ambos se medirán los KPIS en el Modelo de Simulación y se comprobará la viabilidad del layout optimizado, en cuestión de acumulación de entidades y saturación del sistema.

Para comprender mejor el sistema se deberá tener en cuenta los recursos de transporte de los que se dispone en el modelo:

- Carretillas de transporte (Zona de los muelles):
Estas carretillas son de uso manual y se necesita un operario para poder hacer uso de ellas.
Estas en concreto se encargarán del transporte de los muelles de entrada y salida a la centradora y envolvedora 1 y a las líneas de flujo.
- Carretillas de transporte (Zona del ASRS):
Estas carretillas son de uso manual y se necesita un operario para poder hacer uso de ellas.
Estas en concreto se encargarán del transporte de la centradora y envolvedora 2, ASRS, almacén tradicional y las líneas de flujo.
- AGV: Los AGV son vehículos autónomos los cuales no requieren de un operario para su uso por lo que su coste es menor al de las carretillas de transporte. Este se encargará de sacar las entidades de la zona de producción a la centradora y envolvedora 2.

5.2.1 DISTRIBUCIÓN INICIAL

En la distribución inicial la cuál es la que disponía el cliente constaba de los siguientes elementos de transporte:

- Carretillas de transporte:
 - Zona de muelles: 7
 - Zona de ASRS: 8
- AGV:
Hay 1 AGV

Tras simular el modelo durante 8 horas se han obtenido los siguientes resultados:

- % de utilización de vehículos:
 - Carretillas de transporte (Zona de los muelles): 88,7%
 - Carretillas de transporte (Zona del ASRS): 88,5%
 - AGV: 46,7%
- Distancia recorrida por los vehículos:

- Carretillas de transporte (Zona de los muelles): 116.359 metros
- Carretillas de transporte (Zona del ASRS): 132.548 metros
- Carretillas de transporte (TOTAL): 248.907 metros
- AGV (TOTAL): 10.278 metros

En el sistema la ocupación de los recursos tiene un porcentaje no sobrepasa el 90% lo cual es sano ya que se tiene una alta ocupación, pero no está al borde de la saturación de los mismos, dejando un margen para imprevistos. Las colas de los elementos del sistema nunca se saturan siendo la media total de entidades en cola entre 1 y 2.

Teniendo ambas cosas en cuenta se puede considerar este sistema como un sistema estable.

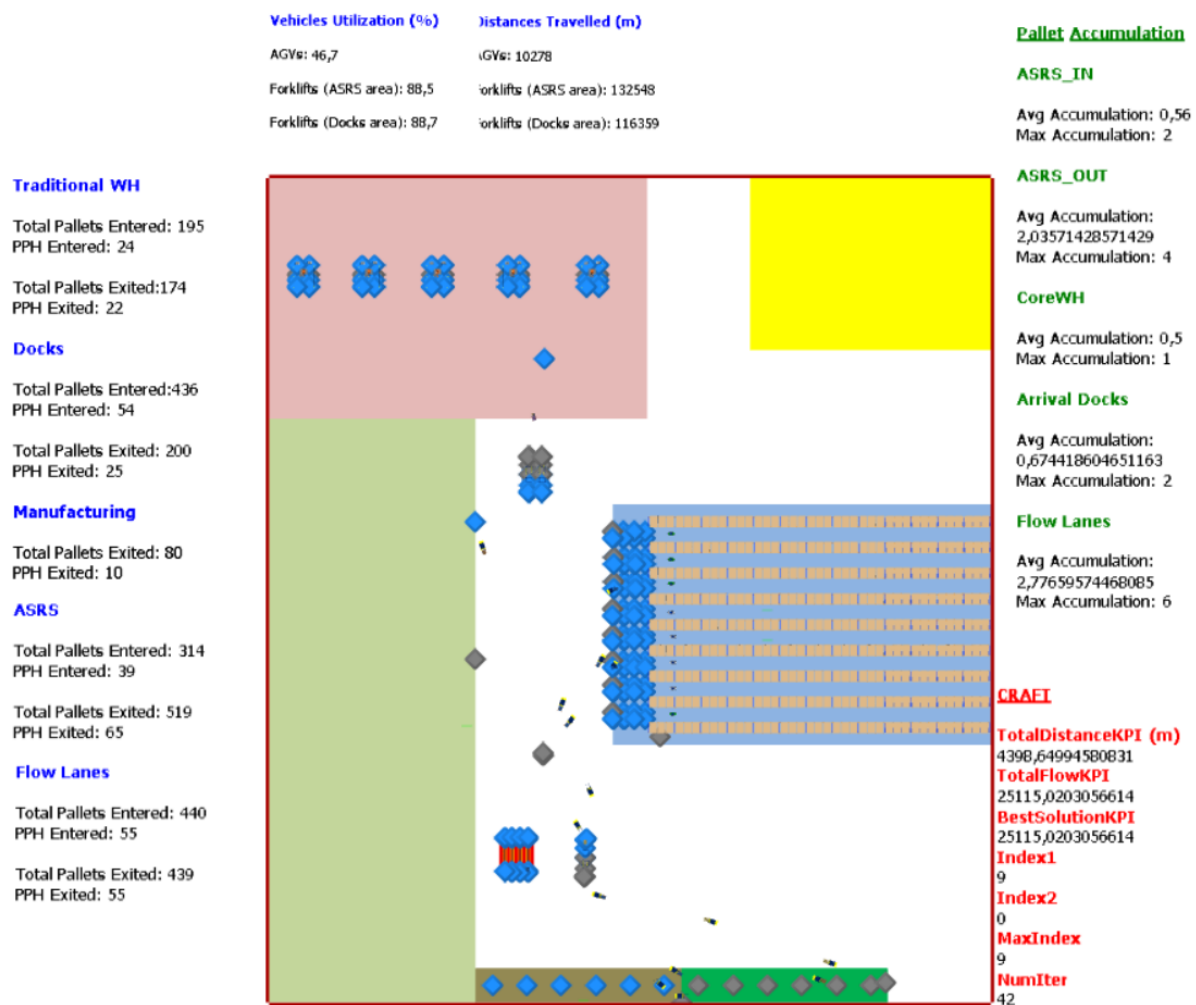


Ilustración 63. Resultados de la simulación de la distribución inicial

5.2.2 DISTRIBUCIÓN OPTIMIZADA

En la distribución optimizada la cuál ha sido generada con el algoritmo de CORELAP se han obtenido los siguientes resultados:

- Carretillas de transporte:
 - Zona de muelles: 3
 - Zona de ASRS: 10
- AGV:
Hay 1 AGV

Tras simular el modelo durante 8 horas se han obtenido los siguientes resultados:

- % de utilización de vehículos:
 - Carretillas de transporte (Zona de los muelles): 87,7%
 - Carretillas de transporte (Zona del ASRS): 87%
 - AGV: 65,4%
- Distancia recorrida por los vehículos
 - Carretillas de transporte (Zona de los muelles): 44.760 metros
 - Carretillas de transporte (Zona del ASRS): 163.079 metros
 - Carretillas de transporte (TOTAL): 207.839 metros
 - AGV (TOTAL): 16.041 metros

En este sistema se ha conseguido reducir en 1% la ocupación de las carretillas de transporte, las distancias recorridas por los vehículos en las 8 horas en las que se ha simulado el modelo han variado con respecto a la distribución inicial de la siguiente manera:

- Carretillas de transporte (TOTAL): - 41.608 metros
- AGV (TOTAL): + 5.763 metros

Y el número de vehículos que se han conseguido reducir con respecto al diseño inicial son los siguientes:

- Carretillas de transporte = 2

- AGV = 0

Y por últimos las colas de los elementos del sistema siguen con una cola media toatl de entidades en ellas de entre 1 y 2.

Por lo que se ha conseguido reducir los costes operacionales de este almacén y aumentar su eficiencia.

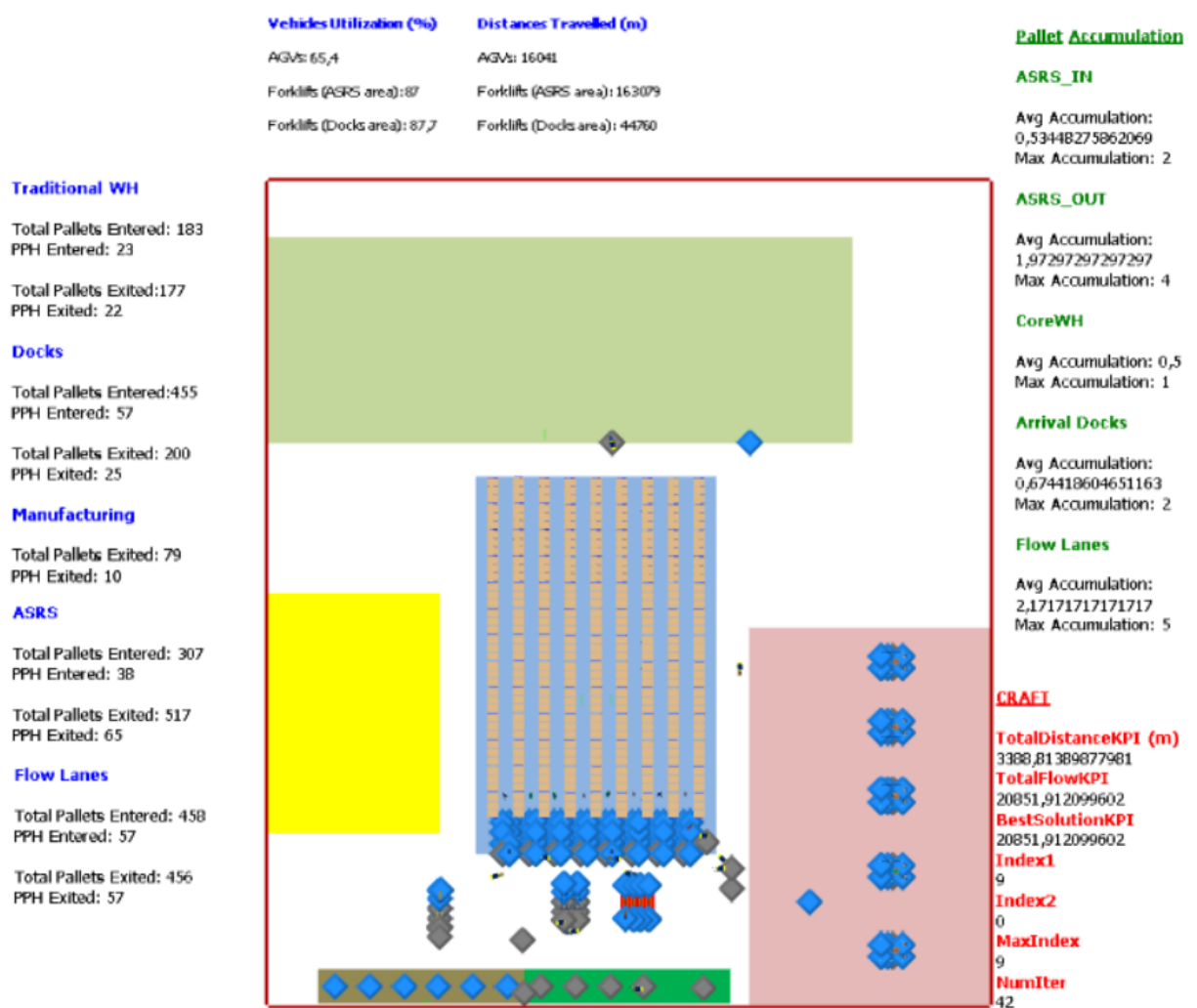


Ilustración 64. Resultados de la simulación de la distribución optimizada

Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

6.1.1 GENERALES

En este estudio, se ha abordado la optimización del diseño de almacenes mediante la aplicación de heurísticas constructivas (CORELAP) y heurísticas de mejora (CRAFT). Los resultados obtenidos han demostrado que ambas técnicas son efectivas para alcanzar soluciones óptimas locales en un tiempo significativamente reducido.

La heurística constructiva CORELAP ha permitido una rápida generación de un diseño inicial eficiente del almacén, al priorizar la cercanía de las áreas con mayor interacción y flujo de materiales. Este enfoque inicial proporciona una base sólida sobre la cual se pueden aplicar técnicas de mejora posteriores.

Por su parte, la heurística de mejora CRAFT ha optimizado el diseño inicial ajustando iterativamente la disposición de las áreas para minimizar los costos de transporte y manejo de materiales. A través de estas iteraciones, CRAFT ha refinado el diseño inicial, alcanzando mejoras notables en la eficiencia operativa del almacén.

Una de las principales ventajas de utilizar estas heurísticas es la capacidad de obtener soluciones viables en un tiempo muy reducido, lo que resulta crucial en entornos empresariales dinámicos donde las decisiones rápidas pueden tener un impacto significativo. Además, la implementación de estos métodos ha demostrado que los cambios en el diseño de los almacenes pueden generar ahorros sustanciales en costos operativos, especialmente a largo plazo.

En resumen, la combinación de las heurísticas CORELAP y CRAFT no solo proporciona un enfoque práctico y eficiente para el diseño de almacenes, sino que también ofrece beneficios económicos tangibles para las empresas. La adopción de estas técnicas puede traducirse en una mayor competitividad y sostenibilidad operativa, al reducir los costos asociados al

manejo y transporte de materiales. Por tanto, se recomienda encarecidamente la consideración de estos métodos en futuros proyectos de optimización de almacenes.

6.1.2 CASO DE USO

Para el caso de uso sacamos como conclusiones el coste que ahorrará si el cliente implementa en el nuevo almacén el diseño propuesto por el algoritmo.

Suponiendo los siguientes costes los cuales han sido calculados por analogía de otros proyectos podemos sacar el ahorro que se podría obtener cada año con esta distribución frente a la inicial:

Costes fijos				
Máquina	Coste	Cantidad	Amortización al año (máximo 20 años)	Coste al año
Carretilla elevadora	20.704 €	2	5%	2.070 €

Costes Humanos				
Operarios	Coste	Cantidad	%Impuestos	Coste al año
Conductor de carretilla	19.700 €	2	30%	51.220 €

Costes variables				
Operarios	Coste por metro recorrido	Total de metros reducidos en una jornada laboral	número de jornadas laborales al año	Coste total al año
Consumible de la Carretilla	0,0025 €	41608	269	27.981 €
Consumible del AGV	0,0001 €	-5763	269	-155 €

Coste Total en 1 año	81.117 €
Coste Total en 20 años	1.622.335 €

Tabla 27. Ahorro total tras la optimización

Ese sería el ahorro final en un año para un almacén, si se realizara para ‘n’ almacenes el ahorro sería esa cantidad por ‘n’.

6.2 TRABAJOS FUTUROS

Como trabajos futuros, se pretende ampliar la biblioteca de optimización incorporando algoritmos adicionales como ALDEP (Automated Layout Design Program), lo que permitirá

explorar nuevas posibilidades y adaptar la solución mejor a otro tipo de distribuciones de almacenes.

Además, se probará la paralelización del proceso de optimización utilizando la librería de Python ‘multiprocessing’ para reducir significativamente los tiempos de ejecución. La implementación de la paralelización promete hacer que las heurísticas sean aún más rápidas y efectivas, facilitando su aplicación en entornos de gran escala y alta complejidad.

Y por último se realizarán ‘stress tests’ los cuales ayudan a hacer el código más robusto y que no de errores al meter cualquier tipo de instancia en el algoritmo.

Capítulo 7. BIBLIOGRAFÍA

- [1] Weng, Li. “Efficient and flexible algorithm for plant layout generation”, West Virginia University 1999. [Efficient and flexible algorithm for plant layout generation \(wvu.edu\)](#)
- [2] Schiffauerova, Dr.Andrea. “Facility location I”, “Facility location II”, “Facility location III”. [Untitled Document \(concordia.ca\)](#)
- [3] Pourvaziri, Hani. “Integrating Facility Layout Design and Aisle Structure in Manufacturing Systems: Formulation and Exact Solution”, Campus de Cézeaux. [Integrating facility layout design and aisle structure in manufacturing systems: Formulation and exact solution - ScienceDirect](#)
- [4] Porcentaje de coste de un trabajador para una empresa. [Coste del trabajador para la empresa: Guía + Fórmula | Personio](#)
- [5] Coste de la licencia de Simio. [Microsoft Word - Student Edition.docx \(simio.com\)](#)
- [6] Coste de ordenador. [Dell XPS 15 7590 15" Core i5 2.4 GHz - SSD 256 GB - 8GB - | Back Market](#)
- [7] Cálculo de amortizaciones. [Cómo calcular la amortización de equipos informáticos \(holded.com\)](#)

ANEXO A

Anexo A

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
I.C.A.I.

PROYECTOS FIN DE MÁSTER
CURSO:

Ficha de proyecto fin de máster
(RELLENAR CON LETRAS DE IMPRENTA EN ORDENADOR)

Titulación y optatividad: BigData y Tecnología Avanzada

Alumno 1º Apellido: Fernández- Figueroa

2º Apellido: Díaz-Trechuelo

Nombre: José Luis

Teléfono de contacto: 629435470

e-mail: joseluisffdt01@gmail.com

Título del Proyecto Fin de Máster: Optimización y Simulación de Layout de un Warehouse.

Director (nombre y dos apellidos): Guillermo Bonmati Arias

Teléfono de contacto: 618133916

e-mail: g.bonmati.arias@accenture.com

Breve descripción del proyecto (5 o 6 líneas):

Optimización del layout de un Warehouse aplicando métodos constructivos y de mejora específicas de layout en Python. Dichos métodos no se encuentran creados en ninguna librería por lo que han sido desarrollado desde cero. Tras la optimización el resultado del layout se simulará en Simio para comparar el escenario baseline (layout inicial) y el layout optimizado.

El documento final del proyecto será subido al Repositorio Institucional de Comillas con acceso público. El alumno podrá solicitar un nivel restringido de acceso (incluido el "cerrado" o "confidencial") que podrá concederse, excepcionalmente, si está plenamente justificado.

The final report of the Project will be uploaded to the Comillas Institutional Repository with public access. The student will be able to ask for a restricted access (even "closed" or "confidential") which will be exceptionally accepted if it is fully justified.

Aceptación del Director (firma y fecha)

**BONMATI
ARIAS
GUILLERMO -
48776189C**

Digitally signed by
BONMATI ARIAS
GUILLERMO -
48776189C
Date: 2025.04.29
09:40:30 +02'00'