



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

PROYECTO FIN DE CARRERA

**APLICACIÓN DE LA TEORÍA DE REDES
COMPLEJAS AL ANÁLISIS DE
VULNERABILIDADES DE UNA RED DE
SATÉLITES**

AUTOR: Antonio Serda Mena

DIRECTOR: Enrique Sevillano Bravo

MADRID, Agosto de 2022

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**Aplicación de la Teoría de Redes Complejas al Análisis de
Vulnerabilidades de una Red de Satélites**
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2021 - 2022 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.:



Fecha: 18 / 08 / 2022

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

SEVILLANO
BRAVO
ENRIQUE -
70065330Q

Firmado
digitalmente por
SEVILLANO
BRAVO ENRIQUE -
70065330Q
Fecha: 2022.08.23
10:43:45 +02'00'

Fdo.:

Fecha: 23 / agosto / 2022

APLICACIÓN DE LA TEORÍA DE REDES COMPLEJAS AL ANÁLISIS DE VULNERABILIDADES DE UNA RED DE SATÉLITES

Autor: Serda Mena, Antonio

Director: Sevillano Bravo, Enrique

Entidad colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Introducción

Ante el desarrollo de nuevos sistemas basados en el Internet de Satélites, se vuelve necesario disponer de métodos capaces de evaluar las conexiones de una red de forma dinámica. Este tipo de constelaciones se basan en la comunicación entre satélites en órbita con el objetivo de cubrir grandes distancias y prescindir, en la medida de lo posible, de infraestructura cableada en tierra. Operando de esta forma, se podrían conseguir ahorros significativos de materias primas, así como una mayor cobertura de las redes de comunicación.

Para evitar la saturación de las órbitas, debido al incremento en la cantidad de desechos espaciales, así como a la aparición de las llamadas «megaconstelaciones», se ha llegado a plantear la interconexión entre satélites no pertenecientes a la misma constelación. De este modo, en aquellos momentos en los que varios satélites contasen con recursos de procesamiento disponibles, podrían generarse conexiones esporádicas entre ellos para transmitir información de una forma más eficiente.

Este nuevo paradigma plantea un nivel de complejidad elevado al poder darse conexiones de forma repentina. Por ello, la representación de todos los elementos de la red en el tiempo, así como sus flujos de información, se vuelven elementos clave para el funcionamiento del sistema. Esto puede realizarse empleando la Teoría de Redes Complejas.

La misma sirve para representar, en forma de grafos, conjuntos de elementos relacionados entre sí mediante vínculos que generan condiciones topológicas no triviales. Dicha teoría tiene su origen a principios del siglo actual y todavía siguen desarrollándose avances dentro de esta.

En este Proyecto se ha representado una hipotética red de satélites, empleando para ello la constelación Globalstar durante el mes de septiembre de 2021, de la que se considera cada satélite como a un nodo de la red y la visibilidad directa entre estos, como un enlace. Se han creado varios programas en MATLAB para determinar los satélites más críticos durante un intervalo de tiempo y el orden de eliminación que generaría la red menos eficiente posible. Esto sirve para determinar cuáles son los satélites más importantes en cada momento y poder tomar medidas en consecuencia de cara al futuro. Para ello se emplean dos métodos distintos.

El primero de ellos es el método «multiplex», basado en el cálculo de las métricas de distintas redes, definidas de forma periódica a lo largo de un intervalo temporal, de las que después se calculan sus valores medios y se decide en consonancia.

El segundo es el «monoplex», basado en el agrupamiento de las distintas redes binarias del método multiplex en una sola red de enlaces ponderados. Los pesos de cada enlace representan la proporción de tiempo de conexión entre dos satélites. Una vez se dispone de la red monoplex, se calculan sus métricas y se decide en consonancia.

Como las métricas de ambos métodos no pueden ser directamente comparadas, es necesario crear otros programas para realizar la comparativa de la efectividad entre ellos. La forma de evaluar esto consiste en comparar el número de coincidencias de las listas de satélites eliminados por cada método y su orden de eliminación con una distribución de probabilidad aleatoria, así como comparar las métricas de cada método con las métricas de las redes recalculadas usando las listas de satélites eliminados de los métodos análogos.

Metodología

Los distintos objetivos del Proyecto fueron surgiendo a medida que este se iba completando, gracias a la metodología de carácter ágil empleada. Tanto la teoría como los aspectos técnicos del mismo fueron reevaluados de forma continua.

Inicialmente, se originó la idea de evaluar los valores promedio de las métricas para todo el intervalo, puesto que este era el método más intuitivo, es decir, el multiplex. Por medio de la investigación sobre casos similares, se descubrió el concepto de «redes complejas multicapa». Gracias a ello se consideró representar la proporción de tiempo total de conexión entre satélites en forma de enlaces ponderados, es decir, por medio del método monoplex. Con ello se obtuvieron dos métodos de análisis diferentes que no tenían por qué ser directamente comparables.

Por último, se desarrollaron las pruebas necesarias para comparar ambos métodos. Primero se identificó la función de la distribución de probabilidad utilizada para determinar si los satélites coincidentes eliminados por ambos métodos seguían una distribución aleatoria. Esto se resolvió mediante el uso de combinatoria. Después se comparó la distribución aleatoria con una distribución experimental, fruto de ejecutar varios análisis de forma iterativa en el tiempo. La prueba de bondad del ajuste realizada fue la prueba χ^2 de Pearson, empleada para comparar distribuciones discretas. Finalmente, se representaron los valores de eficiencia de los distintos métodos y sus versiones recalculadas para determinar la posible superioridad de alguno de los mismos y las ventajas y desventajas de cada uno de ellos.

Resultados

Tras realizar todas las pruebas pertinentes se obtuvieron los siguientes resultados, en donde la métrica de eficiencia es la que debe disminuir al máximo en la red para obtener los mejores resultados de cada método.

- Un mayor nivel de coincidencia entre los satélites eliminados por ambos métodos cuando se ha realizado un número de eliminaciones similar a la mitad de los totales de la red. Esto implica un mayor grado de similitud entre ambos métodos para estas eliminaciones. En las últimas eliminaciones las coincidencias entre ambos métodos tendían a la aleatoriedad. Esto indica un descenso en la eficacia de los métodos a partir de dicho punto.

- Un mayor descenso en la eficiencia del método multiplex que la de su versión recalculada. Esto implica no poder asegurar que el método monoplex ofrezca mejores resultados que el multiplex.
- Un nivel de eficiencia similar entre el método multiplex y su versión recalculada tras las primeras eliminaciones de satélites, así como un aumento de la eficiencia en las primeras eliminaciones para la versión recalculada del método monoplex, encontrándose la original por debajo y descendiendo. Aunque la versión recalculada del multiplex no ofrezca mejores resultados, la versión recalculada del método monoplex tampoco lo hace y además no reduce la eficiencia para estas iteraciones. Esto implica no poder asegurar que el método multiplex ofrezca mejores resultados para las primeras eliminaciones de satélites de la red.
- Un descenso en la eficiencia del método monoplex recalculado, por debajo de la del monoplex, para las eliminaciones centrales de satélites. Esto implica una mayor eficacia del método multiplex sobre el monoplex cuando se ha realizado un número de eliminaciones similar a la mitad de los satélites totales de la red.
- Una mayor variabilidad en los resultados de los experimentos para las últimas eliminaciones de satélites, explicando porqué se da un menor número de coincidencias entre las listas de satélites eliminados según cada método cuando quedan pocos satélites por eliminar.

Conclusiones

En este Proyecto se han desarrollado varios programas en MATLAB que permiten el análisis de redes de satélites, así como su representación en el tiempo y el espacio. Gracias a los mismos se pueden realizar otros estudios relacionados con una mayor facilidad. Todos los programas cuentan con explicaciones de su funcionamiento y en la memoria se detallan los fundamentos teóricos de los mismos.

Todo lo aquí mostrado es una pequeña parte de lo que se espera que constituyan los procesos necesarios para generar redes de Internet de Satélites. Mediante las mismas se pretende obtener una red de comunicaciones más eficiente y rápida, por lo que será necesario emplear una menor cantidad de materias primas y energía, así como generar una menor cantidad de basura espacial para poder seguir utilizando las órbitas disponibles de forma sostenible.

APPLICATION OF COMPLEX NETWORK THEORY TO THE VULNERABILITY ANALYSIS OF A SATELLITE NETWORK

Author: Serda Mena, Antonio

Supervisor: Sevillano Bravo, Enrique

Collaborating institution: ICAI – Universidad Pontificia Comillas

ABSTRACT

Introduction

With the development of new systems based on the Internet of Satellites, it has become necessary to have methods capable of dynamically evaluating the connections of a network. This type of constellation is based on communication between satellites in orbit with the aim of covering large distances and dispensing, as far as possible, with wired infrastructure on the ground. By operating in this way, significant savings in raw materials could be achieved, as well as greater coverage of communication networks.

To avoid the saturation of orbits, due to the increase in the amount of space debris, as well as the appearance of the so-called “megaconstellations”, the interconnection between satellites not belonging to the same constellation has been considered. In this way, when several satellites have processing resources available, sporadic connections between them could be generated to transmit information more efficiently.

This new paradigm poses a high level of complexity as connections can occur suddenly. Therefore, the representation of all network elements in time, as well as their information flows, become key elements for the operation of the system. This can be done using the Complex Network Theory.

This Theory is used to represent, in the form of graphs, sets of elements related to each other by means of links that generate non-trivial topological conditions. It has its origin at the beginning of the present century and advances within it are still being developed.

In this Project a hypothetical satellite network has been represented, using the Globalstar constellation for the month of September 2021, where each satellite is considered as a node of the network and the direct visibility between them, as a link. Several programs have been created in MATLAB to determine the most critical satellites during a time interval and the order of elimination that would generate the least efficient network possible. This is used to determine which satellites are the most important at any given time and to be able to take action accordingly for the future. Two different methods are used for this purpose.

The first one is the “multiplex” method, based on the calculation of the metrics of different networks, defined periodically over a time interval, for which their average values are then calculated and decided accordingly.

The second one is the “monoplex”, based on the grouping of the various binary networks of the multiplex method into a single network of weighted links. The weights of each link

represent the ratio of connection time between two satellites. Once the monoplex network is available, its metrics are calculated and decided accordingly.

As the metrics of the two methods cannot be directly compared, it is necessary to create other programs to compare the effectiveness between them. The way to evaluate this consists of comparing the number of matches of the lists of satellites eliminated by each method and their order of elimination with a random probability distribution, as well as comparing the metrics of each method with the metrics of the networks recalculated using the lists of eliminated satellites of the analogous methods.

Methodology

The different objectives of the project emerged as it was being completed, thanks to the agile methodology employed. Both the theory and the technical aspects of the project were continuously re-evaluated.

Initially, the idea of evaluating the average values of the metrics for the entire interval originated, since this was the most intuitive method, that is, multiplex. Through research on similar cases, the concept of “complex multilayer networks” was discovered. Thanks to this, it was considered to represent the total connection time ratio between satellites in the form of weighted links, that is, by means of the monoplex method. This resulted in two different analysis methods that were not necessarily directly comparable.

Finally, the necessary tests were developed to compare the two methods. First, the probability distribution function used to determine whether the coincident satellites removed by both methods followed a random distribution was identified. This was resolved using combinatorics. The random distribution was then compared to an experimental distribution, being it the result of running several analyses iteratively over time. The goodness-of-fit test performed was Pearson’s χ^2 test, used to compare discrete distributions. Finally, the efficiency values of the different methods and their recalculated versions were plotted to determine the possible superiority of any of them and the advantages and disadvantages of each.

Results

After performing all the relevant tests, the following results were obtained, where the efficiency metric is the one that must decrease to the maximum in the network to obtain the best results for each method.

- A higher level of coincidence between the satellites eliminated by both methods when a number of eliminations similar to half of the network totals have been performed. This implies a higher degree of similarity between both methods for these eliminations. For the last eliminations, the matches between the two methods tended to randomness. This indicates a decrease in the efficiency of the methods from that point on.
- A greater decrease in the efficiency of the multiplex method than that of its recalculated version. This implies that it cannot be assured that the monoplex method gives better results than the multiplex method.
- A similar level of efficiency between the multiplex method and its recalculated version after the first eliminations of satellites, as well as an increase in efficiency in

the first eliminations for the recalculated version of the monoplex method, the original being below and decreasing. Although the recalculated version of the multiplex does not give better results, the recalculated version of the monoplex method does not either and also does not reduce the efficiency for these iterations. This implies that it cannot be assured that the multiplex method gives better results for the first few satellite removals from the network.

- A decrease in the efficiency of the recalculated monoplex method, below that of the monoplex, for the central satellite removals. This implies a higher efficiency of the multiplex method over the monoplex when a number of deletions similar to half of the total satellites in the network have been performed.
- A greater variability in the results of the experiments for the last satellite eliminations, explaining why there is a lower number of coincidences between the lists of eliminated satellites according to each method when there are few satellites left to eliminate.

Conclusions

In this Project several programs, that allow the analysis of satellite networks, as well as their representation in time and space, have been developed in MATLAB. Thanks to these programs, other related studies can be carried out more easily. All the programs have explanations of their operation and the theoretical foundations of each one are detailed in the memory.

Everything shown here is a small part of what is expected to constitute the processes necessary to generate Internet of Satellites networks. The aim is to obtain a more efficient and faster communications network, so it will be necessary to use less raw materials and energy, as well as to generate less space debris in order to continue using the available orbits in a sustainable way.

*A mis abuelos
Juliana y José Antonio.*

*La NASA
la NASA
la NASA
siempre manda cohetes
cohetes
cohetes
que llegan a la Luna
la Luna
la Luna
y se pasan de largo
a veces
a veces.
LOS GANDULES*

Índice

I. Memoria	15
1. Introducción	17
1.1. Contexto actual de la basura espacial	18
1.2. Principales tipos de órbitas terrestres	21
1.3. Internet de Satélites (IoSat)	22
1.4. Teoría de Redes Complejas	23
1.4.1. Métricas básicas	26
1.4.2. Métricas de integración	27
1.4.3. Métricas de segregación	29
1.4.4. Métricas de centralidad	30
1.5. Constelación Globalstar	31
2. Estado tecnológico actual	35
2.1. Bases de datos de satélites	35
2.1.1. UCS Satellite Database: Identificador NORAD	35
2.1.2. Celestrack: <i>Two Line Element</i> (TLE)	35
2.2. MATLAB	37
2.3. Algoritmo SGP4	38
2.4. Teoría de Redes Complejas	38
2.4.1. Teoría de Redes Complejas Multicapa	38
2.4.2. Funciones para el cálculo de métricas	39
2.5. Análisis de colisiones en órbita	39
2.5.1. SOCRATES	39
2.5.2. MASTER	40
3. Modelo desarrollado	41
3.1. Obtención de las posiciones de los satélites	41
3.1.1. Toma de datos	42
3.1.2. Recursos empleados	43
3.1.3. Algoritmo empleado	44
3.1.3.1. Inicio y selección de los parámetros de funcionamiento	44
3.1.3.2. Capa 1 de la estructura de almacenamiento (Sat_Cloud)	46
3.1.3.3. Capa 2 de la estructura de almacenamiento (TLE_Data)	47
3.1.3.4. Capa 2 de la estructura de almacenamiento (Position)	51
3.1.3.5. Disposiciones finales	54
3.2. Análisis de vulnerabilidades de la red de satélites	56
3.2.1. Toma de datos	58
3.2.2. Recursos empleados	59
3.2.3. Algoritmo empleado	59
3.2.3.1. Inicio y selección de los parámetros de funcionamiento	60
3.2.3.2. Creación de la estructura de redes multiplex (Sat_Network)	61

3.2.3.3.	Estructura de redes multiplex (Sat_Network) y (Sat_Multiplex)	63
3.2.3.4.	Estructura de redes monoplex (Sat_Monoplex)	65
3.2.3.5.	Lista de satélites eliminados y nivel de coincidencia en cada iteración .	67
3.2.3.6.	Intercambio de los satélites eliminados por ambos métodos	68
3.2.3.7.	Representación gráfica de la red	73
3.3.	Comparación de los métodos multiplex y monoplex	75
3.3.1.	Toma de datos	77
3.3.2.	Recursos empleados	77
3.3.3.	Algoritmo empleado	78
3.3.3.1.	Realización de múltiples experimentos (Análisis.m)	78
3.3.3.2.	Probabilidad de coincidencia aleatoria entre satélites (Probabilidad.m) .	79
3.3.3.3.	Comparación de los métodos multiplex y monoplex (Comparacion.m) .	79
4.	Análisis de resultados	83
4.1.	Condiciones de experimentación	83
4.2.	Resultados	85
5.	Conclusiones	107
5.1.	Metodología	108
5.2.	Resultados	109
5.3.	Recomendaciones para futuros estudios	109
	Bibliografía	111
II.	Objetivos de Desarrollo Sostenible	117
1.	Alineamiento con los Objetivos de Desarrollo Sostenible	119
1.1.	4 Educación de calidad	120
1.2.	9 Industria, innovación e infraestructura	120
1.3.	12 Producción y consumo responsables	121
III.	Código fuente	123
1.	Obtención de las posiciones de los satélites	125
1.1.	TLE_Satdata.m	125
1.2.	Programa para la obtención de las posiciones de los satélites (Nube.m)	126
2.	Análisis de vulnerabilidades de la red de satélites	133
2.1.	Programa para el análisis de vulnerabilidades de la red de satélites (Red.m) . . .	133
3.	Comparación de métodos de análisis de vulnerabilidades de la red de satélites	149
3.1.	Programa para el almacenamiento de métricas de la red de satélites para distintos intervalos periódicos (Análisis.m)	149
3.2.	Programa para el cálculo de la probabilidad de coincidencia aleatoria entre satélites eliminados por cada uno de los métodos de análisis (Probabilidad.m) .	164
3.3.	Programa para la comparación de la eficacia de los métodos multiplex y monoplex (Comparacion.m)	164

Índice de figuras

1. Objetos en órbita identificados, catalogados y en observación según origen y año [3]	18
2. Número de lanzamientos según tipo y año [7]	19
3. Impacto a hipervelocidad [17]	20
4. Objetos en órbita identificados, catalogados y en observación según órbita y año [18]	21
5. Principales tipos de órbita según altitud [19]	22
6. Principales tipos de red según las características de los enlaces [25]	25
7. Principales métricas básicas en una red [25]	27
8. Esquema de conexión de la constelación Iridium [29]	32
9. Satélite de la constelación Globalstar [30]	32
10. Mapa de cobertura de la constelación Globalstar [31]	33
11. Línea de título de un TLE asociado a la Estación Espacial Internacional [38]	36
12. Línea 1 de un TLE asociado a la Estación Espacial Internacional [38]	36
13. Línea 2 de un TLE asociado a la Estación Espacial Internacional [38]	36
14. Parámetros orbitales [39]	37
15. Representación general de un red multicapa [46]	38
16. 10 colisiones más probables según SOCRATES (04/06/2022-11/06/2022) [49]	40
17. Interfaz de uso de MASTER [51]	40
18. Ejes terrestres [52]	41
19. Formulario de solicitud de datos especiales de CelesTrack [53]	43
20. Fila 1 de la estructura Sat_Cloud	46
21. Filas 1 y 2 de la estructura TLE_Data	48
22. Filas 207 y 208 de la estructura Position	51
23. Diagrama conceptual para el cálculo de la posición de un satélite	52
24. Coordenadas del satélite 25652 durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	54
25. Coordenadas de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	55
26. Coordenadas del satélite 25652 durante el día 01/09/2021	55
27. Coordenadas de la constelación Globalstar durante el día 01/09/2021	55
28. Representación gráfica de la geometría asociada a la función <i>rise-set</i> [55]	56
29. Comparación de eficiencia entre las redes multiplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	69
30. Diferencia de eficiencias entre las redes multiplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	69
31. Estructura Sat_Multiplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	70
32. Estructura Sat_Network para la constelación Globalstar durante el día 01/09/2021 (00:18)	70
33. Estructura Sat_Rec_Multiplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	70

34. Comparación de eficiencia entre las redes monoplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	71
35. Diferencia de eficiencias entre las redes monoplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	71
36. Estructura Sat_Monoplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	72
37. Estructura Sat_Rec_Monoplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)	72
38. Grafo de fuerzas de la constelación Globalstar durante el día 01/09/2021 (00:00)	73
39. Grafo circular de la constelación Globalstar durante el día 01/09/2021 (00:00)	73
40. Grafo de capas de la constelación Globalstar durante el día 01/09/2021 (00:00)	74
41. Posiciones y enlaces de la constelación Globalstar durante el día 01/09/2021 (00:00)	74
42. Grafo de fuerzas de la constelación Globalstar durante el día 01/09/2021	74
43. Grafo circular de la constelación Globalstar durante el día 01/09/2021	75
44. Grafo de capas de la constelación Globalstar durante el día 01/09/2021	75
45. Coordenadas del satélite 25652 durante el mes 09/2021	84
46. Coordenadas de la constelación Globalstar durante el mes 09/2021	84
47. Probabilidad de coincidencia aleatoria entre satélites eliminados para las iteraciones 1-16	85
48. Probabilidad de coincidencia aleatoria entre satélites eliminados para las iteraciones 17-32	86
49. Proporción de coincidencias entre satélites eliminados para las iteraciones 1-16	86
50. Proporción de coincidencias entre satélites eliminados para las iteraciones 16-32	87
51. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 1)	87
52. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 2)	88
53. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 3)	88
54. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 4)	88
55. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 5)	89
56. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 6)	89
57. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 7)	89
58. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 8)	90
59. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 9)	90
60. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 10)	90
61. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 11)	91
62. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 12)	91
63. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 13)	91

64. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 14)	92
65. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 15)	92
66. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 16)	92
67. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 17)	93
68. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 18)	93
69. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 19)	93
70. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 20)	94
71. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 21)	94
72. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 22)	94
73. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 23)	95
74. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 24)	95
75. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 25)	95
76. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 26)	96
77. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 27)	96
78. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 28)	96
79. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 29)	97
80. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 30)	97
81. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 31)	97
82. Valores medios de las distribuciones de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021	99
83. Evolución de los valores medios de eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021	100
84. Evolución de las diferencias de eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021	100
85. Evolución de la eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021	100
86. Evolución de la eficiencia con cada iteración del método multiplex recalculado para la constelación Globalstar durante el mes 09/2021	101
87. Evolución de la eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021	101

88. Evolución de la eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021	101
89. Evolución de la eficiencia con cada iteración del método multiplex recalculado para la constelación Globalstar durante el mes 09/2021	102
90. Evolución de los valores medios de eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021	102
91. Evolución de las diferencias de eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021	103
92. Evolución de la eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021	103
93. Evolución de la eficiencia con cada iteración del método monoplex recalculado para la constelación Globalstar durante el mes 09/2021	103
94. Evolución de la eficiencia con cada iteración del método monoplex y monoplex recalculado para la constelación Globalstar durante el mes 09/2021	104
95. Evolución de la eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021	104
96. Evolución de la eficiencia con cada iteración del método monoplex recalculado para la constelación Globalstar durante el mes 09/2021	104
97. Objetivos de Desarrollo Sostenible	119

Índice de tablas

1. Elementos de la línea de título de un TLE [34] [38]	36
2. Elementos de la línea 1 de un TLE [34] [38]	36
3. Elementos de la línea 2 de un TLE [34] [38]	37
4. Identificadores NORAD de los satélites de la constelación Globalstar a 01/09/2021 [33]	42
5. p-valores de la prueba χ^2 de Pearson, valores medios de las distribuciones aleatoria y experimental y proporción de no aleatoriedad de la distribución experimental	98

Índice de extractos de código

1. Obtención de las posiciones de los satélites: almacenamiento de la lista de identificadores NORAD (PARTE III, sección 1.2, líneas 44-49)	45
2. Obtención de las posiciones de los satélites: selección de fechas de análisis (PARTE III, sección 1.2, líneas 51-67)	45
3. Obtención de las posiciones de los satélites: selección del periodo de muestreo (PARTE III, sección 1.2, líneas 69-75)	46
4. Obtención de las posiciones de los satélites: almacenamiento de identificadores NORAD (PARTE III, sección 1.2, líneas 79-82)	47
5. Obtención de las posiciones de los satélites: almacenamiento de los nombres de los archivos .txt en donde se encuentran contenidos los TLE (PARTE III, sección 1.2, líneas 84-87)	47
6. Obtención de las posiciones de los satélites: almacenamiento de los TLE (PARTE III, sección 1.2, líneas 89-95)	47
7. Obtención de las posiciones de los satélites: análisis de las fechas de los TLE (PARTE III, sección 1.2, líneas 97-116)	48
8. Obtención de las posiciones de los satélites: almacenamiento de los parámetros necesarios para utilizar el modelo SPG4 (PARTE III, sección 1.2, líneas 118-146)	49
9. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 148-155)	49
10. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 159-164)	50
11. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 166-187)	50
12. Obtención de las posiciones de los satélites: asignación de las fechas determinadas por el periodo de muestreo (PARTE III, sección 1.2, líneas 191-203)	52
13. Obtención de las posiciones de los satélites: aplicación del modelo SPG4 y la ley de la palanca inversa para el cálculo de las posiciones y velocidades de cada satélite (PARTE III, sección 1.2, líneas 205-248)	53
14. Análisis de vulnerabilidades de la red de satélites: parámetros de funcionamiento (PARTE III, sección 2.1, líneas 22-59)	60
15. Análisis de vulnerabilidades de la red de satélites: ajuste de fechas del nuevo intervalo (PARTE III, sección 2.1, líneas 92-105)	61
16. Análisis de vulnerabilidades de la red de satélites: almacenamiento de las posiciones de los satélites (PARTE III, sección 2.1, líneas 129-142)	62
17. Análisis de vulnerabilidades de la red de satélites: creación de la matriz de adyacencia inicial (Sat_Network) (PARTE III, sección 2.1, líneas 144-169)	62
18. Análisis de vulnerabilidades de la red de satélites: modificación de las nuevas matrices de adyacencia inicial (Sat_Network) (PARTE III, sección 2.1, líneas 176-191)	63

19. Análisis de vulnerabilidades de la red de satélites: selección del satélite con mayor centralidad de intermediación (Sat_Multiplex) (PARTE III, sección 2.1, líneas 354-375)	65
20. Análisis de vulnerabilidades de la red de satélites: creación de la matriz de pesos (Sat_Monoplex) (PARTE III, sección 2.1, líneas 431-437)	66
21. Análisis de vulnerabilidades de la red de satélites: comparación de los satélites eliminados por ambos métodos (PARTE III, sección 2.1, líneas 628-636)	68
22. Análisis de vulnerabilidades de la red de satélites: modificación de la matriz de adyacencia en la nueva red multiplex (Sat_Rec_Multiplex) (PARTE III, sección 2.1, líneas 649-662)	68
23. Bucle para el cálculo de la probabilidad de coincidencia aleatoria entre satélites eliminados por cada uno de los métodos de análisis (PARTE III, sección 3.2, líneas 15-26)	79
24. Eliminación de filas y columnas innecesarias de las variables de entrada (PARTE III, sección 3.3, líneas 54-64)	80
25. Ejecución de la prueba χ^2 de Pearson entre la distribución de probabilidad aleatoria y la muestra de los múltiples experimentos (PARTE III, sección 3.3, líneas 68-75)	80

Acrónimos

<i>ESA</i>	European Space Agency (Agencia Espacial Europea)
<i>IADC</i>	Inter-Agency Space Debris Coordination Committee
<i>ICAI</i>	Instituto Católico de Artes e Industrias
<i>MIT</i>	Massachusetts Institute of Technology.
<i>NASA</i>	National Aeronautics and Space Administration
<i>NORAD</i>	North American Aerospace Defense Command
<i>OCDE</i>	Organización para la Cooperación y el Desarrollo Económicos
<i>ONU</i>	Organización de las Naciones Unidas
<i>UCS</i>	Union of Concerned Scientists

Símbolos

$[A]$	Matriz de adyacencia (1)
$a_{i,j}$	Elemento de la matriz de adyacencia (1)
b_i	Centralidad de intermediación o <i>betweenness centrality</i> de un nodo (22)
b_i^w	Centralidad de intermediación o <i>betweenness centrality</i> de un nodo en una red ponderada (23)
C	Coficiente global de agrupamiento o <i>global clustering coefficient</i> de la red (20)
C^w	Coficiente global de agrupamiento o <i>global clustering coefficient</i> de una red ponderada (21)
C_i	Coficiente local de agrupamiento o <i>local clustering coefficient</i> de un nodo (18)
C_i^w	Coficiente local de agrupamiento o <i>local clustering coefficient</i> de un nodo en una red ponderada (19)
$[D]$	Matriz de distancias entre nodos (7)
$d_{i,j}$	Distancia o <i>shortest path length</i> entre nodos (5)
$d_{i,j}^w$	Distancia o <i>shortest path length</i> entre nodos en una red ponderada (6)
E	Eficiencia global de la red (12)
E^w	Eficiencia global de una red ponderada (17)
E_i	Eficiencia de un nodo (14)
E_i^w	Eficiencia de un nodo en una red ponderada (15)
k_i	Grado de un nodo (3)
k_i^w	Grado de un nodo en una red ponderada (4)
L	Longitud de trayectoria característica o <i>characteristic path length</i> de la red (12)
L^w	Longitud de trayectoria característica o <i>characteristic path length</i> de una red ponderada (13)
L_i	Longitud de trayectoria o <i>path length</i> de un nodo (10)
L_i^w	Longitud de trayectoria o <i>path length</i> de un nodo en una red ponderada (11)
N	Conjunto de nodos de la red
n	Número de nodos de la red
t_i	Número de triángulos alrededor de un nodo (8)
t_i^w	Número de triángulos alrededor de un nodo en una red ponderada (9)
$[W]$	Matriz de pesos (2)
$w_{i,j}$	Elemento de la matriz de pesos (2)

PARTE I



MEMORIA



Capítulo 1

Introducción

ESTE Proyecto se centra en el desarrollo de una herramienta, programada en MATLAB, para la identificación de las vulnerabilidades de una red en órbita. Para ello se emplea como herramienta matemática principal la «Teoría de Redes Complejas». Mediante los parámetros definidos por la misma, se buscarán los satélites más críticos para la arquitectura de la red en caso de ser inutilizados. Para ello, será necesario modelar la red como un grafo durante distintos instantes de tiempo. Cada uno de los satélites representará un «nodo» de la red, mientras que las conexiones serán asumidas siempre que exista una línea de visión directa entre ellos.

Al tratarse de redes en continuo movimiento, por lo tanto, con la presencia de cambios en su estructura, se evaluarán dos enfoques distintos. El primero, mediante los valores medios de las métricas de las redes modeladas dentro de un intervalo de tiempo, calculadas para periodos de tiempo más pequeños. El segundo, mediante una red única, ponderando las conexiones a lo largo de ese intervalo y calculando después sus métricas. Finalmente, se compararán ambos métodos.

Es decir, para un intervalo dado, como por ejemplo el tiempo que tarda en realizar una órbita la constelación, el primer método modela una red para cada instante del intervalo, por ejemplo cada 6 min. Tras esto, se calculan los valores medios empleando las métricas de cada red. El segundo método, modela sólo una red. En ella, las conexiones son representadas mediante un valor entre 0 y 1; dicho valor ejemplifica la proporción de tiempo del intervalo en donde los satélites se encontraban conectados. Finalmente se comparan los valores medios de las métricas del primer método con las métricas de la red del segundo. También se analizarán las similitudes o diferencias entre los satélites críticos identificados por cada uno de los métodos.

El enfoque principal del Proyecto se encuentra en los satélites ubicados en «órbitas terrestres bajas», también conocidas como «OTB» o «LEO», según sus siglas en inglés. La motivación tras esto reside en el mayor número de objetos presentes en estas órbitas, tanto satélites como basura espacial, el mayor nivel de riesgo de colisión como consecuencia de lo anterior y la presencia de constelaciones y megaconstelaciones de satélites. Además, son este tipo de órbitas las que tienen un mayor interés estratégico para el desarrollo de proyectos, tanto de empresas privadas, como de entes públicos.

La red de satélites escogida para realizar todos los experimentos ha sido la constelación «Globalstar». El intervalo de tiempos seleccionado ha sido el mes de septiembre de 2021.

Este Proyecto surge bajo la idea y el trabajo previo realizado por el Director del mismo, Enrique Sevillano Bravo. En este capítulo se detallan los aspectos principales que han servido como motivación del Proyecto, así como la teoría que lo sustenta.

1.1. Contexto actual de la basura espacial

Desde el comienzo de la carrera espacial en 1957, cerca de 6.200 cohetes han sido lanzados. Estos han sido capaces de situar en órbita alrededor de 12.980 satélites, de los cuales 8.300 todavía siguen en el espacio; sólo en torno a 5.400 de estos últimos se encuentran actualmente operativos. Tanto los restos de cohetes como los satélites no operativos suponen elementos de alto riesgo para el resto de cuerpos en órbita. Estos son capaces de generar grandes cantidades de fragmentos, ya sea al explotar los remanentes de combustible o las baterías o bien, debido a las colisiones con otros fragmentos. Estas colisiones generan a su vez nuevos impactos con un incremento exponencial en lo que se conoce como «Efecto Kessler»[1]. Hasta la fecha se han detectado más de 630 eventos generadores de fragmentos. El número de objetos catalogados y en observación por las agencias de vigilancia asciende a 30.910; además de estos, se estiman mediante modelos estadísticos 36.500 objetos de más de 10 cm de diámetro, 1.000.000 de entre 1 cm y 10 cm y 130 millones de objetos de entre 1 mm y 1 cm. [2]

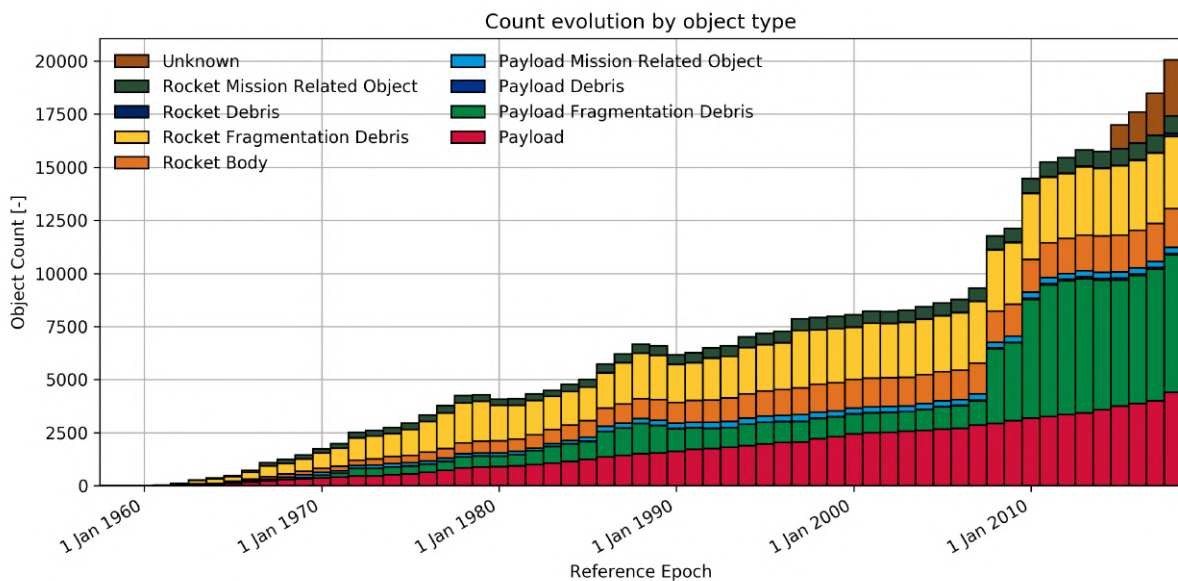


Figura 1. Objetos en órbita identificados, catalogados y en observación según origen y año [3]

Como puede observarse en la Figura 1, la mayor parte de la basura espacial proviene de los impactos con otros fragmentos. Otra fuente importante en aumento constante, aunque no relacionada con las colisiones, son los restos de combustible sólido que quedan en suspensión tras los lanzamientos. Estos restos de óxido de aluminio (Al_2O_3) forman pequeñas partículas que, a la larga, van impactando y dañando los equipos en órbita. Al no poderse evitar este tipo de fragmentos, es importante reducir al máximo las probabilidades de impacto entre otros objetos, así como generar las mínimas cantidades posibles de basura espacial. [3]

La ESA ya se encuentra desarrollando sistemas para la recogida de desechos espaciales. El programa ClearSpace-1 tiene el objetivo de recoger por primera vez un satélite inhabilitado y eliminarlo de órbita a partir de 2025. Según las conclusiones de estudios realizados por la ESA y

la NASA, iniciativas para la eliminación activa de basura espacial serán necesarias para mitigar los efectos que esta genere y evitar que la situación se vuelva incontrolable. [4]

Estos esfuerzos se dan en el contexto de un incremento en el número de lanzamientos, así como en la aparición de las llamadas «megaconstelaciones». Respecto a los nuevos lanzamientos, la gran mayoría son de tipo comercial, siendo satélites pequeños con masas inferiores a la tonelada. Si bien es cierto que cada vez más lanzamientos siguen las «Pautas de Mitigación de Desechos Espaciales»[5] del IADC, las medidas tomadas son aun insuficientes. Mejoras notables pueden observarse en la gestión de los desechos de los cohetes o en la de los satélites situados en órbitas geoestacionarias, sin embargo, aproximadamente el 80 % de los operadores con misiones en órbitas bajas no asumieron medidas para la reducción y eliminación de los desechos asociados en 2019. [6]



Figura 2. Número de lanzamientos según tipo y año [7]

Respecto a las megaconstelaciones, al igual que la mayoría de cuerpos, se sitúan en las órbitas bajas. Estas surgen en los últimos años, en el marco de las corporaciones privadas, con el objetivo de dar cobertura de internet y servicios de telecomunicaciones en aquellas zonas del planeta en donde no exista infraestructura cableada. Los motivos tras la elección de este tipo de órbita son la mayor velocidad de transmisión y los menores costes de lanzamiento; el motivo tras el uso de un elevado número de satélites es la necesidad de generar redes suficientemente malladas y amplias, capaces de transmitir grandes volúmenes de información.

Las tres mayores constelaciones planeadas, Starlink, Kuiper y OneWeb contarían con 42.000, 3.236 y 1.260 satélites respectivamente [8]. En caso de completarse estos proyectos el número de cuerpos en órbita sería dos veces y media superior al actual, sin contar con aquellos que pudieran generarse como fruto de nuevas colisiones. En la actualidad, solamente Starlink y OneWeb tienen satélites lanzados: 1.815 y 394 respectivamente [9].

Si constelaciones como Starlink cumplieren sólo con los requerimientos mínimos legales actuales en relación a los desechos espaciales se estima que los niveles de basura en órbita

crecerían al doble de velocidad que en un escenario base sin megaconstelaciones. Estos requerimientos fijan un límite del 90 % de desechos tras cada misión, «PMD» o *Post-Mission Disposal*, que ha de reentrar en la atmósfera o situarse en una «órbita cementerio»¹. Un límite del 95 % de PMD supondría un 19 % más de desechos, mientras que un límite del 99 % implicaría un escenario neutro. [8]

Las megaconstelaciones suponen actualmente un incremento en la cantidad de desechos en órbita, la saturación de las órbitas bajas, cambios en la composición de la atmósfera [11] y un mayor nivel de contaminación lumínica en el cielo nocturno debido a la reflectancia de los objetos, ya sean satélites o basura espacial; esto último impide la observación astronómica [12].

Según los datos consultados, el panorama respecto a la seguridad espacial no parece muy prometedor. Las colisiones en órbita pueden suponer la muerte de astronautas de la Estación Espacial, la pérdida de funcionalidades como internet, servicios meteorológicos o telecomunicaciones, la interrupción de la investigación científica, la pérdida de órbitas y cuantiosas pérdidas económicas. Para este último ámbito ya existe un informe de la OCDE [13] en donde se evalúan este tipo de costes de forma cualitativa, con gastos que van desde seguros hasta medidas de protección físicas. Estos últimos podrían rondar entre el 5 % y el 10 % del coste de la misión en órbitas geoestacionarias, siendo los mismos superiores en misiones a baja altitud debido a los mayores niveles de riesgo asociados a las órbitas. [14]

Para poner en contexto los efectos físicos de este tipo de colisiones hay que tener en cuenta las velocidades a las que los fragmentos de basura espacial orbitan: entre 7 y 8 km/s. Aun así, la mayoría de colisiones suelen darse a 10 km/s, es decir, 36 000 km/h; algunas han llegado hasta los 15 km/s. Una de estas colisiones con un fragmento de 1 cm de diámetro puede liberar la misma energía que el choque de un coche a 40 km/h [15] [16]. En la Figura 3 puede observarse el impacto de una esfera de aluminio de 1,2 cm de diámetro y 1,7 g contra un bloque de aluminio de 18 cm de grosor.

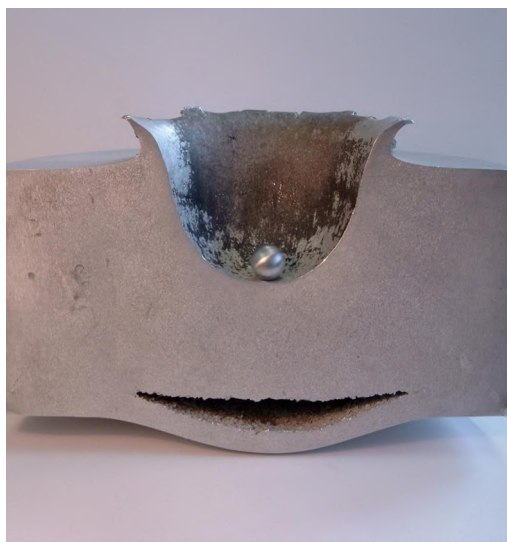


Figura 3. Impacto a hipervelocidad [17]

¹Este tipo de órbitas pueden encontrarse a una mayor altura, para satélites situados en órbitas geoestacionarias, ya que frenar el satélite requeriría más combustible; o situarse aun más cerca de la superficie terrestre, para cuerpos situados en órbitas bajas, en donde el rozamiento con la atmósfera va frenando el material hasta caer, desintegrándolo en el proceso si es suficientemente pequeño. [10]

Como ya se ha comentado, este tipo de colisiones son más probables en las órbitas bajas debido al mayor número de objetos presentes; en la Figura 4 puede apreciarse este fenómeno. Si bien no todas las colisiones producen la destrucción del satélite, la exposición continuada a estas partículas van dañándolo poco a poco. Es por esto por lo que las órbitas bajas cobran un mayor interés para los análisis realizados en este Proyecto. Como conclusión, es necesario tener en cuenta que cada satélite inutilizado pasaría a considerarse automáticamente basura espacial. De ahí la importancia de disponer de sistemas capaces de evaluar el riesgo asociado a cada uno así como la importancia relativa del satélite dentro de la constelación.

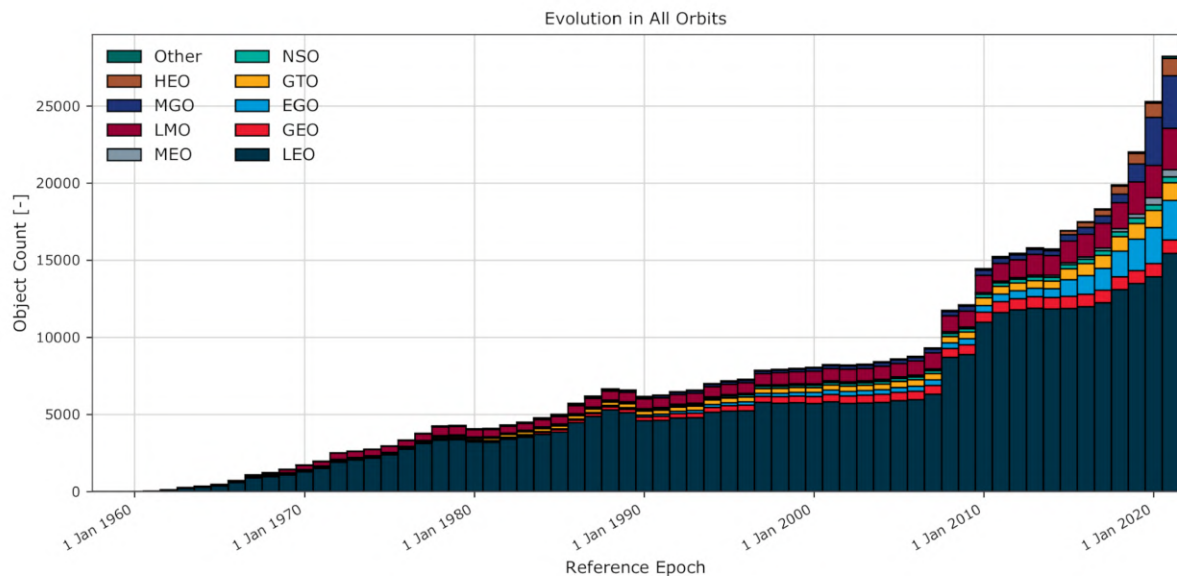


Figura 4. Objetos en órbita identificados, catalogados y en observación según órbita y año [18]

1.2. Principales tipos de órbitas terrestres

Aun teniendo en cuenta la existencia de múltiples tipos de órbita alrededor de la Tierra, estos pueden resumirse en tres principales según su altitud, que a su vez se encuentra inversamente relacionada con la velocidad. El fundamento teórico tras esto reside en la fuerza de atracción de la gravedad de la Tierra sobre los cuerpos de su entorno. La fuerza es inversamente proporcional al cuadrado de la distancia, por ello, cuanto más cerca de la superficie terrestre se encuentre un cuerpo, más velocidad tangencial necesitará para mantener un movimiento estable y no caer. Esto produce un efecto contraintuitivo por el cual es necesario reducir la velocidad de rotación del satélite para poder llevarlo hasta una órbita más baja; una vez allí esté alcanzará una situación de equilibrio aumentando su velocidad. [19]

De esta forma, en las «órbitas terrestres altas», también conocidas como «HEO», según sus siglas en inglés, se encuentran cuerpos orbitando a menor velocidad. Estas son consideradas a partir de alturas superiores a 42 164 km desde el centro de la Tierra; aproximadamente 35 700 km desde la superficie. Justo a esta altura la velocidad de órbita se iguala a la velocidad de rotación de la Tierra, teniendo una gran utilidad para aquellas aplicaciones en las que se requiera una vista fija de la superficie, como por ejemplo en observación meteorológica o telecomunicaciones. La distancia a la que se encuentran permiten la cobertura casi total sobre la cara visible de la Tierra; como consideración, es necesario orbitar alrededor del ecuador para disponer de una vista

fija de la superficie [20]. A estas órbitas se las conoce como «órbitas geosíncronas» o «GEO» según su acrónimo. [19]

Otro tipo de órbita son las «órbitas terrestres medias», también conocidas como «MEO», según sus siglas en inglés. Se encuentran ubicadas entre 1700 y 35 700 km sobre la superficie terrestre; para algunos autores la distancia mínima es de 2000 km [19]. Los periodos de estas órbitas se encuentran entre 2 y 24 h, siendo la banda de 2000 km la más importante por ofrecer periodos de 12 h. Debido a su predictibilidad, suelen ser empleadas por sistemas de navegación como GPS o Galileo, aunque, a diferencia de las órbitas geosíncronas, no es necesario limitarse al ecuador ya que no es posible mantener referencias fijas sobre la Tierra [20]. [21]

Finalmente, las órbitas en las que se centra el Proyecto son las «órbitas terrestres bajas». Se encuentran entre 80 y 1700 km o entre 180 y 2000 km según otros autores [19]; todavía no existe un consenso al respecto. Para este tipo de órbitas se asumen periodos inferiores a 2 h. La mayor proximidad respecto a la superficie terrestre las hace especialmente atractivas para la toma de imágenes en alta definición. También llevan asociadas menores costes de puesta en órbita y poseen una baja latencia en las comunicaciones, sin embargo, no son muy útiles para uso general en telecomunicaciones debido a la dificultad para rastrear la posición de los satélites a esas velocidades; además, al igual que las órbitas medias, pueden tener cualquier inclinación. Por ello suelen emplearse para formar constelaciones de satélites capaces de dar cobertura constante a toda la Tierra actuando como una red. Debido a este tipo de estructuras, es la órbita que cuenta con más satélites activos: 4.078 de los 4.852 que contabilizó la UCS en enero de 2022 [9]. [20]



Figura 5. Principales tipos de órbita según altitud [19]

1.3. Internet de Satélites (IoSat)

Ya se han descrito las principales órbitas y cómo es en las bajas donde se acumula una mayor cantidad de cuerpos. Esto presenta un reto para las misiones que allí se encuentren, siendo este mayor al aumentar la complejidad de los sistemas presentes. Es en las llamadas constelaciones en donde el grado de complejidad puede alcanzar los niveles máximos. Estas pueden estar formados por satélites independientes entre sí, en donde las conexiones de cada satélite se dan exclusivamente con elementos en tierra, o bien realizar conexiones entre satélites para transmitir información de forma inalámbrica. Las tendencias recientes muestran nuevas preferencias por este último tipo de conexión debido a los nuevos usos que pueden llevar asociados. [22]

Sin entrar en un análisis detallado, las conexiones entre satélites permiten una mayor eficiencia al no necesitar una estructura cableada en tierra y poder compartir recursos; también

permiten un mayor grado de flexibilidad, pues las conexiones podrían modificarse en función de las necesidades de cada instante. El nombre por el que se conoce a este tipo de conexiones es «Comunicaciones Inter-Satelitales» o «ISC», según sus siglas en inglés [22]. En el futuro podrían darse incluso las conexiones entre satélites de distintas redes, incrementando aun más la cobertura y la eficiencia en las telecomunicaciones en lo que se conoce como «Internet de Satélites» o «IoSat», según su acrónimo en inglés. Los planteamientos de base son los mismos que los que se encuentran detrás del «Internet de las Cosas» en tierra: conexiones dinámicas distribuidas, generadas de forma colaborativa, según las necesidades y los recursos disponibles en cada momento. [23]

Al producirse las conexiones de forma intermitente, es posible la interrupción de las comunicaciones debido a la saturación de los distintos elementos, sucediendo en caso de disponer de un número insuficiente de conexiones o caminos entre satélites. Los motivos podrían ir desde un aumento en la cantidad de información a procesar, hasta la pérdida de alguno de los satélites debido a colisiones con el entorno; siendo más problemático el efecto cuanto más crítico para la red fuese el satélite inutilizado. Ante dichas situaciones, surge este Proyecto con el objetivo de dar respuesta con antelación a cuáles serían los satélites más importantes de una constelación para su adecuado funcionamiento a lo largo del tiempo.

1.4. Teoría de Redes Complejas

Debido a su funcionamiento, las futuras constelaciones de satélites pueden modelarse como una red compleja. Este tipo de redes están formadas por un conjunto de elementos relacionados entre sí mediante vínculos de difícil representación directa a través de modelos físicos o matemáticos convencionales. Esto se debe a las distintas categorías, características y funciones que puedan encontrarse asociadas a cada elemento y/o a las relaciones heterogéneas que puedan tener entre sí.

Es por ello por lo que se utilizan representaciones simplificadas en forma grafos. En los mismos, se dan condiciones topológicas no triviales para este tipo de redes, es decir: en lugar de modelarse como gráficos de celosía, en donde se genera una cuadrícula con todos los nodos conectados entre sí de igual forma, o grafos aleatorios, en donde las conexiones se encuentran definidas por procesos aleatorios, las distintas conexiones están definidas por fenómenos del mundo real con características concretas. A su vez, dentro de ellas existen diferentes subestructuras con funciones y relaciones específicas, pudiendo darse cambios no lineales en una de las zonas de la red al reaccionar a los de otras regiones. [24]

Esta rama de la teoría de grafos tiene un origen reciente datado a partir de los años 2000. Está basada en el estudio de sucesos reales como las redes neuronales, las redes sociales o internet. A través de ella se puede modelar una red como un conjunto de «nodos» y «enlaces», siendo estos sus dos elementos principales:

- **Nodos:** son aquellos puntos en los que se puede procesar información o producirse algún suceso. Podrían representar neuronas, personas o servidores de internet, según los ejemplos anteriores. En este Proyecto representan a los distintos satélites de la constelación; cada nodo equivale a un satélite.
- **Enlaces:** representan las relaciones dadas entre los nodos. Sólo son considerados entre parejas de nodos; esto significa que cada enlace puede tener solamente una pareja de nodos

asociada. Podrían representar axones, la relación entre dos personas o las conexiones entre servidores, según los ejemplos anteriores. En este Proyecto se asume la aparición de un enlace entre dos satélites durante los instantes en los que tengan visibilidad directa.

Una vez definidos los elementos principales, pueden darse distintos tipos de redes según su estructura. En función del tipo de enlace entre nodos, las redes pueden ser [25]:

- **Binarias:** son aquellas redes en donde todos los enlaces se consideran iguales. Si existe un enlace entre dos nodos se representará mediante un 1 y si no, mediante un 0. En este Proyecto son las redes empleadas para modelar las conexiones durante los periodos más pequeños de tiempo. Por ejemplo: analizando la posición de dos satélites cada 5 s, si hubiese línea de visión entre ellos en el instante inicial, se le asociaría un 1 a ese enlace durante ese periodo; si en el siguiente periodo, en el segundo 5, la Tierra los tapase, se le asociaría un 0 al enlace durante ese último periodo.
- **Ponderadas:** son aquellas redes en las que se asocian características a cada enlace mediante el uso de un escalar. En este Proyecto se utilizan para modelar el tiempo de conexión entre satélites durante intervalos más grandes de tiempo. Son empleadas en el segundo método de análisis y asumen valores decimales entre 0 y 1. Por ejemplo: analizando la posición de dos satélites a lo largo de un intervalo de duración equivalente al periodo orbital, se determina que se encuentran visibles un 70 % del tiempo, por lo tanto, su conexión es representada mediante el valor 0,7. Podrían tener también otros usos, como por ejemplo, representar el ancho de banda entre satélites. A su vez, podrían llevar asociados valores superiores a 1. En el Proyecto no se muestran ejemplos de estas dos situaciones.

Una red ponderada podría representarse como una red binaria asumiendo un umbral inferior a partir del cual el enlace pasaría a considerarse como inexistente. En el caso inverso, una red binaria podría considerarse como ponderada en caso de multiplicar el valor de cada enlace por un escalar, siendo esto equivalente a multiplicar el valor de los enlaces de la red inicial por los de otra red ponderada. Si se sumasen los valores de los enlaces de varias redes binarias se obtendría una red ponderada. Mediante el último planteamiento se obtienen las redes ponderadas empleadas en el segundo método de análisis del Proyecto.

En función de si se tiene en cuenta o no la direccionalidad de los enlaces, las redes pueden ser de otros dos tipos [25]:

- **Dirigidas:** tienen en cuenta la direccionalidad de los enlaces. Podría existir un enlace entre un nodo i y otro j pero no a la inversa; esto representaría la transmisión exclusiva de información desde i hasta j . Aunque en este proyecto no se emplean, podrían utilizarse para representar transmisiones unidireccionales de información entre satélites
- **No dirigidas:** no tienen en cuenta la direccionalidad de los enlaces. Una red dirigida en donde todos los enlaces tuviesen su análogo en la dirección opuesta podría pasar a considerarse como no dirigida. Como en los análisis efectuados en el Proyecto no se conocen las transmisiones de información o son asumidas directamente, todas las redes consideradas son de este tipo.

Las redes pueden tener otras clasificaciones que, aunque no son empleadas a lo largo del Proyecto, pueden ser interesante de definir debido a las futuras posibilidades que puedan ofrecer en este campo [26] :

- **Aleatorias:** su uso es anterior al de la Teoría de Redes Complejas. Se asocia una probabilidad de aparición a los enlaces en función de una distribución binomial. Son útiles para procesos con incertidumbre. Conociendo las conexiones históricas entre satélites, se podrían asociar las probabilidades de enlace entre ellos para proyectar periodos futuros. Aun así, en este Proyecto se ha optado por obtener las posiciones exactas para cada instante de tiempo.
- **Sin escala:** son parecidas a las aleatorias pero con la probabilidad de formar nuevos enlaces siguiendo una «ley potencial» en cada nodo. Aquellos nodos con un mayor número de enlaces tienen una mayor probabilidad de conectarse a nuevos nodos que aparezcan con el tiempo en la red. Son útiles para modelar redes dinámicas. En relación a las constelaciones de satélites, aquellos con recursos infrutilizados podrían conectarse a más satélites gracias a su mayor capacidad de procesamiento disponible, generándose distribuciones de este tipo. Conocer los criterios de conexión evitaría la sobrecarga de los satélites.

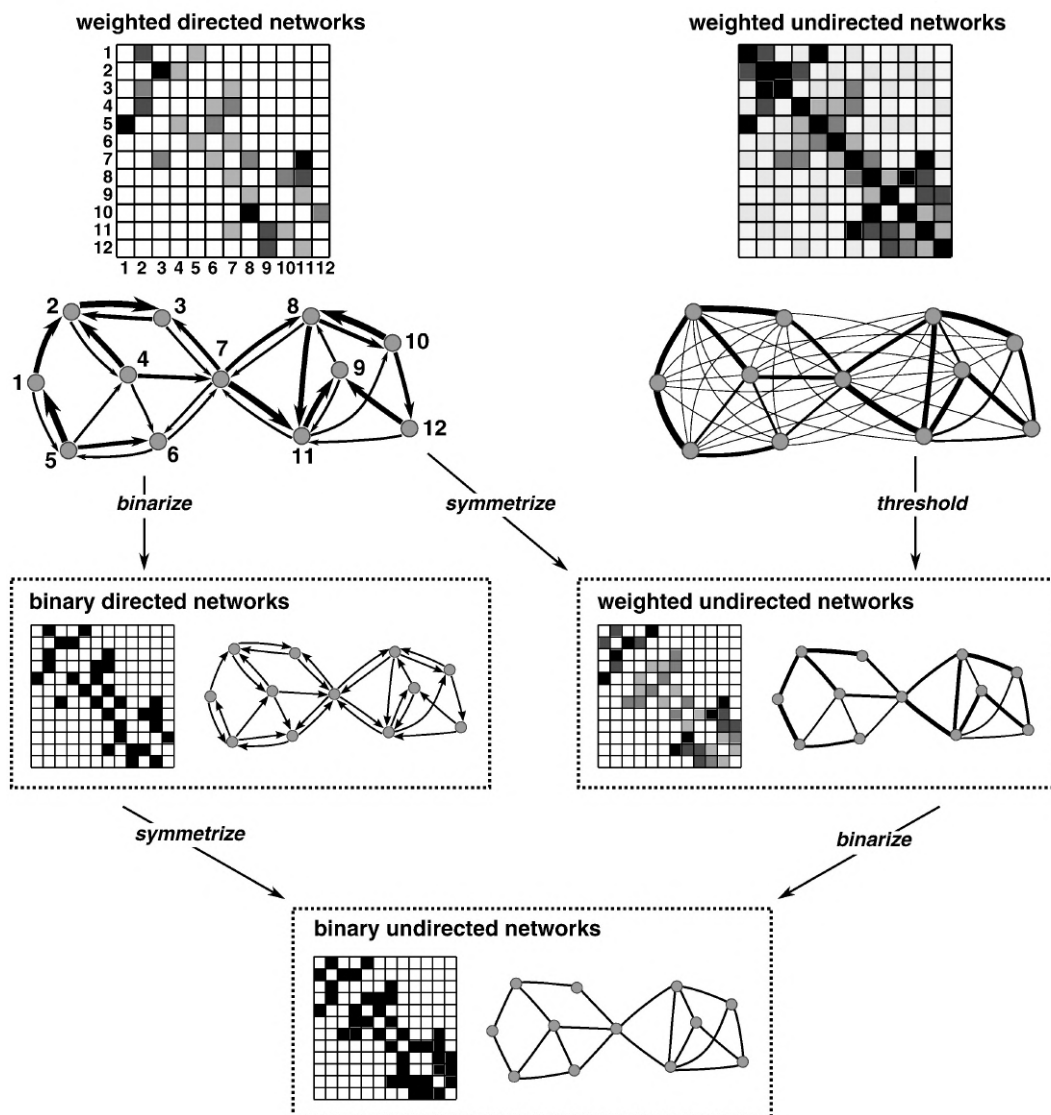


Figura 6. Principales tipos de red según las características de los enlaces [25]

Tras quedar definida la red, existen distintas métricas para analizarla. Para poder calcularlas es necesario representar primero la red como un grafo, pudiendo emplearse dos matrices:

- **Matriz de adyacencia:** es una matriz cuadrada en donde filas y columnas representan los distintos nodos. Mediante el uso de un 0 o un 1 se indica si existe un enlace entre ellos. Siendo n el número de nodos de la red:

$$[A]_{n \times n} = \begin{bmatrix} a_{ij} & \cdots & a_{in} \\ \vdots & \ddots & \vdots \\ a_{nj} & \cdots & a_{nn} \end{bmatrix} \mid a_{ij} = \begin{cases} 0 & \leftrightarrow i = j \cap \nexists \text{ enlace}_{ij} \\ 1 & \leftrightarrow \exists \text{ enlace}_{ij} \end{cases} \quad (1)$$

- **Matriz de pesos:** empleadas solamente en redes ponderadas. Representan las características de cada enlace mediante un escalar.

$$[W]_{n \times n} = \begin{bmatrix} w_{ij} & \cdots & w_{in} \\ \vdots & \ddots & \vdots \\ w_{nj} & \cdots & w_{nn} \end{bmatrix} \mid \begin{cases} w_{ij} = 0 & \leftrightarrow i = j \cap \nexists \text{ enlace}_{ij} \\ w_{ij} \neq 0 & \leftrightarrow \exists \text{ enlace}_{ij} \end{cases} \quad (2)$$

Una vez se dispone de la formulación matemática de los grafos, pueden calcularse las distintas métricas. Debido a su gran variedad, en esta sección solamente se muestran aquellas empleadas durante el Proyecto. [25]

1.4.1. Métricas básicas

La métrica más básica posible es el grado k_i de cada nodo. Indica el número de enlaces conectados a los mismos, en caso de tratarse una red binaria, o la suma de los pesos de los enlaces, en caso de tratarse de una red ponderada. Las dos medidas pueden ser normalizadas si se dividen entre el número total de nodos menos 1. Las redes ponderadas pueden emplear ambos tipos de métrica según la intencionalidad. Siendo N el conjunto de nodos de la red:

- **Grado de un nodo en una red binaria no dirigida:**

$$k_i = \sum_{j \in N} a_{ij} \quad (3)$$

- **Grado de un nodo en una red ponderada no dirigida:**

$$k_i^w = \sum_{j \in N} w_{ij} \quad (4)$$

La siguiente métrica a tener en cuenta es la distancia más corta entre dos nodos, también conocida como *shortest path length*. Esta indica cuántos enlaces se encuentran entre un nodo determinado y otro siguiendo el camino más corto. Para las redes ponderadas se utiliza una función $f(w_{u,v})$ de los pesos de cada enlace. En capítulos posteriores se explica el uso de la función elegida y los motivos tras ello.

- **Distancia o *shortest path length* entre dos nodos:** siendo $g_{i \leftrightarrow j}$ el camino más corto o «geodésica» entre los nodos i y j . Si tal camino entre nodos no existe porque se encuentren desconectados, entonces $d_{ij} = \infty$.

$$d_{ij} = \sum_{a_{uv} \in g_{i \leftrightarrow j}} a_{uv} \quad (5)$$

- **Distancia o *shortest path length* entre dos nodos en una red ponderada:** siendo $g_{i \leftrightarrow j}^w$ el camino más corto, teniendo en cuenta las ponderaciones, entre los nodos i y j . Si tal camino entre nodos no existe porque se encuentren desconectados, entonces $d_{ij}^w = \infty$.

$$d_{ij}^w = \sum_{w_{uv} \in g_{i \leftrightarrow j}^w} f(w_{uv}) \quad (6)$$

- **Matriz de distancias entre dos nodos:** las distancias entre nodos pueden codificarse en forma de matriz.

$$[D]_{n \times n} = \begin{bmatrix} d_{ij} & \cdots & d_{in} \\ \vdots & \ddots & \vdots \\ d_{nj} & \cdots & d_{nn} \end{bmatrix} \mid d_{ij} = \begin{cases} 0 & \leftrightarrow i = j \\ \infty & \leftrightarrow \nexists g_{i \leftrightarrow j} \end{cases} \quad (7)$$

La última de las métricas básicas es el número de triángulos que existen alrededor de un nodo. Si dos nodos se encuentran enlazados a otro nodo inicial y también entre sí se considera a esta estructura como un triángulo del nodo inicial. Para las redes ponderadas se tiene en cuenta la media geométrica de los pesos.

- **Número de triángulos alrededor de un nodo:**

$$t_i = \frac{1}{2} \sum_{j,h \in N} a_{ij} a_{ih} a_{jh} \quad (8)$$

- **Número de triángulos alrededor de un nodo en una red ponderada:**

$$t_i^w = \frac{1}{2} \sum_{j,h \in N} (w_{ij} w_{ih} w_{jh})^3 \quad (9)$$

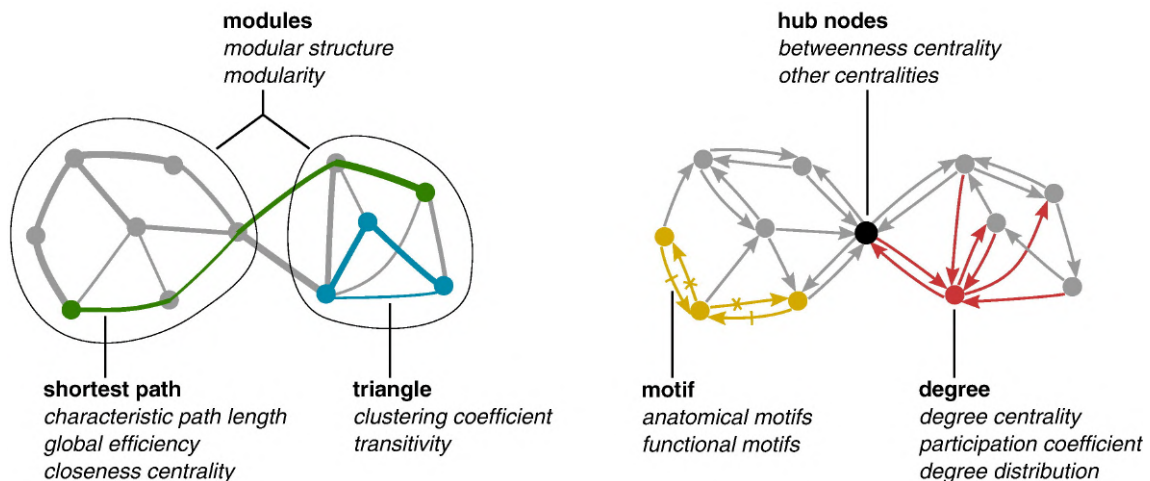


Figura 7. Principales métricas básicas en una red [25]

1.4.2. Métricas de integración

Las métricas de integración son aquellas relacionadas con los recorridos que tomaría la información para atravesar la red. Su métrica más básica es la distancia, a partir de la que se

obtienen el resto de métricas; la mayoría relacionadas con el conjunto de la red en lugar de con sus nodos individuales.

La longitud de trayectoria o *path length* de un nodo surge al promediar las distancias desde el mismo hasta el resto de nodos de la red. Mide y permite comparar el nivel de lejanía general de cada nodo a los demás. Para ello se emplea como denominador el número de nodos menos 1, ya que no se tiene en cuenta el nodo de origen. El proceso es el mismo para las redes ponderadas. Si un nodo de la red se encuentra desconectado, la longitud de trayectoria de todos los nodos será igual a infinito.

■ **Longitud de trayectoria o *path length* de un nodo:**

$$L_i = \sum_{j \in N; j \neq i} \frac{d_{ij}}{n-1} \quad (10)$$

■ **Longitud de trayectoria o *path length* de un nodo en una red ponderada:**

$$L_i^w = \sum_{j \in N; j \neq i} \frac{d_{ij}^w}{n-1} \quad (11)$$

La longitud de trayectoria característica o *characteristic path length* es el promedio de las longitudes de trayectoria de todos los nodos de la red. Es una medida global que permite medir el grado de lejanía global entre todos los nodos de la red en función del número de enlaces entre nodos. Una red de estructura alargada tendrá una longitud de trayectoria característica mayor que otra red más mallada de igual número de nodos. Si la red se divide en dos o más partes desconectadas entre sí su valor será infinito. Este efecto es útil en programación ya que permite detectar si se han producido desconexiones sin tener que comprobar las distancias asociadas a cada nodo una a una, ahorrando tiempo de computación.

■ **Longitud de trayectoria característica o *characteristic path length* de la red:**

$$L = \sum_{i \in N} \frac{L_i}{n} \quad (12)$$

■ **Longitud de trayectoria característica o *characteristic path length* de una red ponderada:**

$$L^w = \sum_{i \in N} \frac{L_i^w}{n} \quad (13)$$

Debido al efecto generado por la desconexión de nodos, en donde la longitud de trayectoria característica pasa a ser infinito, otra medida empleada es la eficiencia global. Esta se calcula mediante la eficiencia de cada nodo², siendo esta la suma de las inversas de las distancias de cada nodo entre el número de nodos menos 1.

■ **Eficiencia de un nodo:**

$$E_i = \sum_{j \in N; j \neq i} \frac{d_{ij}^{-1}}{n-1} \quad (14)$$

²No confundir con la inversa de la longitud de trayectoria L_i^{-1} ni con la eficiencia local del nodo $E_{loc,i}$, calculada en función del grado del nodo y de los nodos adyacentes. La última no se emplea en este Proyecto.

- **Eficiencia de un nodo en una red ponderada:**

$$E_i^w = \sum_{j \in N; j \neq i} \frac{(d_{ij}^w)^{-1}}{n-1} \quad (15)$$

De esta forma es posible ignorar el efecto generado por los nodos desconectados al transformar en 0 la inversa de infinito, pudiéndose sumar todos los valores para determinar y comparar el grado de cercanía entre los nodos de la red. Esta cercanía se traduce como eficiencia ya que, cuanto más alejados se encuentren los nodos entre sí, más se saturarán los enlaces al tener que transmitir mayores cantidades de información a más distancia.

- **Eficiencia global de la red:**

$$E = \sum_{i \in N} \frac{E_i}{n} \quad (16)$$

- **Eficiencia global de una red ponderada:**

$$E^w = \sum_{i \in N} \frac{E_i^w}{n} \quad (17)$$

Durante el Proyecto se evaluará el estado de red mediante el conjunto de las métricas de longitud de trayectoria característica y eficiencia. La primera permite detectar el momento en el que se ha producido una escisión en la red. Después será necesario analizar la matriz de distancias para localizar los distintos grupos formados. La segunda permite analizar de forma comparativa las distintas redes formadas tras la eliminación de un satélite, incluso cuando se han generado redes independientes. De esa forma se puede comprobar si el satélite inutilizado era el más crítico respecto a la transmisión de datos desde cualquier punto de la red. A menor eficiencia, mayor probabilidad de saturación de las redes, así como una mayor probabilidad de fallo al tener que involucrar a más nodos en la transmisión de información.

1.4.3. Métricas de segregación

Las métricas de segregación son aquellas relacionadas con la estructura de la red, centrándose especialmente en la aparición de grupos, módulos o *clusters*. Su métrica más básica es el número de triángulos alrededor de un nodo, a partir del que se obtienen el resto de métricas.

El coeficiente local de agrupamiento o *local clustering coefficient* mide la proporción de triángulos existentes entre el total de triángulos posibles alrededor de un nodo. Esto se traduce como el nivel de agrupamiento entre los vecinos a ese nodo.

- **Coeficiente local de agrupamiento o *local clustering coefficient* de un nodo:³**

$$C_i = \frac{2t_i}{k_i(k_i - 1)} \quad \forall k_i \geq 2 \quad (18)$$

- **Coeficiente local de agrupamiento o *local clustering coefficient* de un nodo en una red ponderada:**

$$C_i^w = \frac{2t_i^w}{k_i(k_i - 1)} \quad \forall k_i \geq 2 \quad (19)$$

³El 2 en el denominador indica los dos sentidos de cada enlace, ya que la redes aquí expuestas no son dirigidas.

Para aquellos casos en donde $k_i < 2$ la mayoría de autores asumen un valor igual a 0, aunque algunos emplean un valor de 1. No existe una respuesta correcta sobre este aspecto ya que, tanto para aquellos nodos con un vecino como para los nodos aislados, se dan indeterminaciones respecto al número de triángulos resueltas solamente mediante convencionalismos. Ahora bien, es necesario tener en cuenta el efecto que esto genera sobre el coeficiente global de agrupamiento.

El coeficiente global de agrupamiento o *global clustering coefficient* pondera los distintos coeficientes locales permitiendo medir el nivel de agrupamiento general entre los nodos de la red; coeficientes altos implican redes muy malladas. A su vez, un mayor coeficiente global de agrupamiento supone disponer de una red con más posibilidades de transmitir información, por lo tanto, más resistente a los fallos.

- **Coeficiente global de agrupamiento o *global clustering coefficient* de la red:**

$$C = \sum_{i \in N} \frac{C_i}{n} \quad (20)$$

- **Coeficiente global de agrupamiento o *global clustering coefficient* de una red ponderada:**

$$C^w = \sum_{i \in N} \frac{C_i^w}{n} \quad (21)$$

Volviendo al valor de los coeficientes locales de agrupamiento para aquellos nodos en donde el grado es menor a 2, si se asume un coeficiente local igual a 0, se asume una penalización al coeficiente global. Esto se traduciría como la representación de una red más propensa a presentar fallos debido al mayor número de nodos con menos de dos vecinos. Este convencionalismo no suele estar enfocado a comparar redes con el mismo origen en donde sus elementos van siendo eliminados, como es el caso de este Proyecto.

Por ello, una opción alternativa sería asociar a los nodos problemáticos el valor de la media de los coeficientes locales de los nodos con dos o más vecinos. De esta forma el coeficiente global no se vería afectado por dichos nodos y se podría seguir calculando en función del mismo número de nodos en el denominador. Como esto sería equivalente a calcular el coeficiente global omitiendo tales nodos, también podrían tomarse sus coeficiente locales como si no fuesen un número. Con ello, se estaría indicando, mediante esa métrica, que su grado es inferior a 2. Para el cálculo del coeficiente global, sería necesario emplear solamente el número de nodos de grado mayor o igual a 2.

La utilidad de un método u otro depende del número de métricas de la red dispuestas a calcularse y de la información deseada mediante cada una de ellas. Para ahorrar tiempo de computación, se asumirán los valores clásicos de coeficientes locales de agrupamiento.

1.4.4. Métricas de centralidad

Las métricas de centralidad son aquellas relacionadas con el nivel de importancia de nodos individuales dentro de la red. La medida más básica de centralidad sería el grado de un nodo. Sin embargo, esta métrica no es la más útil para encontrar el punto más crítico dentro de la red. El nodo con más enlaces no tiene por qué ser aquel por el que pasa más información. Si este fuera eliminado, podría no tener apenas efecto en caso de encontrarse la red suficientemente mallada, es decir, con un coeficiente global de agrupamiento alto.

El nodo más crítico de una red puede definirse de distintas formas equivalentes. Puede ser aquel a través del cual podría pasar la mayor cantidad de información incapaz de transmitirse mediante otros caminos; no confundir con el nodo atravesado por un mayor número de geodésicas. Esto equivale a los nodos que, al ser eliminados, perjudicarían más la métrica de eficiencia en el largo plazo, es decir, tras varias eliminaciones. La métrica empleada para definir a este tipo de nodos es la centralidad de intermediación o *betweenness centrality*. [27]

- **Centralidad de intermediación o *betweenness centrality* de un nodo:** siendo ρ_{hj} el número de geodésicas entre los nodos h y j y $\rho_{hj}(i)$ el número de geodésicas entre los nodos h y j que atraviesan el nodo i .⁴

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{h,j \in N; h \neq j; h \neq i; j \neq i} \frac{\rho_{hj}(i)}{\rho_{hj}} \quad (22)$$

- **Centralidad de intermediación o *betweenness centrality* de un nodo en una red ponderada:** siendo ρ_{hj}^w el número de geodésicas entre los nodos h y j y $\rho_{hj}^w(i)$ el número de geodésicas entre los nodos h y j que atraviesan el nodo i . Las geodésicas tiene que haber sido calculadas en función de la ponderación de los enlaces.

$$b_i^w = \frac{1}{(n-1)(n-2)} \sum_{h,j \in N; h \neq j; h \neq i; j \neq i} \frac{\rho_{hj}^w(i)}{\rho_{hj}^w} \quad (23)$$

Si se sigue el criterio de eliminación prioritaria de este tipo de nodos, es más probable fragmentar la red en distintos módulos independientes, aunque también existen otros métodos para ello cuando se dan ciertos tipos específicos de red. [28]

1.5. Constelación Globalstar

Una vez explicados los criterios con los que se ha de modelar la red, es necesario escoger un grupo de satélites para realizar los experimentos presentados en el Proyecto. Lo ideal sería seleccionar una o varias constelaciones capaces de generar Comunicaciones Inter-Satelitales. Es necesario que sus datos sean públicamente accesibles y actualizados con cierta frecuencia. Además, deberían disponer de un número de satélites suficientemente alto como para poder extraer conclusiones estadísticas; alrededor de 30 serían suficientes, según el «Teorema del Límite Central». Respecto al número máximo, no existe un límite, aunque cuanto más elevado sea, mayor será el tiempo de computación. Por este último motivo cualquiera de las megaconstelaciones quedaría descartada, aun así los métodos aquí expuestos podrían ser igualmente aplicables.

Uno de los candidatos considerados por cumplir con las especificaciones fue la constelación Iridium. Dicha constelación dispone de 66 satélites operativos, con 4 enlaces de comunicación por cada satélite [29]. Sin embargo, tras realizar los primeros experimentos se comprobó como el grado de complejidad ofrecido por la red era demasiado bajo, por lo que fue descartada. Al tratarse de una red lo suficientemente mallada, con un patrón de movimiento regular, el nivel de criticidad de todos los satélites era parecido, por lo que se obtenían resultados aleatorios con escaso interés.

⁴Es necesario tener en cuenta que, según esta formulación, cada geodésica se contaría dos veces, una para cada sentido.

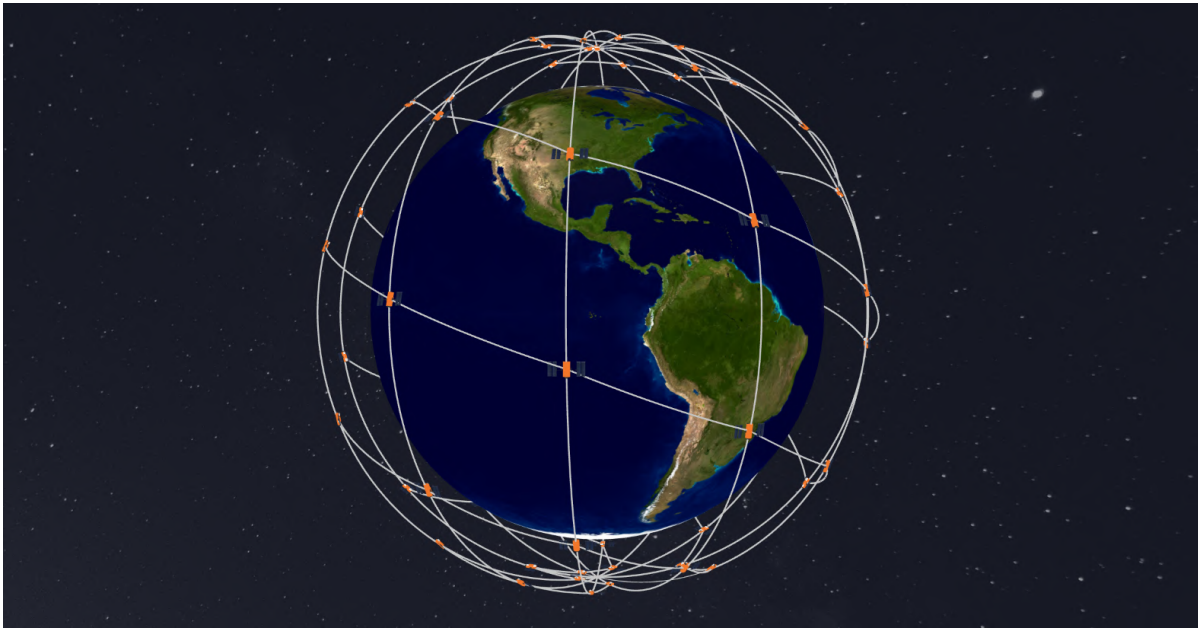


Figura 8. Esquema de conexión de la constelación Iridium [29]

Ya que actualmente no se encuentran suficientes constelaciones que cumplan dichas condiciones, se optó por emplear una constelación sin Comunicaciones Inter-Satelitales: Globalstar. La justificación tras esto reside en las múltiples posibilidades futuras del Internet de Satélites. En caso de producirse enlaces entre distintas constelaciones, gran parte de las conexiones se darían en momentos puntuales debido a la independencia entre la dirección de las órbitas; aun así estas podrían producirse con cierta periodicidad, ya que los satélites dispondrían de periodos parecidos al encontrarse a altitudes similares.



Figura 9. Satélite de la constelación Globalstar [30]

La constelación Globalstar cuenta con 32 satélites en órbita alrededor del ecuador [31]. A diferencia de Iridium, los satélites no se comunican entre sí, empleando en su lugar 24 estaciones terrestres conectadas a la red de comunicaciones tradicional [32]. La constelación es capaz de

recibir señales de más del 80 % de la superficie terrestre; ninguno de los satélites orbita sobre los polos. El periodo orbital de la mayoría de sus satélites es de 114,1 min, siendo algo menor en 6 de ellos y mayor en 1 [33]. Los motivos tras la elección de esta constelación son la suficiente antigüedad de la misma, la fiabilidad de los datos al respecto, el gran número de estudios realizados sobre su funcionamiento y la suficiente complejidad que presenta su movimiento.

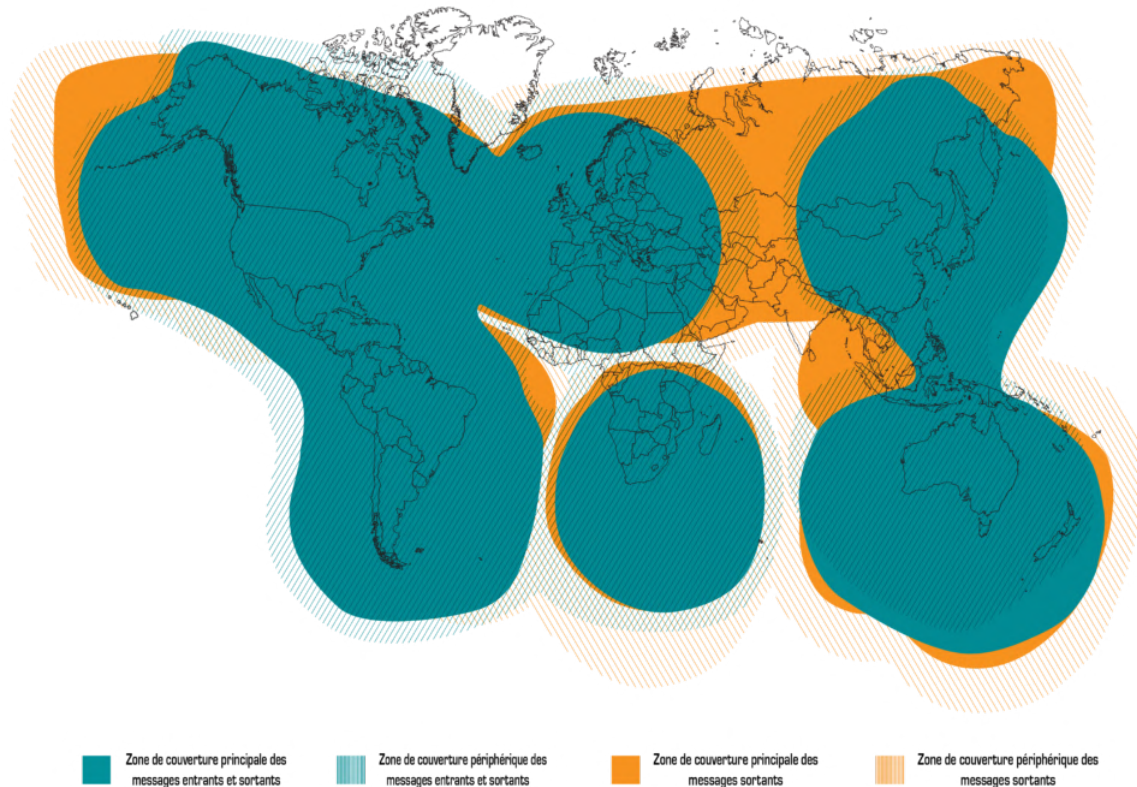


Figura 10. Mapa de cobertura de la constelación Globalstar [31]

Como conclusión, es necesario recalcar que el análisis de vulnerabilidades podrá realizarse con cualquier constelación en el futuro. Se ha escogido esta ya que es de las más fiables para extraer conclusiones en la actualidad, por todo lo expresado con anterioridad. Aun así, este Proyecto pretende demostrar cómo es posible realizar un análisis de vulnerabilidades de cualquier red en órbita de la que se pueda acceder a sus datos de posición en el tiempo. Todo ello con el objetivo de facilitar las conexiones entre satélites y así disponer de redes más eficientes en el futuro.

Capítulo 2

Estado tecnológico actual

EN este capítulo se describen brevemente las tecnologías mediante las cuales el desarrollo de este proyecto ha sido viable.

2.1. Bases de datos de satélites

El primer paso para poder realizar los análisis es la obtención de los identificadores y las posiciones de los satélites seleccionados. La forma más simple de representar esta información es mediante la codificación en formato de «Elementos de Dos Líneas» o *Two Line Element*; «TLE» según sus siglas en inglés. Para ello, existen diversas páginas de libre acceso en internet.

2.1.1. UCS Satellite Database: Identificador NORAD

Uno de los identificadores más empleados, para diferenciar los distintos cuerpos artificiales en órbita, son los identificadores NORAD. Estos están formados por un número de 5 cifras, asociado a cada objeto en función del orden de lanzamiento o descubrimiento; el número 1 fue la lanzadera de la sonda Sputnik 1. Debido a las limitaciones de observación, el tamaño mínimo para ser denominado con uno de estos números es 10 cm de diámetro, siendo la organización NORAD la que recoge la información en su catálogo de satélites «SATCAT». [34] [35]

Gracias a este identificador es fácil localizar la información relativa a los satélites escogidos; el mismo también permite rastrear basura espacial. Una vez se ha seleccionado la constelación a analizar, Globalstar en este caso, se han de buscar los identificadores asociados a cada uno de sus satélites. Para ello, la forma más simple es mediante la base de datos de la UCS.

Esta pertenece a la *Union of Concerned Scientists*, una organización sin ánimo de lucro fundada en 1969 por científicos y estudiantes del MIT [36]. Su base de datos permite, de forma sencilla, buscar los nombres de satélites activos y obtener en una lista sus identificadores NORAD, así como otros parámetros [21]. La información se actualiza cada cuatro meses; a su vez, se dispone de un documento en donde pueden encontrarse todos los cambios realizados hasta el momento [37]. En este Proyecto se ha empleado la información del mes de septiembre de 2021. [33]

2.1.2. Celestrack: *Two Line Element* (TLE)

El siguiente paso es obtener la información relativa a la posición de cada satélite a lo largo de un intervalo de tiempo. Como ya se ha indicado antes, la codificación de la misma se da en

Campo	Columnas	Contenido
1	01	Número de línea
2	03-07	Número de catálogo del satélite (identificador NORAD)
3	09-16	Inclinación (grados)
4	18-25	Longitud del nodo ascendente (grados)
5	27-33	Excentricidad (punto decimal asumido)
6	35-42	Argumento del periastro (grados)
7	44-51	Anomalía media (grados)
8	53-63	Movimiento medio (revoluciones por día)
9	64-68	Número de revolución en la época astronómica (revoluciones)
10	69	Suma de comprobación (módulo 10)

Tabla 3. Elementos de la línea 2 de un TLE [34] [38]

Los datos obtenidos mediante un TLE son los relativos al momento en el que fueron tomados. Como curiosidad, debido a las limitaciones de codificación de la época astronómica, así como el gran incremento de cuerpos en órbita, se están planteando nuevos modelos para representar estos parámetros [40]. En la siguiente imagen pueden observarse algunos de los parámetros orbitales referidos a las coordenadas del satélite.

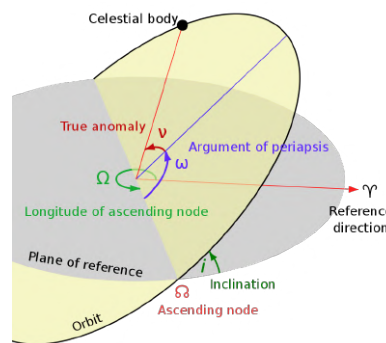


Figura 14. Parámetros orbitales [39]

Para conseguir los TLE de cada satélite es necesario conocer sus identificadores NORAD. Una vez se dispone de los mismos, se selecciona un intervalo de tiempo y se solicitan los archivos TLE de la constelación a la página web de CelesTrack. Dicha página, con 37 años de antigüedad, fue creada por el Dr. T.S. Kelso, el cual ha trabajado durante su carrera para la Fuerza Aérea de los Estados Unidos, así como otras organizaciones dedicadas a la operación espacial; actualmente sigue siendo el gestor de la página [41]. En ella pueden encontrarse, de forma pública y gratuita, los archivos TLE de numerosos cuerpos en órbita, así como otros recursos relacionados con la operación espacial. [42]

2.2. MATLAB

Una vez se dispone de los archivos TLE se puede comenzar a analizar los datos. El recurso empleado para ello ha sido MATLAB. MATLAB es una plataforma de programación y cálculo numérico con un lenguaje propio. Dicha plataforma permite la compatibilidad con otros programas; a su vez, cuenta con soporte y con una comunidad activa que ha generado diversas librerías de acceso público y gratuito. Se ha decidido emplear este recurso debido a los motivos mencionados anteriormente y al acceso a las licencias del que dispone la Universidad Pontificia Comillas. Se ha utilizado la versión R2021a. [43]

2.3. Algoritmo SGP4

Una de las limitaciones del formato TLE es la disponibilidad exclusiva de los datos para el instante en el que fueron tomados, indicado por la época astronómica. Por este motivo es necesario emplear modelos de perturbaciones simplificadas, los cuales permiten obtener la posición y la velocidad de cada satélite para cualquier instante de tiempo en función de los datos del TLE. Existen diversos modelos; el escogido en este Proyecto es el SGP4.

Las siglas «SPG» provienen de *Simplified General Perturbation*. El modelo SGP4 permite extrapolar los parámetros de orbitación de un satélite en el presente para obtener su posición y velocidad, tanto en el futuro como en el pasado. Este modelo no tiene en cuenta las perturbaciones gravitacionales que puedan producir otros cuerpos sobre el satélite, por lo que su precisión es de 1 km en la época astronómica y aumenta a un ritmo de entre 1 y 3 km por día; el modelo tampoco tiene en cuenta los movimientos con origen humano o las colisiones. [44]

Debido a su buena relación entre precisión y simplicidad es adecuado para las órbitas bajas con periodos inferiores a 225 min. A su vez, gracias a la abundancia de datos disponibles sobre la constelación Globalstar, de la que puede obtenerse más de un TLE al día por cada satélite, se puede considerar la ausencia de inconvenientes respecto a la precisión.

Para el uso de dicho modelo existen diversas librerías, programadas en MATLAB, con el algoritmo; la mayoría de reciente creación. La librería empleada fue la versión 1.1.1.0 de SPG4 de Maysam Mahooti [45], escogida por su simplicidad, su compatibilidad con cualquier versión de MATLAB y sus reseñas positivas.

2.4. Teoría de Redes Complejas

La Teoría de Redes Complejas es el elemento central de este Proyecto. Como ya se ha indicado en la sección 1.4, dicha teoría es de estudio reciente y actualmente continúan dándose avances en la materia.

2.4.1. Teoría de Redes Complejas Multicapa

Una de estas áreas de avance son las denominadas «Redes Multicapa». Su fundamento se basa en la agregación de varios grafos distintos, constituyendo cada uno de ellos una capa de la red. Las conexiones entre los nodos de una capa y los de las otras pueden ser representadas mediante tensores en lugar de matrices.

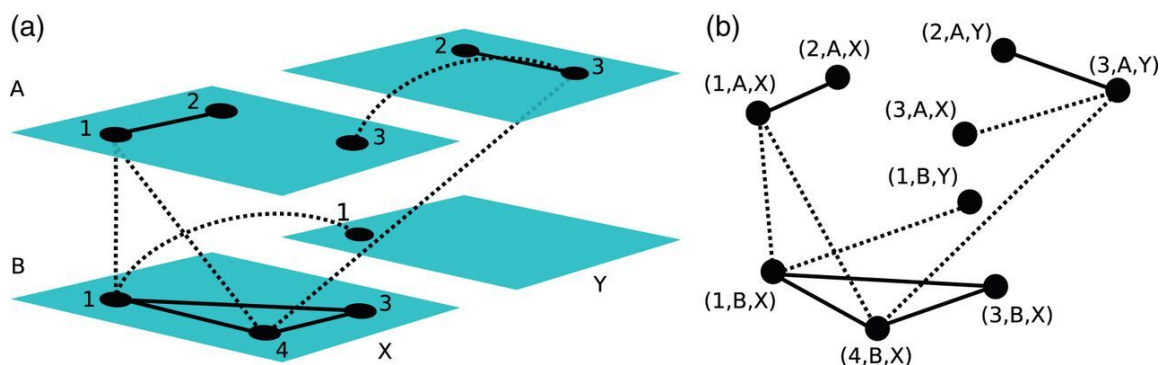


Figura 15. Representación general de un red multicapa [46]

La evolución en el tiempo de las conexiones entre los distintos satélites podría considerarse como un tipo concreto de estas redes, denominado como «Redes Temporales», en donde cada grafo asociado a un instante temporal constituiría una capa. Una de las formas de representarlas sería mediante enlaces unidireccionales, dirigidos desde los instantes inmediatamente anteriores a cada instante hasta el mismo satélite en el instante inmediatamente posterior. En pos de la simplicidad, no se asumen este tipo de conexiones en el Proyecto, ponderando las métricas de cada red de forma conjunta durante el primer método de análisis de vulnerabilidades. [47]

Otra forma posible de representar este tipo de redes se basa en agregar todas las capas de la red para pasar de una red «multiplex» a una red «monoplex». En ella los enlaces serían de tipo ponderado. Este método se corresponde con el segundo método, empleado en el Proyecto, para el análisis de vulnerabilidades. [46]

2.4.2. Funciones para el cálculo de métricas

Con el objetivo de facilitar el cálculo de los coeficientes locales de agrupamiento, tanto para redes binarias como ponderadas, mostrados en las ecuaciones (18) y (19) de la subsección 1.4.3 Métricas de segregación, se han empleado las funciones correspondientes de la librería *Brain Connectivity Toolbox* [48].

Dicha librería, programada en MATLAB, está principalmente orientada a realizar estudios neurológicos en donde se modelen redes complejas, aunque se ha empleado en multitud de proyectos de todo tipo desde su aparición. La misma está basada en el trabajo previo realizado por Mikail Rubinov y Olaf Sporns [25], autores a su vez de la librería.

Las funciones a utilizar son las denominada como «clustering_coef_bu.m» y «clustering_coef_wu.m», respectivamente. Se ha decidido hacer uso de las mismas ya que MATLAB no dispone de ninguna función diseñada específicamente para esas tareas, siendo un cálculo complejo de efectuar de la forma más eficiente posible.

2.5. Análisis de colisiones en órbita

Una vez se conocen los satélites críticos de la constelación para un intervalo de tiempo determinado, existen distintos recursos para determinar las posibles colisiones que puedan sufrir. Mediante los mismos, podrían automatizarse distintos procesos para asociar niveles de riesgo adicional a cada satélite.

2.5.1. SOCRATES

«SOCRATES» es un acrónimo de *Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space*. Este es un servicio gratuito, ofrecido por CelesTrack, capaz de calcular futuras colisiones entre cuerpos en órbita de los que se pueda disponer de un TLE, es decir, con un diámetro mayor a los 10 cm. La plataforma cuenta con distintas opciones para la búsqueda de las mismas, así como con un servicio de alertas gracias al que se podría asociar un aviso automático sobre el riesgo potencial al que estarían expuestos los satélites más vulnerables de la constelación. [49]

Action	NORAD Catalog Number	Name	Days Since Epoch	Max Probability	Dilution Threshold (km)	Min Range (km)	Relative Velocity (km/sec)
				Start (UTC)	TCA (UTC)	Stop (UTC)	
TLE Data	37214	FENGYUN 3B [+]	3.201	4.880E-02	0.002	0.009	5.459
	35189	FENGYUN 1C DEB [-]	16.327	2022 Jun 07 07:44:18.457	2022 Jun 07 07:44:19.373	2022 Jun 07 07:44:20.289	
TLE Data	45374	STARLINK-1300 [+]	6.189	4.162E-02	0.002	0.010	4.646
	47741	STARLINK-2154 [+]	6.184	2022 Jun 09 20:17:18.338	2022 Jun 09 20:17:19.414	2022 Jun 09 20:17:20.490	
TLE Data	46068	STARLINK-1532 [+]	6.118	1.224E-02	0.005	0.018	4.646
	47668	STARLINK-2059 [+]	6.388	2022 Jun 10 06:19:09.903	2022 Jun 10 06:19:10.979	2022 Jun 10 06:19:12.055	
TLE Data	50470	ONEWEB-0390 [+]	4.133	4.514E-03	0.010	0.022	15.555
	38584	BREEZE-M DEB [-]	6.533	2022 Jun 08 02:28:47.044	2022 Jun 08 02:28:47.365	2022 Jun 08 02:28:47.687	
TLE Data	43100	SUPERVIEW-1 04 [+]	3.800	5.295E-03	0.008	0.023	12.303
	44402	LEMUR-2-WANLI [+]	3.804	2022 Jun 07 22:31:13.063	2022 Jun 07 22:31:13.469	2022 Jun 07 22:31:13.876	
TLE Data	52691	STARLINK-3945 [+]	4.693	6.130E-03	0.006	0.027	0.001
	52696	STARLINK-4025 [+]	4.693	2022 Jun 08 07:52:11.431	2022 Jun 08 11:32:28.005	2022 Jun 08 17:35:05.294	
TLE Data	45044	STARLINK-1132 [+]	4.725	5.270E-03	0.007	0.028	4.645
	46679	STARLINK-1773 [+]	4.531	2022 Jun 08 06:31:40.105	2022 Jun 08 06:31:41.181	2022 Jun 08 06:31:42.257	
TLE Data	45767	STARLINK-1511 [+]	4.000	3.913E-03	0.008	0.033	4.645
	47829	STARLINK-2416 [+]	3.663	2022 Jun 07 19:52:42.794	2022 Jun 07 19:52:43.870	2022 Jun 07 19:52:44.947	
TLE Data	31698	TERRASAR-X [+]	1.621	3.573E-03	0.008	0.035	0.001
	36605	TANDEM-X [+]	1.554	2022 Jun 04 00:00:00.000	2022 Jun 05 16:29:40.582	2022 Jun 08 21:04:14.860	
TLE Data	42780	LEMUR-2-LISASAURUS [+]	0.409	3.511E-03	0.008	0.035	0.029
	42789	SKCUBE [+]	1.128	2022 Jun 04 11:55:49.668	2022 Jun 04 11:58:43.813	2022 Jun 04 12:01:37.957	

Figura 16. 10 colisiones más probables según SOCRATES (04/06/2022-11/06/2022) [49]

2.5.2. MASTER

«MASTER» es un acrónimo de *Meteoroid and Space Debris Terrestrial Environment Reference*. Este es un programa gratuito ofrecido por la ESA como parte de la plataforma *Space Debris User Portal*. Su objetivo es calcular la probabilidad de impacto con basura espacial para órbitas, puntos en el espacio o distribuciones de área. El programa es capaz de calcular probabilidades de impacto con cuerpos de hasta 1 μm gracias a los modelos estadísticos y las fuentes que emplea. Uno de sus usos potenciales podría ser calcular el nivel relativo de desgaste de cada satélite de la red debido a pequeños impactos. [50]

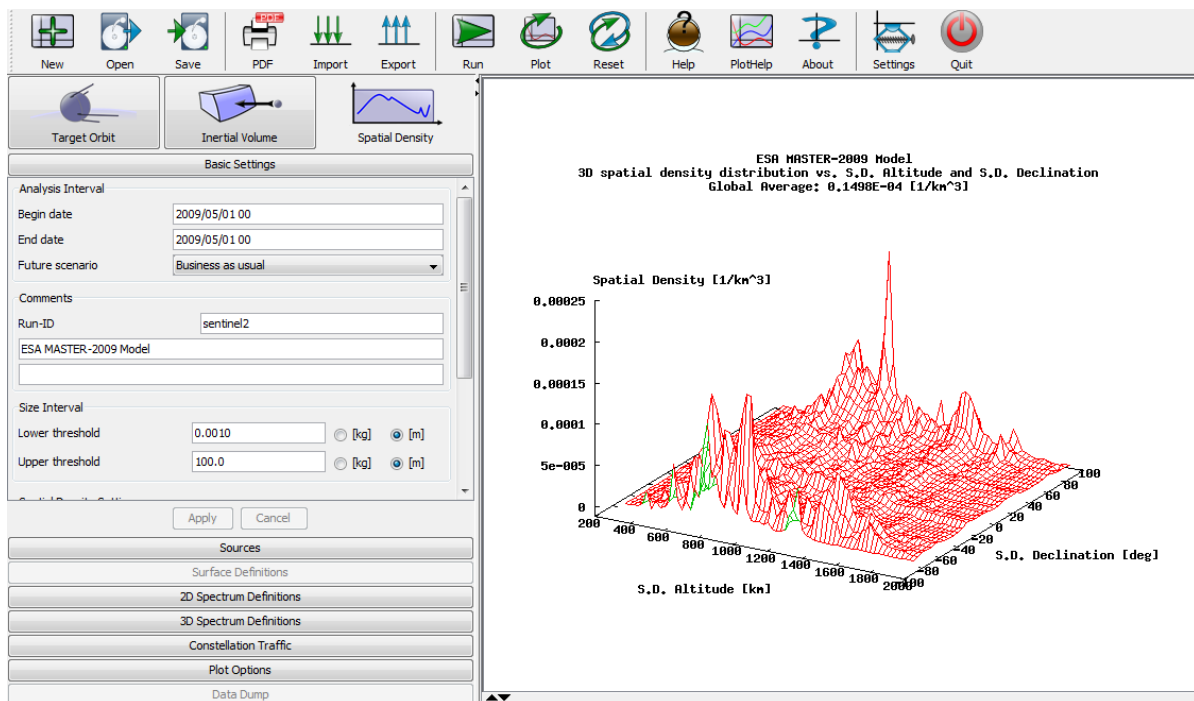


Figura 17. Interfaz de uso de MASTER [51]

Capítulo 3

Modelo desarrollado

EN este capítulo se explican todos los aspectos técnicos relevantes del proyecto, así como los pasos seguidos en las diferentes fases del mismo.

3.1. Obtención de las posiciones de los satélites

Para poder realizar los análisis es necesario disponer de un programa capaz de ubicar los satélites en el espacio a lo largo de un intervalo de tiempo. Dicha posición queda representada por los ejes cartesianos no inerciales x' , y' y z' , colineales a la rotación de la Tierra, encontrándose x' e y' en el mismo plano que el ecuador y siendo x' perpendicular a una recta tangente al meridiano de Greenwich⁵; el origen de coordenadas se encuentra en el centro de masas de la Tierra⁶. [52]

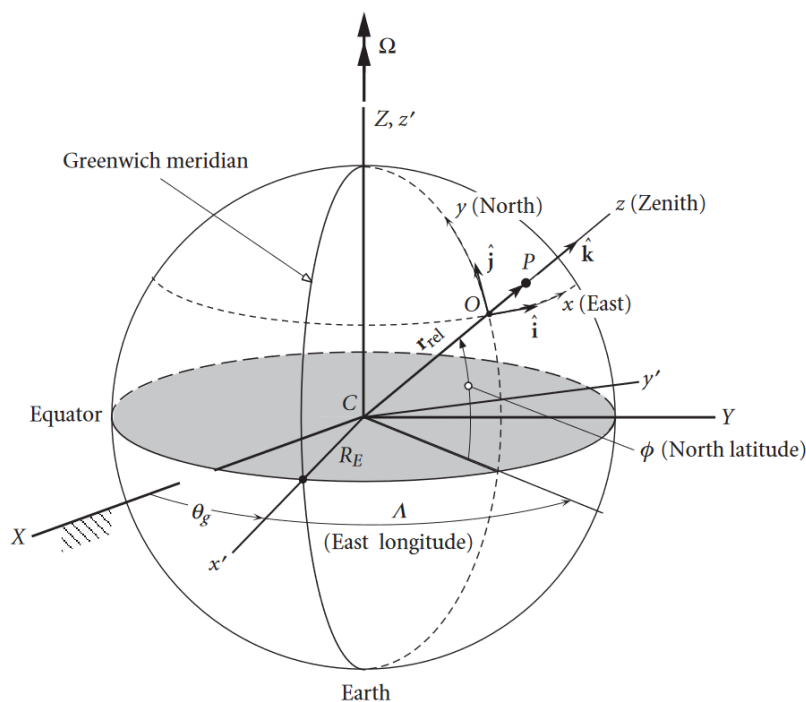


Figura 18. Ejes terrestres [52]

⁵A partir de este punto los ejes x' , y' y z' serán denominados, durante el resto del Proyecto, como ejes x , y y z , en pos de la simplicidad.

⁶Como el sistema de coordenadas representa la posición relativa entre la Tierra y cada satélite, realmente el origen de coordenadas se encontraría en el centro de masas formado por ambos cuerpos, pero al ser $m_{Tierra} \gg m_{Sat}$ se asume el origen en el de la Tierra. También se asume la ausencia de efectos gravitacionales de otros cuerpos.

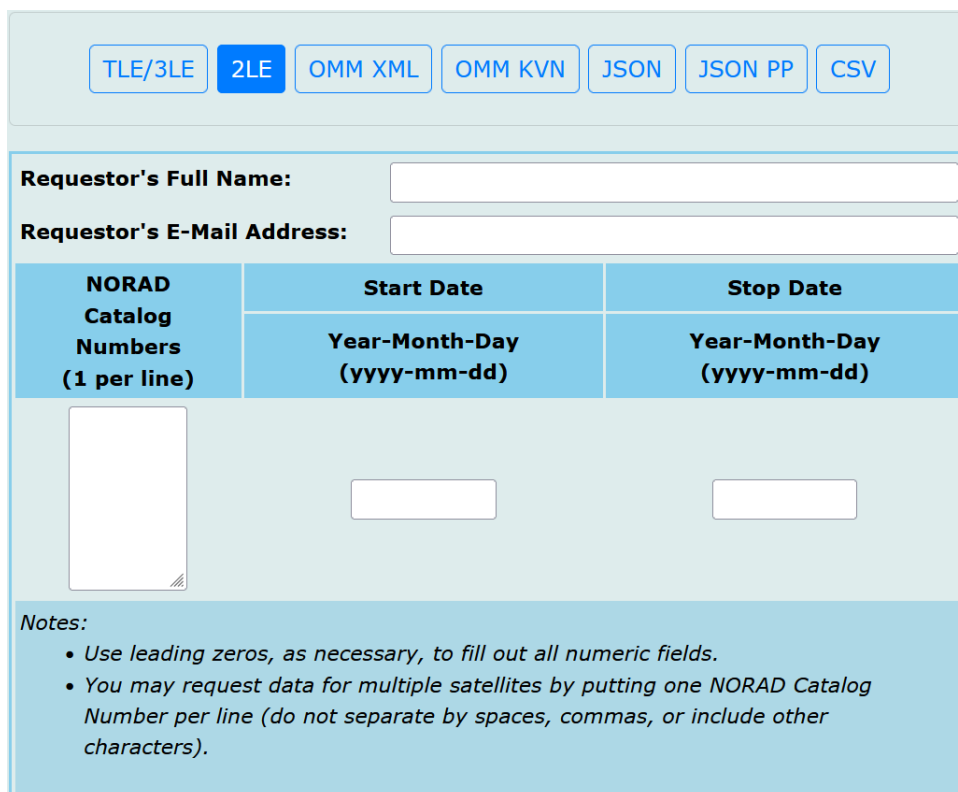
3.1.1. Toma de datos

El primer paso es obtener los datos de la constelación a representar. Para ello, se ha acudido a la base de datos de satélites de la UCS a partir de septiembre de 2021 [33], se han seleccionado los satélites de la constelación Globalstar (columna A) y se han copiado sus identificadores NORAD (columna AA). Los identificadores de la constelación son:

Nombre del satélite y nombres alternativos	Identificador NORAD
Globalstar M037 (Globalstar 16)	25652
Globalstar M065 (Globalstar 65)	31571
Globalstar M066 (Globalstar 66)	32265
Globalstar M067 (Globalstar 67)	32263
Globalstar M069 (Globalstar 69)	31573
Globalstar M070 (Globalstar 70)	32264
Globalstar M071 (Globalstar 71)	31576
Globalstar M072 (Globalstar 72)	31574
Globalstar M073 (Globalstar 73, Globalstar 2-6)	37193
Globalstar M074 (Globalstar 74, Globalstar 2-2)	37189
Globalstar M075 (Globalstar 75, Globalstar 2-5)	37192
Globalstar M076 (Globalstar 76, Globalstar 2-3)	37190
Globalstar M077 (Globalstar 77, Globalstar 2-4)	37191
Globalstar M078 (Globalstar 95, Globalstar 2-23)	39076
Globalstar M079 (Globalstar 79, Globalstar 2-1)	37188
Globalstar M080 (Globalstar 80, Globalstar 2-14)	38041
Globalstar M081 (Globalstar 81, Globalstar 2-11)	37743
Globalstar M082 (Globalstar 82, Globalstar 2-15)	38042
Globalstar M083 (Globalstar 83, Globalstar 2-7)	37739
Globalstar M084 (Globalstar 84, Globalstar 2-13)	38040
Globalstar M085 (Globalstar 85, Globalstar 2-10)	37742
Globalstar M086 (Globalstar 86, Globalstar 2-18)	38045
Globalstar M088 (Globalstar 88, Globalstar 2-8)	37740
Globalstar M089 (Globalstar 89, Globalstar 2-12)	37744
Globalstar M090 (Globalstar 90, Globalstar 2-17)	38044
Globalstar M091 (Globalstar 91, Globalstar 2-9)	37741
Globalstar M092 (Globalstar 92, Globalstar 2-16)	38043
Globalstar M093 (Globalstar 87, Globalstar 2-20)	39073
Globalstar M094 (Globalstar 93, Globalstar 2-21)	39074
Globalstar M095 (Globalstar 96, Globalstar 2-24)	39077
Globalstar M096 (Globalstar 94, Globalstar 2-22)	39075
Globalstar M097 (Globalstar 78, Globalstar 2-19)	39072

Tabla 4. Identificadores NORAD de los satélites de la constelación Globalstar a 01/09/2021 [33]

Una vez se dispone de los identificadores en forma de columna se acude a la web CelesTrack, concretamente a la sección de solicitud de datos especiales [53]. Es necesario seleccionar los datos en formato TLE o «2LE», según la página. En este caso la fecha de inicio ha sido 2021-09-01 y la del final, 2021-09-31. Los datos serán enviados a la dirección de correo especificada en el formulario, en forma de archivos .txt, obteniéndose un archivo distinto por cada indentificador NORAD.



TLE/3LE 2LE OMM XML OMM KVN JSON JSON PP CSV

Requestor's Full Name:

Requestor's E-Mail Address:

NORAD Catalog Numbers (1 per line)	Start Date	Stop Date
	Year-Month-Day (yyyy-mm-dd)	Year-Month-Day (yyyy-mm-dd)
<input type="text"/>	<input type="text"/>	<input type="text"/>

Notes:

- Use leading zeros, as necessary, to fill out all numeric fields.
- You may request data for multiple satellites by putting one NORAD Catalog Number per line (do not separate by spaces, commas, or include other characters).

Figura 19. Formulario de solicitud de datos especiales de CelesTrack [53]

Para facilitar la organización, los archivos .txt se guardarán en una carpeta llamada «SatelliteTLE». La lista de identificadores NORAD se guardará en un archivo .txt llamado «Globalstar.txt», usando el mismo formato que el empleado al introducirla en el formulario de CelesTrack.

3.1.2. Recursos empleados

Tras conseguir los datos de la constelación a analizar, se vuelve necesaria la utilización de funciones para transformar los mismos en coordenadas sobre ejes cartesianos en función del tiempo. Se ha empleado el modelo de perturbaciones simplificadas SPG4, codificado en las funciones de la librería SPG4 de Maysam Mahooti [45]. Una vez descargadas se almacenan en una carpeta llamada «SPG4», facilitando la organización.

Las funciones a emplear serán dos: «spg4.m» y «TLE_Satdata.m» (PARTE III, sección 1.1), obtenida tras modificar el programa «test_spg4.m» del paquete.

La función «spg4.m» permite aplicar el modelo de perturbaciones simplificadas, requiriendo dos entradas: el tiempo en minutos para el que se desea obtener las coordenadas, a partir del instante en el que fue tomado el TLE, y los parámetros orbitales contenidos en el TLE. Las salidas son dos vectores de tres elementos cada uno, con la posición y la velocidad del satélite, referenciados para los ejes x , y y z .

La función «TLE_Satdata.m» permite transformar los datos de cada línea de un TLE en parámetros orbitales, siendo las entradas la primera y la segunda línea del mismo. La salida de la función es una estructura en donde se muestran dichos parámetros de forma que la función «spg4.m» sea capaz de leerlos, siendo esta estructura la segunda entrada de dicha función.

Otro elemento a considerar para el funcionamiento del programa desarrollado es la «Parallel Computing Toolbox» de MATLAB [54]. Si se desea que el programa funcione a una mayor velocidad, será necesario instalarla para el uso de bucles «parfor». El funcionamiento de estos bucles es el mismo que el de un bucle «for», a excepción de la utilización de distintos núcleos del ordenador en paralelo. Su velocidad de ejecución será un número de veces superior equivalente al número de núcleo del que disponga el sistema. Aun así, es necesario que el tiempo de ejecución del interior de los bucles sea superior al tiempo de partición de los datos para que compense su uso; es decir, es recomendable usarlos sólo cuando se de un número suficiente de operaciones complejas en su interior. El principal inconveniente de tales bucles es la imposibilidad de usar, dentro de ellos, datos provenientes de distintas iteraciones. Si no se dispone de dicha herramienta bastará con sustituir los bucles «parfor» por bucles «for».

3.1.3. Algoritmo empleado

El programa creado con anterioridad por el director de este Proyecto, Enrique Sevillano Bravo, obtenía la posición de los satélites mediante funciones programadas en JavaScript. Debido a la limitación de las fuentes, de las que se obtenía solamente la latitud y longitud de cada satélite, la altitud tenía que darse como una aproximación, en donde a todos los satélites de la constelación se les asignaba la misma. Este programa permitía una mayor inmediatez, rapidez y simplicidad a cambio de precisión.

Como en este Proyecto se pretende poder evaluar las posiciones de los satélites de distintas constelaciones de cara al futuro, es necesario poder representar los distintos planos de altitud de cada uno, pudiendo asumir un mayor grado de complejidad. A su vez, se pretende ofrecer todo en un único lenguaje de programación, facilitando su comprensión a aquellas personas que no dispongan de conocimientos sobre otros lenguajes.

Al no haber encontrado un programa capaz de representar la posición de todos los satélites de una constelación a lo largo del tiempo, ha sido necesario crear uno desde cero a partir de los recursos presentados con anterioridad.

La organización general del programa está basada en el uso de estructuras. Las principales ventajas son la claridad con la que pueden visualizarse los datos, la posibilidad de agrupar datos en distinto formato y la facilidad para añadir nuevos apartados en el código mediante el uso de bucles «for».

3.1.3.1. Inicio y selección de los parámetros de funcionamiento

Una vez instalada la «Parallel Computing Toolbox» y disponiendo de la lista de identificadores NORAD «Globalstar.txt», las carpetas «SatelliteTLE» y «SPG4», con sus respectivas funciones, y la nueva función «TLE_Satdata.m», se introducen todos los archivos en la carpeta que contenga al programa principal. Es necesario seleccionar las carpetas y añadir sus elementos al directorio principal mediante la acción «Add to Path».

Tras eliminar las variables anteriores en el espacio de trabajo y comenzar a contar el tiempo de ejecución mediante la función «tic», se definen los parámetros de funcionamiento del programa. En primer lugar, se han de almacenar, en un vector, los identificadores de los satélites de la constelación, para ello se introduce el nombre del archivo .txt correspondiente en una variable «list_title». Tras esto, se indican las fechas de inicio y final del intervalo para el que se quieren calcular las posiciones. Finalmente, se indica en otra variable «period» el periodo de muestreo.

Los identificadores NORAD se ordenan de forma ascendente para facilitar su posterior lectura. También se cuenta el número de satélites de la lista; para que el programa pueda funcionar en versiones más antiguas de MATLAB se emplea la función «size» en lugar de «height».

```
% List of wanted satellites [1]
list = fopen(list_title); % Open text file
NORAD_Cloud_Struct = textscan(list, '%s'); % Text scan data
fclose(list); % Close file
n_Sat_Cloud = size(NORAD_Cloud_Struct{1,1},1); % Number of satellites wanted
NORAD_Cloud = sort(NORAD_Cloud_Struct{1,1}); % Sort NORAD numbers of wanted satellites
```

Código 1. Obtención de las posiciones de los satélites: almacenamiento de la lista de identificadores NORAD (PARTE III, sección 1.2, líneas 44-49)

Después, se ha de seleccionar el intervalo de fechas para el que quiere realizarse el análisis. Es posible introducir desde el número de segundos hasta el año. Si se introduce un número superior a los especificados en los comentarios, el programa funcionará con normalidad, pero será más difícil su interpretación. En caso de ser la fecha de inicio posterior a la de final, el programa se detendrá y mostrará un mensaje de error. Es necesario tener en cuenta la existencia de TLE solamente hasta el año 2056, debido a sus limitaciones de codificación.

```
% Date interval selection
% Initial [2]
Date_Cloud.Epoch_Day_0 = round(datenum(Date_Cloud.y_0,Date_Cloud.mo_0,Date_Cloud.d_0,
    Date_Cloud.h_0,Date_Cloud.m_0,Date_Cloud.s_0) - datenum(Date_Cloud.y_0,0,0,0,0),8); %
    Day of the year epoch
Date_Cloud.Full_Epoch_0 = Date_Cloud.y_0*1000 + Date_Cloud.Epoch_Day_0;
Date_Cloud.Date_0 = datetime(Date_Cloud.y_0,Date_Cloud.mo_0,Date_Cloud.d_0,Date_Cloud.h_0,
    Date_Cloud.m_0,Date_Cloud.s_0);
% Final [3]
Date_Cloud.Epoch_Day_f = round(datenum(Date_Cloud.y_f,Date_Cloud.mo_f,Date_Cloud.d_f,
    Date_Cloud.h_f,Date_Cloud.m_f,Date_Cloud.s_f) - datenum(Date_Cloud.y_f,0,0,0,0),8); %
    Day of the year epoch
Date_Cloud.Full_Epoch_f = Date_Cloud.y_f*1000 + Date_Cloud.Epoch_Day_f;
Date_Cloud.Date_f = datetime(Date_Cloud.y_f,Date_Cloud.mo_f,Date_Cloud.d_f,Date_Cloud.h_f,
    Date_Cloud.m_f,Date_Cloud.s_f);
% Time interval
Date_Cloud.t_max = split(between(Date_Cloud.Date_0,Date_Cloud.Date_f,'Time'),'Time');

% Date comprobation
if Date_Cloud.t_max <= 0
    disp('Incorrect date interval')
    pause
end
```

Código 2. Obtención de las posiciones de los satélites: selección de fechas de análisis (PARTE III, sección 1.2, líneas 51-67)

El último parámetro de funcionamiento es el periodo de muestreo para el que se desea obtener las posiciones. Gracias a este, pueden representarse las posiciones para los mismos instantes de tiempo, con independencia de cuando haya sido tomado el TLE. Con ello se genera un vector columna de fechas, en forma de malla regular, al que irán asociados el resto de los parámetros de cada satélite.

```

% Time period between positions [minutes] [4]
T_Cloud_minutes = minutes(period); % [minutes]

% Date grid array
d_grid_Cloud = Date_Cloud.Date_0:T_Cloud_minutes:Date_Cloud.Date_f;
d_grid_Cloud = d_grid_Cloud';
n_d_grid_Cloud = size(d_grid_Cloud,1);

```

Código 3. Obtención de las posiciones de los satélites: selección del periodo de muestreo (PARTE III, sección 1.2, líneas 69-75)

3.1.3.2. Capa 1 de la estructura de almacenamiento (Sat_Cloud)

A partir de este punto, la información se almacena dentro de estructuras anidadas; la estructura principal se ha denominado como «Sat_Cloud». Esta se encuentra formada por varias capas, con distintos campos en cada una. El número de fila hace referencia al orden ascendente de los identificadores NORAD de los satélites, es decir, cada fila de esta capa almacena los datos de un satélite. Los campos de esta estructura son:

- **NORAD:** indica el identificador NORAD del satélite asociado a cada línea.
- **txt_Names:** indica el nombre del archivo .txt en donde están contenidos los distintos TLE de cada satélite.
- **TLE:** almacena los TLE de cada satélite en un vector columna de celdas. Cada celda del vector se corresponde con una fila de un TLE.
- **TLE_Data:** contiene distintos datos de posición del satélite dentro de una estructura. Se encuentran asociados a los tiempos en los que fueron tomados los TLE, siempre que se encuentren dentro del intervalo de tiempo establecido para la toma de datos.
- **Position:** contienen los datos de posición del satélite, para cada instante de tiempo del intervalo, dentro de una estructura.

Fields	NORAD	txt_Names	TLE	TLE_Data	Position
1	25652	'sat25652.txt'	154x1 cell	1x2 struct	1x241 struct

Figura 20. Fila 1 de la estructura Sat_Cloud

Los distintos campos van siendo añadidos mediante distintos bucles «for», obteniendo un mayor grado de claridad dentro del código. Otro motivo para ir añadiendo los campos de forma individual es el menor tiempo necesario de computación; al estar generando vectores de tamaño creciente con cada iteración del bucle, MATLAB necesita realizar una copia del vector cada vez que un nuevo dato es añadido. Por este motivo, se crea uno solo al inicio con los identificadores NORAD, a los que se irán asociando el resto de parámetros sin necesidad de ejecutar el proceso anteriormente descrito.

```

% Generate NORAD names column
for i=1:n_Sat_Cloud
    Sat_Cloud(i).NORAD = str2double(NORAD_Cloud(i,1)); % NORAD names column (double)
end

```

Código 4. Obtención de las posiciones de los satélites: almacenamiento de identificadores NORAD (PARTE III, sección 1.2, líneas 79-82)

El siguiente campo está formado por los nombres de los archivos .txt en donde se encuentran guardados los TLE asociados a cada satélite; estos se utilizan para localizar dichos archivos. Los TLE de cada satélite se guardan en el campo adyacente en forma de vectores de celdas; cada línea ocupará una celda distinta.

```

% Generate .txt archive names column
for i=1:n_Sat_Cloud
    Sat_Cloud(i).txt_Names = strcat('sat',char(NORAD_Cloud(i,1)),'.txt'); % .txt archive names column (char)
end

```

Código 5. Obtención de las posiciones de los satélites: almacenamiento de los nombres de los archivos .txt en donde se encuentran contenidos los TLE (PARTE III, sección 1.2, líneas 84-87)

```

% Store TLE of each satellite into a cell column of cell columns
for i=1:n_Sat_Cloud
    list = fopen(Sat_Cloud(i).txt_Names); % Open text file
    Sat_Cloud(i).TLE = textscan(list,'%s','Delimiter','\n'); % Text scan data
    fclose(list); % Close file
    Sat_Cloud(i).TLE = Sat_Cloud(i).TLE{1,1}; % TLE cell column of cell columns
end

```

Código 6. Obtención de las posiciones de los satélites: almacenamiento de los TLE (PARTE III, sección 1.2, líneas 89-95)

3.1.3.3. Capa 2 de la estructura de almacenamiento (TLE_Data)

Una vez se dispone de los TLE almacenados, se puede proceder a transformarlos en posiciones y velocidades. Estas se encontrarán asociadas al instante en el que fue tomado el TLE, representado por una fecha. Cada conjunto de datos se encontrará en una fila de la estructura, siempre que se encuentren dentro del intervalo de tiempos seleccionado inicialmente. Los campos de esta estructura son:

- **Year:** año.
- **Day:** día y fracción del mismo.
- **Date:** fecha en su formato correspondiente de MATLAB.
- **Diff_minutes:** diferencia de tiempo, en minutos, respecto a la fecha inicial del intervalo.
- **satdata:** estructura con los datos necesarios para aplicar el modelo de perturbaciones simplificadas SPG4.

- **Position_xyz**: vector columna con la posición en ejes x , y y z del satélite para la fecha dada. Se da en km.
- **Velocity_xyz**: vector columna con la velocidad en ejes x , y y z del satélite para la fecha dada. Se da en km/h.
- **TLE_Diff_minutes**: diferencia de tiempo, en minutos, respecto al TLE anterior.
- **Exp_Position_xyz**: vector columna con la posición esperada para la fecha dada según el TLE anterior, aplicando el modelo SPG4. Ejes x , y y z . Se da en km.
- **Exp_Velocity_xyz**: vector columna con la velocidad esperada para la fecha dada según el TLE anterior, aplicando el modelo SPG4. Ejes x , y y z . Se da en km/h.
- **Error_Position_km**: escalar con la diferencia, en km, entre la posición obtenida mediante el TLE presente y el anterior, aplicando el modelo SPG4.
- **Error_Velocity_kmh**: escalar con la diferencia, en km/h, entre la velocidad obtenida mediante el TLE presente y el anterior, aplicando el modelo SPG4.

Fields	Year	Day	Date	Diff_minutes	satdata	Position_xyz	Velocity_xyz	TLE_Diff_minutes	Exp_Position_xyz	Exp_Velocity_xyz	Error_Position_km	Error_Velocity_kmh
1	2021	244.0801	1x1 datetime	115.3528	1x1 struct	[-3.1314e+03...	[3.4289;-3.75...					
2	2021	244.8612	1x1 datetime	1.2402e+03	1x1 struct	[-4.9215e+03...	[-0.1765;-7.0...	1.1248e+03	[-4.9215e+03;-1.0...	[-0.1765;-7.0106;-...	0.0629	5.9492e-05

Figura 21. Filas 1 y 2 de la estructura TLE_Data

Para obtener los parámetros a introducir en el modelo SPG4 (satdata), así como las fechas de referencia para cada parámetro, es necesario crear dos bucles anidados: el primero, para ir cambiando de satélite y el segundo, para ir alternando de dos en dos las filas de sus TLE.

```
% Select TLE data according to the date interval selection
for i=1:n_Sat_Cloud
    n_TLE = size(Sat_Cloud(i).TLE,1);
    j = 1;
    k = 1;
    flag = 0; % Activated if TLE dates are over maximum interval date
    while j <= n_TLE && flag == 0
        % Load the odd rows of each TLE
        line = Sat_Cloud(i).TLE{j,1};

        % Full year of the TLE
        line_year_2d = str2num(line(19:20)); % Obtain the last 2 digits of the TLE
        if line_year_2d > 57 % Year between 1957 and 1999
            line_year = 1900 + line_year_2d;
        else % Year between 2000 and 2056
            line_year = 2000 + line_year_2d;
        end

        % Day
        line_day = str2num(line(21:32));
```

Código 7. Obtención de las posiciones de los satélites: análisis de las fechas de los TLE (PARTE III, sección 1.2, líneas 97-116)

Como se muestra en el extracto de código anterior, el segundo bucle tendrá una variable auxiliar «flag», con el objetivo de indicar que la fecha del último TLE es superior a la fecha

máxima del intervalo y así poder dejar de ejecutarse, ahorrando tiempo. Una vez se conocen las fechas de cada TLE, se va comprobando que estas se encuentren dentro del intervalo especificado; es necesario un orden cronológico dentro de los archivos. Si la última fecha es superior a la máxima especificada, la variable «flag» tomará el valor de 1 y la ejecución del bucle se interrumpirá. Si el TLE cumple con las especificaciones de fecha, se introducirán sus líneas en la función TLE_Satdata, obteniéndose los parámetros requeridos para el modelo SPG4.

```

% Create a structure with the TLE data if the epoch is contained in the date
interval
if line_year >= Date_Cloud.y_0 && line_day >= Date_Cloud.Epoch_Day_0
    if line_year <= Date_Cloud.y_f && line_day <= Date_Cloud.Epoch_Day_f
        % TLE obtention year
        Sat_Cloud(i).TLE_Data(k).Year = line_year;

        % TLE obtention day of the year
        Sat_Cloud(i).TLE_Data(k).Day = line_day;

        % Date of TLE obtention
        [mon,day,hr,minute,sec] = days2mdh(line_year,line_day);
        Sat_Cloud(i).TLE_Data(k).Date = datetime(line_year,mon,day,hr,minute,sec);

        % Time difference from initial date
        Sat_Cloud(i).TLE_Data(k).Diff_minutes = minutes(split(between(Date_Cloud.
            Date_0,Sat_Cloud(i).TLE_Data(k).Date,'time'),'time')));

        % Structure of the data obtained from the TLE for each epoch
        TLE_Line1 = Sat_Cloud(i).TLE{j,1};
        TLE_Line2 = Sat_Cloud(i).TLE{j+1,1};
        Sat_Cloud(i).TLE_Data(k).satdata = TLE_Satdata(TLE_Line1,TLE_Line2);

        k = k+1;
    else
        flag = 1;
    end
end
end
j = j+2;
end
end
end

```

Código 8. Obtención de las posiciones de los satélites: almacenamiento de los parámetros necesarios para utilizar el modelo SPG4 (PARTE III, sección 1.2, líneas 118-146)

Habiendo determinado qué TLE son los necesarios, se procede a transformar las variables de la estructura satdata en posiciones y velocidades. Esto se consigue mediante la función «spg4.m», ya mencionada. El tiempo introducido en la función es 0, representando el momento presente en el que fueron tomados los datos. Como la obtención de los datos de distintos satélites es independiente, puede emplearse un bucle de tipo «parfor» para acelerar el proceso.

```

% Position and velocity in xyz for each TLE of each satellite
parfor i=1:n_Sat_Cloud % Parallel Computing Toolbox needed: it allows a faster execution.
    Change "parfor" by "for" if Parallel Computing Toolbox is not installed
    for j=1:1:size(Sat_Cloud(i).TLE_Data,2)
        [pos, vel] = spg4(0,Sat_Cloud(i).TLE_Data(j).satdata);
        Sat_Cloud(i).TLE_Data(j).Position_xyz = pos; % [km]
        Sat_Cloud(i).TLE_Data(j).Velocity_xyz = vel; % [km/h]
    end
end
end

```

Código 9. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 148-155)

A partir de este punto, se añaden dos secciones de uso no indispensable para el resto del funcionamiento del programa. La primera, es la diferencia de tiempos entre TLE. Esta puede ser empleada para calcular la siguiente sección, en donde se calculan las posiciones y velocidades esperadas en cada fecha según el TLE anterior, así como el error respecto a las originales.

```
% Time between TLE of the same satellite [minutes]
for i=1:n_Sat_Cloud
    for j=2:1:size(Sat_Cloud(i).TLE_Data,2)
        Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes = minutes(split(between(Sat_Cloud(i).
            TLE_Data(j-1).Date,Sat_Cloud(i).TLE_Data(j).Date,'time'),'time')));
    end
end
```

Código 10. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 159-164)

Disponiendo de la diferencia de tiempos, se calculan las posiciones y velocidades según el TLE anterior. Tras esto, se calcula el módulo del vector diferencia entre las posiciones y velocidades originales y las obtenidas mediante el modelo de perturbaciones. Si el módulo es superior al umbral de incertidumbre del modelo SPG4 significará que el satélite ha sido desplazado de su órbita habitual. Este umbral se ha fijado en 5 km por cada día que haya pasado, siendo una aproximación muy simplificada, ya que el incremento del error no es lineal con el tiempo [44]. Puesto que, de la constelación Globalstar, se dispone de sus TLE para periodos inferiores a un día, se puede considerar válida la aproximación, siendo recomendado cambiar la función para tiempos superiores.

```
% Expected position [km] and velocity [km/h] in xyz for according to the previous TLE of
each satellite
k = 1;
for i=1:n_Sat_Cloud
    for j=2:1:size(Sat_Cloud(i).TLE_Data,2)
        % Expected position and velocity
        [pos, vel] = sgp4(Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes,Sat_Cloud(i).TLE_Data(j)
            -1).satdata);
        Sat_Cloud(i).TLE_Data(j).Exp_Position_xyz = pos; % [km]
        Sat_Cloud(i).TLE_Data(j).Exp_Velocity_xyz = vel; % [km/h]
        Sat_Cloud(i).TLE_Data(j).Error_Position_km = norm(Sat_Cloud(i).TLE_Data(j).
            Position_xyz - Sat_Cloud(i).TLE_Data(j).Exp_Position_xyz); % [km]
        Sat_Cloud(i).TLE_Data(j).Error_Velocity_kmh = norm(Sat_Cloud(i).TLE_Data(j).
            Velocity_xyz - Sat_Cloud(i).TLE_Data(j).Exp_Velocity_xyz); % [km/h]

        % Simplified position error function (change function for TLE_Diff_minutes > 1440
minutes [1 day])
        Error_Max_km = Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes*(5/1440); % [km] 5 km of
maximum error / day (1440 minutes)

        % Location index matrix for displaced satellites
        if Sat_Cloud(i).TLE_Data(j).Error_Position_km > Error_Max_km
            Error_Index(k,1) = i; % Satellite index
            Error_Index(k,2) = j; % TLE index
            k = k+1;
        end
    end
end
```

Código 11. Obtención de las posiciones de los satélites: obtención de las posiciones y velocidades para las fechas de cada TLE (PARTE III, sección 1.2, líneas 166-187)

Estas dos últimas secciones pueden ser omitidas, en caso de no requerirse, mediante la función de comentarios de MATLAB. Así puede ahorrarse tiempo de ejecución del programa.

3.1.3.4. Capa 2 de la estructura de almacenamiento (Position)

Las posiciones empleadas para generar los grafos son las determinadas para cada una de las fechas, situadas en el tiempo de forma regular según el periodo de muestreo, dentro del intervalo seleccionado en la sección de inicio. Estas se muestran dentro de la subestructura «Position», en donde cada una de las asociadas a los distintos satélites tendrá el mismo número de filas que las demás, siendo cada una equivalente a una de las fechas marcadas según el intervalo de muestreo. Los campos de esta estructura son:

- **Date:** fecha para la que se muestran posición y velocidad.
- **Date_Char:** fecha en formato vector de caracteres para facilitar su visualización en la estructura.
- **Time_interval_Backwards:** tiempo, en minutos, desde el siguiente TLE hasta la fecha de la línea.
- **Position_Backwards:** vector columna con la posición del satélite, en ejes x , y y z , para la fecha de la línea, aplicando el modelo SPG4 a partir del siguiente TLE. Se da en km.
- **Velocity_Backwards:** vector columna con la velocidad del satélite, en ejes x , y y z , para la fecha de la línea, aplicando el modelo SPG4 a partir del siguiente TLE. Se da en km/h.
- **Time_interval_Forward:** tiempo, en minutos, desde el TLE anterior hasta la fecha de la línea.
- **Position_Forward:** vector columna con la posición del satélite, en ejes x , y y z , para la fecha de la línea, aplicando el modelo SPG4 a partir del TLE anterior. Se da en km.
- **Velocity_Forward:** vector columna con la velocidad del satélite, en ejes x , y y z , para la fecha de la línea, aplicando el modelo SPG4 a partir del TLE anterior. Se da en km/h.
- **Position_Mean:** vector columna con la posición media entre los vectores de posición anteriores, aplicando la ley inversa de la palanca. Se da en km.
- **Velocity_Mean:** vector columna con la velocidad media entre los vectores de velocidad anteriores, aplicando la ley inversa de la palanca. Se da en km/h.

F...	Date	Date_Char	Time_interval_Backwards	Position_Backwards	Velocity_Backwards	Time_interval_Forward	Position_Forward	Velocity_Forward	Position_Mean	Velocity_Mean
207	1x1 datetime	'01-Sep-2021 20:36...		-4.1749[-4.7481e+03;679.92... [-1.2022;-7.0540;-0.1...		1.1206e+03[-4.7481e+03;679...	[-1.2022;-7.0540;-...	[-4.7481e+03;6...	[-1.2022;-7.054...	
208	1x1 datetime	'01-Sep-2021 20:42...		-880.2833[-4.9159e+03;-1.850... [0.2781;-6.8748;-1.9...		1.8251[-4.9159e+03;-1.8...	[0.2781;-6.8748;-1...	[-4.9159e+03;-1...	[0.2781;-6.8748...	

Figura 22. Filas 207 y 208 de la estructura Position

El primer paso consiste en asignar las fechas de la malla, creada en la sección de inicio, a cada línea de la estructura.

```

% Date collumn
for i=1:n_Sat_Cloud
    for j=1:1:n_d_grid_Cloud
        Sat_Cloud(i).Position(j).Date = d_grid_Cloud(j,1);
    end
end

% Convert date to text to simplify data view
for i=1:n_Sat_Cloud
    for j=1:1:n_d_grid_Cloud
        Sat_Cloud(i).Position(j).Date_Char = datestr(Sat_Cloud(i).Position(j).Date);
    end
end
end

```

Código 12. Obtención de las posiciones de los satélites: asignación de las fechas determinadas por el periodo de muestreo (PARTE III, sección 1.2, líneas 191-203)

Como el modelo de perturbaciones simplificadas SPG4 tiene un error asociado al tiempo pasado desde que se toma la información, se han calculado las posiciones y velocidades desde los TLE anterior y posterior a la fecha. Los valores calculados tendrán un mayor nivel de precisión para aquellas fechas más cercanas a un TLE, por ello se han ponderado los valores en concordancia con este principio. El eje vertical representa, de forma simplificada, la posición del satélite; el eje horizontal, el tiempo; en rojo se muestra la estimación de posición del modelo SPG4 para el TLE obtenido en el instante 1; en azul, la estimación, hacia atrás en el tiempo, para el TLE obtenido en el instante 2; finalmente, la curva morada representa la ponderación de ambas posiciones, pudiéndose considerar como una estimación más fiable y logrando evitar escalones en la posición al cambiar el TLE de referencia.

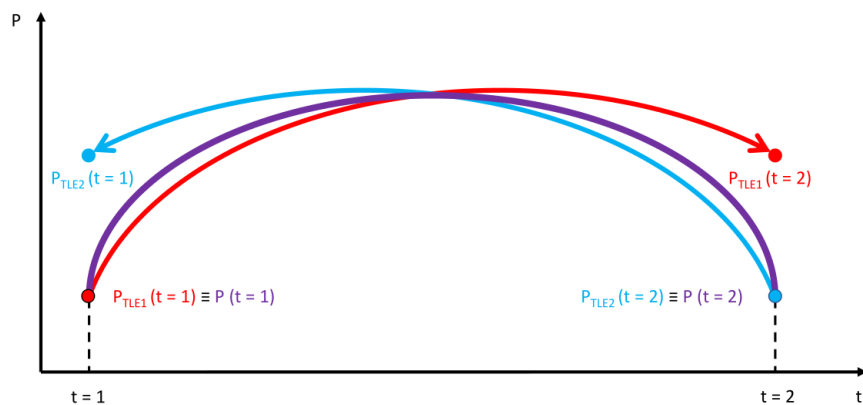


Figura 23. Diagrama conceptual para el cálculo de la posición de un satélite

La función de ponderación para la posición es la mostrada a continuación, pudiéndose sustituir las posiciones por valores de velocidad para obtener su análoga.

$$P(t = n) = \frac{|t_{TLE1 \rightarrow n}| P_{TLE2}(t = n) + |t_{TLE2 \rightarrow n}| P_{TLE1}(t = n)}{|t_{TLE1 \rightarrow n}| + |t_{TLE2 \rightarrow n}|} \quad (24)$$

Para calcular los tiempos de cada fecha respecto a la de los TLE anterior y posterior, se emplea el tiempo desde el siguiente TLE hasta la fecha presente con signo negativo «tb» para indicar cuando se ha sobrepasado este último. Cuando el signo cambia, significa que la fecha

es posterior a dicho TLE, por lo que se cambia el TLE de referencia por el siguiente, tanto en la comparación hacia delante, como en la comparación hacia atrás; el TLE posterior pasará a ser el nuevo TLE anterior en las siguientes iteraciones. Si no hay TLE anterior o posterior, por encontrarse la fecha justo al principio o al final del intervalo de tiempo, se emplearán los mismos valores de tiempo en ambos campos. Las posiciones y velocidades se calculan en función de estos tiempos y después, se calcula la media ponderada según la ecuación (24).

```

% Position and velocity
parfor i=1:n_Sat_Cloud % Parallel Computing Toolbox needed: it allows a faster execution.
    Change "parfor" by "for" if Parallel Computing Toolbox is not installed
    k = 1; % Backwards index
    r = 1; % Forward index
    for j=1:1:n_d_grid_Cloud
        % Time between current date in the grid and the TLE that was obtained after that
        % date (time origin in TLE date)
        tb = minutes(split(between(Sat_Cloud(i).TLE_Data(k).Date,Sat_Cloud(i).Position(j).
            Date,'time'),'time'));

        % Time between current date in the grid and the TLE that was obtained before that
        % date (time origin in TLE date)
        tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(j).
            Date,'time'),'time'));

        % If the tb is positive that means that the current date in the grid is after the
        % TLE, so it changes the time origin to the next satellite's TLE
        if tb > 0 && k == size(Sat_Cloud(i).TLE_Data,2) && r < size(Sat_Cloud(i).TLE_Data
            ,2)
            r = r+1;
            tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(
                j).Date,'time'),'time'));
        end
        if tb > 0 && k < size(Sat_Cloud(i).TLE_Data,2)
            if k > 1
                r = r+1;
            end
            k = k+1;
            tb = minutes(split(between(Sat_Cloud(i).TLE_Data(k).Date,Sat_Cloud(i).Position(
                j).Date,'time'),'time'));
            tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(
                j).Date,'time'),'time'));
        end

        % Position and velocity according to tb
        [posb, velb] = sgp4(tb,Sat_Cloud(i).TLE_Data(k).satdata);
        Sat_Cloud(i).Position(j).Time_interval_Backwards = tb;
        Sat_Cloud(i).Position(j).Position_Backwards = posb;
        Sat_Cloud(i).Position(j).Velocity_Backwards = velb;

        % Position and velocity according to tf
        [posf, velf] = sgp4(tf,Sat_Cloud(i).TLE_Data(r).satdata);
        Sat_Cloud(i).Position(j).Time_interval_Forward = tf;
        Sat_Cloud(i).Position(j).Position_Forward = posf;
        Sat_Cloud(i).Position(j).Velocity_Forward = velf;

        % Mean position and velocity according
        pos = (abs(tb)*posf + abs(tf)*posb) / (abs(tb) + abs(tf)); % Inverse levering between
        % time and position
        vel = (abs(tb)*velf + abs(tf)*velb) / (abs(tb) + abs(tf)); % Inverse levering between
        % time and velocity
        Sat_Cloud(i).Position(j).Position_Mean = pos;
        Sat_Cloud(i).Position(j).Velocity_Mean = vel;
    end
end
end

```

Código 13. Obtención de las posiciones de los satélites: aplicación del modelo SPG4 y la ley de la palanca inversa para el cálculo de las posiciones y velocidades de cada satélite (PARTE III, sección 1.2, líneas 205-248)

3.1.3.5. Disposiciones finales

Habiendo completado la estructura `Sat_Cloud`, se guardan las variables a emplear durante la aplicación de la Teoría de Redes Complejas. Estas se guardan en un archivo denominado «`Cloud_Variables`», las mismas son:

- **n_Sat_Cloud**: número de satélites analizados.
- **n_d_grid_Cloud**: número de fechas analizadas.
- **Date_Cloud**: estructura con las fechas seleccionadas para el intervalo de análisis.
- **T_Cloud_minutes**: periodo de muestreo.
- **Sat_Cloud**: estructura con los datos de posición de los satélites a lo largo del intervalo.

El programa utiliza, en su última línea, la función «`toc`», permitiendo medir y mostrar el tiempo de ejecución total. Esto es de vital importancia de cara a optimizar su ejecución o para evaluar la magnitud de la información capaz de procesar por el sistema.

Finalmente, se incluye un apartado opcional para dibujar las órbitas de los satélites a lo largo del tiempo. Este permite seleccionar un satélite para un intervalo a determinar, la constelación entera para el mismo intervalo, un satélite para todo el intervalo inicial y la constelación entera para todo el intervalo inicial.

En las figuras puede observarse como existe un pequeño solape entre posiciones debido a la complejidad asociada a determinar el periodo orbital con exactitud. También puede contemplarse como los satélites se agrupan en distintas órbitas cercanas, gracias a ello son capaces de cubrir las áreas especificadas en la sección 1.5 Constelación Globalstar con un menor nivel de riesgo, disponiendo de unos espacios determinados en lugar de cubrir toda la superficie terrestre. Las zonas de los polos quedan libres, al no ser atravesadas por ninguna órbita, por lo que no disponen de cobertura. Por último, puede apreciarse como los satélites van desplazando su órbita a lo largo del día, sin recorrer las mismas posiciones de la elipse dos veces seguidas, a excepción de los extremos del eje mayor; es en estos puntos en donde se agrupan un mayor número de puntos de posición ya que el satélite reduce su velocidad.

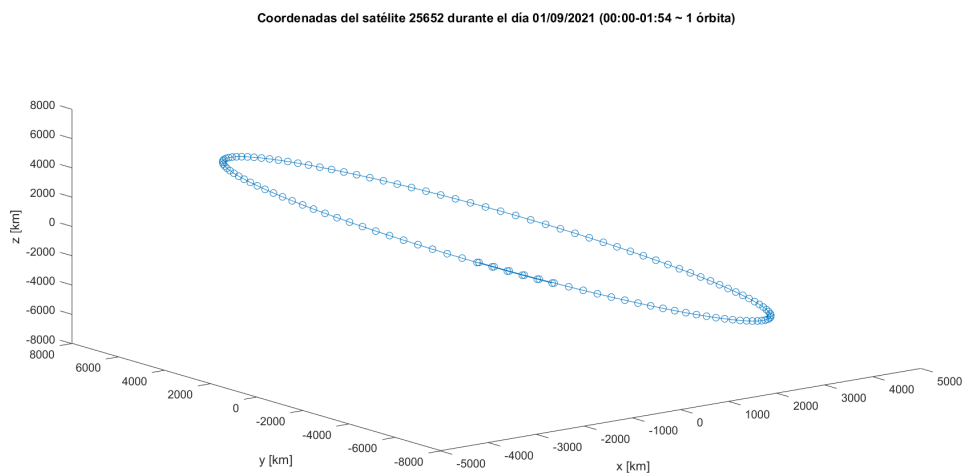


Figura 24. Coordenadas del satélite 25652 durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Coordenadas de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 ~ 1 órbita)

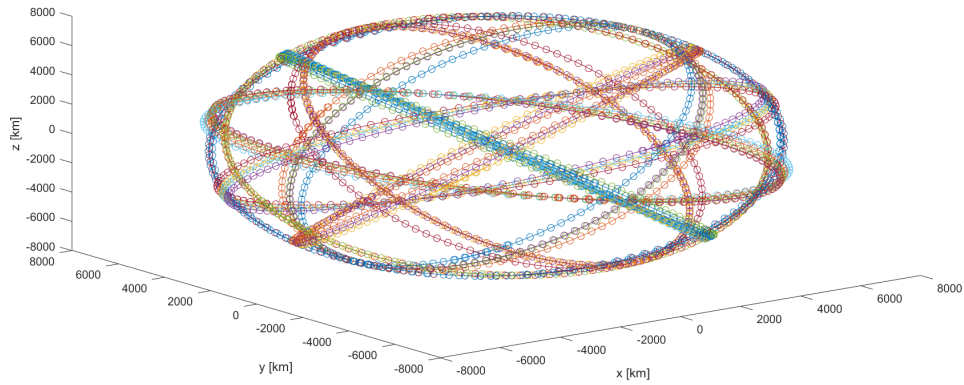


Figura 25. Coordenadas de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Coordenadas del satélite 25652 durante el día 01/09/2021

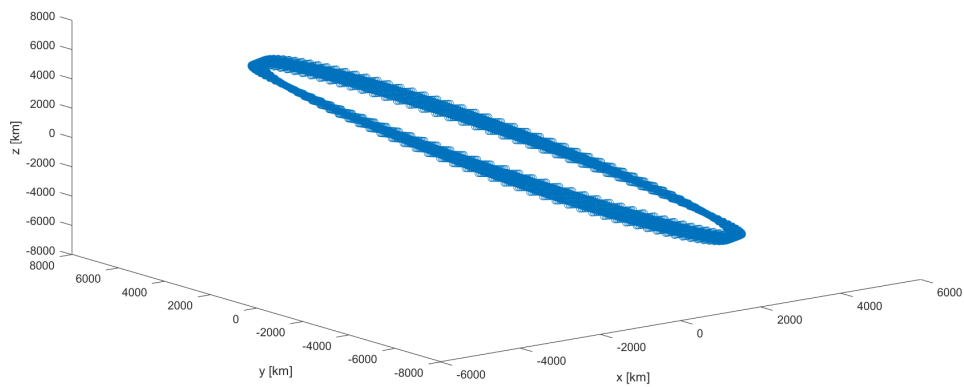


Figura 26. Coordenadas del satélite 25652 durante el día 01/09/2021

Coordenadas de la constelación Globalstar durante el día 01/09/2021

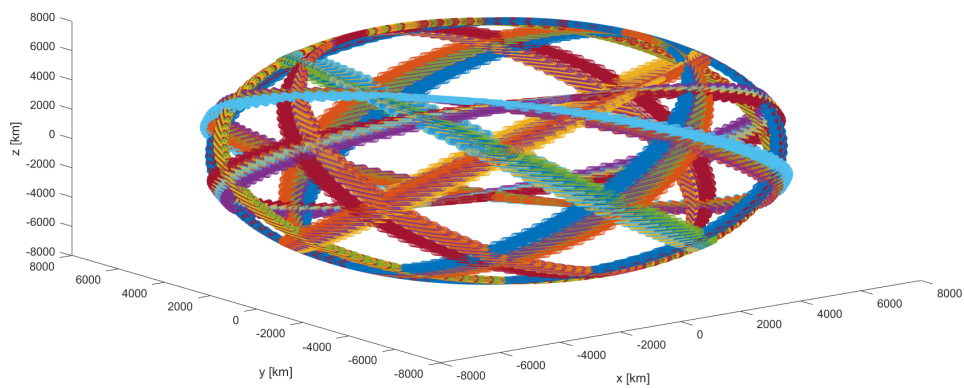


Figura 27. Coordenadas de la constelación Globalstar durante el día 01/09/2021

3.2. Análisis de vulnerabilidades de la red de satélites

Los dos métodos desarrollados para realizar el análisis de vulnerabilidades son descritos en esta sección. Ambos se basan en asumir conexiones entre satélites siempre que exista visibilidad directa entre ellos. Los satélites serán los nodos de la red y se considerarán los enlaces sólo entre parejas de estos. La función empleada para la determinación de la visibilidad es la expresión geométrica asociada a la función *rise-set* [55], relacionada con el cálculo de periodos temporales en los que dos satélites cuentan con visibilidad.

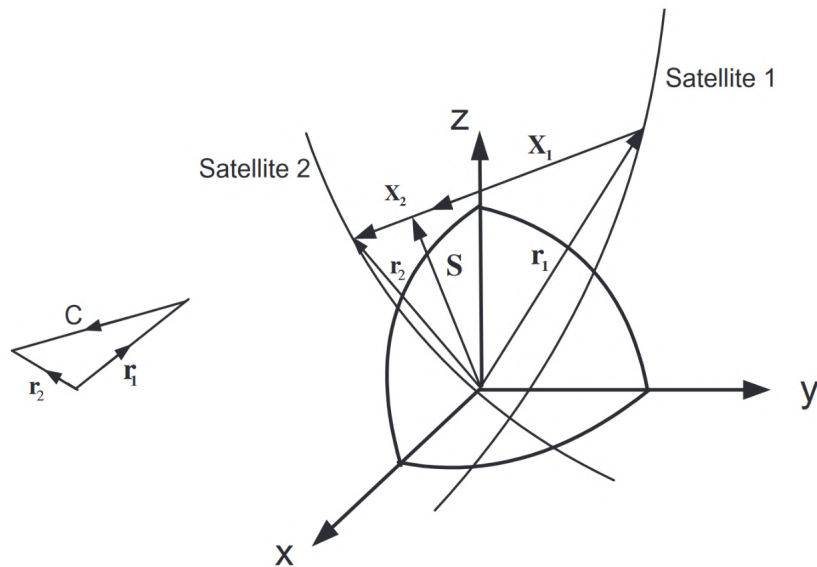


Figura 28. Representación gráfica de la geometría asociada a la función *rise-set* [55]

Para determinar la visibilidad entre dos satélites, es necesario disponer de un triángulo cuyos vértices sean estos dos junto con el centro de la Tierra. Se ha de calcular la altura S del triángulo desde la base formada entre los dos satélites C , siendo perpendicular a la misma y pasando por el centro de la Tierra. Si esta altura es menor que el radio de la Tierra junto con una distancia de margen, empleada para representar la irregularidad del planeta y el efecto de la atmósfera, se considerará que la Tierra estará situada entre los dos satélites, por lo que no existirá un enlace entre ambos durante ese instante.

El radio de la Tierra asumido es de 6378,137 km, siendo el utilizado para representar al semieje mayor por el «Sistema Geodésico Mundial 1984» o «WGS84» [56], según sus siglas en inglés. La distancia adicional considerada es de 60 km, al ser la empleada por otros autores ante el cálculo de periodos de visibilidad entre satélites [57].

Para calcular la altitud de cada satélite se emplea el módulo de su vector de posición:

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (25)$$

La distancia entre satélites es el módulo del vector resta entre sus dos vectores de posición:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (26)$$

Para calcular la altura del triángulo es necesario disponer primero del semiperímetro:

$$sp_{ij} = \frac{r_i + r_j + d_{ij}}{2} \quad (27)$$

La altura viene determinada por la fórmula:

$$S_{ij} = \frac{2\sqrt{sp_{ij}(sp_{ij} - d_{ij})(sp_{ij} - r_i)(sp_{ij} - r_j)}}{d_{ij}} \quad (28)$$

Los enlaces entre satélites se guardarán en matrices de adyacencia (1) para cada instante de tiempo, al principio de ambos métodos.

La representación de las vulnerabilidades en la red se da por el efecto generado tras la eliminación de sus elementos. Al realizar un «ataque» sobre una red compleja, el mismo puede darse sobre sus nodos o sobre sus enlaces [27]. El caso de los enlaces sería aplicable siempre que estos tuviesen una representación capaz de materializarse; ejemplos de ello serían los axones entre neuronas al deteriorarse, las conexiones entre servidores de internet al cortarse o incluso, las relaciones entre personas, en una red social, al desaparecer.

En el caso de los satélites, la única posibilidad de impedir un enlace concreto sería eclipsando la conexión con otro cuerpo orbitando entre ellos. Este caso no se presenta durante el Proyecto, en donde se busca evaluar las redes en situaciones con altas incertidumbres, enfocándose en un posible uso del IoSat, ya comentado en la sección 1.3 Internet de Satélites (IoSat), aunque pudiéndose aplicar en cualquier caso en donde se de una red de satélites; para una red con un movimiento más regular, como por ejemplo Iridium, podría ser más fácil darse dicha situación. Por este motivo, el programa no asume la eliminación de enlaces concretos, sino únicamente la de los nodos.

Otro de los aspectos relevantes a comentar es el estado de los satélites cuando se asume su eliminación de la red. Al principio, se barajó la posibilidad de desarrollar dos programas distintos, uno de ellos omitiendo aquellos satélites eliminados tras cada iteración, y otro en donde se diera la posibilidad de representar los satélites eliminados como satélites aislados, sin posibilidad de disponer de ninguna conexión, pero presentes en la red. Finalmente, se decidió prescindir de la segunda opción ya que se hacía necesario disponer de un mecanismo adicional para poder diferenciar aquellos satélites eliminados de los que quedaban aislados como consecuencia de la pérdida de otros satélites críticos. Además, dejando los satélites, sólo serían comparables aquellas redes con el mismo número de nodos desde el inicio, ya que, al comparar los parámetros, estos se verían distorsionados por ir referidos a la cantidad de satélites eliminados.

Al ejecutar ambos métodos, el programa irá localizando el satélite más crítico, simulará su eliminación y repetirá el proceso con una red sin ese satélite. De esta forma se consigue un algoritmo más eficaz que al evaluar la criticidad de los nodos una única vez en la primera iteración; estos últimos ataques son conocidos como ataques de «distribución inicial», mientras que los empleados en el Proyecto, como de «distribución recalculada» [27]. Volver a generar una nueva red tras cada eliminación permite evaluar correctamente qué enlaces desaparecieron, sin asumir ninguna duplicidad de los mismos al poder estar compartidos por varios nodos en la red inicial.

La métrica empleada para evaluar, de forma comparativa, el estado de la red es la eficiencia global E , mostrada en las ecuaciones (16) y (17). Como ya se ha indicado en la subsección 1.4.2 Métricas de integración, este parámetro permite evaluar la facilidad relativa con la que tenderían a saturarse los enlaces de la misma al transmitir información, por necesitar atravesar un mayor número de nodos. Otras métricas importantes son la longitud de trayectoria L_i (10) (11) y la longitud de trayectoria característica L (12) (13), evaluando también dichas distancias entre nodos, pero tendiendo a infinito cuando se produce una escisión en la red.

Para determinar la criticidad de cada satélite se ha decidido emplear la centralidad de intermediación b_i , mostrada en las ecuaciones (22) y (23) de la subsección 1.4.4 Métricas de centralidad. Esta mide la proporción de geodésicas que exclusivamente pueden pasar por el nodo al que va referida la métrica. Según estudios ya realizados [27], es la forma más eficaz de evaluar la criticidad de los nodos; aun así, el programa cuenta con la opción de realizar el análisis de vulnerabilidades en función del grado de cada nodo (3) (4), del coeficiente de agrupamiento local (18) (19) y de la eficiencia de cada nodo (14) (15).

Las características principales de los dos métodos son las mencionadas con anterioridad. Las particularidades de cada uno de los métodos para el análisis de vulnerabilidades son:

- **Red multiplex cuasitemporal:** se determinan los enlaces de las redes binarias para cada instante de tiempo del intervalo; toman valores de 0 o 1. Se calculan las métricas para cada una de las redes complejas binarias. Se obtienen los valores medios de cada una, así como sus desviaciones estándar. El método es más lento, aunque permite observar la evolución en el tiempo de cada una de las métricas, también permite añadir ponderación a los enlaces en función de otras variables.
- **Red monoplex:** se determinan los enlaces de las redes binarias para cada instante de tiempo del intervalo; toman valores de 0 o 1. Se suman cada una de las matrices de adyacencia y se normaliza el valor de los enlaces resultantes; se obtiene una única matriz de pesos en donde sus elementos toman valores entre 0 y 1, dependiendo de la proporción de tiempo durante el intervalo en la que esos satélites se encuentren visibles. Se calculan las métricas para redes complejas ponderadas. El método es más simple y rápido, ya que sólo requiere calcular las métricas de una red, sin embargo se pierde la dimensión temporal dentro del intervalo, así como la posibilidad de disponer de enlaces ponderados en función de otras variables, como por ejemplo el ancho de banda de la conexión.

Finalmente, se obtiene la lista con el orden de eliminación de los satélites para cada uno de los métodos y se recalculan las métricas intercambiando los métodos. Con esto se pretende comprobar si hay algún método de obtención de satélites críticos con el que se consiguen mejores resultados, ya que comparar directamente las métricas de cada uno de ellos no tendría el grado de rigurosidad adecuado.

3.2.1. Toma de datos

Para poder realizar el análisis es necesario introducir los datos de la red de satélites en el formato de la estructura Sat_Cloud. Si se desea emplear otro formato, los requerimientos principales serían disponer de la distribución regular en el tiempo de las posiciones de cada satélite, así como una fecha asociada a cada una de ellas. Las otras variables requeridas se muestran en la subsección 3.1.3.5 Disposiciones finales: «n_Sat_cloud», «n_d_grid_Cloud», «Date_Cloud» y «T_Cloud_minutes».

3.2.2. Recursos empleados

Al igual que en la sección 3.1.2 Recursos empleados para la obtención de las posiciones de los satélites, el programa hace uso de la Parallel Computing Toolbox [54] de MATLAB; el funcionamiento es el mismo que en el programa anterior.

Para el cálculo de las métricas relacionadas con la Teoría de Redes Complejas, se hace uso de los «Algoritmos de gráficas y redes» de MATLAB, los cuales vienen incluidos en la versión utilizada de dicho programa a lo largo de este Proyecto.

Como dentro de estas funciones no se incluyen aquellas relativas a el cálculo de los coeficientes locales de agrupamiento, se han empleado las funciones «clustering_coef_bu.m» y «clustering_coef_wu.m» de la librería *Brain Connectivity Toolbox* de Mikail Rubinov y Olaf Sporns [48]. Dichas funciones reciben como entrada la matriz de adyacencia (1), para el caso de las redes binarias, o la matriz de pesos (2), para el caso de las redes ponderadas. En el caso de la matriz de pesos, por especificaciones de la función, es necesario introducir valores normalizados entre 0 y 1; no habrá ningún tipo de problema con esto ya que es como se ha decidido representar por defecto esta matriz. Las salidas de dichas funciones son sendos vectores columna con los valores de los coeficientes locales de agrupamiento para cada satélite.

Al igual que en la sección 3.1.2 Recursos empleados para la obtención de las posiciones de los satélites, las funciones se almacenarán en una carpeta denominada «Brain Connectivity Toolbox», facilitando la organización. Para poder utilizarlas será necesario seleccionar y añadir el contenido de la carpeta al directorio principal mediante la acción «Add to Path».

3.2.3. Algoritmo empleado

El programa creado con anterioridad por el director de este Proyecto, Enrique Sevillano Bravo, obtenía la mayoría de las métricas presentadas en este programa, centrándose exclusivamente en las redes binarias generadas en cada instante del intervalo; es decir, las redes multiplex cusitemporales del primer método, presentado al comienzo de esta sección. Con ello se disponía de la evolución en el tiempo de cada una de las métricas.

En este Proyecto se perfecciona el funcionamiento de dicho programa, añadiendo la posibilidad de evaluar las vulnerabilidades de forma iterativa tras eliminar los satélites en orden de criticidad. Para ello, se ofrecen distintas y nuevas opciones como la centralidad de intermediación (22) (23), el grado (3) (4), el coeficiente de agrupamiento local (18) (19) o la longitud de trayectoria (10) (11).

También se desarrolla el segundo método explicado al comienzo de esta sección, en donde se genera una red monoplex para todo el intervalo. Finalmente, se intercambian las listas con el orden de los satélites eliminados por ambos métodos y así poder comprobar cual ha resultado más efectivo en función de la métrica de eficiencia de cada uno de ellos.

Al igual que en el programa para la obtención de las posiciones de los satélites, la organización general está basada en el uso de estructuras. Con esto se pueden eliminar con facilidad los bloques de aquellas métricas de las que no se desea calcular sus valores, con el correspondiente ahorro de tiempo que conlleva.

3.2.3.1. Inicio y selección de los parámetros de funcionamiento

El nuevo programa comienza limpiando el espacio de trabajo anterior y cargando las variables almacenadas por el programa para la obtención de las posiciones de los satélites, ya descritas en la subsección 3.1.3.5 Disposiciones finales.

Al igual que en el programa anterior, se dispone de una sección de selección de parámetros. En ella se puede seleccionar: el nombre de la nueva lista de satélites a analizar, la cual puede contener menos satélites de los posicionados con el programa anterior; un nuevo intervalo para calcular las métricas de las redes, el cual ha de estar contenido en el intervalo para el que fueron calculadas las posiciones; un múltiplo que permite reducir el periodo de muestreo por el que se definen las posiciones de los satélites, ahorrando tiempo de ejecución; una proporción de satélites a eliminar para el análisis; un índice para determinar el método de asignación de criticidad a los satélites; y las constantes astronómicas.

```
% Loading of satellite positions and related variables
load('Cloud_Variables');

% List of wanted satellites [1]
list_title = 'Globalstar.txt';

% Date interval selection
% Initial [2]
Date_Network.s_0 = 0; % Second 0-59
Date_Network.m_0 = 0; % Minute 0-59
Date_Network.h_0 = 0; % Hour 0-23
Date_Network.d_0 = 1; % Day 1-28/29/30/31 depending on the month
Date_Network.mo_0 = 9; % Month 1-12
Date_Network.y_0 = 2021; % Year 1957-2056
% Final [3]
Date_Network.s_f = 0; % Second 0-59
Date_Network.m_f = 54; % Minute 0-59
Date_Network.h_f = 1; % Hour 0-23
Date_Network.d_f = 1; % Day 1-28/29/30/31 depending on the month
Date_Network.mo_f = 9; % Month 1-12
Date_Network.y_f = 2021; % Year 1957-2056

% Time period between positions multiplum [4]
mult_period = 1; % Time between measures must be a multiplum of the time grid interval of
    the cloud (must be an integer > 1)

% Maximum number of iterations [5]
p = 0.1; % Portion of the total number of satellites to eliminate (from 0 to 1)

% Method for determining vulnerability [6]
method = 1;
% 1 -> Max betweenes centrality
% 2 -> Max node degree
% 3 -> Min local clustering coefficient
% 4 -> Max node efficiency

%Astronomical constants [7]
R_Earth = 6378.137; % Earth's radius [km] (WGS-84)
dR = 60; % Extra radius [km]
```

Código 14. Análisis de vulnerabilidades de la red de satélites: parámetros de funcionamiento (PARTE III, sección 2.1, líneas 22-59)

Una vez seleccionados los parámetros de ejecución, los ajustes iniciales en los que se asignan los satélites a representar, así como los límites del intervalo de fechas, funcionan de la misma forma que en el programa anterior. La única excepción se da al generar la malla de fechas a las que se encontrará asignada cada una de las redes en formato multiplex.

Al generar la malla, es necesario ajustar la diferencia entre cada una de las fechas en concordancia con el múltiplo elegido anteriormente. Otro factor importante consiste en asegurar que la primera fecha del intervalo se corresponde con una fecha con posiciones asignadas. Para ello, se emplean las variables «t_diff_minutes», con la que se mide la diferencia en minutos desde el instante inicial de la estructura de posiciones hasta la primera fecha del intervalo seleccionado, y «t_diff_adjusted_minutes», con la que se mide la diferencia en minutos desde el instante inicial de la estructura de posiciones hasta la primera fecha de dicha estructura que se encuentre dentro del nuevo intervalo de tiempo seleccionado.

```
% Time period between positions [minutes] [4]
if mult_period < 1
    disp('Incorrect multiplum')
    pause
end
T_Network_minutes = T_Cloud_minutes*round(mult_period); % [minutes]

% Date grid array
t_diff_minutes = minutes(minutes(split(between(Date_Cloud.Date_0,Date_Network.Date_0,'Time')
,'Time')));
t_diff_adjusted_minutes = T_Network_minutes*ceil(datenum(t_diff_minutes)/datenum(
T_Network_minutes));
d_Network_0 = Date_Cloud.Date_0 + t_diff_adjusted_minutes;
d_grid_Network = d_Network_0:T_Network_minutes:Date_Network.Date_f;
d_grid_Network = d_grid_Network';
n_d_grid_Network = size(d_grid_Network,1);
```

Código 15. Análisis de vulnerabilidades de la red de satélites: ajuste de fechas del nuevo intervalo (PARTE III, sección 2.1, líneas 92-105)

3.2.3.2. Creación de la estructura de redes multiplex (Sat_Network)

Para evaluar ambos métodos es necesario disponer de una estructura en donde se almacenen las distintas métricas para cada instante de tiempo. A dicha estructura se la ha denominado como «Sat_Network» y posee un número de filas equivalente al número de fechas mostradas en el intervalo. Los campos con los que cuenta son:

- **Date:** fecha para la que se muestran las métricas de cada instante.
- **Date_Char:** fecha en formato vector de caracteres para facilitar su visualización en la estructura.
- **Nodes:** estructura en donde están contenidos los identificadores NORAD de cada satélite junto a su vector de posiciones.
- **Iteration:** estructura en donde se guardan las métricas asociadas a cada red multiplex. Cada fila representa los datos tras la eliminación de un satélite y los anteriores, siendo la primera la que representa a la red completa.

Mediante distintos bucles «for», se van creando cada uno de los campos de la estructura de igual forma que en el programa anterior. Para situar las posiciones de los satélites, es necesario tener en cuenta el múltiplo de reducción del periodo «mult_period», por medio del cual se van omitiendo aquellas fechas de la estructura de posiciones del programa anterior que no deban ser guardadas.

```

% Position of each satellite for each date
for i=1:n_d_grid_Network
    for j=1:n_Sat_Network
        for k=1:n_Sat_Cloud
            if Sat_Cloud(k).NORAD == Sat_Network(i).Nodes(j).NORAD
                for m=1:round(mult_period):n_d_grid_Cloud
                    if Sat_Cloud(k).Position(m).Date == Sat_Network(i).Date
                        Sat_Network(i).Nodes(j).Position_Mean = Sat_Cloud(k).Position(m).
                            Position_Mean;
                    end
                end
            end
        end
    end
end
end
end
end

```

Código 16. Análisis de vulnerabilidades de la red de satélites: almacenamiento de las posiciones de los satélites (PARTE III, sección 2.1, líneas 129-142)

Una vez se dispone de las posiciones, se procede a calcular la visibilidad entre satélites aplicando el método descrito al comienzo de esta sección. Si existe visibilidad, se guardará un 1 en el elemento correspondiente de la matriz de adyacencia (1), la cual se encontrará en la primera columna de la estructura «Iteration». El resto de columnas de dicha estructura se rellenan en los bucles posteriores.

```

% Adjacency matrix for each date for the initial state
parfor i=1:n_d_grid_Network
    for j=1:n_Sat_Network
        x1 = Sat_Network(i).Nodes(j).Position_Mean(1,1);
        y1 = Sat_Network(i).Nodes(j).Position_Mean(2,1);
        z1 = Sat_Network(i).Nodes(j).Position_Mean(3,1);
        r1_km = sqrt(x1^2 + y1^2 + z1^2); % Satellite height
        for k=1:n_Sat_Network
            % Triangle between Earth's center and 2 satellites
            x2 = Sat_Network(i).Nodes(k).Position_Mean(1,1);
            y2 = Sat_Network(i).Nodes(k).Position_Mean(2,1);
            z2 = Sat_Network(i).Nodes(k).Position_Mean(3,1);
            r2_km = sqrt(x2^2 + y2^2 + z2^2); % Height of the rest of the satellites
            d_km = sqrt((x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2); % Distance between satellites
            sp_km = (r1_km + r2_km + d_km)/2; % Triangle semiperimeter
            h_km = (2/d_km)*sqrt(sp_km*(sp_km-d_km)*(sp_km-r1_km)*(sp_km-r2_km)); %
                Triangle height formula

            % If triangle's height is higher than Earth's radius (+60 km extra) satellites
            are visible [7]
            if h_km >= R_Earth + dR
                Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 1;
            else
                Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 0;
            end
        end
    end
end
end
end

```

Código 17. Análisis de vulnerabilidades de la red de satélites: creación de la matriz de adyacencia inicial (Sat_Network) (PARTE III, sección 2.1, líneas 144-169)

La matriz de adyacencia (1) se guarda de forma separada a los bucles de los dos métodos ya que es utilizada por ambos. Si se desea no calcular alguno de ellos bastará con aplicar la función comentar de MATLAB sobre su bucle correspondiente, sin afectar a la matriz inicial.

3.2.3.3. Estructura de redes multiplex (Sat_Network) y (Sat_Multiplex)

Las métricas de las redes multiplex se calculan dentro de un único bucle «for» en el que el índice se encuentra asociado a las distintas iteraciones en las que se van eliminando satélites críticos. Cada una se corresponde con una de las filas de la estructura «Iteration» de dentro de la estructura «Sat_Network».

Antes de mostrar los bucles para calcular las métricas de las redes para cada instante de tiempo, se sitúa una sentencia condicional con la que, si la iteración es superior a la primera, se elimina el satélite más crítico determinado en la iteración anterior. El índice empleado para localizarlo es la variable «s», a través de la que se eliminan las filas y las columnas pertinentes de la matriz de adyacencia. A partir de ese punto, se vuelven a calcular todas las métricas en función de la nueva matriz.

```

% Changes for each iteration
if it>1
    % Changes in NORAD names list
    Sat_Multiplex(it).Missed_Satellite = NORAD_Network(s,:); % Missed satellite
    NORAD_Network(s,:) = [];

    % Changes in indexes
    n_Sat_Network = n_Sat_Network-1;

    % Changes in adjacency matrixes for each date
    for i=1:n_d_grid_Network
        Sat_Network(i).Iteration(it).Adjacency_Matrix = Sat_Network(i).Iteration(it-1).
            Adjacency_Matrix;
        Sat_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
        Sat_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
    end
end
end

```

Código 18. Análisis de vulnerabilidades de la red de satélites: modificación de las nuevas matrices de adyacencia inicial (Sat_Network) (PARTE III, sección 2.1, líneas 176-191)

Para calcular las métricas se emplea un bucle «parfor» en el que el índice principal recorre cada una de las fechas determinadas por el periodo de muestreo. Estas se guardan en variables auxiliares que finalmente son almacenadas dentro de la estructura «Iteration» de dentro de la estructura «Sat_Network». La misma contiene las variables:

- **Adjacency_Matrix:** matriz de adyacencia (1).
- **Satellites:** estructura con las métricas de los distintos satélites.
- **Network:** matriz de adyacencia guardada en el formato de estructura de red de MATLAB. Contiene los identificadores de cada nodo y los enlaces existentes entre ellos, definidos por parejas de estos.
- **Distances:** matriz de distancias (7).
- **n_Edges:** número de enlaces de la red.
- **Characteristic_Path_Length:** longitud de trayectoria característica de la red (12).
- **Efficiency:** eficiencia de la red (16).
- **Global_Clustering_Coefficient:** coeficiente global de agrupamiento de la red (20).

Dentro de la estructura «Network» se almacenan, en distintas variables, las métricas de cada uno de los satélites. Estas son:

- **NORAD**: identificador de cada satélite.
- **Node_Degree**: grado del nodo (3).
- **Path_Length**: longitud de trayectoria del nodo (10).
- **Efficiency**: eficiencia del nodo (14).
- **Local_Clustering_Coefficient**: coeficiente local de agrupamiento (20). Se emplea la función «Clustering_coef_bu.m», descrita en la subsección 3.2.2 Recursos empleados.
- **Betweenness_Centrality**: centralidad de intermediación sin normalizar. No tiene en cuenta el número de nodos en el denominador, a diferencia del valor normalizado, el cual es de uso extendido. Se calcula empleando la función «centrality» de MATLAB, siendo necesario multiplicar el valor obtenido por 2, ya que al emplear como entrada una red no dirigida sólo tiene en cuenta los enlaces en una dirección, pero no en la recíproca.
- **Norm_Betweenness_Centrality**: centralidad de intermediación normalizada o centralidad de intermediación (22).

Una vez se dispone de las métricas de la red para cada instante de tiempo del intervalo, se procede a calcular sus valores medios y desviaciones estándar. Estos se almacenan en la estructura «Sat_Multiplex». La misma dispone de las variables explicadas con anterioridad, mostrando sus medias y desviaciones para cada iteración del bucle «for» inicial; cada fila representa una iteración tras la que se ha eliminado un satélite. Es decir, en cada fila se muestra una estructura «Satellites» (con los valores de las métricas asociadas a cada satélite), los valores relacionados con las métricas globales y una variable «Missed_Satellite», de tipo celda, en la que se indica el satélite eliminado en la iteración anterior por ser el más crítico.

Conociendo los valores medios, es posible identificar cuál es el satélite más crítico durante ese intervalo. Para escoger el criterio de selección se emplea una sentencia condicional de tipo «switch». Como índice se utiliza la variable «method», representando a las opciones de:

- **1**: máxima centralidad de intermediación (22).
- **2**: máximo grado (3).
- **3**: mínimo coeficiente de agrupamiento local (18).
- **4**: máxima eficiencia local (14).

Mediante las funciones «find» y «max» o «min» se localiza el índice del satélite más crítico y se almacena en la variable «s», siendo aquella empleada, al principio del proceso, para localizar el identificador NORAD y eliminar las filas correspondientes de la matriz de adyacencia en la siguiente iteración.

Si varios satélites tienen un mismo valor máximo para la métrica evaluada, se emplearán otras métricas de forma sucesiva para determinar qué satélite se ha de eliminar, como por ejemplo el grado máximo y el coeficiente de agrupamiento local para el extracto de código mostrado. Si volviesen a ser iguales se escogería el primero de los índices.

```

case 1 % Satellite with higher centrality
    s = find(Av_Norm_Betweenness_Centrality==max(Av_Norm_Betweenness_Centrality)); % Allows
        to find more than one index
    % If 2 satellites have the same centrality the one with higher degree is chosen
    if length(s) > 1
        Degree_aux = zeros(1,length(s));
        for i=1:length(s)
            Degree_aux(i) = Av_Degree(s(i));
        end
        s = find(Av_Degree==max(Degree_aux));
        % If 2 satellites have the same degree the one with lower weighted LCC is chosen
        if length(s) > 1
            LCC_aux = zeros(1,length(s));
            for i=1:length(s)
                LCC_aux(i) = Av_LCC(s(i));
            end
            s = find(Av_LCC==min(LCC_aux));
            % If 2 satellites have the same LCC the first one is chosen
            if length(s) > 1
                s = s(1);
            end
        end
    end
end
end

```

Código 19. Análisis de vulnerabilidades de la red de satélites: selección del satélite con mayor centralidad de intermediación (Sat_Multiplex) (PARTE III, sección 2.1, líneas 354-375)

Respecto a la opción del mínimo coeficiente de agrupamiento local (18), es necesario tener en cuenta que el objetivo de la misma es encontrar aquellos satélites rodeados por vecinos con pocas interconexiones, buscando que, al eliminarlos, estos queden aislados. Como se explica en la sección 1.4.3 Métricas de segregación, según la mayoría de autores, a aquellos nodos con menos de 2 vecinos se les ha de asignar un valor de 0 a esta métrica. Siguiendo este criterio, los primeros nodos en ser eliminados serían aquellos aislados o con sólo un enlace, es decir, aquellos que generarían un impacto mínimo en el resto de la red

Para solucionar esto, sería necesario sustituir los valores de 0, en este tipo de nodos, por los valores medios de coeficiente de agrupamiento local del resto de nodos, tal y como se explica al final de la sección mencionada con anterioridad. Así el efecto sería obviado y el coeficiente global no se vería afectado.

En este programa no se contempla dicha opción ya que consumiría tiempo de computación y este fenómeno no abarca el elemento central del Proyecto. Si se fuese a realizar el análisis centrándose en el mínimo coeficiente de agrupamiento local, sería necesario programar la obtención de estos valores; al usarse como opción secundaria, en el caso de que coincidan valores de centralidad o grado empleando los otros criterios, no se daría dicho efecto ya que antes se habrán seleccionado los nodos con mayor grado.

3.2.3.4. Estructura de redes monoplex (Sat_Monoplex)

Para configurar la red monoplex correspondiente con el intervalo especificado es necesario disponer de una matriz de pesos inicial. Esta se genera sumando todas las matrices de adyacencia de la primera iteración de la estructura «Sat_Network», cada una de ellas correspondiente con un instante de tiempo. Finalmente, se normalizan los pesos entre 0 y 1, dividiendo los valores iniciales entre el número de instantes.

```

% Weight matrix for the whole period
W = zeros(n_Sat,n_Sat);
for i=1:n_d_grid_Network
    W = W + Sat_Network(i).Iteration(1).Adjacency_Matrix;
end
W = W/n_d_grid_Network;
Sat_Monoplex(1).Weight_Matrix = W;

```

Código 20. Análisis de vulnerabilidades de la red de satélites: creación de la matriz de pesos (Sat_Monoplex) (PARTE III, sección 2.1, líneas 431-437)

El resto de métricas asociadas a la red se calculan en función de esta matriz. Todas ellas se guardan en la estructura Sat_Monoplex, muy parecida a la estructura Sat_Multiplex, pero sin disponer de las desviaciones estándar de cada métrica y con las métricas específicas de las redes ponderadas. Cada una de sus filas se corresponde con una iteración, tras la que se determina el satélite más crítico y se elimina para la siguiente. Los campos de la primera capa de esta estructura son:

- **Weight_Matrix:** matriz de pesos (2). Valores entre 0 y 1.
- **Satellites:** estructura con las métricas de los distintos satélites.
- **Network:** matriz de pesos guardada en el formato de estructura de red de MATLAB. Contiene los identificadores de cada nodo y los enlaces existentes entre ellos, definidos por parejas de estos.
- **Network_Inv:** matriz de pesos inversa guardada en el formato de estructura de red de MATLAB. Los valores de la diagonal han de ser siempre 0. Se emplea para calcular de forma correcta las métricas de longitud de trayectoria, eficiencia y centralidad de intermediación.
- **Distances:** matriz de distancias (7). Se obtiene empleando «Network_Inv». Valores entre 1 e infinito.
- **Distances_Inv:** matriz de distancias inversa (7). Se obtiene invirtiendo «Distances». Valores entre 0 y 1. Los valores de la diagonal han de ser siempre 0
- **n_Edges:** número de enlaces de la red.
- **Characteristic_Path_Length_Wei:** longitud de trayectoria característica de la red ponderada (13). Se obtiene empleando «Distances».
- **Efficiency:** eficiencia de la red ponderada (17). Se obtiene empleando «Distances_Inv».
- **Global_Clustering_Coefficient:** coeficiente global de agrupamiento de la red ponderada (21). Se obtiene empleando «Weight_Matrix ».
- **Missed_Satellite:** indica el satélite eliminado en la iteración anterior por ser el más crítico.

Dentro de la estructura «Satellites» se almacenan las siguientes variables:

- **NORAD:** identificador de cada satélite.
- **Node_Degree:** grado del nodo sin tener en cuenta las ponderaciones de los enlaces (3).

- **Node_Degree_Wei:** grado del nodo teniendo en cuenta las ponderaciones de los enlaces (4).
- **Path_Length_Wei:** longitud de trayectoria del nodo en una red ponderada (11).
- **Efficiency_Wei:** eficiencia del nodo en una red ponderada (15).
- **Local_Clustering_Coefficient_Wei:** coeficiente local de agrupamiento en una red ponderada (21). Se emplea la función «Clustering_coef_wu.m», descrita en la subsección 3.2.2 Recursos empleados.
- **Betweenness_Centrality_Wei:** centralidad de intermediación sin normalizar en una red ponderada. Se calcula empleando la función «centrality» de MATLAB. Al igual que en la red multiplex, se ha de multiplicar el valor obtenido por 2, ya que al emplear como entrada una red no dirigida sólo tiene en cuenta los enlaces en una dirección, pero no en la recíproca.
- **Norm_Betweenness_Centrality_Wei:** centralidad de intermediación normalizada o centralidad de intermediación en una red ponderada (23).

Como los valores de pesos se encuentran entre 0 y 1, es necesario invertirlos para calcular la matriz de distancias. Si esta operación no se da, los valores de distancias podrán ser valores decimales o superiores a 1, por lo tanto, al calcular las métricas de eficiencia, en donde es necesario sumar las inversas de las distancias, se verían distorsionadas por los valores superiores a 1 obtenidos tras invertir aquellos números decimales que pudiesen encontrarse previamente en la matriz de distancias.

Con el objetivo de mantener la coherencia entre las métricas, se usan las distancias obtenidas de la matriz de pesos invertida para calcular la longitud de trayectoria y la centralidad de intermediación. La justificación tras esto es que, cuanto más «débil» sea el enlace entre dos satélites, a más «distancia» se encontrarán el uno del otro. Los coeficientes de agrupamiento pueden calcularse empleando la matriz de pesos original.

Una vez se conocen estos detalles, el cálculo de las métricas se realiza de igual modo que en la estructura «Sat_Network», con la diferencia de que, en este caso, las métricas son las de una red ponderada y no se calculan para distintos instantes. El sistema de selección del satélite más crítico sigue el mismo procedimiento que en la estructura «Sat_Multiplex». A su vez, al principio del bucle inicial «for» se encuentra otro bucle en donde se eliminan las filas y columnas pertinentes de la matriz de pesos, de forma similar al procedimiento realizado con la matriz de adyacencia en la estructura «Sat_Network».

3.2.3.5. Lista de satélites eliminados y nivel de coincidencia en cada iteración

Disponiendo de los datos de las dos redes, multiplex y monoplex, se pueden comparar cuántos satélites eliminados coinciden tras cada iteración del algoritmo de búsqueda del nodo más crítico. Como las redes son de distinto tipo, no tendría sentido comparar sus métricas directamente, aunque sí que se puede comparar si ambos métodos dan resultados significativamente parecidos en cuanto al orden de eliminación de satélites.

Para ello se guardan las variables «Missed_Satellite» de las estructuras «Sat_Multiplex» y «Sat_Monoplex» en dos vectores columna. Tras esto se puede comprobar los satélites coincidentes mediante el siguiente código:

```

% Level of coincidence
Coincidence = zeros(it_max-1,1);
for i=1:it_max-1
    for j=1:i
        for k=1:i
            Coincidence(i) = Coincidence(i) + isequal(Missed_Satellites_Monoplex(j),
                Missed_Satellites_Multiplex(k));
        end
    end
end
end

```

Código 21. Análisis de vulnerabilidades de la red de satélites: comparación de los satélites eliminados por ambos métodos (PARTE III, sección 2.1, líneas 628-636)

Si se quiere saber la proporción de coincidencia normalizada entre 0 y 1 bastará con dividir el número de satélites coincidentes de la fila entre el número de fila. No puede haber más satélites coincidentes que el número de satélites eliminado en esa iteración.

Para poder extraer conclusiones estadísticas de la comparación será necesario realizar múltiples repeticiones del experimento; esto se muestra en la siguiente sección. Aun así, con las listas de satélites eliminados se puede comprobar si la lista obtenida de un método da mejores resultados en el otro y viceversa.

3.2.3.6. Intercambio de los satélites eliminados por ambos métodos

Esta sección del código permite comprobar si alguno de los dos métodos ofrece mejores resultados cuando se emplea su lista de orden de eliminación de satélites en el otro método. Para ello se vuelven a calcular las estructuras «Sat_Network», «Sat_Multiplex» y «Sat_Monoplex» a partir de la segunda iteración. En lugar de determinar los satélites más vulnerables en cada una, se van eliminando según el orden de aparición en las listas «Missed_Satellites_Multiplex» y «Missed_Satellites_Monoplex» mediante el código:

```

% Changes in NORAD names list
s = find(str2double(NORAD_Network_Rec)==Missed_Satellites_Monoplex(it-1));
Sat_Rec_Multiplex(it).Missed_Satellite = NORAD_Network_Rec(s,:); % Missed satellite
NORAD_Network_Rec(s,:) = [];

% Changes in indexes
n_Sat_Network = n_Sat_Network - 1;

% Changes in adjacency matrix
for i=1:n_d_grid_Network
    Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix = Sat_Rec_Network(i).Iteration(it-1).
        Adjacency_Matrix;
    Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
    Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
end

```

Código 22. Análisis de vulnerabilidades de la red de satélites: modificación de la matriz de adyacencia en la nueva red multiplex (Sat_Rec_Multiplex) (PARTE III, sección 2.1, líneas 649-662)

Para la red monoplex se realiza la misma acción de forma análoga, empleando la lista «Missed_Satellites_Multiplex» y editando la matriz de pesos. Las estructuras en las que se

guarda la nueva información se denominan «Sat_Rec_Network», «Sat_Rec_Multiplex» y «Sat_Rec_Monoplex».

Finalmente, el programa almacena en distintas variables la eficiencia de las redes para cada una de las iteraciones. También permite observar de forma gráfica su evolución, así como la diferencia entre los métodos originales y sus versiones recalculadas.

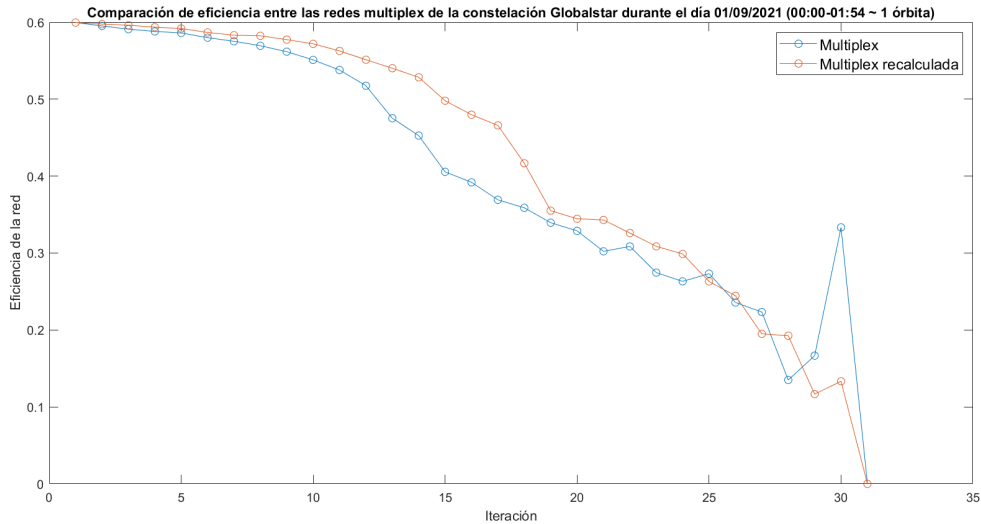


Figura 29. Comparación de eficiencia entre las redes multiplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

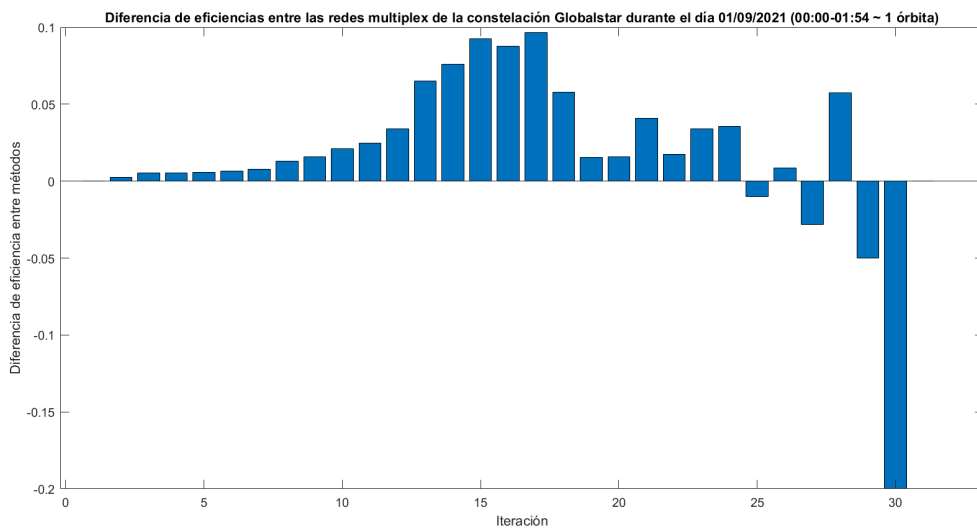


Figura 30. Diferencia de eficiencias entre las redes multiplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

En este caso, puede observarse como el método original es más efectivo hasta haber realizado el 75 % de las eliminaciones. Es necesario tener en cuenta que, a partir de cierto punto, pueden producirse escisiones en la red capaces de afectar notablemente a las métricas. En este ejemplo, la primera escisión se ha producido en la décima iteración, pudiendo detectarse al localizar la iteración en donde la longitud de trayectoria característica media tiende a infinito. Sin embargo,

estas escisiones no se dan a lo largo de todo el intervalo; para el instante de las 00:18, no se generan satélites o grupos de satélites incomunicados hasta la decimosexta eliminación. En la red recalculada, la primera escisión no se produce hasta la decimoquinta iteración. Esto puede explicar la relativa estabilidad que presenta su eficiencia hasta ese punto.

Fields	Satellites	Av_n_Edges	Std_n_Edges	Av_Characteristic_Path_Length	Std_Characteristic_Path_Length	Av_Efficiency	Std_Efficiency
1	1x32 struct	143.2500	7.7723	1.9822	0.0599	0.5996	0.0146
2	1x31 struct	132.7500	7.3690	2.0030	0.0640	0.5952	0.0151
3	1x30 struct	122.8000	6.7325	2.0232	0.0630	0.5910	0.0147
4	1x29 struct	114.1500	6.1153	2.0406	0.0640	0.5881	0.0146
5	1x28 struct	107.3500	5.7790	2.0646	0.0819	0.5861	0.0163
6	1x27 struct	99.0500	4.9255	2.1074	0.0991	0.5800	0.0173
7	1x26 struct	92.9500	4.4895	2.1680	0.1373	0.5752	0.0193
8	1x25 struct	85.3000	4.4141	2.2123	0.1690	0.5695	0.0217
9	1x24 struct	79	4.4839	2.3034	0.2710	0.5618	0.0283
10	1x23 struct	73.3000	4.9215	Inf	NaN	0.5511	0.0455
11	1x22 struct	66.9500	5.0729	Inf	NaN	0.5379	0.0605

Figura 31. Estructura Sat_Multiplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Fields	Adjacency_Matrix	Satellites	Network	Distances	n_Edges	Characteristic_Path_Length	Efficiency	Global_Clustering_Coefficient
1	32x32 double	1x32 struct	1x1 graph	32x32 dou...	150	1.9516	0.6089	0.5779
2	31x31 double	1x31 struct	1x1 graph	31x31 dou...	139	1.9785	0.6032	0.5867
3	30x30 double	1x30 struct	1x1 graph	30x30 dou...	129	2.0046	0.5981	0.5983
4	29x29 double	1x29 struct	1x1 graph	29x29 dou...	119	2.0320	0.5932	0.6009
5	28x28 double	1x28 struct	1x1 graph	28x28 dou...	112	2.0529	0.5915	0.6227
6	27x27 double	1x27 struct	1x1 graph	27x27 dou...	102	2.0798	0.5852	0.6236
7	26x26 double	1x26 struct	1x1 graph	26x26 dou...	97	2.1538	0.5797	0.6555
8	25x25 double	1x25 struct	1x1 graph	25x25 dou...	91	2.1633	0.5807	0.6788
9	24x24 double	1x24 struct	1x1 graph	24x24 dou...	85	2.2500	0.5739	0.7019
10	23x23 double	1x23 struct	1x1 graph	23x23 dou...	81	2.3281	0.5704	0.7310
11	22x22 double	1x22 struct	1x1 graph	22x22 dou...	74	2.3550	0.5673	0.7266
12	21x21 double	1x21 struct	1x1 graph	21x21 dou...	70	2.2810	0.5802	0.7326
13	20x20 double	1x20 struct	1x1 graph	20x20 dou...	63	2.3263	0.5741	0.7305
14	19x19 double	1x19 struct	1x1 graph	19x19 dou...	58	2.3743	0.5716	0.7361
15	18x18 double	1x18 struct	1x1 graph	18x18 dou...	51	2.9216	0.5322	0.8008
16	17x17 double	1x17 struct	1x1 graph	17x17 dou...	48	Inf	0.4118	0.8989
17	16x16 double	1x16 struct	1x1 graph	16x16 dou...	42	Inf	0.4083	0.9060

Figura 32. Estructura Sat_Network para la constelación Globalstar durante el día 01/09/2021 (00:18)

Fields	Satellites	Av_n_Edges	Std_n_Edges	Av_Characteristic_Path_Length	Std_Characteristic_Path_Length	Av_Efficiency	Std_Efficiency
1	1x32 struct	143.2500	7.7723	1.9822	0.0599	0.5996	0.0146
2	1x31 struct	132.9500	7.6191	1.9892	0.0630	0.5975	0.0153
3	1x30 struct	123.5500	7.4796	1.9937	0.0651	0.5962	0.0160
4	1x29 struct	114.7000	6.9214	2.0074	0.0670	0.5936	0.0162
5	1x28 struct	106.7500	6.5845	2.0189	0.0714	0.5919	0.0169
6	1x27 struct	97.7500	5.9813	2.0437	0.0713	0.5867	0.0165
7	1x26 struct	90.9500	5.4529	2.0751	0.0874	0.5831	0.0176
8	1x25 struct	84.2500	5.0039	2.0842	0.0764	0.5824	0.0160
9	1x24 struct	76.4500	4.2978	2.1089	0.0716	0.5775	0.0148
10	1x23 struct	69.5500	4.1609	2.1466	0.0859	0.5719	0.0165
11	1x22 struct	62.6000	3.3779	2.2074	0.1092	0.5628	0.0174
12	1x21 struct	56.9000	3.1772	2.3288	0.1829	0.5513	0.0221
13	1x20 struct	51	2.8098	2.4416	0.2506	0.5402	0.0266
14	1x19 struct	45.5000	2.5026	2.5626	0.2962	0.5285	0.0278
15	1x18 struct	40.4000	2.4149	Inf	NaN	0.4980	0.0584
16	1x17 struct	36.0500	2.9105	Inf	NaN	0.4798	0.0718

Figura 33. Estructura Sat_Rec_Multiplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Los valores de eficiencia son capaces de incrementar, especialmente en las últimas iteraciones, debido a la eliminación de aquellos satélites con una eficiencia menor a la media aunque con los valores más altos de centralidad de intermediación. Esto permite una mayor probabilidad de obtener escisiones en la red y un mayor descenso de la eficiencia en el largo plazo a cambio de una disminución de la eficiencia menos acusada entre ciertas iteraciones.

El modelo de redes monoplex se comporta de forma similar aunque cuenta con algunas diferencias notables.

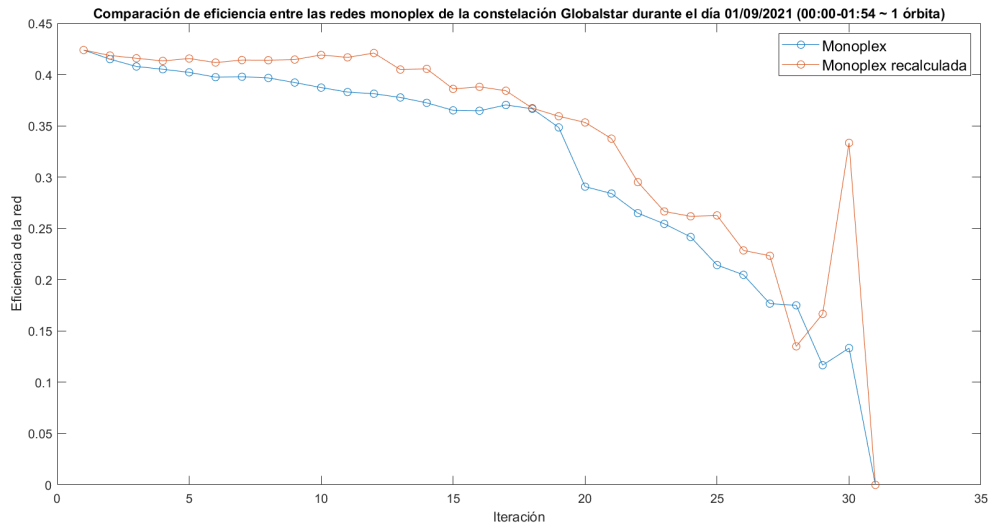


Figura 34. Comparación de eficiencia entre las redes monoplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

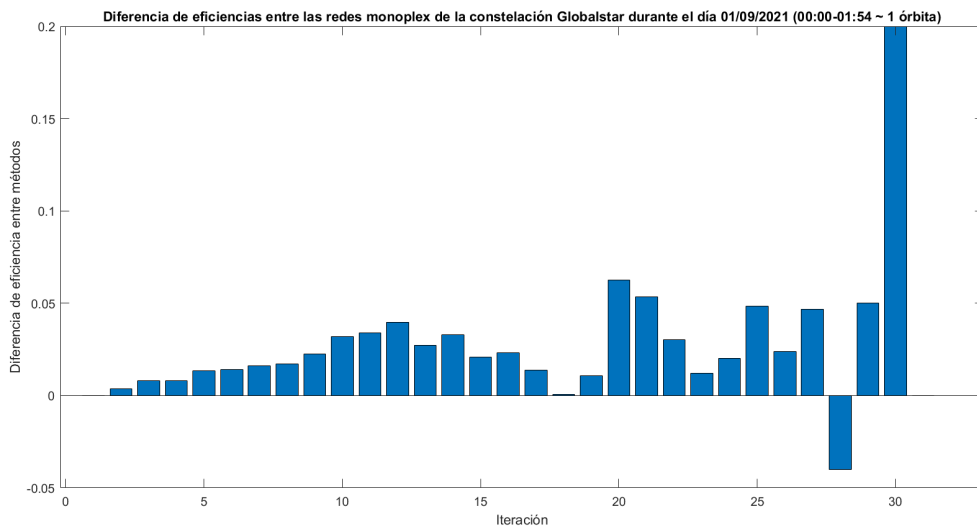


Figura 35. Diferencia de eficiencias entre las redes monoplex de la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

La primera diferencia son las distintas escalas entre los valores de eficiencia. Se puede comprobar como las redes multiplex y monoplex no parten de valores iniciales iguales. Por ello,

no es recomendable comparar directamente ambos tipos de redes, ya que realmente no se da una equivalencia entre las redes binarias y las ponderadas.

Otra diferencia se da en el momento a partir del que se producen las primeras escisiones en la red. En las redes monoplex, estas se dan a partir de la vigésima iteración, para la red original, y a partir de la vigesimosegunda, para la recalculada. El motiva tras esto es que, al tratarse de una red ponderada, no se tienen en cuenta las escisiones que pueden aparecer y desaparecer a lo largo del intervalo, sólo aquellas dadas de forma continua. Es a partir de estas escisiones en donde la eficiencia empieza a disminuir de forma abrupta, al no existir enlaces a través de los que alcanzar determinados satélites aislados.

Fields	Weight_Matrix	Satellites	Network	Network_Inv	Distances	Distances_Inv	n_Edges	Characteristic_Path_Length_Wei	Efficiency_Wei
1	32x32 double	1x32 struct	1x1 graph	1x1 graph	32x32 dou...	32x32 double	257	2.9042	0.4239
2	31x31 double	1x31 struct	1x1 graph	1x1 graph	31x31 dou...	31x31 double	242	2.9716	0.4150
3	30x30 double	1x30 struct	1x1 graph	1x1 graph	30x30 dou...	30x30 double	227	3.0210	0.4078
4	29x29 double	1x29 struct	1x1 graph	1x1 graph	29x29 dou...	29x29 double	211	3.0534	0.4051
5	28x28 double	1x28 struct	1x1 graph	1x1 graph	28x28 dou...	28x28 double	197	3.0887	0.4021
6	27x27 double	1x27 struct	1x1 graph	1x1 graph	27x27 dou...	27x27 double	181	3.1320	0.3974
7	26x26 double	1x26 struct	1x1 graph	1x1 graph	26x26 dou...	26x26 double	167	3.1690	0.3979
8	25x25 double	1x25 struct	1x1 graph	1x1 graph	25x25 dou...	25x25 double	155	3.1938	0.3967
9	24x24 double	1x24 struct	1x1 graph	1x1 graph	24x24 dou...	24x24 double	141	3.2445	0.3922
10	23x23 double	1x23 struct	1x1 graph	1x1 graph	23x23 dou...	23x23 double	129	3.3167	0.3873
11	22x22 double	1x22 struct	1x1 graph	1x1 graph	22x22 dou...	22x22 double	116	3.3826	0.3829
12	21x21 double	1x21 struct	1x1 graph	1x1 graph	21x21 dou...	21x21 double	104	3.4713	0.3813
13	20x20 double	1x20 struct	1x1 graph	1x1 graph	20x20 dou...	20x20 double	93	3.5522	0.3777
14	19x19 double	1x19 struct	1x1 graph	1x1 graph	19x19 dou...	19x19 double	83	3.6386	0.3724
15	18x18 double	1x18 struct	1x1 graph	1x1 graph	18x18 dou...	18x18 double	73	3.8440	0.3651
16	17x17 double	1x17 struct	1x1 graph	1x1 graph	17x17 dou...	17x17 double	65	3.8841	0.3647
17	16x16 double	1x16 struct	1x1 graph	1x1 graph	16x16 dou...	16x16 double	58	3.9083	0.3703
18	15x15 double	1x15 struct	1x1 graph	1x1 graph	15x15 dou...	15x15 double	49	4.0280	0.3666
19	14x14 double	1x14 struct	1x1 graph	1x1 graph	14x14 dou...	14x14 double	41	5.2153	0.3485
20	13x13 double	1x13 struct	1x1 graph	1x1 graph	13x13 dou...	13x13 double	36	Inf	0.2907
21	12x12 double	1x12 struct	1x1 graph	1x1 graph	12x12 dou...	12x12 double	31	Inf	0.2840

Figura 36. Estructura Sat_Monoplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Fields	Weight_Matrix	Satellites	Network	Network_Inv	Distances	Distances_Inv	n_Edges	Characteristic_Path_Length_Wei	Efficiency_Wei
1	32x32 double	1x32 struct	1x1 graph	1x1 graph	32x32 dou...	32x32 double	257	2.9042	0.4239
2	31x31 double	1x31 struct	1x1 graph	1x1 graph	31x31 dou...	31x31 double	239	2.9586	0.4186
3	30x30 double	1x30 struct	1x1 graph	1x1 graph	30x30 dou...	30x30 double	222	2.9780	0.4159
4	29x29 double	1x29 struct	1x1 graph	1x1 graph	29x29 dou...	29x29 double	206	3.0131	0.4133
5	28x28 double	1x28 struct	1x1 graph	1x1 graph	28x28 dou...	28x28 double	192	3.0267	0.4156
6	27x27 double	1x27 struct	1x1 graph	1x1 graph	27x27 dou...	27x27 double	177	3.0832	0.4117
7	26x26 double	1x26 struct	1x1 graph	1x1 graph	26x26 dou...	26x26 double	164	3.1079	0.4141
8	25x25 double	1x25 struct	1x1 graph	1x1 graph	25x25 dou...	25x25 double	149	3.1331	0.4140
9	24x24 double	1x24 struct	1x1 graph	1x1 graph	24x24 dou...	24x24 double	136	3.1832	0.4146
10	23x23 double	1x23 struct	1x1 graph	1x1 graph	23x23 dou...	23x23 double	123	3.1956	0.4191
11	22x22 double	1x22 struct	1x1 graph	1x1 graph	22x22 dou...	22x22 double	112	3.2496	0.4167
12	21x21 double	1x21 struct	1x1 graph	1x1 graph	21x21 dou...	21x21 double	100	3.3297	0.4210
13	20x20 double	1x20 struct	1x1 graph	1x1 graph	20x20 dou...	20x20 double	90	3.5848	0.4048
14	19x19 double	1x19 struct	1x1 graph	1x1 graph	19x19 dou...	19x19 double	79	3.6168	0.4055
15	18x18 double	1x18 struct	1x1 graph	1x1 graph	18x18 dou...	18x18 double	70	4.3117	0.3860
16	17x17 double	1x17 struct	1x1 graph	1x1 graph	17x17 dou...	17x17 double	62	4.5182	0.3880
17	16x16 double	1x16 struct	1x1 graph	1x1 graph	16x16 dou...	16x16 double	54	4.7286	0.3842
18	15x15 double	1x15 struct	1x1 graph	1x1 graph	15x15 dou...	15x15 double	46	5.2883	0.3672
19	14x14 double	1x14 struct	1x1 graph	1x1 graph	14x14 dou...	14x14 double	39	5.5428	0.3594
20	13x13 double	1x13 struct	1x1 graph	1x1 graph	13x13 dou...	13x13 double	34	5.4820	0.3533
21	12x12 double	1x12 struct	1x1 graph	1x1 graph	12x12 dou...	12x12 double	28	5.5224	0.3375
22	11x11 double	1x11 struct	1x1 graph	1x1 graph	11x11 dou...	11x11 double	23	Inf	0.2951

Figura 37. Estructura Sat_Rec_Monoplex para la constelación Globalstar durante el día 01/09/2021 (00:00-01:54 \cong 1 órbita)

Finalmente, en ambos métodos puede comprobarse como las versiones recalculadas no ofrecen resultados mejores, salvo en las últimas iteraciones, en donde estos pueden producirse de forma casual por el fenómeno ya comentado acerca de la eficiencia individual de cada satélite.

3.2.3.7. Representación gráfica de la red

Por último, el programa cuenta con una sección destinada a generar distintos grafos con el objetivo de representar la red. Para utilizarla con las redes multiplex es necesario indicar el número de iteración y el índice del instante temporal a representar; para las red monoplex, sólo hace falta el número de iteración y el programa ya la representa para todo el intervalo.

Los grafos representan los enlaces mediante líneas y la centralidad de intermediación de cada satélite, mediante una escala de color. El grafo de fuerzas sitúa los nodos más cercanos en grupos; el circular sitúa los nodos en un círculo; el de capas permite visualizar con facilidad las relaciones entre nodos. Finalmente, el gráfico tridimensional permite observa las posiciones y enlaces en el espacio; este último gráfico no existe para redes monoplex ya que los satélites se desplazan a lo largo del intervalo.

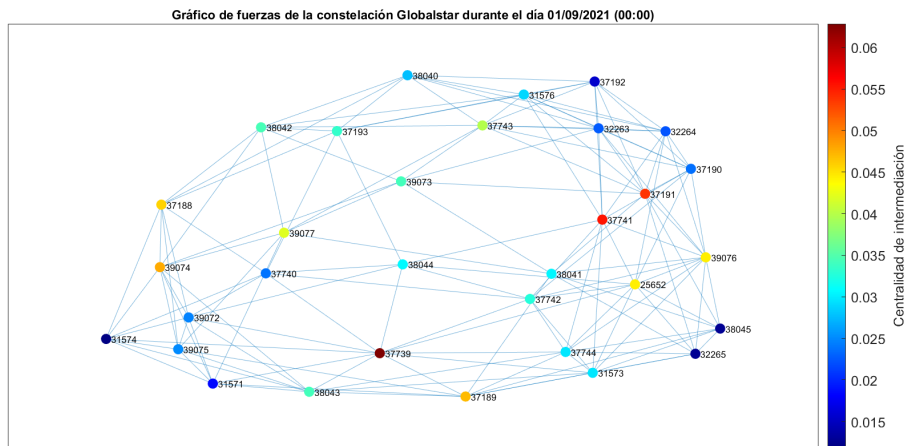


Figura 38. Grafo de fuerzas de la constelación Globalstar durante el día 01/09/2021 (00:00)

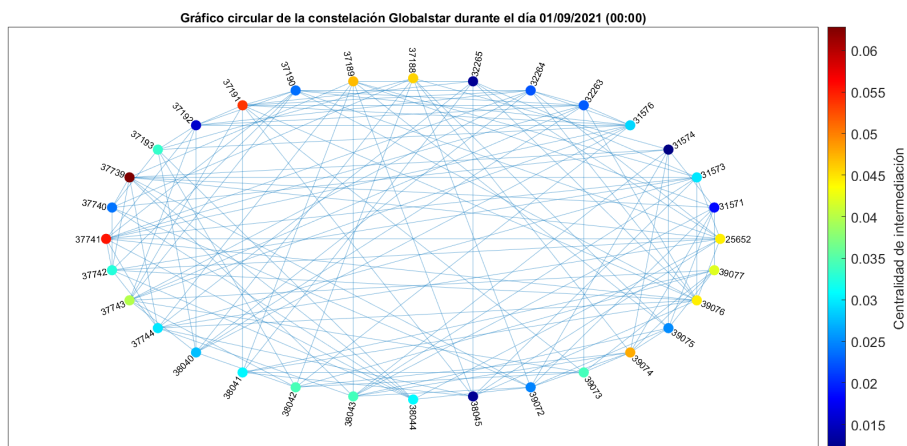


Figura 39. Grafo circular de la constelación Globalstar durante el día 01/09/2021 (00:00)

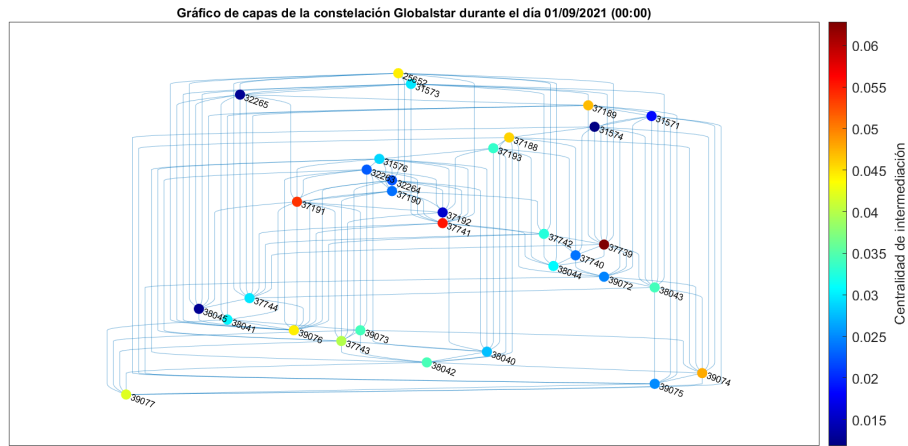


Figura 40. Grafo de capas de la constelación Globalstar durante el día 01/09/2021 (00:00)

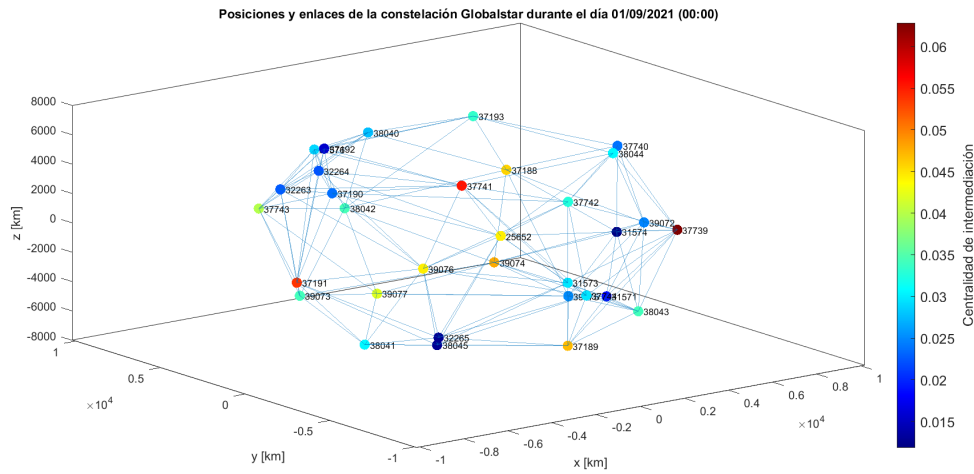


Figura 41. Posiciones y enlaces de la constelación Globalstar durante el día 01/09/2021 (00:00)

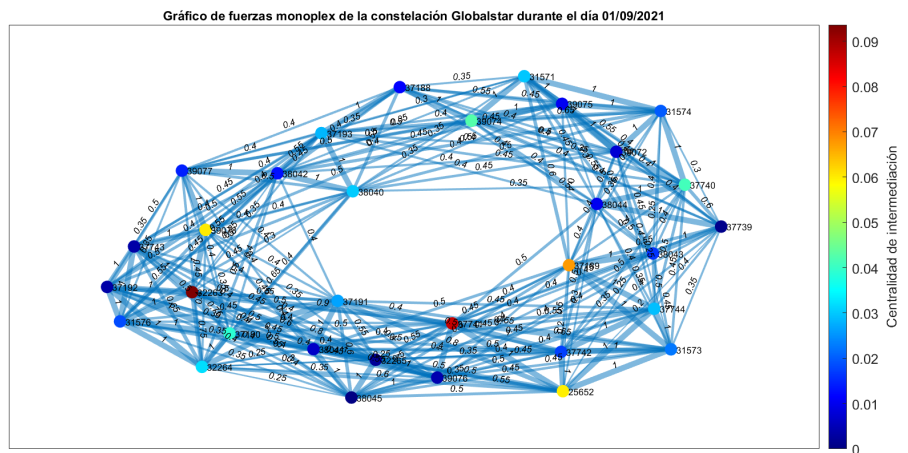


Figura 42. Grafo de fuerzas de la constelación Globalstar durante el día 01/09/2021

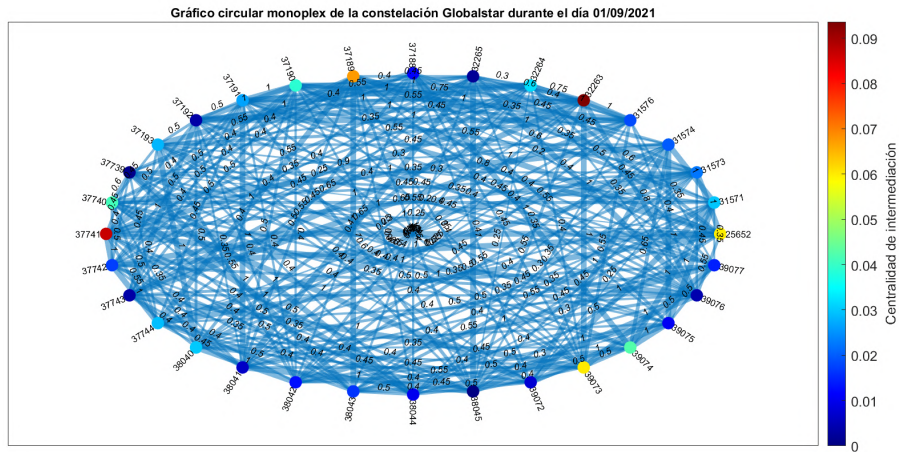


Figura 43. Grafo circular de la constelación Globalstar durante el día 01/09/2021

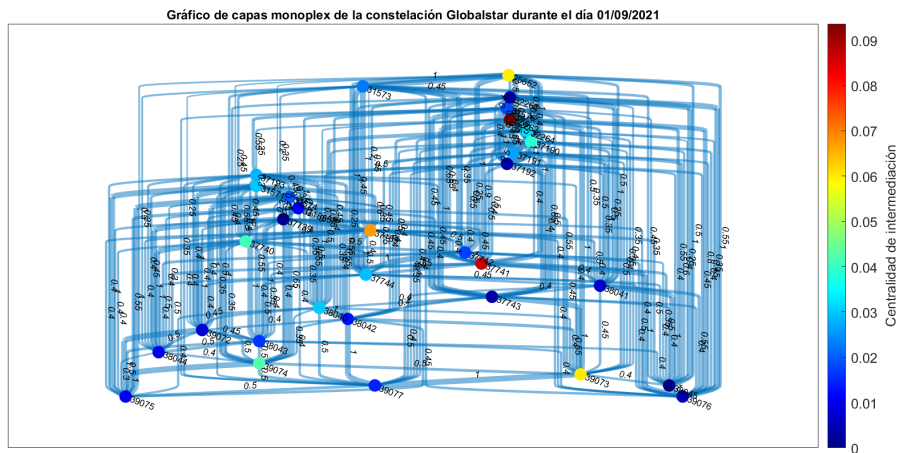


Figura 44. Grafo de capas de la constelación Globalstar durante el día 01/09/2021

Estos grafos son útiles para disponer de forma conjunta de la información sobre los enlaces y la centralidad. La principal ventaja que ofrecen los de tipo multiplex frente a los monoplex es su sencillez, pese a esto, los segundos presentan información relativa a todo el intervalo de tiempo de forma conjunta, por lo que serían más adecuados para formar una primera impresión de la red en caso de que el método fuera válido.

3.3. Comparación de los métodos multiplex y monoplex

El procedimiento para comparar ambos métodos se basa en evaluar la función de probabilidad aleatoria de coincidencia de satélites y compararla con las proporciones de coincidencia reales tras realizar varios experimentos. Otra de las funciones aportadas consiste en la visualización de la evolución de la eficiencia de cada uno de los métodos, así como sus diferencias entre los métodos originales y los recalculados.

Respecto a la diferencia de eficiencias, se ha decidido mostrar sólo esta, así como los valores individuales de eficiencia de cada método, ya que cada experimento podría presentar valores

en escalas distintas de eficiencia absoluta, pudiendo no tener sentido la comparación entre las mismas si provienen de distintos métodos. La diferencia de eficiencias es la métrica necesaria para comprobar si alguno de los métodos recalculados ofrece mejores resultados que el original al realizar múltiples experimentos, ya que todas las eficiencias se encuentran entre 0 y 1.

Respecto a la comparación de coincidencias, la función de probabilidad aleatoria representa la probabilidad de coincidencia al sacar dos elementos de dos grupos de elementos iguales e independientes. Se desea obtener la probabilidad P de que aparezca un número n de parejas cuando se ha sacado un número it de elementos de cada grupo de s elementos totales. s representa el número total de satélites; it , la iteración del programa anterior o el número de satélites eliminados; n , el número de coincidencias. Para cada iteración, el número máximo de coincidencias alcanzables es el número de satélites eliminados.

Para facilitar la comprensión de esta función, se ha decidido realizar una analogía utilizando un ejemplo en el que se emplean barajas de cartas. Se dispondría de dos barajas distintas y separadas, ambas con el mismo número de cartas. En cada iteración se saca, de forma aleatoria, una carta de cada baraja y se añade a un montón central: en la primera iteración se saca una carta de cada baraja, en la segunda iteración se vuelve a sacar otra carta de cada baraja, quedando cuatro cartas en el montón central; se va realizando este proceso sucesivamente hasta que no queden cartas en las barajas. En cada iteración it habría un número it de parejas en el montón central, de las cuales podrían coincidir un número n de parejas, siempre inferior al número de parejas totales it del montón. Se quiere obtener la probabilidad de aparición de cada número n de parejas para cada iteración it del proceso. La función de probabilidad no es convencional ya que la probabilidad de obtener parejas iguales en el montón va aumentando a medida que se reduce el número de cartas en cada baraja y aumenta el número de cartas del montón.

Para entender la analogía, una baraja representaría al método multiplex y la otra, al monoplex. Las cartas sacadas en cada iteración representarían el identificador del satélite eliminado según cada método en esa iteración. Las cartas del montón son los satélites eliminados por ambos métodos hasta ese momento, contando la eliminación de la última iteración. Para calcular la función de probabilidad es necesario dividir el número de casos favorables entre el número de casos posibles.

■ **Casos posibles:**

$$\binom{s}{it} \binom{s}{it} \quad (29)$$

■ **Casos favorables:**

$$\binom{s}{n} \binom{s-n}{it-n} \binom{s-n-(it-n)}{it-n} \quad (30)$$

El número de casos posibles viene representado por todas las combinaciones posibles, en cada ronda it , al sacar cartas de uno de los montones con s cartas, multiplicado por otro elemento de s sobre it , ya que así se representarían la otra baraja. La multiplicación representa el número total de posibilidades al juntar el efecto de las dos barajas.

Para el número de casos favorables hay que tener en cuenta que, en la iteración it , habrá salido un número it de cartas de cada baraja. Estas quedan divididas en n cartas coincidentes e $it - n$ cartas no coincidentes de cada baraja; es decir: n parejas coincidentes e $it - n$ parejas no coincidentes. Se desea obtener el número de casos en los que n parejas son coincidentes.

El número de casos favorables queda representado por tres términos multiplicados. El primero representa todas las combinaciones posibles de las n parejas coincidentes para el total de las s cartas de cada baraja. El segundo representa la forma de sacar, de los $s - n$ elementos que quedan en la primera baraja, $it - n$ elementos, es decir, los elementos no coincidentes de esa iteración. El tercero representa la forma de sacar, de los elementos restantes de la segunda baraja (s cartas de la baraja, menos n elementos coincidentes en esa iteración, menos los elementos no coincidentes $it - n$), los elementos no coincidentes $it - n$ de la iteración. La multiplicación de los términos muestra la probabilidad conjunta de los tres sucesos.

La probabilidad de obtener n parejas coincidentes en la iteración it para dos grupos iguales e independientes de s satélites presenta la siguiente función⁷:

$$P(n, it) = \frac{\binom{s}{n} \binom{s-n}{it-n} \binom{s-it}{it-n}}{\binom{s}{it} \binom{s}{it}} \quad (31)$$

Disponiendo de la distribución de probabilidad para cada una de las iteraciones, se compara la misma con la muestra de los experimentos realizados mediante el primer programa presentado en esta sección. Para ello se emplea la prueba χ^2 de Pearson para dos muestras [58]. Esta permite la comprobación de la bondad del ajuste entre dos muestras de datos, provenientes de distribuciones discretas, con el objetivo de rechazar o no la hipótesis nula de que ambas siguen la misma distribución de probabilidad para un nivel de significación α determinado, generalmente del 5 %.

3.3.1. Toma de datos

La toma de datos se realiza de la misma forma que en el apartado de la sección anterior 3.2.1 Recursos empleados para el análisis de vulnerabilidades de la red de satélites. En este caso, el intervalo de tiempo ha de abarcar un espacio suficientemente amplio como para contener en su interior las fechas de todos los intervalos para los que se va a repetir el experimento presentado en la sección anterior.

3.3.2. Recursos empleados

Los recursos empleados son los mismos que en el apartado de la sección anterior 3.2.2 Recursos empleados para el análisis de vulnerabilidades de la red de satélites. El primero de los programas es una versión modificada del programa para el análisis de vulnerabilidades de la red de satélites.

A su vez, para la comparación entre los distintos métodos de análisis de redes se emplea una función para obtener una muestra de datos discretos a partir de una distribución de probabilidad. El programa escogido es la función «discretesample.m» de la librería de Dahua Lin [59]. En ella se introduce un vector con la distribución de probabilidad, en donde sus elementos han de sumar 1, y el número de datos a obtener. A la salida se obtiene un vector con el número de elementos determinado a la entrada en el que todos ellos siguen la distribución indicada.

Finalmente, ya que MATLAB no dispone de ninguna función con la que comparar dos muestras de distribuciones discretas no determinadas, se emplea la función «chi2test2.m» de la librería de James Meldrum [60]. Esta recibe dos muestras de datos distintas y el nivel de significación α requerido. En la salida se obtiene un valor binario, en el que 0 indica que no se

⁷Esta función fue planteada por Miguel Solana Cires, alumno de la Facultad de Ciencias Matemáticas de la Universidad Complutense de Madrid.

puede rechazar la hipótesis nula y 1, que esta es rechazada para el nivel de significación indicado. La hipótesis nula indica que ambas muestras provienen de la misma distribución. Otra de las variables obtenidas a la salida es el p-valor de la prueba [61].

3.3.3. Algoritmo empleado

Para la comparación de los métodos de análisis se emplean tres programas diferentes.

El primero de ellos, «Análisis.m», tiene el objetivo de repetir los experimentos de análisis realizados en la sección anterior, 3.2 Análisis de vulnerabilidades de la red de satélites, un número determinado de veces para distintos intervalos repartidos de forma regular en el tiempo y así poder obtener una muestra de datos. Con ello se pretende almacenar las métricas de eficiencia de cada uno de los métodos, así como las de sus versiones recalculadas, junto con el número de coincidencias de satélites tras cada iteración.

El segundo programa, «Probabilidad.m», tiene el objetivo de calcular la distribución de probabilidad de la ocurrencia de cierto número de coincidencias según el número de eliminaciones de satélites realizadas.

Finalmente, el último programa, «Comparacion.m», está diseñado para comparar la distribución de probabilidad aleatoria con los valores de los múltiples experimentos realizados. Con ello se pretende comprobar si las coincidencias de satélites eliminados son aleatorias o responden a algún tipo de distribución distinta. A su vez, se muestran los distintos valores de eficiencia y las diferencias existentes entre los métodos originales y los recalculados para verificar si alguna de las versiones ofrece resultados superiores.

3.3.3.1. Realización de múltiples experimentos (Análisis.m)

La misión de este programa es repetir iterativamente los análisis presentados en el programa de la sección anterior. A cada una de estas repeticiones se le ha denominado como un «experimento».

Al final de cada experimento, los datos sobre el número de satélites coincidentes en cada iteración, durante la comparación de los métodos multiplex y monoplex, son almacenados en una matriz «An_Coincidence»; los datos relativos a la eficiencia, así como a la diferencia de eficiencias entre los métodos originales y los recalculados, también son almacenados en las matrices «An_Efficiency_Multiplex», «An_Efficiency_Multiplex_Rec», «An_Efficiency_Multiplex_Dif», «An_Efficiency_Monoplex», «An_Efficiency_Monoplex_Rec» y «An_Efficiency_Monoplex_Dif».

Cada columna de las matrices representa los datos de un experimento, mientras que cada fila representa una iteración en la que se ha localizado el satélite más crítico y ha sido eliminado para calcular la nueva red siguiente.

Los bucles de iteración de cada uno de los métodos de análisis (multiplex, monoplex, multiplex recalculado y monoplex recalculado), así como algunas de las líneas de inicio y configuración del programa anterior, se encuentran contenidas por un bucle más amplio en donde el índice marca el número de experimento realizado. Los dos elementos añadidos respecto al programa anterior son el número máximo de experimentos a realizar y los parámetros de

tiempo para definir cada uno de los experimentos, además del almacenamiento de las métricas pertinentes.

Los parámetros de tiempo constan de una fecha inicial del intervalo general, marcada en cada una de sus variables por el sufijo 0, una duración de cada experimento, marcada por el sufijo d , y un periodo entre experimentos, marcado por el sufijo i . Al principio del bucle general se asigna la fecha inicial y final del experimento. Al final del bucle general se suma, a la fecha inicial, el valor del periodo entre experimentos. La primera fecha inicial es la fecha de inicio del intervalo general. Para conseguir la fecha final de cada experimento se le suma, a la fecha inicial del mismo, los valores de duración.

A la hora de ejecutar este programa, es necesario tener en cuenta que es el que posee un tiempo de ejecución más largo de todos los presentados en este Proyecto.

3.3.3.2. Probabilidad de coincidencia aleatoria entre satélites (Probabilidad.m)

Para ejecutar este programa solamente es necesario indicar el número de satélites totales. Tras esto, se realizan los cálculos explicados al comienzo de esta sección. La salida es una matriz en donde cada fila representa el número de coincidencias, siendo la primera fila la que representa la probabilidad de que no ocurra ninguna coincidencia; cada columna representa una iteración, es decir la eliminación de otro satélite.

Debido a la función «nchoosek» de MATLAB, empleada para calcular combinaciones, es necesario omitir aquellos casos en donde el número de elementos a agrupar sea superior al número de elementos totales.

```

%% Probability calculation loop
for it = 1:s
    for n = 1:it+1 % One additional initial row for representing 0 coincidences
        if it-(n-1) <= s-it % Only calculates combinations if the number of elements in
            each group is inferior to the total number of elements
            F_O(n,it) = nchoosek(s,(n-1))*nchoosek(s-(n-1),it-(n-1))*nchoosek(s-it,it-(n-1)
                );
        else
            F_O(n,it) = 0;
        end
        P_O(it) = nchoosek(s,it)*nchoosek(s,it);
        Probability(n,it) = F_O(n,it)/P_O(it);
    end
end
end

```

Código 23. Bucle para el cálculo de la probabilidad de coincidencia aleatoria entre satélites eliminados por cada uno de los métodos de análisis (PARTE III, sección 3.2, líneas 15-26)

3.3.3.3. Comparación de los métodos multiplex y monoplex (Comparacion.m)

Este es el programa que permite comparar la eficacia de cada uno de los métodos de análisis de redes. Para su uso es necesario disponer de los datos de coincidencias entre satélites eliminados por iteración y de los valores de eficiencia para varios experimentos realizados en distintos intervalos de tiempo; también es necesaria la distribución de probabilidad aleatoria de coincidencias entre satélites. Dichos datos son obtenidos mediante los dos programas comentados en las subsecciones previas, cargándose sus variables, en el espacio de trabajo, al principio de la ejecución de este programa.

Las únicas variables de entrada manual son dos índices inicial y final en relación al número de iteración. Estos sirven para seleccionar el intervalo de visualización de los distintos valores de eficiencia.

Una vez cargadas todas las variables necesarias, el programa calcula los valores medios de eficiencia para cada iteración a partir de los valores de cada experimento. Tras esto, se ajustan algunas de las variables eliminando filas y columnas. Los motivos para ello son la aparición de filas vacías en la última iteración, debido al cálculo de la eficiencia empleando solamente un satélite. Para el caso de la probabilidad aleatoria, se eliminan los datos de coincidencia para el número máximo de eliminaciones ya que será siempre 1.

```
% Elimination of last rows (empty)
An_Efficiency_Multiplex(n_it+1,:) = [];
An_Efficiency_Multiplex_Rec(n_it+1,:) = [];
An_Efficiency_Multiplex_Dif(n_it+1,:) = [];
An_Efficiency_Monoplex(n_it+1,:) = [];
An_Efficiency_Monoplex_Rec(n_it+1,:) = [];
An_Efficiency_Monoplex_Dif(n_it+1,:) = [];

% Elimination of last row and collumn (empty)
Probability(n_it+2,:) = [];
Probability(:,n_it+1) = [];
```

Código 24. Eliminación de filas y columnas innecesarias de las variables de entrada (PARTE III, sección 3.3, líneas 54-64)

Para la comparación entre métodos según el número de coincidencias de satélites eliminados se calculan los valores medios y las desviaciones estándar de la cantidad de coincidencias por cada iteración; estos datos se pueden revisar en los vectores «Av_Coincidence» y «Std_Coincidence».

También se realiza la prueba χ^2 de Pearson entre la distribución de probabilidad aleatoria y la muestra de los múltiples experimentos; esto se hace mediante la función «chi2test2». Como la misma sólo permite la comparación entre dos muestras, sin poder emplear una distribución de probabilidad, es necesario crear una muestra para la distribución aleatoria. Con el objetivo de mantener cierta coherencia y suponiendo un número de muestras suficiente en la realización de los experimentos anteriores, la muestra de la distribución aleatoria se genera empleando el mismo número de datos que los experimentos realizados. Para ello se emplea la función «discretesample».

```
% Pearson's chi-squared test
Probability_Sample = zeros(n_it,n_Experiments);
h = zeros(n_it,1);
p = zeros(n_it,1);
for i=1:n_it
    Probability_Sample(i,:) = discretesample(Probability(:,i),n_Experiments)-1;
    [h(i,1),p(i,1)] = chi2test2(An_Coincidence(i,:)',Probability_Sample(i,:)',0.05);
end
```

Código 25. Ejecución de la prueba χ^2 de Pearson entre la distribución de probabilidad aleatoria y la muestra de los múltiples experimentos (PARTE III, sección 3.3, líneas 68-75)

Es necesario tener en cuenta que la función de muestreo tiene en cuenta la probabilidad de ocurrencia de cada índice, por lo que el primer número es el 1 y no el 0, a diferencia de la

función de probabilidad aleatoria calculada en el programa anterior. Por ello, es necesario restar una unidad a cada número obtenido de la muestra.

Tras esto, se calculan las frecuencias de aparición de las coincidencias de la muestra experimental y se normalizan sus valores entre 0 y 1. Esto permite representarlas de forma conjunta a la distribución de probabilidad, mediante un histograma por cada iteración.

Finalmente, se calculan los valores medios de la distribución aleatoria y de la experimental. También se calcula un valor entre 0 y 1 que representa lo cerca que se encuentra la media de los valores experimentales de los valores máximos o de ser aleatoria en cada iteración. Los valores medios de las distribuciones son representados en una gráfica junto a los máximos valores posibles.

Respecto a la representación de la eficiencia, el programa muestra distintas gráficas para los métodos multiplex y monoplex; en ellas se representan todos los experimentos realizados. Para cada uno de los métodos se genera:

- Una gráfica con la evolución de los valores medios de eficiencia para cada iteración, comparando el método original con el recalculado.
- Un gráfico de caja para cada iteración representando la diferencia entre los valores de eficiencia entre el método original y el recalculado. Valores positivos indican que la eficiencia del método recalculado tras cada iteración es superior a la del método original, es decir, el método recalculado sería menos eficaz para determinar satélites más críticos.
- Un gráfico de caja para cada iteración representando la eficiencia del método original.
- Un gráfico de caja para cada iteración representando la eficiencia del método recalculado.
- Los dos elementos anteriores representados uno al lado del otro.
- Un gráfico dinámico en el que se representa la evolución de la eficiencia de cada uno de los experimentos empleando el método original. El gráfico permite seleccionar cada una de las curvas y ver sus valores en cada iteración de forma sencilla.
- Un gráfico dinámico en el que se representa la evolución de la eficiencia de cada uno de los experimentos empleando el método recalculado.

Capítulo 4

Análisis de resultados

EN el capítulo anterior se ha explicado el funcionamiento de los programas necesarios para obtener las posiciones de los satélites de una red a lo largo del tiempo, analizar sus satélites más críticos de forma iterativa y comprobar la validez de cada uno de los métodos para ello: multiplex y monoplex. En este capítulo se emplean dichos programas para dar respuesta a las siguientes cuestiones:

- **Validez de cada uno de los métodos:** comprobar si ambos métodos ofrecen resultados coherentes entre sí. Si ofreciesen los mismos resultados se daría el máximo nivel de coherencia, por lo que ambos serían válidos. Si los satélites coincidentes eliminados por cada método siguiesen una distribución aleatoria se daría el mínimo nivel coherencia, por lo que se podría descartar el valor de realizar análisis de vulnerabilidades a lo largo de un intervalo de tiempo en lugar de para instantes concretos.
- **Superioridad de alguno de los métodos sobre el otro:** si la lista con el orden de eliminación de los satélites según un método fuese aplicada en el otro y sus métricas ofreciesen mejores resultados que con los satélites de la lista original significaría que ese método es más eficaz que el original y que el original no está determinando los satélites más críticos de la forma más eficaz.
- **Usos preferentes de cada uno de los métodos:** cada uno puede presenta ventajas e inconvenientes que lo pueden hacer más adecuado para unas funciones u otras.

Con el objetivo de aportar validez estadística a los resultados se han realizado varios experimentos a lo largo del tiempo. Las condiciones de los mismos se explican en la siguiente sección.

4.1. Condiciones de experimentación

El primer paso consiste en determinar cuál ha de ser el intervalo para la realización de cada experimento. Como los satélites siguen un movimiento periódico alrededor de su órbita puede considerarse este periodo como el intervalo de cada experimento.

Pese a esto, los satélites no siguen la misma órbita de forma continua, rotando la misma alrededor del eje mayor de la elipse. A su vez, no todos los satélites de la constelación tienen el mismo periodo orbital, siendo ligeramente distinto entre ellos y pudiendo variar en función de las perturbaciones gravitacionales o desplazamientos intencionados de cada satélite. Para simplificar

los cálculos y no asumir una precisión inferior al minuto, el periodo orbital considerado es de 1 h y 54 min [33].

Por estos motivos, se puede escoger el periodo orbital como intervalo de experimentación, ya que no se producirán cambios significativos entre las conexiones de una órbita y las siguientes, pero sí se producirán cambios en el largo plazo, permitiendo la replicación de los experimentos para la obtención de distintas muestras aleatorias.

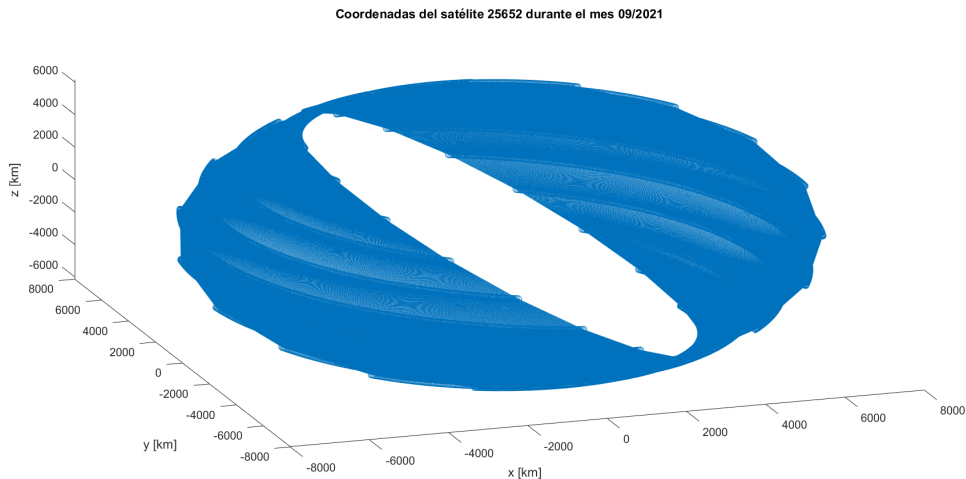


Figura 45. Coordenadas del satélite 25652 durante el mes 09/2021

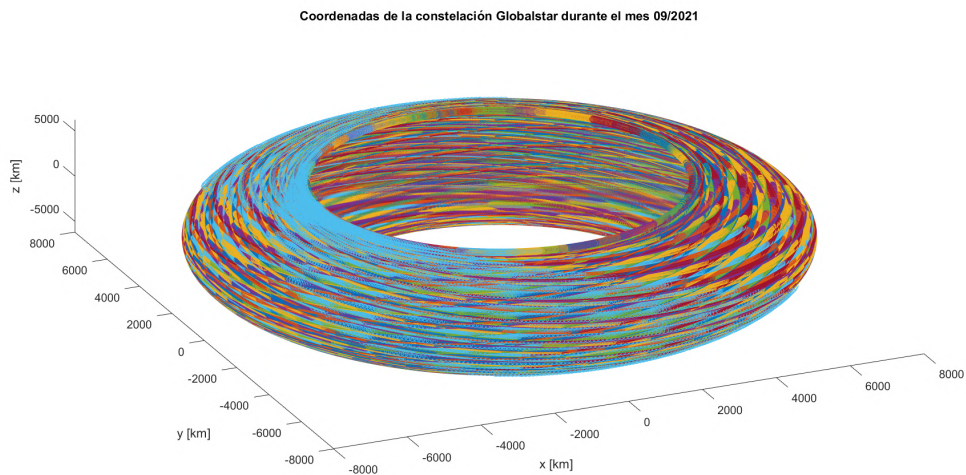


Figura 46. Coordenadas de la constelación Globalstar durante el mes 09/2021

En las imágenes puede comprobarse como el efecto de rotación de las órbitas hace que la constelación acabe cubriendo el total de la superficie terrestre, a excepción de los polos. De este modo pueden considerarse como distintas las órbitas a lo largo del mes y por lo tanto, también las redes que se forman entre satélites.

El siguiente paso consiste en elegir el periodo de muestreo de la posición de los satélites. El periodo escogido ha sido de 6 min ya que el periodo orbital es múltiplo de este número y de esta forma se facilita representar el intervalo del experimento lo más cercano a este sin perder

simplicidad. Por cada experimento se dispone de 19 medidas. Es necesario tener en cuenta que el tiempo de computación de todos los programas es directamente proporcional a este número, siendo seis veces más lento si se escogiera un periodo de 1 min.

Para aplicar la prueba χ^2 de Pearson es necesario disponer de una muestra con al menos 25 elementos, siendo el ideal a partir de 50. Ya que se dispone de los datos de un mes y, por mantener la coherencia, los experimentos se realizan a intervalos regulares, se ha decidido realizar 3 experimentos por día, obteniendo un total de 90 experimentos; los mismos se han de realizar cada 8 h.

Disponiendo de dichas especificaciones, se obtienen los datos de posiciones para todo el mes de septiembre y se ejecutan los programas para la comparación de los métodos multiplex y monoplex.

4.2. Resultados

Para poder evaluar la validez de los métodos se ha de comparar una muestra proveniente de la distribución de probabilidad aleatoria con la muestra de experimentos en relación al número de satélites eliminados coincidentes en cada iteración. Para ello se ha de calcular primero la función de probabilidad aleatoria para 32 eliminaciones mediante el programa detallado en la subsección 3.3.3.2 Probabilidad de coincidencia aleatoria entre satélites (Probabilidad.mat).

Las columnas representan cada iteración y las filas, el número de coincidencias, correspondiéndose la fila 1 con 0 coincidencias.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.9688	0.8770	0.7367	0.5694	0.4009	0.2541	0.1428	0.0699	0.0291	0.0100	0.0027	5.5790e-04	7.8106e-05	6.4908e-06	2.4040e-07	1.6637e-09
2	0.0313	0.1210	0.2456	0.3644	0.4358	0.4355	0.3683	0.2632	0.1573	0.0771	0.0301	0.0089	0.0019	2.5444e-04	1.8030e-05	4.2590e-07
3	0	0.0020	0.0175	0.0631	0.1453	0.2475	0.3315	0.3583	0.3147	0.2231	0.1253	0.0540	0.0170	0.0036	4.4174e-04	2.3957e-05
4	0	0	2.0161e-04	0.0031	0.0174	0.0574	0.1315	0.2263	0.3023	0.3172	0.2602	0.1637	0.0761	0.0246	0.0050	5.2173e-04
5	0	0	0	2.7809e-05	6.7039e-04	0.0054	0.0239	0.0707	0.1512	0.2429	0.2974	0.2762	0.1901	0.0929	0.0299	0.0055
6	0	0	0	0	4.9658e-06	1.7215e-04	0.0019	0.0108	0.0398	0.1029	0.1943	0.2719	0.2800	0.2065	0.1032	0.0317
7	0	0	0	0	0	1.1035e-06	5.1993e-05	7.3472e-04	0.0053	0.0238	0.0729	0.1586	0.2489	0.2787	0.2151	0.1067
8	0	0	0	0	0	0	2.9710e-07	1.8254e-05	3.2472e-04	0.0029	0.0153	0.0544	0.1340	0.2317	0.2765	0.2177
9	0	0	0	0	0	0	0	9.5072e-08	7.3800e-06	1.6113e-04	0.0017	0.0106	0.0431	0.1183	0.2212	0.2756
10	0	0	0	0	0	0	0	0	3.5652e-08	3.4102e-06	8.9518e-05	0.0011	0.0080	0.0364	0.1095	0.2177
11	0	0	0	0	0	0	0	0	0	1.5501e-08	1.7904e-06	5.5538e-05	7.9780e-04	0.0065	0.0328	0.1067
12	0	0	0	0	0	0	0	0	0	0	7.7505e-09	1.0629e-06	3.8397e-05	6.3004e-04	0.0057	0.0317
13	0	0	0	0	0	0	0	0	0	0	0	4.4288e-09	7.1105e-07	2.9533e-05	5.4691e-04	0.0055
14	0	0	0	0	0	0	0	0	0	0	0	0	2.8787e-09	5.3454e-07	2.5242e-05	5.2173e-04
15	0	0	0	0	0	0	0	0	0	0	0	0	0	2.1212e-09	4.5075e-07	2.3957e-05
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.7677e-09	4.2590e-07
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.6637e-09
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 47. Probabilidad de coincidencia aleatoria entre satélites eliminados para las iteraciones 1-16

Probability																
33x32 double																
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2.4040e-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1.8030e-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	4.4174e-04	6.4908e-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0.0050	2.5444e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0.0299	0.0036	7.8106e-05	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0.1032	0.0246	0.0019	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0.2151	0.0929	0.0170	5.5790e-04	0	0	0	0	0	0	0	0	0	0	0	0
10	0.2765	0.2065	0.0761	0.0089	0	0	0	0	0	0	0	0	0	0	0	0
11	0.2212	0.2787	0.1901	0.0540	0.0027	0	0	0	0	0	0	0	0	0	0	0
12	0.1095	0.2317	0.2800	0.1637	0.0301	0	0	0	0	0	0	0	0	0	0	0
13	0.0328	0.1183	0.2489	0.2762	0.1253	0.0100	0	0	0	0	0	0	0	0	0	0
14	0.0057	0.0364	0.1340	0.2719	0.2602	0.0771	0	0	0	0	0	0	0	0	0	0
15	5.4691e-04	0.0065	0.0431	0.1586	0.2974	0.2231	0.0291	0	0	0	0	0	0	0	0	0
16	2.5242e-05	6.3004e-04	0.0080	0.0544	0.1943	0.3172	0.1573	0	0	0	0	0	0	0	0	0
17	4.5075e-07	2.9533e-05	7.9780e-04	0.0106	0.0729	0.2429	0.3147	0.0699	0	0	0	0	0	0	0	0
18	1.7677e-09	5.3454e-07	3.8397e-05	0.0011	0.0153	0.1029	0.3023	0.2632	0	0	0	0	0	0	0	0
19	0	2.1212e-09	7.1105e-07	5.5538e-05	0.0017	0.0238	0.1512	0.3583	0.1428	0	0	0	0	0	0	0
20	0	0	2.8787e-09	1.0629e-06	8.9518e-05	0.0029	0.0398	0.2263	0.3683	0	0	0	0	0	0	0
21	0	0	0	4.4288e-09	1.7904e-06	1.6113e-04	0.0053	0.0707	0.3315	0.2541	0	0	0	0	0	0
22	0	0	0	0	7.7505e-09	3.4102e-06	3.2472e-04	0.0108	0.1315	0.4355	0	0	0	0	0	0
23	0	0	0	0	0	1.5501e-08	7.3800e-06	7.3472e-04	0.0239	0.2475	0.4009	0	0	0	0	0
24	0	0	0	0	0	0	3.5652e-08	1.8254e-05	0.0019	0.0574	0.4358	0	0	0	0	0
25	0	0	0	0	0	0	0	9.5072e-08	5.1993e-05	0.0054	0.1453	0.5694	0	0	0	0
26	0	0	0	0	0	0	0	0	2.9710e-07	1.7215e-04	0.0174	0.3644	0	0	0	0
27	0	0	0	0	0	0	0	0	0	1.1035e-06	6.7039e-04	0.0631	0.7367	0	0	0
28	0	0	0	0	0	0	0	0	0	0	4.9658e-06	0.0031	0.2456	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	2.7809e-05	0.0175	0.8770	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	2.0161e-04	0.1210	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0020	0.9688	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0313	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figura 48. Probabilidad de coincidencia aleatoria entre satélites eliminados para las iteraciones 17-32

Una vez se dispone de la distribución de probabilidad el programa presentado en la subsección 3.3.3.3 Comparación de los métodos multiplex y monoplex (Comparacion.mat) genera una muestra de 90 elementos de la misma. Dicha muestra se compara con la muestra de los experimentos mediante la prueba χ^2 de Pearson. También se representa la distribución aleatoria junto a la obtenida tras normalizar los resultados de los experimentos en cada iteración.

Coincidence																
32x31 double																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.7889	0.5333	0.2889	0.1222	0.0333	0	0	0	0	0	0	0	0	0	0	0
2	0.2111	0.4333	0.6222	0.6778	0.5778	0.3778	0.1111	0.0556	0.0111	0	0	0	0	0	0	0
3	0	0.0333	0.0889	0.1778	0.3222	0.3889	0.3333	0.2222	0.0889	0.0444	0	0	0	0	0	0
4	0	0	0	0.0222	0.0444	0.2000	0.4111	0.3222	0.2667	0.1000	0.0778	0.0333	0.0111	0.0111	0	0
5	0	0	0	0	0.0222	0.0333	0.1333	0.2889	0.2556	0.2556	0.1667	0.0333	0.0222	0	0.0111	0
6	0	0	0	0	0	0	0.0111	0.1000	0.2444	0.2222	0.1667	0.1889	0.0667	0.0111	0	0.0111
7	0	0	0	0	0	0	0	0.0111	0.1111	0.2778	0.2667	0.2333	0.1889	0.0556	0	0
8	0	0	0	0	0	0	0	0	0.0222	0.0778	0.2556	0.2778	0.2889	0.1444	0.0444	0
9	0	0	0	0	0	0	0	0	0	0.0222	0.0444	0.1444	0.2222	0.3111	0.1556	0.0222
10	0	0	0	0	0	0	0	0	0	0	0.0222	0.0889	0.1333	0.2444	0.2889	0.1889
11	0	0	0	0	0	0	0	0	0	0	0	0	0.0556	0.1111	0.2222	0.2333
12	0	0	0	0	0	0	0	0	0	0	0	0	0.0111	0.1111	0.1444	0.2556
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1222	0.1111
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0111	0.1444
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0333
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 49. Proporción de coincidencias entre satélites eliminados para las iteraciones 1-16

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0.0111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0.0111	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0.0111	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0.0222	0.0222	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0.1889	0.0111	0.0222	0.0111	0	0	0	0	0	0	0	0	0	0	0	0
11	0.2333	0.1889	0.0333	0.0111	0	0	0	0	0	0	0	0	0	0	0	0
12	0.2556	0.2111	0.1556	0.0444	0.0333	0.0111	0	0	0	0	0	0	0	0	0	0
13	0.1111	0.1333	0.1778	0.1333	0.0444	0.0111	0.0111	0	0	0	0	0	0	0	0	0
14	0.1444	0.2222	0.1778	0.2000	0.1000	0.0333	0	0	0	0	0	0	0	0	0	0
15	0.0333	0.1000	0.1333	0.1667	0.2444	0.1556	0.0667	0.0111	0	0	0	0	0	0	0	0
16	0	0.0889	0.2222	0.2444	0.2667	0.3111	0.2222	0.0778	0	0	0	0	0	0	0	0
17	0	0.0111	0.0556	0.1000	0.1333	0.1889	0.2667	0.1667	0.0222	0	0	0	0	0	0	0
18	0	0	0.0111	0.0778	0.1111	0.1000	0.1556	0.2889	0.1889	0	0	0	0	0	0	0
19	0	0	0	0.0111	0.0556	0.1000	0.0778	0.1889	0.3556	0.1111	0	0	0	0	0	0
20	0	0	0	0	0.0111	0.0556	0.0889	0.0778	0.1667	0.3889	0	0	0	0	0	0
21	0	0	0	0	0	0.0333	0.0778	0.0667	0.1000	0.2000	0.2556	0	0	0	0	0
22	0	0	0	0	0	0	0.0333	0.0778	0.0667	0.1222	0.3556	0	0	0	0	0
23	0	0	0	0	0	0	0	0.0444	0.0778	0.0667	0.1778	0.3111	0	0	0	0
24	0	0	0	0	0	0	0	0	0.0222	0.0889	0.0778	0.3556	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0.0222	0.1000	0.1889	0.4222	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0.0111	0.0556	0.3444	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0.0222	0.0667	0.1111	0.6000	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0.0222	0.1000	0.2333	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0.0222	0.1444	0.7667	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0.0222	0.1778	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0556	0.9333
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0667

Figura 50. Proporción de coincidencias entre satélites eliminados para las iteraciones 16-32

En las figuras anteriores se pueden observar las distintas proporciones de satélites eliminados coincidentes en cada iteración de los 90 experimentos. Puede comprobarse como no tienen los mismos valores que la distribución aleatoria, aun así es necesario realizar un contraste de hipótesis para asegurar que no sigan la misma distribución y en qué grado.

Con el objetivo de comprender mejor la evolución del número de coincidencias y sus distribuciones, se han representado los histogramas para cada iteración.

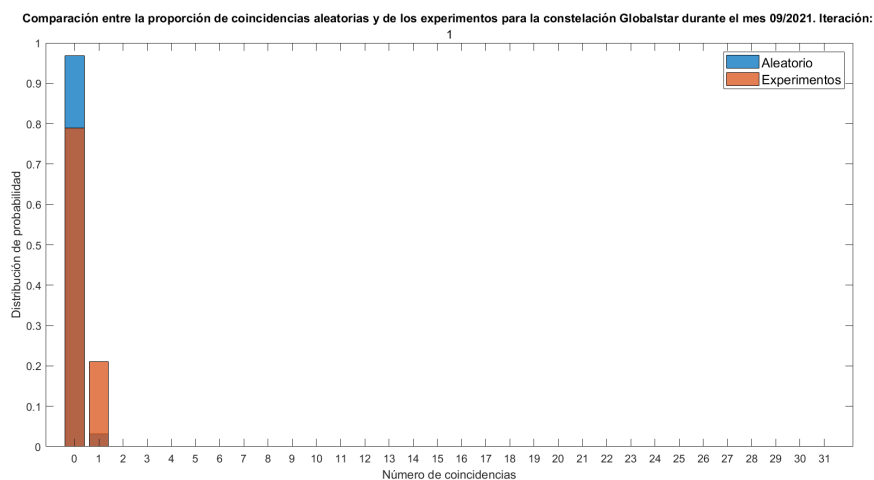


Figura 51. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 1)

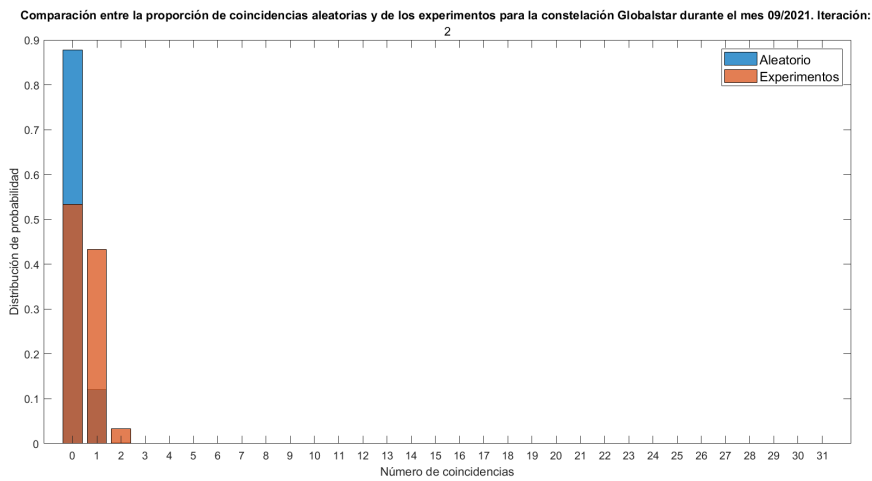


Figura 52. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 2)

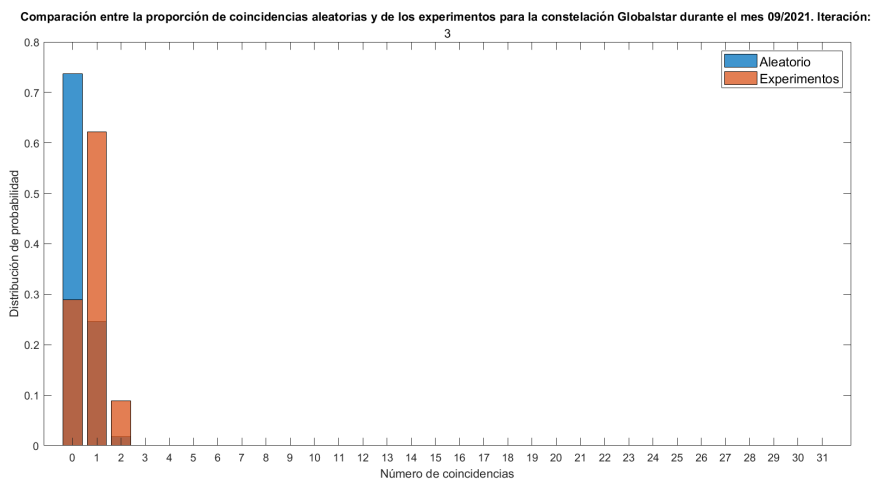


Figura 53. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 3)

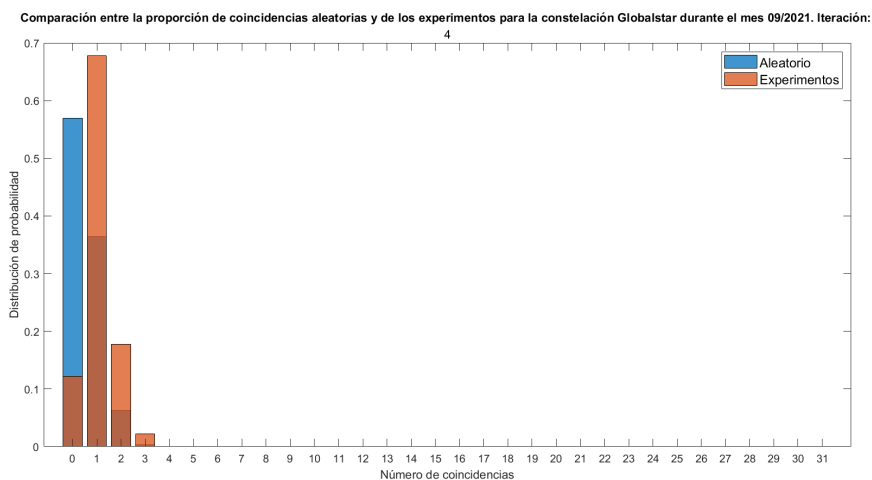


Figura 54. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 4)

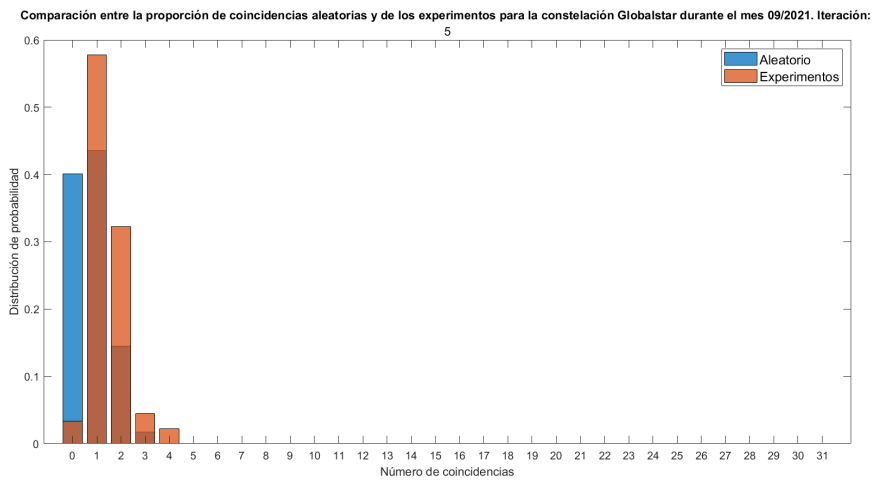


Figura 55. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 5)

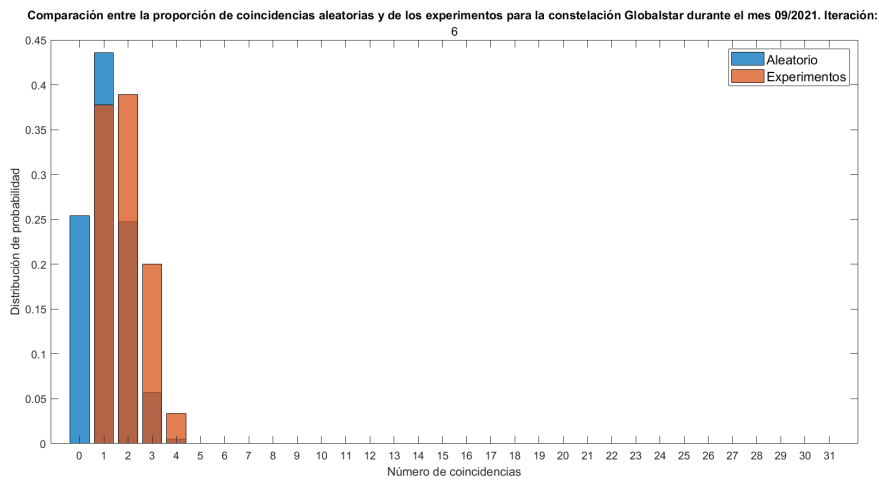


Figura 56. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 6)

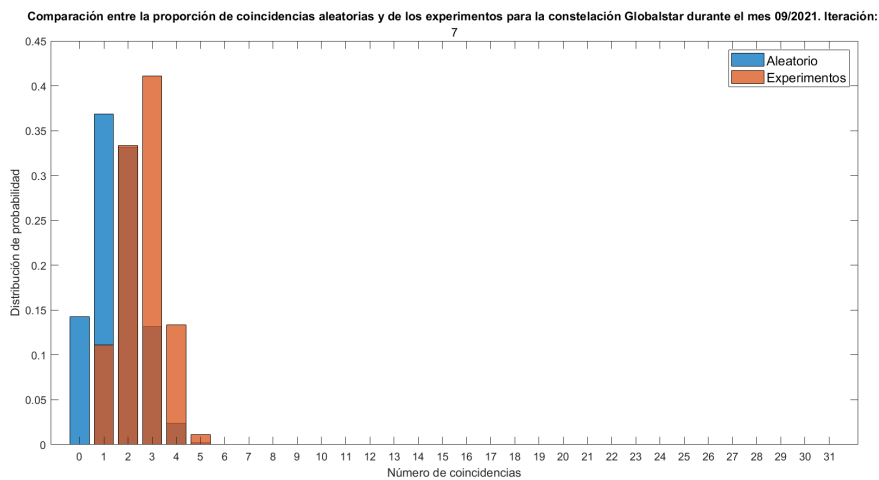


Figura 57. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 7)

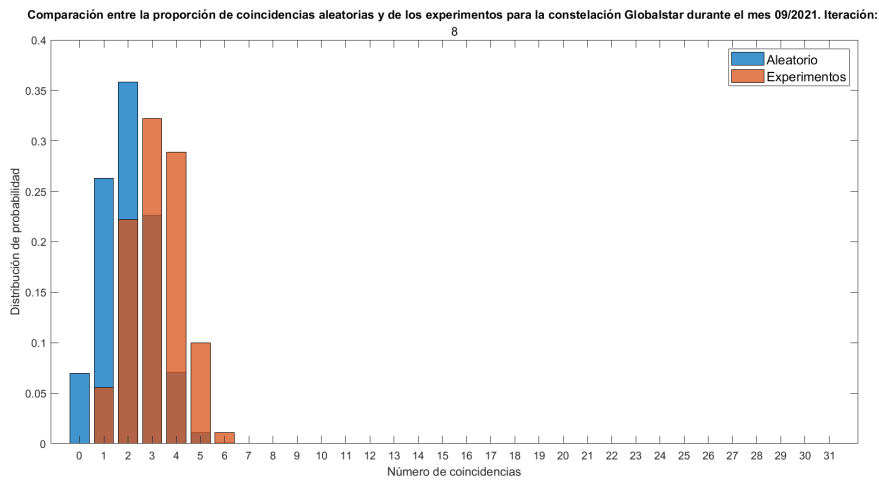


Figura 58. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 8)

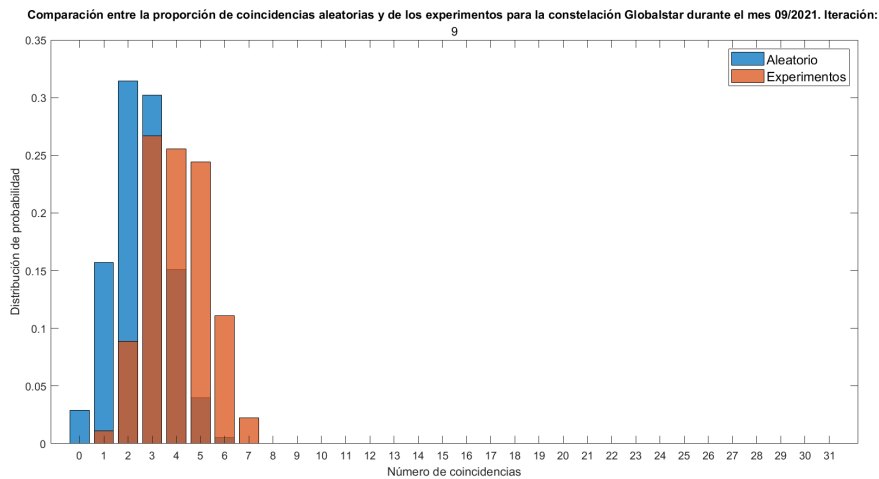


Figura 59. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 9)

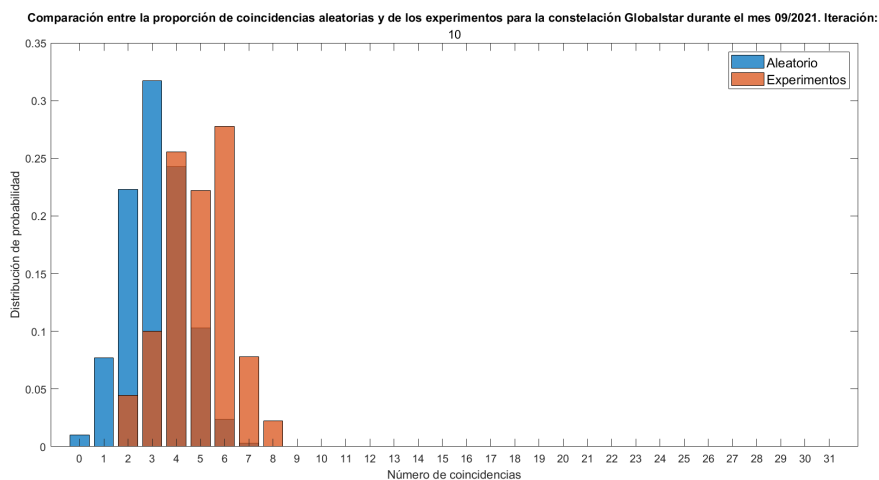


Figura 60. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 10)

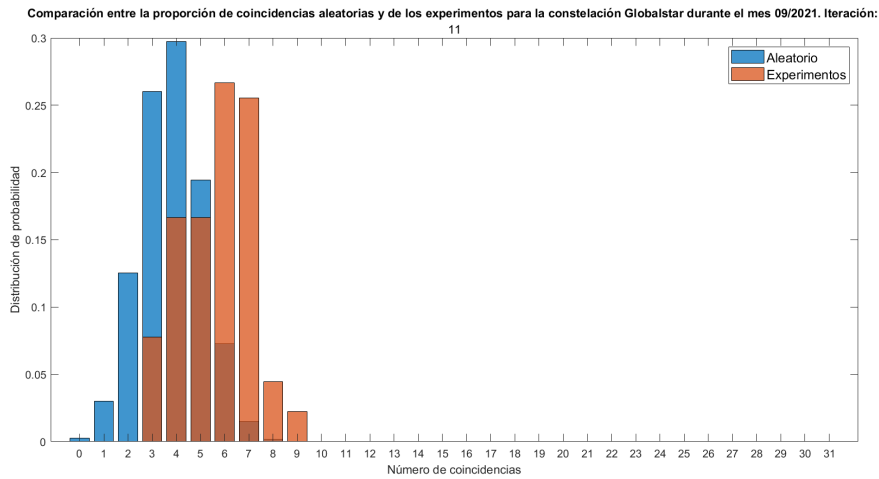


Figura 61. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 11)

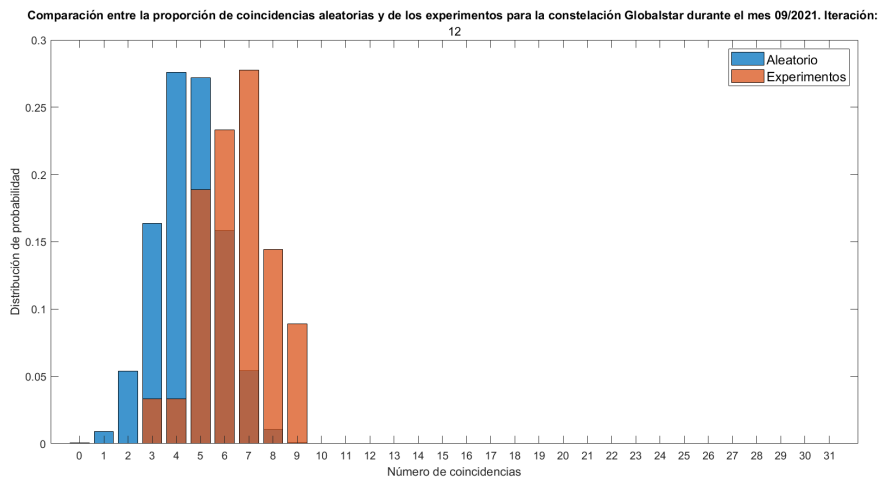


Figura 62. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 12)

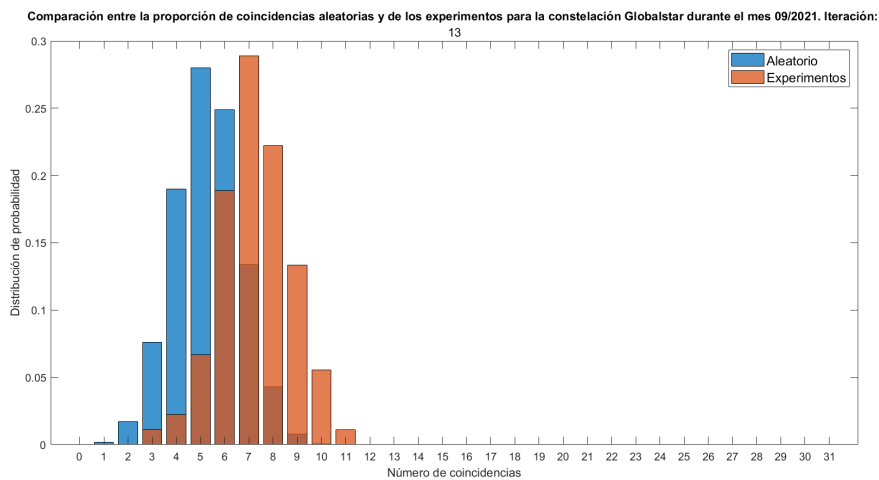


Figura 63. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 13)

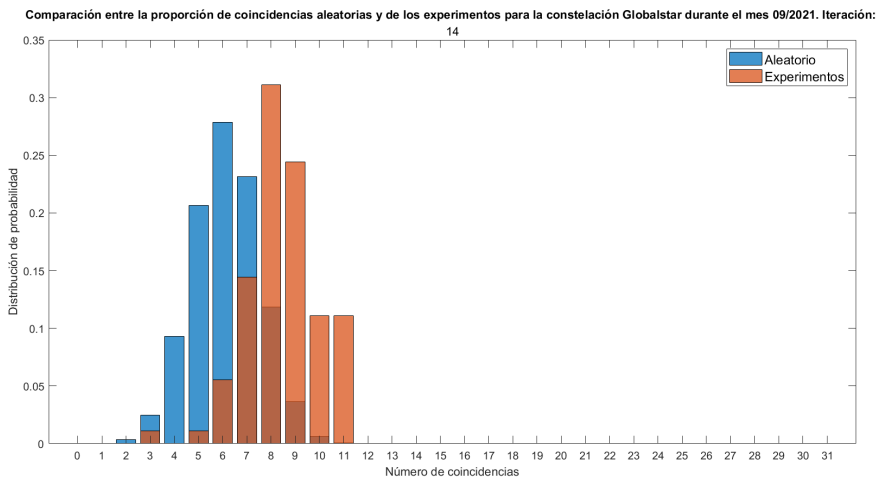


Figura 64. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 14)

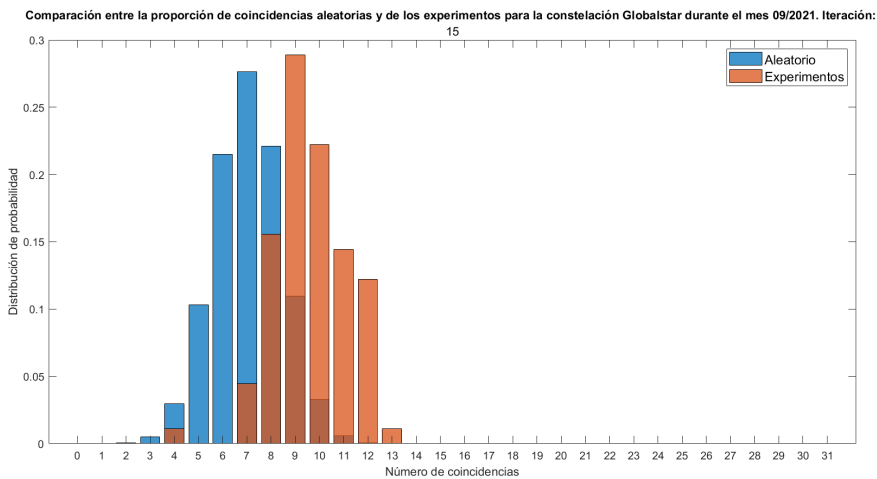


Figura 65. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 15)

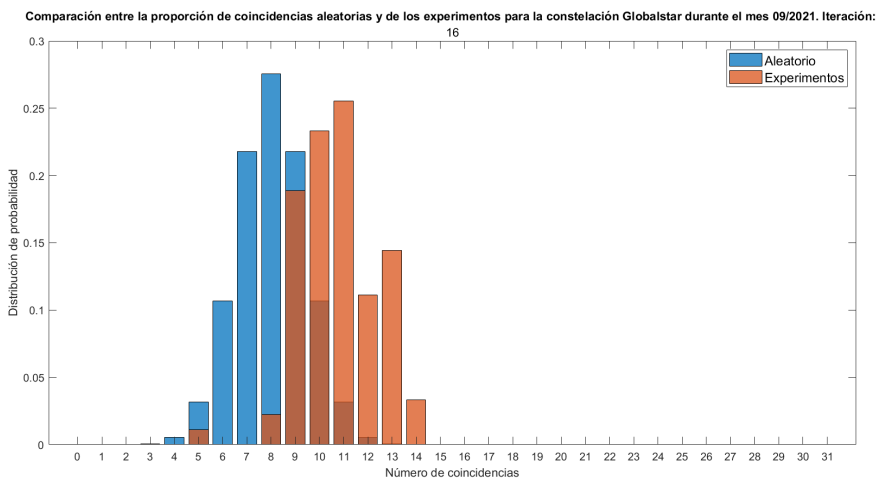


Figura 66. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 16)

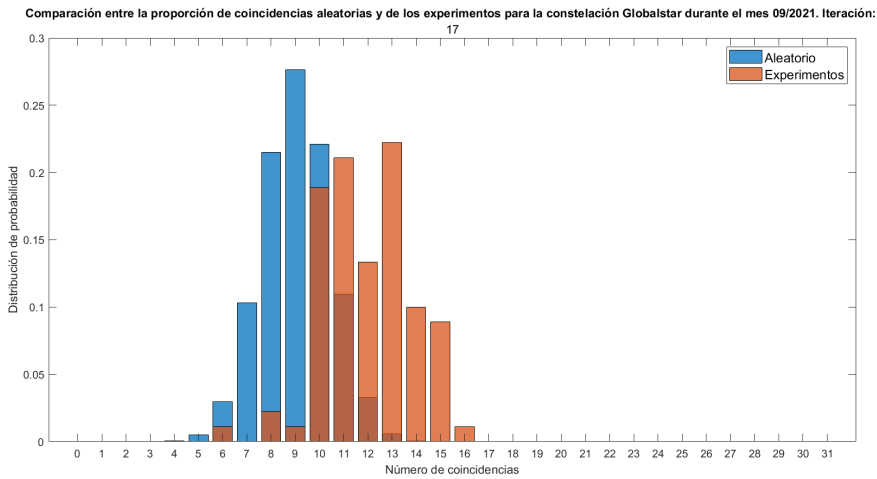


Figura 67. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 17)

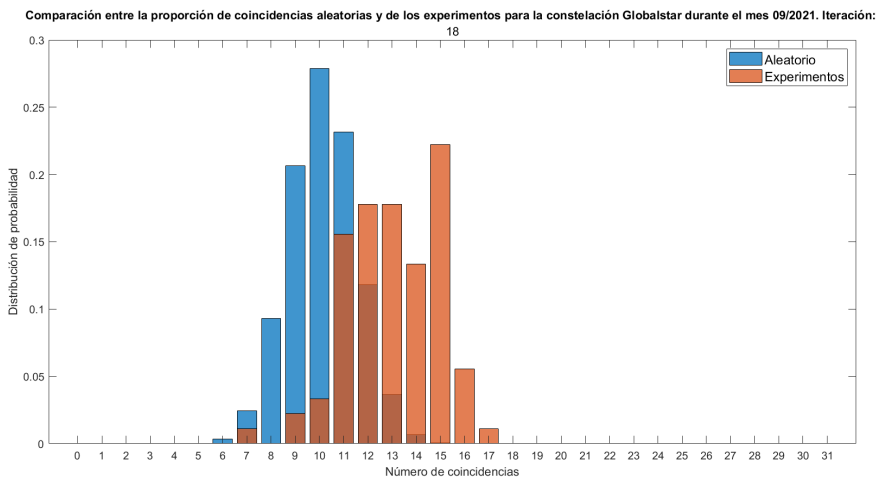


Figura 68. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 18)

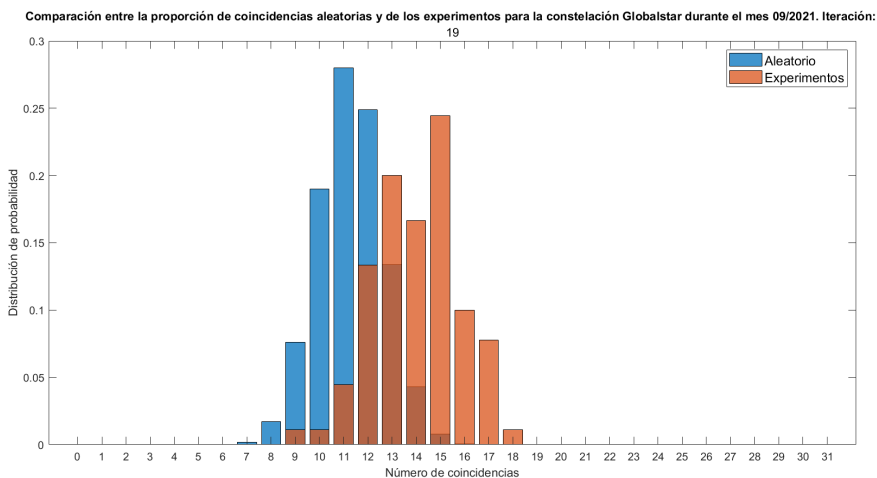


Figura 69. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 19)

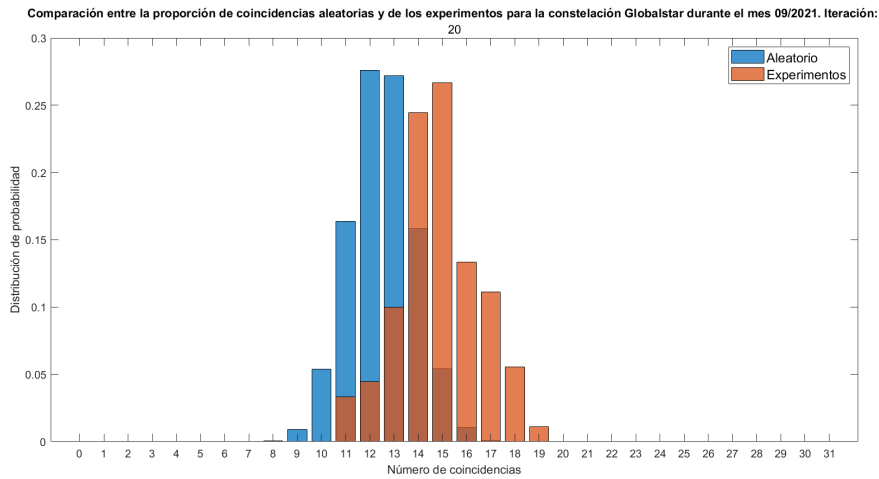


Figura 70. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 20)

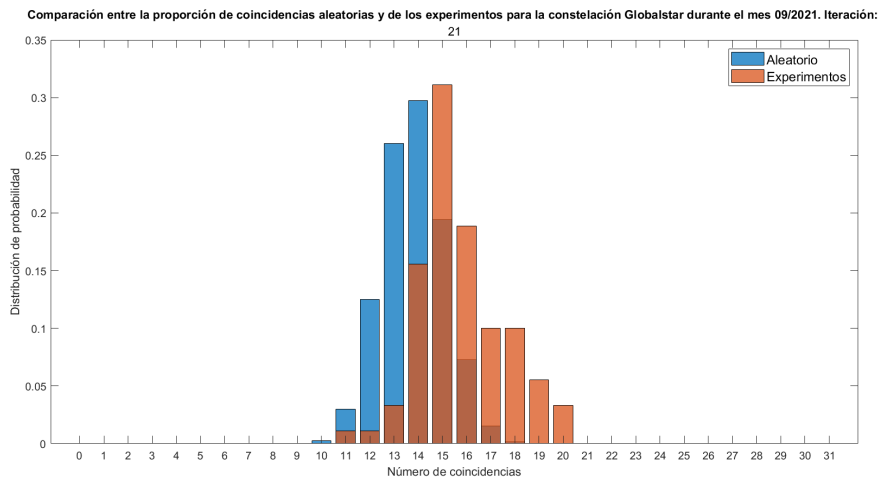


Figura 71. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 21)

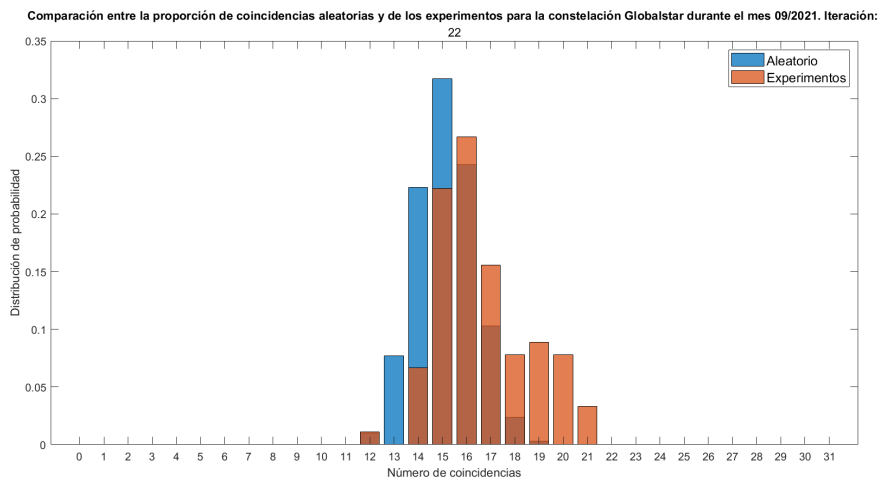


Figura 72. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 22)

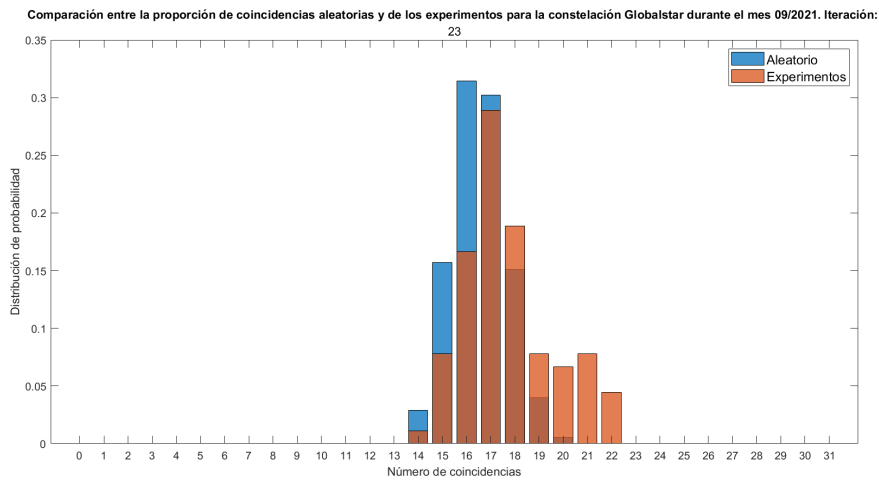


Figura 73. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 23)

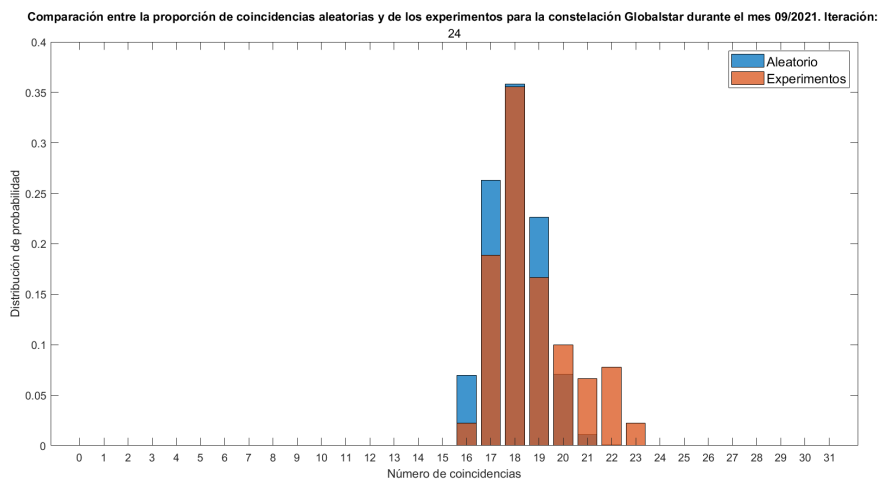


Figura 74. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 24)

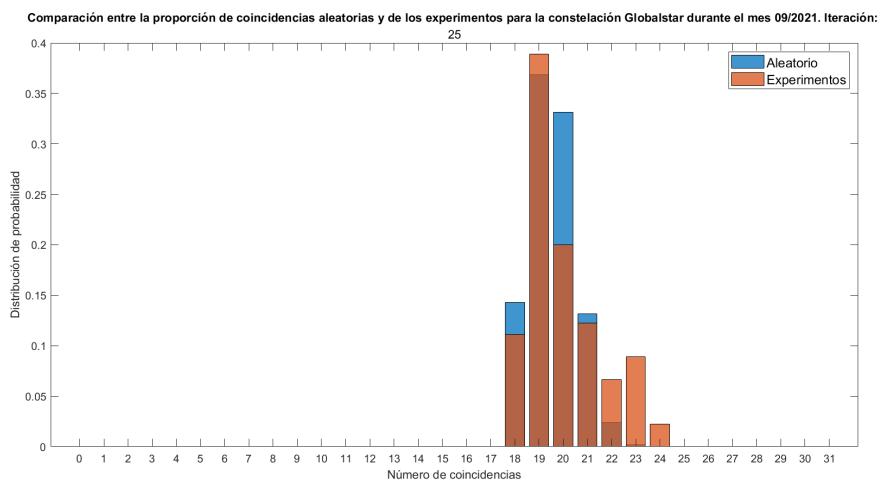


Figura 75. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 25)

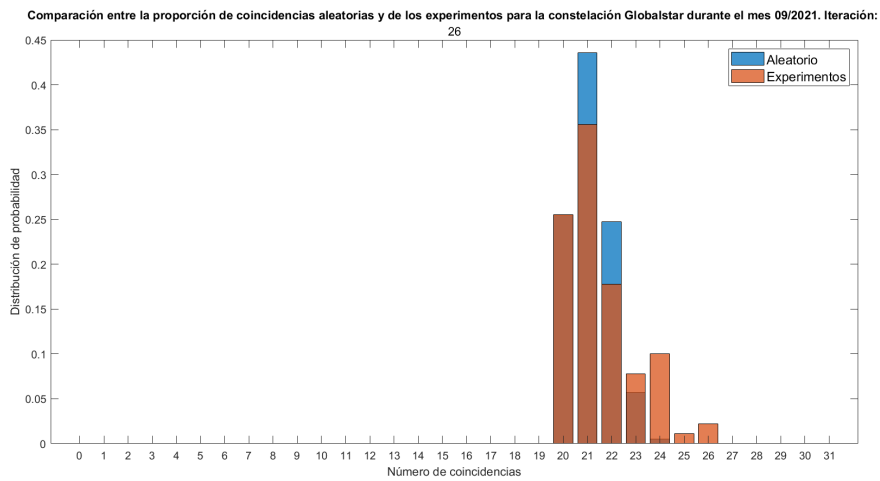


Figura 76. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 26)

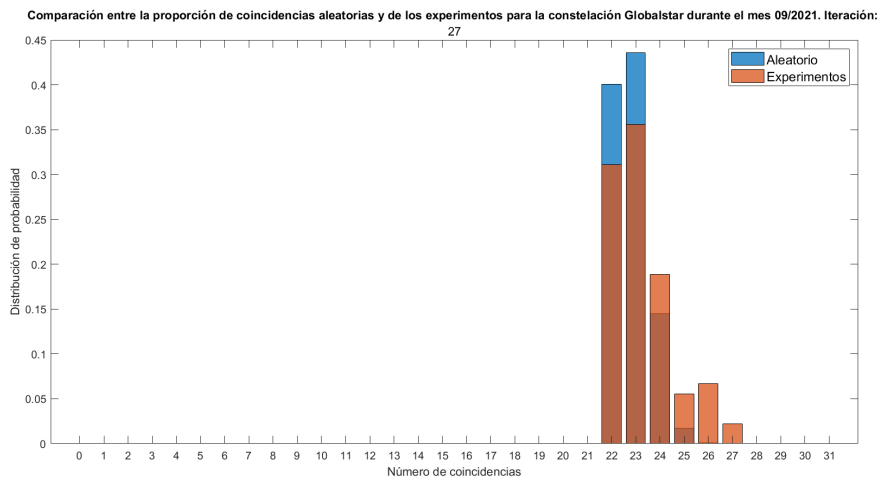


Figura 77. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 27)

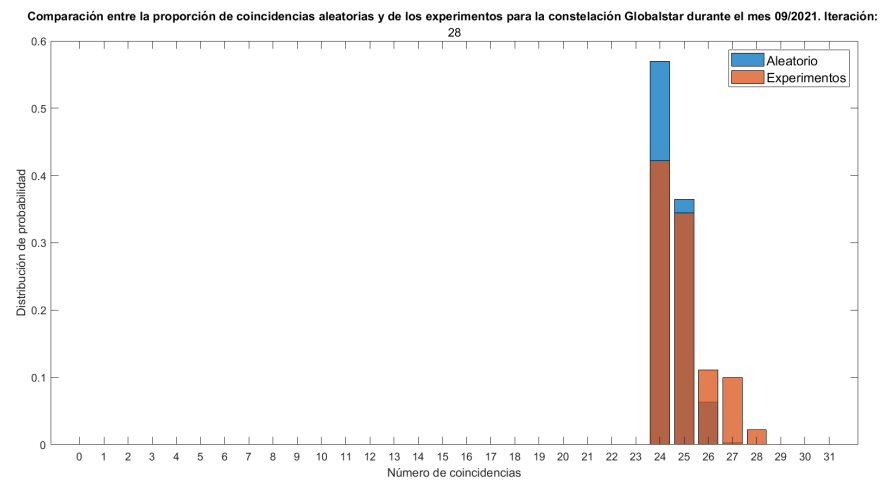


Figura 78. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 28)

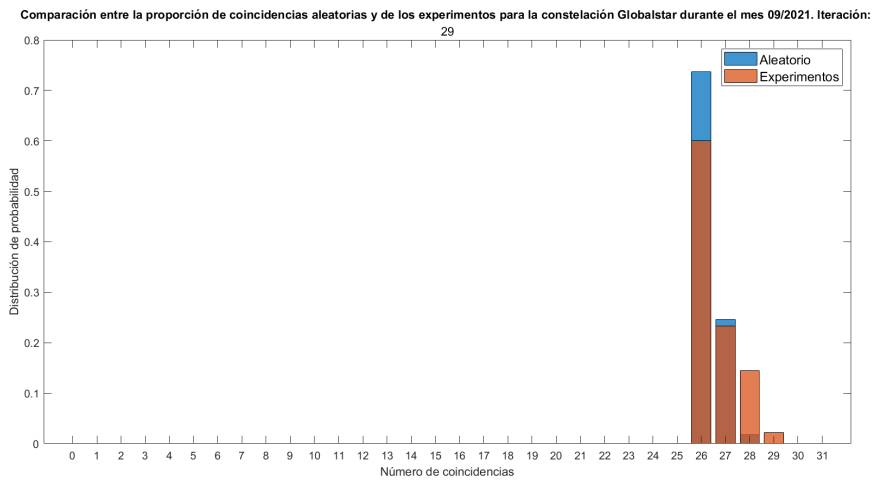


Figura 79. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 29)

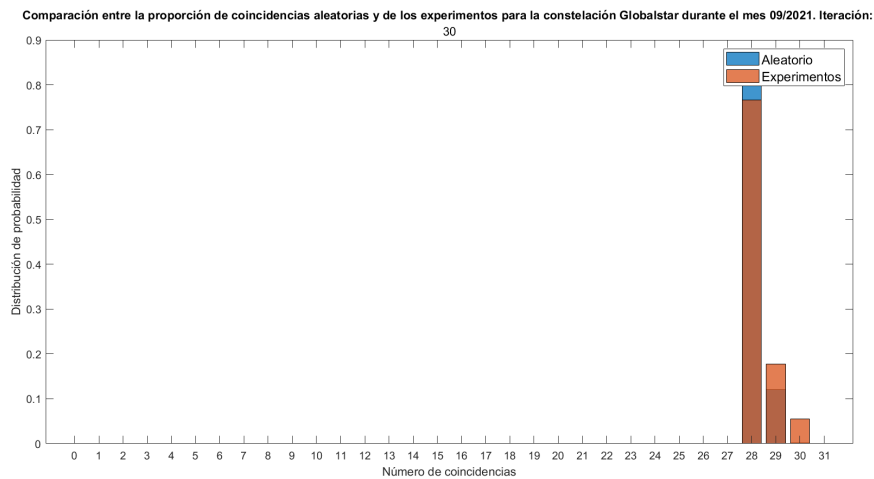


Figura 80. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 30)

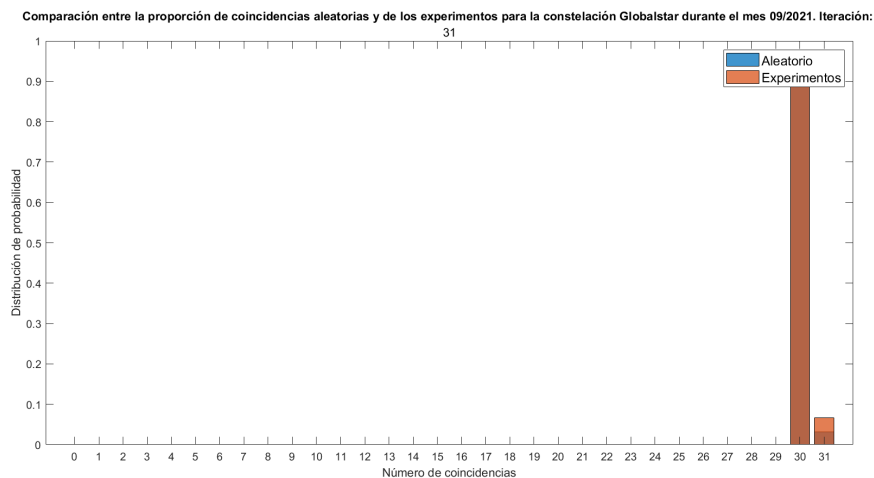


Figura 81. Comparación entre la proporción de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021 (Iteración 31)

Puede observarse como es en los valores de las iteraciones centrales en donde las distribuciones presentan un mayor grado de diferenciación. Entre los valores 9 y 21 se da una mayor aparición de coincidencias que producen distribuciones experimentales asimétricas hacia la izquierda, es decir, con colas de valores bajos en dicho lado. Estas diferencias respecto a las distribuciones aleatorias se traducen como p-valores más bajos en la prueba χ^2 de Pearson.

Iteración	p-valor	Media aleatoria	Media experimental	Proporción
1	1.9629e-05	0.0313	0.2111	0.1857
2	7.6635e-07	0.1250	0.5000	0.2000
3	4.7151e-12	0.2813	0.8000	0.1908
4	6.7239e-07	0.5000	1.1000	0.1714
5	2.9670e-11	0.7813	1.4444	0.1572
6	2.6875e-07	1.1250	1.8889	0.1567
7	2.2340e-08	1.5313	2.6000	0.1954
8	1.2130e-05	2.0000	3.1889	0.1981
9	7.1712e-12	2.5313	4.0556	0.2356
10	1.2158e-11	3.1250	4.9111	0.2598
11	2.5712e-10	3.7813	5.6778	0.2627
12	2.1955e-09	4.5000	6.4778	0.2637
13	4.2599e-10	5.2813	7.2667	0.2572
14	2.5535e-15	6.1250	8.4556	0.2959
15	6.4393e-15	7.0313	9.6222	0.3251
16	0	8.0000	10.7556	0.3444
17	5.8509e-14	9.0313	11.9556	0.3670
18	1.4766e-14	10.1250	13.0444	0.3707
19	3.3318e-13	11.2813	14.0222	0.3551
20	7.3419e-13	12.5000	14.8556	0.3141
21	1.5875e-10	13.7813	15.7778	0.2766
22	2.8104e-05	15.1250	16.6556	0.2226
23	0.0021	16.5313	17.7222	0.1841
24	0.0035	18.0000	18.7556	0.1259
25	0.0039	19.5313	20.0000	0.0857
26	0.0149	21.1250	21.5333	0.0838
27	0.0127	22.7813	23.2778	0.1177
28	0.0156	24.5000	24.9556	0.1302
29	0.0037	26.2813	26.5889	0.1132
30	0.0074	28.1250	28.2889	0.0874
31	0.01273	30.0313	30.0667	0.0366

Tabla 5. p-valores de la prueba χ^2 de Pearson, valores medios de las distribuciones aleatoria y experimental y proporción de no aleatoriedad de la distribución experimental

Estos p-valores representan las probabilidades de que los valores estadísticos calculados para cada iteración sean posibles si las hipótesis nulas fueran ciertas, es decir, que ambas distribuciones fuesen las mismas. Como ninguno de ellos supera el nivel de significación α especificado del 5 %, pueden descartarse todas las hipótesis nulas.

Ahora bien, para los últimos valores puede observarse como las distribuciones experimentales se encuentran más cerca de las distribuciones aleatorias. Los factores de proporción de la última

columna de la tabla, que representan lo cerca que se encuentra la media de la distribución experimental de la de la aleatoria (0) o de los valores máximos de coincidencia (1), adoptan valores más bajos. A su vez, los p-valores son más altos.

Esto indica un mayor grado de coincidencias para los valores de iteraciones medias. Al principio, cada método tiende a eliminar unos satélites muy concretos, que no tienen por que ser iguales, para después ir aumentando las similitudes. Finalmente, los últimos satélites eliminados por cada método presentan un grado de coincidencias medias muy similar al de las distribuciones aleatorias, como puede verse en la siguiente figura en donde se representan los valores medios de cada distribución.

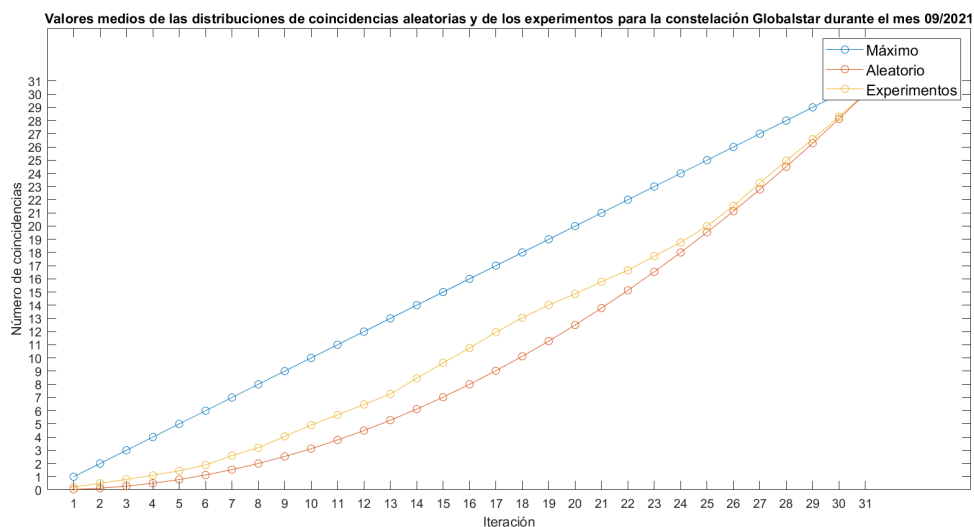


Figura 82. Valores medios de las distribuciones de coincidencias aleatorias y de los experimentos para la constelación Globalstar durante el mes 09/2021

La principal implicación de estos fenómenos es la validez de ambos métodos de eliminación de satélites, multiplex y monoplex, para un número de eliminaciones menor a aproximadamente el 75 % de los satélites totales de la red.

Respecto a la superioridad de alguno de los métodos frente al otro, se han empezado comparando los valores de eficiencia del método multiplex con los de su versión recalculada.

En relación a los valores medios, no existe ningún punto en donde los del método recalculado superen a los del original. Esto implica que el método monoplex no es más eficaz que el método multiplex encontrando satélites críticos. Puede observarse como las diferencias entre eficiencias van aumentando ligeramente a medida que aumenta el número de satélites eliminados, así como aumenta la dispersión de sus valores. La causa tras esto puede ser la menor efectividad de ambos métodos para localizar vulnerabilidades una vez se realizan más del 66 % de las eliminaciones, reflejada en la mayor dispersión de eficiencias finales mostrada en las últimas figuras.

El método multiplex original presenta un mayor nivel de consistencia respecto a la desviación estándar de las eficiencias, con casos más similares entre sí, independientemente de la fecha en la que fue calculada la red.

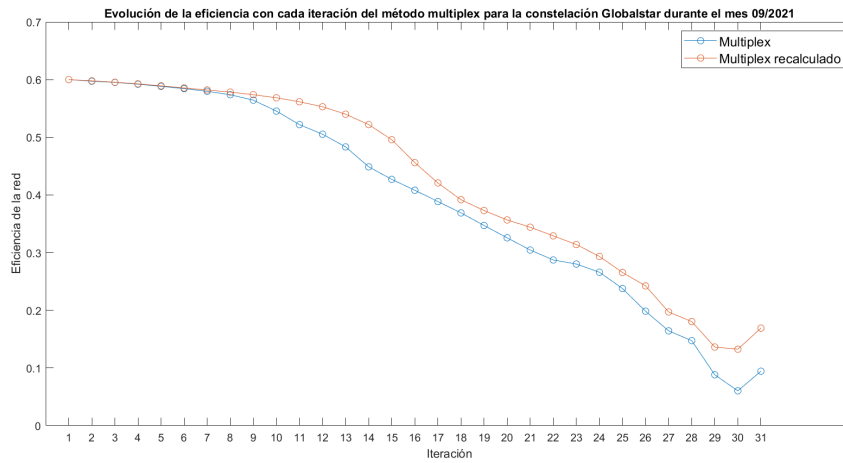


Figura 83. Evolución de los valores medios de eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021

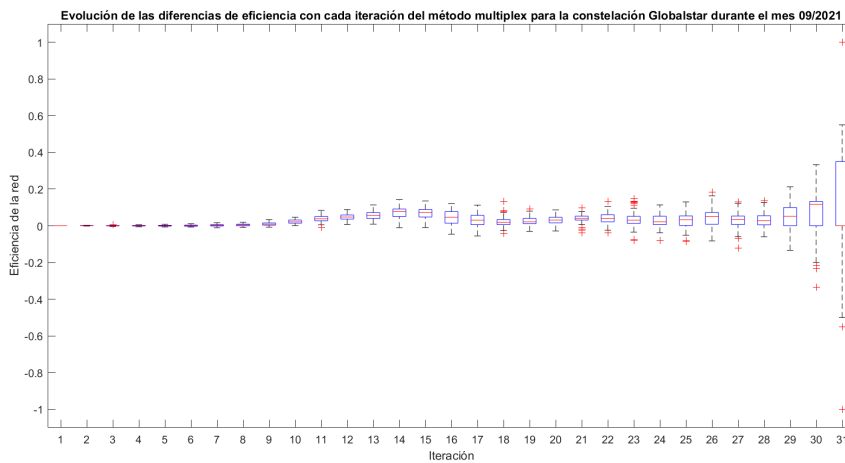


Figura 84. Evolución de las diferencias de eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021

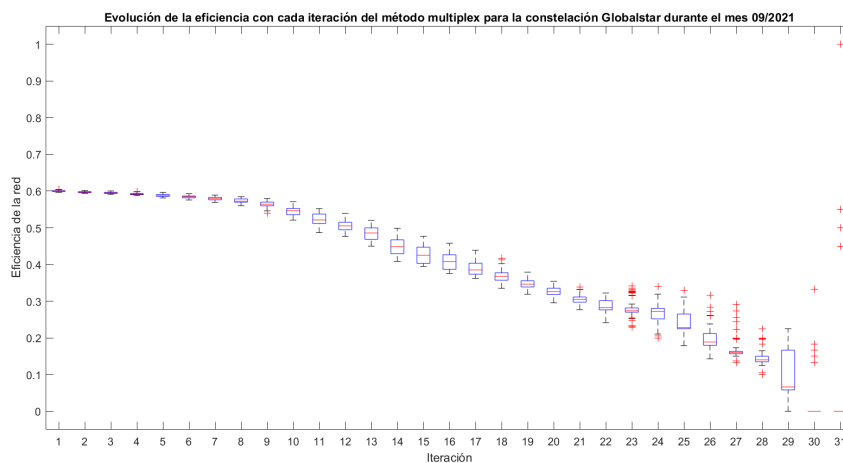


Figura 85. Evolución de la eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021

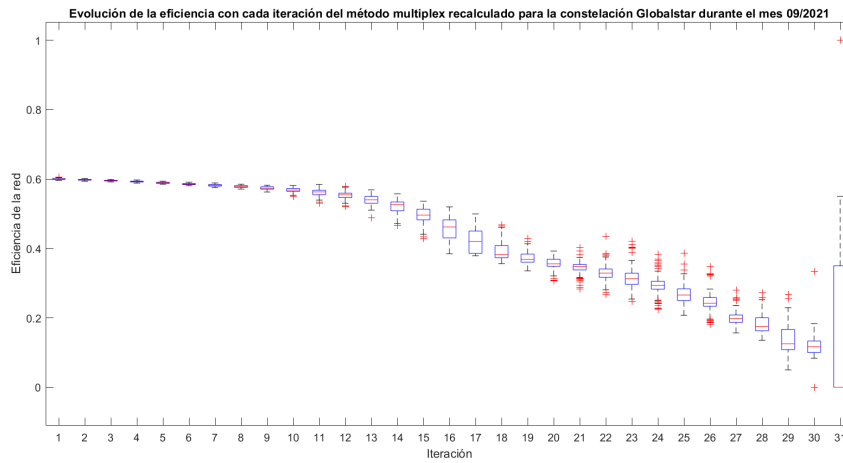


Figura 86. Evolución de la eficiencia con cada iteración del método multiplex recalculado para la constelación Globalstar durante el mes 09/2021

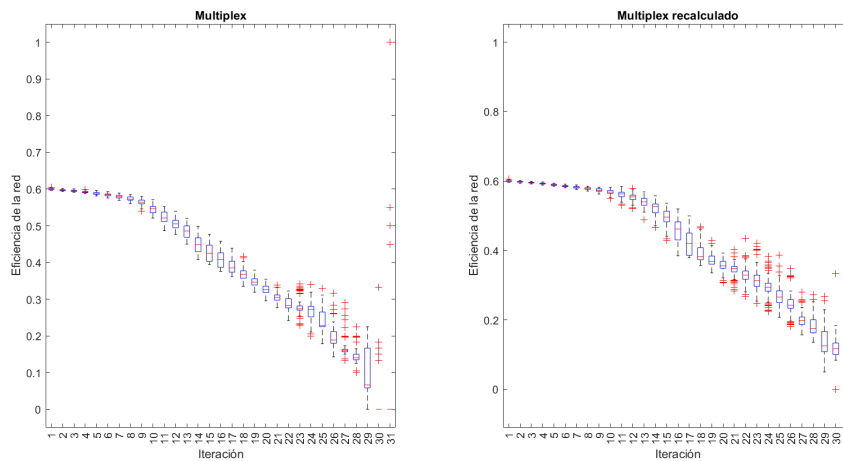


Figura 87. Evolución de la eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021

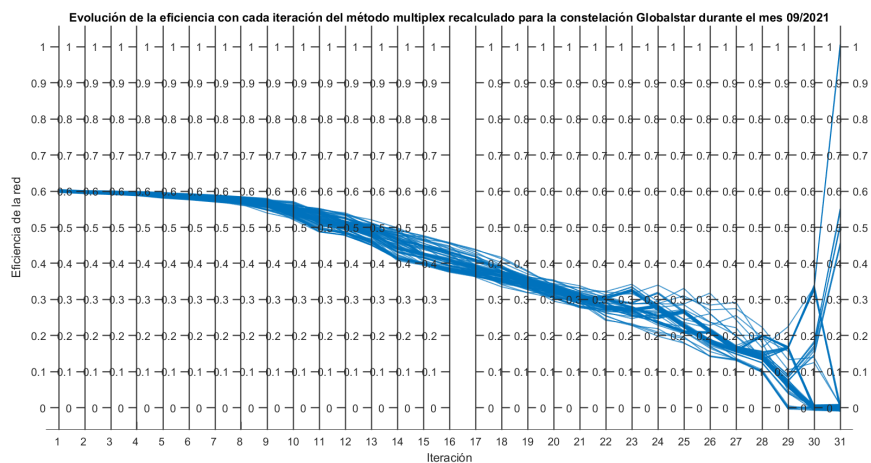


Figura 88. Evolución de la eficiencia con cada iteración del método multiplex para la constelación Globalstar durante el mes 09/2021

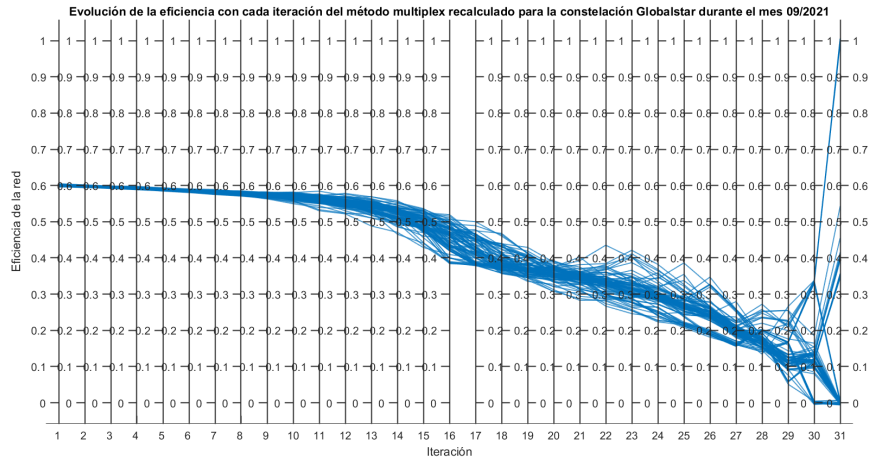


Figura 89. Evolución de la eficiencia con cada iteración del método multiplex recalculado para la constelación Globalstar durante el mes 09/2021

Al analizar los valores de eficiencia del método monoplex respecto a sus valores recalculados puede observarse como el método recalculado ofrece valores ascendentes hasta la decimotercera eliminación, en donde descienden bruscamente por debajo de los valores del método original. Esto puede deberse a la eliminación de satélites con valores bajos de eficiencia hasta la llegada de un punto en donde se producen escisiones críticas en la red o se pierden ciertos enlaces de mayor importancia al haber seguido un criterio coherente de eliminación de satélites con mayor centralidad para producir este tipo de vulnerabilidades.

En las dos figuras finales puede observarse como el método original presenta hasta cinco escalones en donde la eficiencia decae tras la decimoquinta iteración, mientras que el método recalculado presenta un solo escalón definido a partir de la decimotercera. A su vez, el método original presenta más picos de subida de la eficiencia tras dichas iteraciones sin producirse después un mayor descenso de la misma.

La principal implicación de estos fenómenos es la mayor eficacia del método multiplex para encontrar los satélites críticos de la red en el largo plazo.

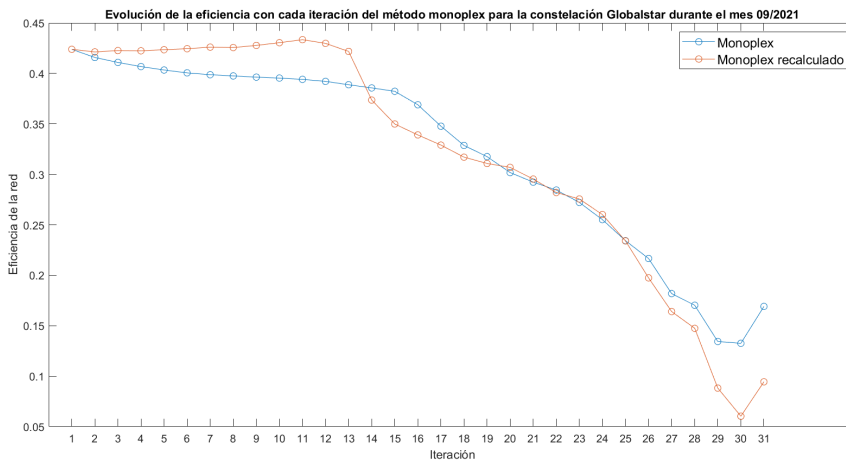


Figura 90. Evolución de los valores medios de eficiencia con cada iteración del método multiplex y multiplex recalculado para la constelación Globalstar durante el mes 09/2021

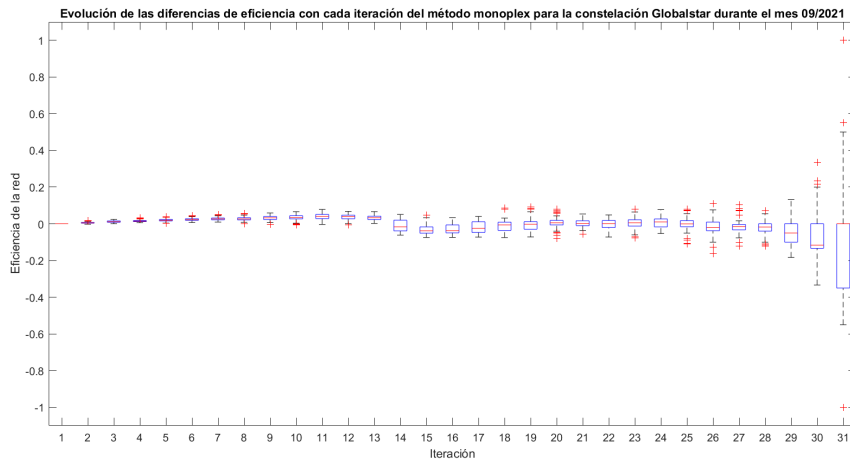


Figura 91. Evolución de las diferencias de eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021

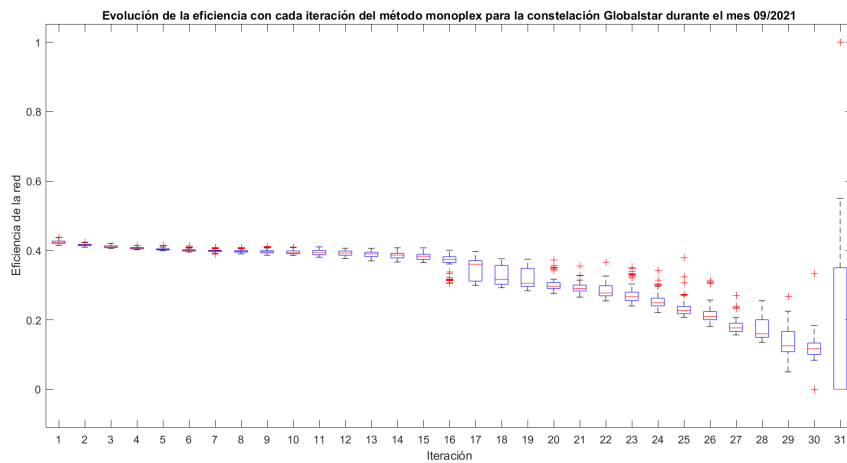


Figura 92. Evolución de la eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021

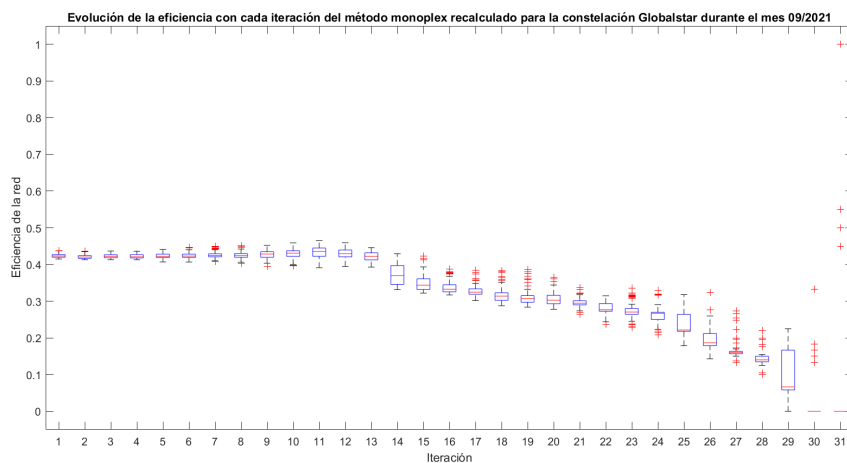


Figura 93. Evolución de la eficiencia con cada iteración del método monoplex recalculado para la constelación Globalstar durante el mes 09/2021

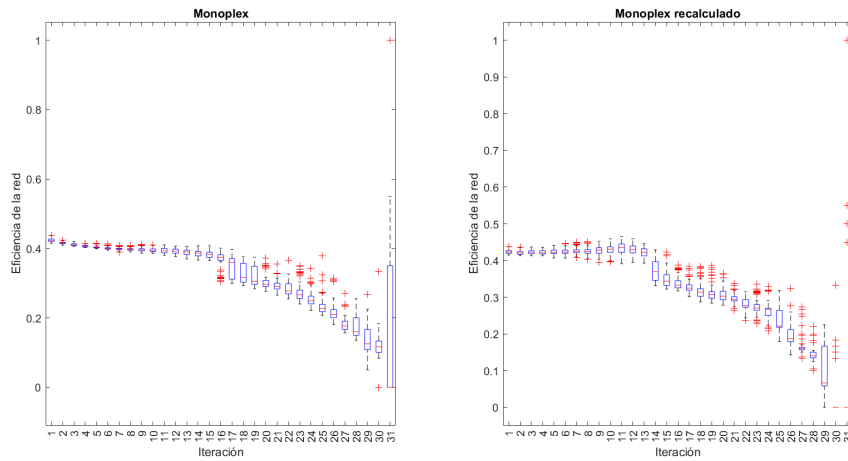


Figura 94. Evolución de la eficiencia con cada iteración del método monoplex y monoplex recalculado para la constelación Globalstar durante el mes 09/2021

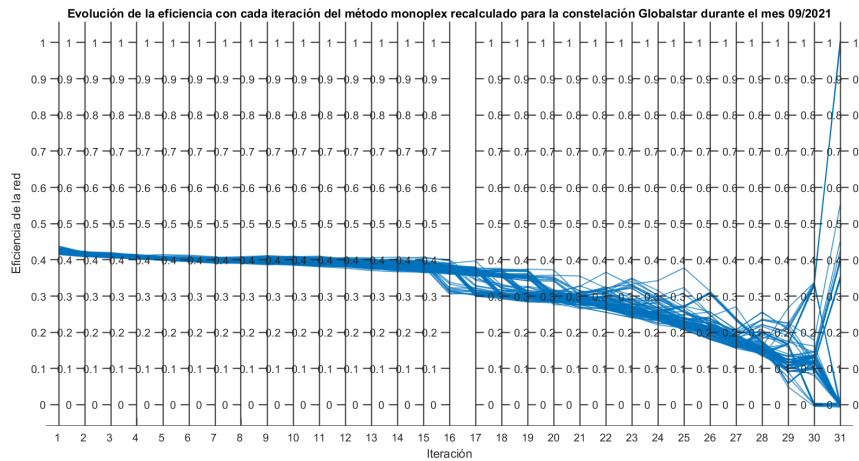


Figura 95. Evolución de la eficiencia con cada iteración del método monoplex para la constelación Globalstar durante el mes 09/2021

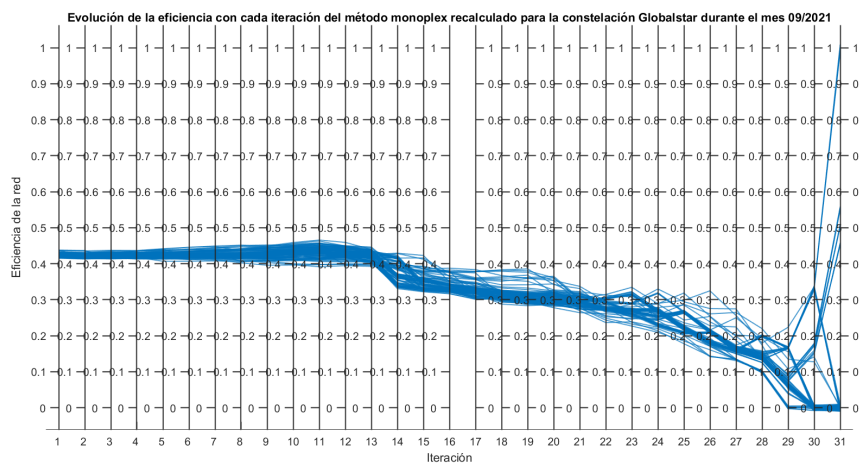


Figura 96. Evolución de la eficiencia con cada iteración del método monoplex recalculado para la constelación Globalstar durante el mes 09/2021

Al no presentar ninguna ventaja el método monoplex sobre el multiplex, empleando los criterios de medida de este último, y ofrecer mejores resultados el multiplex sobre el monoplex a partir del 40 % de eliminaciones, empleando los criterios del monoplex, se puede determinar que el método multiplex presenta un mayor nivel de efectividad determinando el conjunto de los satélites más críticos al eliminar múltiples nodos de la red.

No se puede asegurar que ninguno de ellos presente una ventaja frente al otro en las iteraciones iniciales, aunque el método monoplex presenta menores diferencias de eficiencia cuando se le aplican las métricas del multiplex en estas primeras iteraciones. Por ello, el método monoplex puede ser más útil para representar la constelación de forma completa, compactando el intervalo temporal en forma de enlaces ponderados. Este método requiere realizar un menor número de cálculos y permite visualizar todo el intervalo temporal al mismo tiempo.

El método multiplex necesita realizar un mayor número de cálculos, por lo que es más lento. Aun así, dispone de una dimensión adicional al ser posible asignar pesos a los enlaces para representar la importancia de los mismos u otros factores. Ante la duda de la eficacia del método monoplex para múltiples eliminaciones, la recomendación principal es emplear el método multiplex para la realización del análisis de vulnerabilidades y el monoplex, si se desea representar el estado general de la red en un solo grafo.

Para las últimas eliminaciones puede comprobarse como aparecen múltiples ramificaciones en la eficiencia de la red. A su vez, las coincidencias entre satélites eliminados por ambos métodos tienden a la aleatoriedad. Esto se debe a múltiples formas que pueden adoptar las redes generadas con pocos satélites, las cuales pueden presentar grandes diferencias entre sí tanto en su estructura como en sus variaciones a lo largo de una órbita. Aun así, la eficiencia tiende a acabar siendo nula en la mayoría de los casos, por lo que el análisis sigue siendo válido.

Capítulo 5

Conclusiones

A lo largo de este Proyecto se han desarrollado varios programas diseñados para analizar las vulnerabilidades de una red, localizando en el orden correcto sus satélites más críticos. El punto de partida ha sido el programa desarrollado con anterioridad por el director de este Proyecto, Enrique Sevillano Bravo, con el objetivo de ampliarlo en profundidad.

Ante la falta de herramientas flexibles para el análisis de redes de satélites, han ido surgiendo de forma orgánica distintos programas y planteamientos para el análisis.

El primer paso consistió en desarrollar un programa capaz de representar todos los satélites en el tiempo y el espacio. Para ello se decidió emplear el formato TLE a la entrada por ser el más común. Ya que sí se disponía de funciones, programadas en MATLAB, para el cálculo de la posición y la velocidad de satélites a partir de un TLE, se decidió emplear las mismas.

Como el objetivo de este programa es equilibrar precisión con sencillez de uso, no se han tenido en cuenta perturbaciones gravitacionales, aplicando el modelo SPG4 y suponiendo órbitas bajas de corta duración. Aun así, ya que no se pretende usar el programa para predecir órbitas, sino para analizar con precisión aquellas ya realizadas, se desarrolló e incorporó un método para calcular las posiciones intermedias entre TLE, de los que se conoce la posición con una mayor precisión. Para ello, se obtienen las posiciones a partir de sendos TLE, hacia delante y hacia atrás en el tiempo y se ponderan los valores, tal y como se indica en la ecuación (24).

Gracias a la sencillez de las variables de entrada, en donde sólo hay que indicar los identificadores de los satélites, el intervalo de representación y el periodo de muestreo, así como disponer de sus TLE en formato de archivo de texto, el programa puede emplearse como plataforma para otras aplicaciones relacionadas con satélites. Por último, gracias a las funciones de procesamiento en paralelo, el programa está diseñado para ser un número de veces más rápido equivalente al número de núcleos del sistema en donde se ejecute.

El segundo paso consistió en la ampliación del programa de análisis de vulnerabilidades ya existente. Para ello se redefinió la estructura general del programa, creando distintos bloques para cada métrica y guardando los valores en forma de estructuras. Esto permite un mejor entendimiento y una mayor facilidad para añadir otros bloques. Los datos de entrada son los del programa anterior para disponer de las posiciones de los nodos con la mayor precisión posible.

Durante el desarrollo del programa se plantearon distintas opciones para la representación de la red. Al principio se barajó la idea de mantener los satélites eliminados, pero sin enlazar. Esto se descartó en pos de diferenciar aquellos satélites que quedasen aislados tras varias eliminaciones.

Otra de las ideas surgidas fue la representación de la red mediante el método monoplex. Con el objetivo de facilitar los cálculos, se comprimían las redes binarias de cada instante del intervalo en una sola red con enlaces ponderados, representando la proporción de tiempo en la que los satélites se encuentran conectados.

Una vez planteados los dos métodos de análisis, el programa evalúa la criticidad de la red en función de la métrica de centralidad de intermediación de cada satélite, ya que es la que ha demostrado tener una mayor eficacia al evaluar la importancia de los nodos en función de la eficiencia de la red en estudios realizados previamente [27]. Aun así, el programa ofrece la posibilidad de realizar los mismos análisis en función de otras métricas como el grado, el coeficiente local de agrupamiento o la eficiencia de cada nodo. Gracias a esto se pueden efectuar comparaciones entre la efectividad de cada una de ellas para el mismo análisis, aunque no sea el objetivo principal de este Proyecto.

Finalmente, se evalúa la efectividad de los métodos multiplex y monoplex. Como ambos métodos tienen métricas con distintas escalas, la comparación entre las mismas no se puede realizar de forma directa. Por ello, fue necesario generar otra serie de programas con el objetivo de verificar la validez de cada uno de los métodos. Estos programas sirven para realizar múltiples experimentos de forma periódica y realizar una comparación estadística.

5.1. Metodología

La metodología seguida a lo largo del Proyecto ha sido de carácter ágil. Se partió desde el objetivo inicial de desarrollar un programa capaz de analizar una hipotética red de satélites basada en el Internet de Satélites. A medida que se fueron concretando los aspectos técnicos necesarios fueron surgiendo las ideas para cada uno de los programas.

Como ya se ha mostrado a lo largo del Proyecto, se han planteado ciertas hipótesis iniciales descartadas o aceptadas en pos de una mayor eficiencia. Numerosas evaluaciones del estado del Proyecto y su relación con los aspectos teóricos han sido realizadas a lo largo del mismo. El empleo del método monoplex para el análisis de la red es uno de estos ejemplos.

Al principio, se partió de la idea de evaluar los valores medios de las métricas para todo el intervalo por ser la forma más intuitiva, es decir, del método multiplex. Gracias a la búsqueda de información sobre casos previos parecidos se descubrió el concepto de «redes complejas multicapa». A partir de ese punto se consideró, de forma novedosa, representar la proporción de tiempo total de conexión entre satélites en forma de enlaces ponderados, obteniendo dos métodos distintos de análisis que no tenían por qué ser equivalentes.

Finalmente, fue necesario diseñar las pruebas empleadas para comparar ambos métodos. El primer elemento a desarrollar fue la función de la distribución de probabilidad utilizada para determinar si los satélites coincidentes eliminados por ambos métodos seguían una distribución aleatoria, definida, en este caso, por un problema de combinatoria. Para realizar la prueba de bondad del ajuste entre la distribución aleatoria y la distribución experimental se determinó que el método más adecuado era la prueba χ^2 de Pearson ya que es la empleada para comparar

distribuciones discretas. Por último, se representaron los valores de eficiencia de los distintos métodos y sus versiones recalculadas para determinar la superioridad de alguno de ellos y las ventajas y desventajas de cada uno.

5.2. Resultados

Los resultados de estas pruebas indican:

- Un mayor nivel de coincidencia entre los satélites eliminados por ambos métodos cuando se ha realizado un número de eliminaciones similar a la mitad de los totales de la red. Esto implica un mayor grado de similitud entre ambos métodos para estas eliminaciones. En las últimas eliminaciones las coincidencias entre ambos métodos tienden a la aleatoriedad. Esto indica un descenso en la eficacia de los métodos a partir de dicho punto.
- Un mayor descenso en la eficiencia del método multiplex que la de su versión recalculada. Esto implica no poder asegurar que el método monoplex ofrezca mejores resultados que el multiplex.
- Un nivel de eficiencia similar entre el método multiplex y su versión recalculada en las primeras iteraciones, así como un aumento de la eficiencia en las primeras iteraciones para la versión recalculada del método monoplex, encontrándose la original por debajo y descendiendo. Aunque la versión recalculada del multiplex no ofrezca mejores resultados, la versión recalculada del método monoplex tampoco lo hace y además no reduce la eficiencia para estas iteraciones. Esto implica no poder asegurar que el método multiplex ofrezca mejores resultados para las primeras eliminaciones de satélites de la red. Debido al aumento de la eficiencia en el método monoplex recalculado y la similitud de eficiencias en el multiplex sería conveniente evaluar en estudios posteriores este fenómeno.
- Un descenso en la eficiencia del método monoplex recalculado, por debajo de la del monoplex, para las eliminaciones centrales de satélites. Esto implica una mayor eficacia del método multiplex sobre el monoplex cuando se ha realizado un número de eliminaciones similar a la mitad de los satélites totales de la red.
- Un mayor número de casos posibles, respecto a la eficiencia de la red, para las últimas eliminaciones de satélites. Esto indica una mayor variabilidad en los resultados de los experimentos, explicando porqué se da un menor número de coincidencias entre las listas de satélites eliminados según cada método en dichas iteraciones.

5.3. Recomendaciones para futuros estudios

En función de los resultados de este Proyecto se recomienda:

- Evaluar, usando los programas presentados, la eficacia de realizar el mismo análisis empleando otras métricas locales como el grado, el coeficiente local de agrupamiento y la eficiencia de cada satélite.
- Evaluar la eficacia de un modelo mixto, usando ambos métodos de análisis de vulnerabilidades, en donde las primeras eliminaciones las determina el método monoplex y las intermedias, el multiplex.

- Analizar las vulnerabilidades de otras redes de satélites para obtener una valoración general de la eficacia del método aquí presentado.
- Desarrollar un modelo, a partir del método multiplex, que incluya enlaces ponderados para representar un ancho de banda variable en el tiempo.
- Desarrollar un modelo capaz de incluir las probabilidades de impacto de cada satélite con otros cuerpos en órbita empleando para ello plataformas y programas como SOCRATES y MASTER, ya presentadas en la sección 2.5 Análisis de colisiones en órbita.

Bibliografía

- [1] **ESA.** *The Kessler Effect and how to stop it*. Recuperado de https://www.esa.int/Enabling_Support/Space_Engineering_Technology/The_Kessler_Effect_and_how_to_stop_it
- [2] **ESA. (4 de abril de 2022).** *Space debris by the numbers*. Recuperado de https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers
- [3] **ESA.** *About space debris*. Recuperado de https://www.esa.int/Safety_Security/Space_Debris/About_space_debris
- [4] **ESA. (2019)** *ESA commissions world's first space debris removal*. Recuperado de https://www.esa.int/Safety_Security/Clean_Space/ESA_commissions_world_s_first_space_debris_removal
- [5] **IADC. (2021)** *IADC Space Debris Mitigation Guidelines Rev 3. [Archivo PDF]*. Recuperado de https://www.iadc-home.org/documents_public/view/id/172#u
- [6] **ESA. (2021)** *ESA's Space Environment Report 2021*. Recuperado de https://www.esa.int/Safety_Security/Space_Debris/ESA_s_Space_Environment_Report_2021
- [7] **ESA. (2021).** *We're launching more than ever*. Recuperado de https://www.esa.int/ESA_Multimedia/Images/2021/03/We_re_launching_more_than_ever
- [8] **Brian Patrick Hardy (2020).** *Long-term effects of satellite megaconstellations on the debris environment in low earth orbit. [Tesis de Máster]*. University of Illinois at Urbana-Champaign. <http://hdl.handle.net/2142/108004>
- [9] **UCS. (2022)** *UCS Satellite Database. [Actualizado a 1 de enero de 2022]*. Recuperado de <https://www.ucsus.org/resources/satellite-database>
- [10] **NASA. (2019)** *Where Do Old Satellites Go When They Die?* Recuperado de <https://spaceplace.nasa.gov/spacecraft-graveyard/en/>
- [11] **Aaron C. Boley, Michael Byers. (2021).** *Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth*. <https://doi.org/10.1038/s41598-021-89909-7>

- [12] **Stefano Gallozzi, Marco Scardia, Michele Maris. (2020).** *Concerns about ground based astronomical observations: a step to safeguard the astronomical sky.*
<https://doi.org/10.48550/arXiv.2001.10952>
- [13] **OECD (2020).** *Space sustainability: The economics of space debris in perspective .*
<https://doi.org/10.1787/a339de43-en>
- [14] **ESA. (2020).** *The cost of space debris.* Recuperado de
https://www.esa.int/Safety_Security/Space_Debris/The_cost_of_space_debris
- [15] **NASA.** *Astromaterials Research & Exploration Science: NASA ORBITAL DEBRIS PROGRAM OFFICE. Frequently Asked Questions.* Recuperado de
<https://orbitaldebris.jsc.nasa.gov/faq/#>
- [16] **ESA. (2021).** *The impact of space debris.* Recuperado de
https://www.esa.int/ESA_Multimedia/Images/2021/03/The_impact_of_space_debris
- [17] **ESA. (2009).** *Hypervelocity impact sample.* Recuperado de
https://www.esa.int/ESA_Multimedia/Images/2009/02/Hypervelocity_impact_sample
- [18] **ESA. (2021)** *ESA's Annual Space Environment Report. [Archivo PDF].* Recuperado de
https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf
- [19] **NASA. (2009).** *Catalog of Earth Satellite Orbits.* Recuperado de
<https://earthobservatory.nasa.gov/features/OrbitsCatalog>
- [20] **ESA. (2020).** *Types of orbits.* Recuperado de
https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits
- [21] **UCS. (2017).** *UCS Satellite Database User's Manual 1-1-17. [Archivo PDF].* Recuperado de
<https://s3.amazonaws.com/ucs-documents/nuclear-weapons/sat-database/4-11-17-update/User+Guide+1-1-17+wAppendix.pdf>
- [22] **Radhika Radhakrishnan, William W. Edmonson, Fatemeh Afghah, Ramón Martínez Rodríguez-Osorio, Frank Pinto, Scott C. Burleigh. (2016).** *Survey of Inter-Satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View.*
<https://doi.org/10.1109/COMST.2016.2564990>
- [23] **Joan A. Ruiz de Azúa, Anna Calveras, Adriano Camps. (2018).** *Internet of Satellites (IoSat): Analysis of Network Models and Routing Protocol Requirements.*
<https://doi.org/10.1109/ACCESS.2018.2823983>
- [24] **Réka Albert, Albert-László Barabási. (2002).** *Statistical mechanics of complex networks.*
<https://doi.org/10.1103/RevModPhys.74.47>
- [25] **Mikail Rubinov, Olaf Sporns. (2009).** *Complex network measures of brain connectivity: Uses and interpretations.*
<https://doi.org/10.1016/j.neuroimage.2009.10.003>

- [26] **Marcus Kaiser. (2011).** *A tutorial in connectome analysis: topological and spatial features of brain networks.*
<https://doi.org/10.1016/j.neuroimage.2011.05.025>
- [27] **Petter Holme, Beom Jun Kim, Chang No Yoon, Seung Kee Han. (2002).** *Attack vulnerability of complex networks.*
<https://doi.org/10.1103/PhysRevE.65.056109>
- [28] **Yilin Shen, Nam P. Nguyen, Ying Xuan, My T. Thai. (2012).** *On the Discovery of Critical Links and Nodes for Assessing Network Vulnerability.*
<https://doi.org/10.1109/TNET.2012.2215882>
- [29] **Iridium. (2022).** *Iridium Network.* Recuperado de
<https://www.iridium.com/network/>
- [30] **SpaceNews. (2013).** *Globalstar Says It Is Poised To Reverse Six-year Slide.* Recuperado de
<https://spacenews.com/35451globalstar-says-it-is-poised-to-reverse-six-year-slide/>
- [31] **E-SAT. Le réseau satellitaire Globalstar.** Recuperado de
<https://www.e-sat.fr/reseaux-equipements/globalstar>
- [32] **Globalstar. (2022).** *Tecnología Satelital impulsada por la Red Satelital Globalstar.* Recuperado de
<https://www.globalstar.com/es-es/about/our-technology>
- [33] **UCS. (2021).** *UCS Satellite Database. [Actualizado a 1 de septiembre de 2021].* Recuperado de
<https://www.ucsusa.org/resources/satellite-database>
- [34] **CelesTrack, T.S. Kelso. (2022).** *Frequently Asked Questions: Two-Line Element Set Format.* Recuperado de
<http://celestrak.com/columns/v04n03/>
- [35] **Space-Track.Org.** *Frequently Asked Questions (FAQ).* Recuperado de
<https://www.space-track.org/documentation#>
- [36] **UCS. About.** Recuperado de
<https://www.ucsusa.org/about>
- [37] **UCS. (2022).** *Changes to the UCS Satellite Database for 1-1-2022 Release. [Archivo PDF].* Recuperado de
<https://www.ucsusa.org/sites/default/files/2022-02/changes-to-satellite-database-for-1-1-2022-release.pdf>
- [38] **Wikipedia. (2022).** *Two-line element set.* Recuperado de
https://en.wikipedia.org/wiki/Two-line_element_set
- [39] **Wikipedia. (2007).** *File:Orbit1.svg.* Recuperado de
<https://en.wikipedia.org/wiki/File:Orbit1.svg>
- [40] **CelesTrack, T.S. Kelso. (2022).** *A New Way to Obtain GP Data (aka TLEs).* Recuperado de
<http://celestrak.com/NORAD/documentation/gp-data-formats.php>

- [41] **CelesTrack, T.S. Kelso. (2022).** *Dr. T.S. Kelso, CelesTrack*. Recuperado de <https://celestrak.com/webmaster.php>
- [42] **CelesTrack, T.S. Kelso.** *CelesTrack*. Recuperado de <https://celestrak.com/>
- [43] **MathWorks.** *MATLAB*. Recuperado de https://es.mathworks.com/products/matlab.html?s_tid=hp_products_matlab
- [44] **T.S. Kelso. (2007).** *Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides*. <https://celestrak.com/publications/AAS/07-127/>
- [45] **Meysam Mahooti. (2020).** *SGP4*. Recuperado de <https://www.mathworks.com/matlabcentral/fileexchange/62013-sgp4>
- [46] **Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter. (2014).** *Multilayer networks*. <https://doi.org/10.1093/comnet/cnu016>
- [47] **Marcello Tomasini. (2015).** *An Introduction to Multilayer Networks*. <http://dx.doi.org/10.13140/RG.2.2.16830.18243>
- [48] **Mikail Rubinov, Olaf Sporns. (2010).** *Brain Connectivity Toolbox*. Recuperado de <https://sites.google.com/site/bctnet/home>
- [49] **CelesTrack, T.S. Kelso. (2021).** *SOCRATES*. Recuperado de <https://celestrak.com/SOCRATES/>
- [50] **ESA.** *Space Debris User Portal*. Recuperado de <https://sdup.esoc.esa.int/>
- [51] **ESA. (2017).** *ESA's MASTER software tool*. Recuperado de https://www.esa.int/ESA_Multimedia/Images/2013/04/ESA_s_MASTER_software_tool
- [52] **Howard D. Curtis. (2004).** *Orbital Mechanics for Engineering Students*. <https://archive.org/details/p020160718380095698873>
- [53] **CelesTrack, T.S. Kelso.** *NORAD General Perturbation Data Sets Historical Archives. Special Data Request Form*. Recuperado de <https://celestrak.com/NORAD/archives/request.php>
- [54] **MathWorks.** *Parallel Computing Toolbox*. Recuperado de <https://es.mathworks.com/products/parallel-computing.html>
- [55] **M.A.Sharafa, M.E.Awadb, I.A.Hassanc, R.Ghoneimd, W.N.Ahmed. (2012).** *Visual contact for two satellites orbits under J_2 -gravity*. <https://doi.org/10.1016/j.nrjag.2012.11.007>
- [56] **ESA, Navipedia. (2011).** *Reference Frames in GNSS*. Recuperado de https://gssc.esa.int/navipedia/index.php/Reference_Frames_in_GNSS

- [57] **David Puig Puig. (2019).** *Study of the interlink between small satellites in a constellation [Tesis de Grado]*. Universitat Politècnica de Catalunya.
<http://hdl.handle.net/2117/171092>
- [58] **Wikipedia. (2022).** *Pearson's chi-squared test*. Recuperado de
https://en.wikipedia.org/wiki/Pearson's_chi-squared_test
- [59] **Dahua Lin. (2022).** *Sampling from a discrete distribution*. Recuperado de
<https://www.mathworks.com/matlabcentral/fileexchange/21912-sampling-from-a-discrete-distribution>
- [60] **James Meldrum. (2022).** *CHI2TEST2*. Recuperado de
<https://www.mathworks.com/matlabcentral/fileexchange/41025-chi2test2>
- [61] **Wikipedia. (2022).** *p-value*. Recuperado de
<https://en.wikipedia.org/wiki/P-value>

PARTE II



OBJETIVOS DE DESARROLLO SOSTENIBLE



Capítulo 1

Alineamiento con los Objetivos de Desarrollo Sostenible

ESTE Proyecto tiene como objetivo último la creación de un modelo de análisis de vulnerabilidades de una red de satélites para dar una respuesta anticipada ante la creación de constelaciones basadas en el Internet de Satélites. Para ello se han creado varios programas en MATLAB que permiten representar la red en el espacio y el tiempo, determinar los satélites más críticos con su orden de eliminación y analizar la validez de los modelos empleados.

Puesto que los programas se muestran de forma abierta y su finalidad es mejorar la eficiencia de elementos que afectan de forma material a las personas, siendo estos los satélites, ya que orbitan alrededor de la superficie terrestre, el Proyecto se encuentra en concordancia con algunos de los «Objetivos de Desarrollo Sostenible». Estos son diecisiete objetivos, adoptados por los Estados Miembros de la ONU en 2015, cuyo cumplimiento está planeado para 2030. Los mismos se muestran en la siguiente figura:



Figura 97. Objetivos de Desarrollo Sostenible

El Proyecto esta relacionado con los siguientes Objetivos:

- **4 Educación de calidad.**
- **9 Industria, innovación e infraestructura.**
- **12 Producción y consumo responsables.**

1.1. 4 Educación de calidad

El cuarto Objetivo tiene como finalidad principal «garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos», siendo esta una de las claves para permitir la movilidad socioeconómica. Entre las metas del mismo se encuentran:

- «4.3 De aquí a 2030, asegurar el acceso igualitario de todos los hombres y las mujeres a una formación técnica, profesional y superior de calidad, incluida la enseñanza universitaria»
- «4.4 De aquí a 2030, aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento»
- «4.7 De aquí a 2030, asegurar que todos los alumnos adquieran los conocimientos teóricos y prácticos necesarios para promover el desarrollo sostenible, entre otras cosas mediante la educación para el desarrollo sostenible y los estilos de vida sostenibles, los derechos humanos, la igualdad de género, la promoción de una cultura de paz y no violencia, la ciudadanía mundial y la valoración de la diversidad cultural y la contribución de la cultura al desarrollo sostenible»

Al ofrecer varios programas de código abierto y libre acceso en Internet, con sus debidas explicaciones de funcionamiento, este Proyecto se encuentra relacionado con los apartados presentados con anterioridad. Siguiendo las descripciones detalladas en la memoria, así como los comentarios presentes en el código, es posible entender los principios de funcionamiento de los programas y la teoría que los sustenta.

El nivel de complejidad y detalle de los mismos los hace compatibles con el aprendizaje relacionado asociado a ciertos estudios técnicos, sirviendo alguno de sus programas como plataforma sobre la que desarrollar otros proyectos de forma sencilla. Además, la memoria cuenta con una sección bibliográfica completa, gracias a la que se puede profundizar en todos los conocimientos presentados en la misma.

1.2. 9 Industria, innovación e infraestructura

El noveno Objetivo tiene como finalidad principal «construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación». Estas tienen una repercusión directa en una utilización eficiente de los recursos materiales del planeta. De forma adicional y debido a la aparición de la COVID-19, las tecnologías de la información se encontraron dentro de las principales formas de respuesta ante el problema del aislamiento; para hacer uso de las mismas es necesario desarrollar infraestructuras específicas, encontrándose directamente relacionadas con este Objetivo. Entre las metas del mismo se encuentran:

- «9.4 De aquí a 2030, modernizar la infraestructura y reconvertir las industrias para que sean sostenibles, utilizando los recursos con mayor eficacia y promoviendo la adopción de tecnologías y procesos industriales limpios y ambientalmente racionales, y logrando que todos los países tomen medidas de acuerdo con sus capacidades respectivas»
- «9.5 Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación y aumentando considerablemente, de aquí a 2030, el número de personas que trabajan en investigación y desarrollo por millón de habitantes y los gastos de los sectores público y privado en investigación y desarrollo»
- «9.c Aumentar significativamente el acceso a la tecnología de la información y las comunicaciones y esforzarse por proporcionar acceso universal y asequible a Internet en los países menos adelantados de aquí a 2020»

Gracias a los procesos presentados en este Proyecto es posible determinar aquellos satélites de una red que van a necesitar mayores requerimientos de capacidad de procesamiento y conexión. Con ello es posible crear redes dinámicas en el tiempo capaces de compartir recursos, aunque procedan de sistemas diferentes. Esto permite comunicaciones más eficientes y constituye una de las herramientas capaces de facilitar la aparición de este tipo de comunicaciones.

1.3. 12 Producción y consumo responsables

El duodécimo Objetivo tiene como finalidad principal «garantizar modalidades de consumo y producción sostenibles». Los motivos tras esto son los recursos limitados así como el mantenimiento del medio ambiente. Entre las metas del mismo se encuentran:

- «12.2 De aquí a 2030, lograr la gestión sostenible y el uso eficiente de los recursos naturales»
- «12.5 De aquí a 2030, reducir considerablemente la generación de desechos mediante actividades de prevención, reducción, reciclado y reutilización»
- «12.8 De aquí a 2030, asegurar que las personas de todo el mundo tengan la información y los conocimientos pertinentes para el desarrollo sostenible y los estilos de vida en armonía con la naturaleza»

En el caso de este Proyecto, una mayor eficiencia en las comunicaciones entre satélites implica un menor uso de los recursos necesarios para producirlos, menores cantidades de energía para situarlos y una menor cantidad de cuerpos y desechos espaciales orbitando alrededor de la Tierra. Esto se traduce en un cielo más despejado, siendo beneficiosos para la investigación científica, la seguridad espacial y el disfrute, por parte de todos los habitantes del planeta, del cielo nocturno.

PARTE III



CÓDIGO FUENTE



Capítulo 1

Obtención de las posiciones de los satélites

1.1. TLE_Satdata.m

```
1 %% TLE_Satdata.m
2
3 function Satdata = TLE_Satdata(TLE_Line1,TLE_Line2)
4 %TLE_Satdata: transform TLE lines into a structure
5 % test_sgp4.m modified. Reference:
6 % Meysam Mahooti (2021). SGP4 (https://www.mathworks.com/matlabcentral/fileexchange/62013-sgp4), MATLAB Central File Exchange. Retrieved October 23, 2021.
7 format long g
8
9 ge = 398600.8; % Earth gravitational constant
10 TWOPI = 2*pi;
11 MINUTES_PER_DAY = 1440;
12 MINUTES_PER_DAY_SQUARED = (MINUTES_PER_DAY * MINUTES_PER_DAY);
13 MINUTES_PER_DAY_CUBED = (MINUTES_PER_DAY * MINUTES_PER_DAY_SQUARED);
14
15 % 19-32 04236.56031392 Element Set Epoch (UTC)
16 % 3-7 25544 Satellite Catalog Number
17 % 9-16 51.6335 Orbit Inclination (degrees)
18 % 18-25 344.7760 Right Ascension of Ascending Node (degrees)
19 % 27-33 0007976 Eccentricity (decimal point assumed)
20 % 35-42 126.2523 Argument of Perigee (degrees)
21 % 44-51 325.9359 Mean Anomaly (degrees)
22 % 53-63 15.70406856 Mean Motion (revolutions/day)
23 % 64-68 32890 Revolution Number at Epoch
24
25 % read first line
26 tline = TLE_Line1;
27 Cnum = tline(3:7); % Catalog Number (NORAD)
28 SC = tline(8); % Security Classification
29 ID = tline(10:17); % Identification Number
30 year = str2num(tline(19:20)); % Year
31 doy = str2num(tline(21:32)); % Day of year
32 epoch = str2num(tline(19:32)); % Epoch
33 TD1 = str2num(tline(34:43)); % first time derivative
34 TD2 = str2num(tline(45:50)); % 2nd Time Derivative
35 ExTD2 = tline(51:52); % Exponent of 2nd Time Derivative
36 BStar = str2num(tline(54:59)); % Bstar/drag Term
37 ExBStar = str2num(tline(60:61)); % Exponent of Bstar/drag Term
38 BStar = BStar*1e-5*10^ExBStar;
39 Etype = tline(63); % Ephemeris Type
40 Enum = str2num(tline(65:end)); % Element Number
41
42 % read second line
43 tline = TLE_Line2;
44 i = str2num(tline(9:16)); % Orbit Inclination (degrees)
45 raan = str2num(tline(18:25)); % Right Ascension of Ascending Node (degrees)
46 e = str2num(strcat('0.',tline(27:33))); % Eccentricity
47 omega = str2num(tline(35:42)); % Argument of Perigee (degrees)
```

```

48 M = str2num(tline(44:51)); % Mean Anomaly (degrees)
49 no = str2num(tline(53:63)); % Mean Motion
50 a = ( ge/(no*2*pi/86400)^2 )^(1/3); % semi major axis (m)
51 rNo = str2num(tline(65:end)); % Revolution Number at Epoch
52
53 satdata.epoch = epoch;
54 satdata.norad_number = Cnum;
55 satdata.bulletin_number = ID;
56 satdata.classification = SC; % almost always 'U'
57 satdata.revolution_number = rNo;
58 satdata.ephemeris_type = Etype;
59 satdata.xmo = M * (pi/180);
60 satdata.xnodeo = raan * (pi/180);
61 satdata.omegao = omega * (pi/180);
62 satdata.xincl = i * (pi/180);
63 satdata.eo = e;
64 satdata.xno = no * TWOPI / MINUTES_PER_DAY;
65 satdata.xndt2o = TD1 * 1e-8 * TWOPI / MINUTES_PER_DAY_SQUARED;
66 satdata.xnnd6o = TD2 * TWOPI / MINUTES_PER_DAY_CUBED;
67 satdata.bstar = BStar;
68
69 Satdata = satdata;
70 end

```

1.2. Programa para la obtención de las posiciones de los satélites (Nube.m)

```

1 %% Nube.m
2 % Antonio Serda Mena. (2022). Application of Complex Network Theory to the Vulnerability
   Analysis of a Satellite Network
3 % Obtains satellites positions
4
5 %% Considerations
6 % Parallel Computing Toolbox needed for faster execution: parfor loops used
7 % SPG4 folder needed
8 % Reference: Meysam Mahooti (2021). SGP4 (https://www.mathworks.com/matlabcentral/fileexchange/62013-sgp4), MATLAB Central File Exchange. Retrieved October 23, 2021.
9 % SPG4 -> Add to Path -> Selected Folders and Subfolders
10 % SatelliteTLE -> Add to Path -> Selected Folders and Subfolders
11
12 %% Initialization
13 close all force
14 clear
15 clc
16 tic; % Starts clock to measure time of execution
17
18 %% Functioning parameters
19
20 % List of wanted satellites [1]
21 list_title = 'Globalstar.txt';
22
23 % Date interval selection
24 % Initial [2]
25 Date_Cloud.s_0 = 0; % Second 0-59
26 Date_Cloud.m_0 = 0; % Minute 0-59
27 Date_Cloud.h_0 = 0; % Hour 0-23
28 Date_Cloud.d_0 = 1; % Day 1-28/29/30/31 depending on the month
29 Date_Cloud.mo_0 = 9; % Month 1-12
30 Date_Cloud.y_0 = 2021; % Year 1957-2056
31 % Final [3]
32 Date_Cloud.s_f = 0; % Second 0-59
33 Date_Cloud.m_f = 0; % Minute 0-59
34 Date_Cloud.h_f = 0; % Hour 0-23
35 Date_Cloud.d_f = 2; % Day 1-28/29/30/31 depending on the month
36 Date_Cloud.mo_f = 9; % Month 1-12
37 Date_Cloud.y_f = 2021; % Year 1957-2056
38
39 % Time period between positions [minutes] [4]
40 period = 6;

```

```

41
42 %% Initial adjustments
43
44 % List of wanted satellites [1]
45 list = fopen(list_title); % Open text file
46 NORAD_Cloud_Struct = textscan(list, '%s'); % Text scan data
47 fclose(list) ; % Close file
48 n_Sat_Cloud = size(NORAD_Cloud_Struct(1,1),1); % Number of satellites wanted
49 NORAD_Cloud = sort(NORAD_Cloud_Struct(1,1)); % Sort NORAD numbers of wanted satellites
50
51 % Date interval selection
52 % Initial [2]
53 Date_Cloud.Epoch_Day_0 = round(datenum(Date_Cloud.y_0,Date_Cloud.mo_0,Date_Cloud.d_0,
    Date_Cloud.h_0,Date_Cloud.m_0,Date_Cloud.s_0) - datenum(Date_Cloud.y_0,0,0,0,0,0),8); %
    Day of the year epoch
54 Date_Cloud.Full_Epoch_0 = Date_Cloud.y_0*1000 + Date_Cloud.Epoch_Day_0;
55 Date_Cloud.Date_0 = datetime(Date_Cloud.y_0,Date_Cloud.mo_0,Date_Cloud.d_0,Date_Cloud.h_0,
    Date_Cloud.m_0,Date_Cloud.s_0);
56 % Final [3]
57 Date_Cloud.Epoch_Day_f = round(datenum(Date_Cloud.y_f,Date_Cloud.mo_f,Date_Cloud.d_f,
    Date_Cloud.h_f,Date_Cloud.m_f,Date_Cloud.s_f) - datenum(Date_Cloud.y_f,0,0,0,0,0),8); %
    Day of the year epoch
58 Date_Cloud.Full_Epoch_f = Date_Cloud.y_f*1000 + Date_Cloud.Epoch_Day_f;
59 Date_Cloud.Date_f = datetime(Date_Cloud.y_f,Date_Cloud.mo_f,Date_Cloud.d_f,Date_Cloud.h_f,
    Date_Cloud.m_f,Date_Cloud.s_f);
60 % Time interval
61 Date_Cloud.t_max = split(between(Date_Cloud.Date_0,Date_Cloud.Date_f, 'Time'),'Time');
62
63 % Date comprobation
64 if Date_Cloud.t_max <= 0
65     disp('Incorrect date interval')
66     pause
67 end
68
69 % Time period between positions [minutes] [4]
70 T_Cloud_minutes = minutes(period); % [minutes]
71
72 % Date grid array
73 d_grid_Cloud = Date_Cloud.Date_0:T_Cloud_minutes:Date_Cloud.Date_f;
74 d_grid_Cloud = d_grid_Cloud';
75 n_d_grid_Cloud = size(d_grid_Cloud,1);
76
77 %% TLE loading
78
79 % Generate NORAD names column
80 for i=1:n_Sat_Cloud
81     Sat_Cloud(i).NORAD = str2double(NORAD_Cloud(i,1)); % NORAD names column (double)
82 end
83
84 % Generate .txt archive names column
85 for i=1:n_Sat_Cloud
86     Sat_Cloud(i).txt_Names = strcat('sat',char(NORAD_Cloud(i,1)),'.txt'); % .txt archive names
        column (char)
87 end
88
89 % Store TLE of each satellite into a cell column of cell columns
90 for i=1:n_Sat_Cloud
91     list = fopen(Sat_Cloud(i).txt_Names); % Open text file
92     Sat_Cloud(i).TLE = textscan(list, '%s','Delimiter','\n'); % Text scan data
93     fclose(list) ; % Close file
94     Sat_Cloud(i).TLE = Sat_Cloud(i).TLE{1,1}; % TLE cell column of cell columns
95 end
96
97 % Select TLE data according to the date interval selection
98 for i=1:n_Sat_Cloud
99     n_TLE = size(Sat_Cloud(i).TLE,1);
100     j = 1;
101     k = 1;
102     flag = 0; % Activated if TLE dates are over maximum interval date
103     while j <= n_TLE && flag == 0
104         % Load the odd rows of each TLE
105         line = Sat_Cloud(i).TLE{j,1};
106
107         % Full year of the TLE
108         line_year_2d = str2num(line(19:20)); % Obtain the last 2 digits of the TLE

```

```

109     if line_year_2d > 57 % Year between 1957 and 1999
110         line_year = 1900 + line_year_2d;
111     else % Year between 2000 and 2056
112         line_year = 2000 + line_year_2d;
113     end
114
115     % Day
116     line_day = str2num(line(21:32));
117
118     % Create a structure with the TLE data if the epoch is contained in the date interval
119     if line_year >= Date_Cloud.y_0 && line_day >= Date_Cloud.Epoch_Day_0
120         if line_year <= Date_Cloud.y_f && line_day <= Date_Cloud.Epoch_Day_f
121             % TLE obtention year
122             Sat_Cloud(i).TLE_Data(k).Year = line_year;
123
124             % TLE obtention day of the year
125             Sat_Cloud(i).TLE_Data(k).Day = line_day;
126
127             % Date of TLE obtention
128             [mon,day,hr,minute,sec] = days2mdh(line_year,line_day);
129             Sat_Cloud(i).TLE_Data(k).Date = datetime(line_year,mon,day,hr,minute,sec);
130
131             % Time difference from initial date
132             Sat_Cloud(i).TLE_Data(k).Diff_minutes = minutes(split(between(Date_Cloud.
                Date_0,Sat_Cloud(i).TLE_Data(k).Date,'time'),'time')));
133
134             % Structure of the data obtained from the TLE for each epoch
135             TLE_Line1 = Sat_Cloud(i).TLE{j,1};
136             TLE_Line2 = Sat_Cloud(i).TLE{j+1,1};
137             Sat_Cloud(i).TLE_Data(k).satdata = TLE_Satdata(TLE_Line1,TLE_Line2);
138
139             k = k+1;
140         else
141             flag = 1;
142         end
143     end
144     j = j+2;
145 end
146 end
147
148 % Position and velocity in xyz for each TLE of each satellite
149 parfor i=1:n_Sat_Cloud % Parallel Computing Toolbox needed: it allows a faster execution.
    Change "parfor" by "for" if Parallel Computing Toolbox is not installed
150     for j=1:1:size(Sat_Cloud(i).TLE_Data,2)
151         [pos, vel] = sgp4(0,Sat_Cloud(i).TLE_Data(j).satdata);
152         Sat_Cloud(i).TLE_Data(j).Position_xyz = pos; % [km]
153         Sat_Cloud(i).TLE_Data(j).Velocity_xyz = vel; % [km/h]
154     end
155 end
156
157 %% Detection of displaced satellites (Optional)
158
159 % Time between TLE of the same satellite [minutes]
160 for i=1:n_Sat_Cloud
161     for j=2:1:size(Sat_Cloud(i).TLE_Data,2)
162         Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes = minutes(split(between(Sat_Cloud(i).
            TLE_Data(j-1).Date,Sat_Cloud(i).TLE_Data(j).Date,'time'),'time')));
163     end
164 end
165
166 % Expected position [km] and velocity [km/h] in xyz for according to the previous TLE of each
    satellite
167 k = 1;
168 for i=1:n_Sat_Cloud
169     for j=2:1:size(Sat_Cloud(i).TLE_Data,2)
170         % Expected position and velocity
171         [pos, vel] = sgp4(Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes,Sat_Cloud(i).TLE_Data(j-1)
            .satdata);
172         Sat_Cloud(i).TLE_Data(j).Exp_Position_xyz = pos; % [km]
173         Sat_Cloud(i).TLE_Data(j).Exp_Velocity_xyz = vel; % [km/h]
174         Sat_Cloud(i).TLE_Data(j).Error_Position_km = norm(Sat_Cloud(i).TLE_Data(j).
            Position_xyz - Sat_Cloud(i).TLE_Data(j).Exp_Position_xyz); % [km]
175         Sat_Cloud(i).TLE_Data(j).Error_Velocity_kmh = norm(Sat_Cloud(i).TLE_Data(j).
            Velocity_xyz - Sat_Cloud(i).TLE_Data(j).Exp_Velocity_xyz); % [km/h]
176

```

III. CÓDIGO FUENTE 1. OBTENCIÓN DE LAS POSICIONES DE LOS SATÉLITES

```

177     % Simplified position error function (change function for TLE_Diff_minutes > 1440
178     minutes [1 day])
179     Error_Max_km = Sat_Cloud(i).TLE_Data(j).TLE_Diff_minutes*(5/1440); % [km] 5 km of
180     maximum error / day (1440 minutes)
181     % Location index matrix for displaced satellites
182     if Sat_Cloud(i).TLE_Data(j).Error_Position_km > Error_Max_km
183         Error_Index(k,1) = i; % Satellite index
184         Error_Index(k,2) = j; % TLE index
185         k = k+1;
186     end
187 end
188 %% Position in date grid
189 % Date collumn
190 for i=1:n_Sat_Cloud
191     for j=1:1:n_d_grid_Cloud
192         Sat_Cloud(i).Position(j).Date = d_grid_Cloud(j,1);
193     end
194 end
195 % Convert date to text to simplify data view
196 for i=1:n_Sat_Cloud
197     for j=1:1:n_d_grid_Cloud
198         Sat_Cloud(i).Position(j).Date_Char = datestr(Sat_Cloud(i).Position(j).Date);
199     end
200 end
201 % Position and velocity
202 parfor i=1:n_Sat_Cloud % Parallel Computing Toolbox needed: it allows a faster execution.
203     Change "parfor" by "for" if Parallel Computing Toolbox is not installed
204     k = 1; % Backwards index
205     r = 1; % Forward index
206     for j=1:1:n_d_grid_Cloud
207         % Time between current date in the grid and the TLE that was obtained after that date
208         % (time origin in TLE date)
209         tb = minutes(split(between(Sat_Cloud(i).TLE_Data(k).Date,Sat_Cloud(i).Position(j).Date
210         , 'time'), 'time')));
211
212         % Time between current date in the grid and the TLE that was obtained before that date
213         % (time origin in TLE date)
214         tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(j).Date
215         , 'time'), 'time')));
216
217         % If the tb is positive that means that the current date in the grid is after the TLE,
218         % so it changes the time origin to the next satellite's TLE
219         if tb > 0 && k == size(Sat_Cloud(i).TLE_Data,2) && r < size(Sat_Cloud(i).TLE_Data,2)
220             r = r+1;
221             tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(j).
222             Date, 'time'), 'time')));
223         end
224         if tb > 0 && k < size(Sat_Cloud(i).TLE_Data,2)
225             if k > 1
226                 r = r+1;
227             end
228             k = k+1;
229             tb = minutes(split(between(Sat_Cloud(i).TLE_Data(k).Date,Sat_Cloud(i).Position(j).
230             Date, 'time'), 'time')));
231             tf = minutes(split(between(Sat_Cloud(i).TLE_Data(r).Date,Sat_Cloud(i).Position(j).
232             Date, 'time'), 'time')));
233         end
234
235         % Position and velocity according to tb
236         [posb, velb] = sgp4(tb,Sat_Cloud(i).TLE_Data(k).satdata);
237         Sat_Cloud(i).Position(j).Time_interval_Backwards = tb;
238         Sat_Cloud(i).Position(j).Position_Backwards = posb;
239         Sat_Cloud(i).Position(j).Velocity_Backwards = velb;
240
241         % Position and velocity according to tf
242         [posf, velf] = sgp4(tf,Sat_Cloud(i).TLE_Data(r).satdata);
243         Sat_Cloud(i).Position(j).Time_interval_Forward = tf;
244         Sat_Cloud(i).Position(j).Position_Forward = posf;
245         Sat_Cloud(i).Position(j).Velocity_Forward = velf;

```

```

241
242     % Mean position and velocity according
243     pos = (abs(tb)*posf + abs(tf)*posb)/(abs(tb) + abs(tf)); % Inverse leveraging between
           time and position
244     vel = (abs(tb)*velf + abs(tf)*velb)/(abs(tb) + abs(tf)); % Inverse leveraging between
           time and velocity
245     Sat_Cloud(i).Position(j).Position_Mean = pos;
246     Sat_Cloud(i).Position(j).Velocity_Mean = vel;
247     end
248 end
249
250 %% Saving of workspace variables
251 save('Cloud_Variables', 'n_Sat_Cloud', 'Date_Cloud', 'T_Cloud_minutes', 'Sat_Cloud');
252
253 %% Plot (Optional)
254
255 % Manual time interval selection
256 j_min = 1; % Minimum date index
257 j_max = 20; % Maximum date index
258 j_period = 1; % Sampling period index
259
260 % 1 satellite: manual time interval selection
261 i = 1; % Satellite index
262 for j=j_min:j_period:j_max
263     x1(j) = Sat_Cloud(1).Position(j).Position_Mean(1,1);
264     y1(j) = Sat_Cloud(1).Position(j).Position_Mean(2,1);
265     z1(j) = Sat_Cloud(1).Position(j).Position_Mean(3,1);
266 end
267 figure
268 plot3(x1,y1,z1,'-o');
269 title('Coordinates of the satellite 25652 during day 01/09/2021 (00:00-01:54 ~ 1 orbit)')
270 xlabel('x [km]')
271 ylabel('y [km]')
272 zlabel('z [km]')
273
274 % Constellation: manual time interval selection
275 figure
276 for i=1:1:n_Sat_Cloud
277     for j=j_min:j_period:j_max
278         x2(i,j) = Sat_Cloud(i).Position(j).Position_Mean(1,1);
279         y2(i,j) = Sat_Cloud(i).Position(j).Position_Mean(2,1);
280         z2(i,j) = Sat_Cloud(i).Position(j).Position_Mean(3,1);
281     end
282     if i > 1
283         hold on
284     end
285     plot3(x2(i,:),y2(i,:),z2(i,:),'-o');
286 end
287 hold off
288 title('Coordinates of the Globalstar constellation during day 01/09/2021 (00:00-01:54 ~ 1
           orbit)')
289 xlabel('x [km]')
290 ylabel('y [km]')
291 zlabel('z [km]')
292
293 % 1 satellite: whole time interval
294 i = 1; % Satellite index
295 for j=1:j_period:n_d_grid_Cloud
296     x3(j) = Sat_Cloud(1).Position(j).Position_Mean(1,1);
297     y3(j) = Sat_Cloud(1).Position(j).Position_Mean(2,1);
298     z3(j) = Sat_Cloud(1).Position(j).Position_Mean(3,1);
299 end
300 figure
301 plot3(x3,y3,z3,'-o');
302 title('Coordinates of the satellite 25652 during day 01/09/2021')
303 xlabel('x [km]')
304 ylabel('y [km]')
305 zlabel('z [km]')
306
307 % Constellation: whole time interval
308 figure
309 for i=1:1:n_Sat_Cloud
310     for j=1:j_period:n_d_grid_Cloud
311         x4(i,j) = Sat_Cloud(i).Position(j).Position_Mean(1,1);
312         y4(i,j) = Sat_Cloud(i).Position(j).Position_Mean(2,1);

```



```
313         z4(i,j) = Sat_Cloud(i).Position(j).Position_Mean(3,1);
314     end
315     if i > 1
316         hold on
317     end
318     plot3(x4(i,:),y4(i,:),z4(i,:), '-o');
319 end
320 hold off
321 title('Coordinates of the Globalstar constellation during day 01/09/2021')
322 xlabel('x [km]')
323 ylabel('y [km]')
324 zlabel('z [km]')
325
326 %% Stops clock to measure time of execution
327 toc
```


Capítulo 2

Análisis de vulnerabilidades de la red de satélites

2.1. Programa para el análisis de vulnerabilidades de la red de satélites (Red.m)

```
1 %% Red.m
2 % Antonio Serda Mena. (2022). Application of Complex Network Theory to the Vulnerability
  Analysis of a Satellite Network
3 % Analyzes the satellite network using Complex Network Theory
4
5 %% Considerations
6 % Parallel Computing Toolbox needed for faster execution: parfor loops used
7 % Functions from the Brain Connectivity Toolbox needed:
8 % clustering_coef_bu.m
9 % clustering_coef_wu.m
10 % Reference: Complex network measures of brain connectivity: Uses and interpretations. Rubinov
  M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
11 % Files: https://sites.google.com/site/bctnet/
12 % It is needed to have satellite position first, using the program Nube.mat
13
14 %% Initialization
15 close all force
16 clear
17 clc
18 tic; % Starts clock to measure time of execution
19
20 %% Functioning parameters
21
22 % Loading of satellite positions and related variables
23 load('Cloud_Variables');
24
25 % List of wanted satellites [1]
26 list_title = 'Globalstar.txt';
27
28 % Date interval selection
29 % Initial [2]
30 Date_Network.s_0 = 0; % Second 0-59
31 Date_Network.m_0 = 0; % Minute 0-59
32 Date_Network.h_0 = 0; % Hour 0-23
33 Date_Network.d_0 = 1; % Day 1-28/29/30/31 depending on the month
34 Date_Network.mo_0 = 9; % Month 1-12
35 Date_Network.y_0 = 2021; % Year 1957-2056
36 % Final [3]
37 Date_Network.s_f = 0; % Second 0-59
38 Date_Network.m_f = 54; % Minute 0-59
39 Date_Network.h_f = 1; % Hour 0-23
40 Date_Network.d_f = 1; % Day 1-28/29/30/31 depending on the month
41 Date_Network.mo_f = 9; % Month 1-12
42 Date_Network.y_f = 2021; % Year 1957-2056
43
44 % Time period between positions multiplum [4]
```

```

45 mult_period = 1; % Time between measures must be a multiplum of the time grid interval of the
    cloud (must be an integer > 1)
46
47 % Maximum number of iterations [5]
48 p = 1; % Portion of the total number of satellites to elliminate (from 0 to 1)
49
50 % Method for determining vulnerability [6]
51 method = 1;
52 % 1 -> Max betweenes centrality
53 % 2 -> Max node degree
54 % 3 -> Min local clustering coefficient
55 % 4 -> Max node efficiency
56
57 %Astronomical constants [7]
58 R_Earth = 6378.137; % Earth's radius [km] (WGS-84)
59 dR = 60; % Extra radius [km]
60
61 %% Initial adjustments
62
63 % List of wanted satellites [1]
64 list = fopen(list_title); % Open text file
65 NORAD_Network_Struct = textscan(list, '%s'); % Text scan data
66 fclose(list); % Close file
67 n_Sat = size(NORAD_Network_Struct{1,1},1); % Number of satellites wanted
68 NORAD = sort(NORAD_Network_Struct{1,1}); % Sort NORAD numbers of wanted satellites
69 NORAD_Double = str2double(NORAD);
70 NORAD_Network = NORAD;
71 NORAD_Monoplex = NORAD; % NORAD numbers of wanted satellites for a graph representing the
    whole period
72 n_Sat_Network = n_Sat;
73
74 % Date interval selection
75 % Initial [2]
76 Date_Network.Epoch_Day_0 = round(datum(Date_Network.y_0,Date_Network.mo_0,Date_Network.d_0,
    Date_Network.h_0,Date_Network.m_0,Date_Network.s_0) - datum(Date_Network.y_0,0,0,0,0,0)
    ,8); % Day of the year epoch
77 Date_Network.Full_Epoch_0 = Date_Network.y_0*1000 + Date_Network.Epoch_Day_0;
78 Date_Network.Date_0 = datetime(Date_Network.y_0,Date_Network.mo_0,Date_Network.d_0,
    Date_Network.h_0,Date_Network.m_0,Date_Network.s_0);
79 % Final [3]
80 Date_Network.Epoch_Day_f = round(datum(Date_Network.y_f,Date_Network.mo_f,Date_Network.d_f,
    Date_Network.h_f,Date_Network.m_f,Date_Network.s_f) - datum(Date_Network.y_f,0,0,0,0,0)
    ,8); % Day of the year epoch
81 Date_Network.Full_Epoch_f = Date_Network.y_f*1000 + Date_Network.Epoch_Day_f;
82 Date_Network.Date_f = datetime(Date_Network.y_f,Date_Network.mo_f,Date_Network.d_f,
    Date_Network.h_f,Date_Network.m_f,Date_Network.s_f);
83 % Time interval
84 Date_Network.t_max = split(between(Date_Network.Date_0,Date_Network.Date_f,'time'),'time');
85
86 % Date comprobation
87 if Date_Network.t_max <= 0 || Date_Network.Date_0 < Date_Cloud.Date_0 || Date_Network.Date_f >
    Date_Cloud.Date_f
88     disp('Incorrect date interval')
89     pause
90 end
91
92 % Time period between positions [minutes] [4]
93 if mult_period < 1
94     disp('Incorrect multiplum')
95     pause
96 end
97 T_Network_minutes = T_Cloud_minutes*round(mult_period); % [minutes]
98
99 % Date grid array
100 t_dif_minutes = minutes(minutes(split(between(Date_Cloud.Date_0,Date_Network.Date_0,'Time'),'
    Time')));
101 t_dif_adjusted_minutes = T_Network_minutes*ceil(datum(t_dif_minutes)/datum(
    T_Network_minutes));
102 d_Network_0 = Date_Cloud.Date_0 + t_dif_adjusted_minutes;
103 d_grid_Network = d_Network_0:T_Network_minutes:Date_Network.Date_f;
104 d_grid_Network = d_grid_Network';
105 n_d_grid_Network = size(d_grid_Network,1);
106
107 % Set maximum number of iterations [5]
108 it_max = round(n_Sat*p); % Translates the percentage of eliminated satellites into an integer
    
```

```

109
110 %% Sat_Network initialization
111
112 % Date collumn
113 for i=1:n_d_grid_Network
114     Sat_Network(i).Date = d_grid_Network(i);
115 end
116
117 % Convert date to text to simplify data view
118 for i=1:n_d_grid_Network
119     Sat_Network(i).Date_Char = datestr(Sat_Network(i).Date);
120 end
121
122 % Generate NORAD names column for each date
123 for i=1:n_d_grid_Network
124     for j=1:n_Sat_Network
125         Sat_Network(i).Nodes(j).NORAD = NORAD_Double(j,1); % NORAD names column (double)
126     end
127 end
128
129 % Position of each satellite for each date
130 for i=1:n_d_grid_Network
131     for j=1:n_Sat_Network
132         for k=1:n_Sat_Cloud
133             if Sat_Cloud(k).NORAD == Sat_Network(i).Nodes(j).NORAD
134                 for m=1:round(mult_period):n_d_grid_Cloud
135                     if Sat_Cloud(k).Position(m).Date == Sat_Network(i).Date
136                         Sat_Network(i).Nodes(j).Position_Mean = Sat_Cloud(k).Position(m).
                            Position_Mean;
137                     end
138                 end
139             end
140         end
141     end
142 end
143
144 % Adjacency matrix for each date for the initial state
145 parfor i=1:n_d_grid_Network
146     for j=1:n_Sat_Network
147         x1 = Sat_Network(i).Nodes(j).Position_Mean(1,1);
148         y1 = Sat_Network(i).Nodes(j).Position_Mean(2,1);
149         z1 = Sat_Network(i).Nodes(j).Position_Mean(3,1);
150         r1_km = sqrt(x1^2 + y1^2 + z1^2); % Satellite height
151         for k=1:n_Sat_Network
152             % Triangle between Earth's center and 2 satellites
153             x2 = Sat_Network(i).Nodes(k).Position_Mean(1,1);
154             y2 = Sat_Network(i).Nodes(k).Position_Mean(2,1);
155             z2 = Sat_Network(i).Nodes(k).Position_Mean(3,1);
156             r2_km = sqrt(x2^2 + y2^2 + z2^2); % Height of the rest of the satellites
157             d_km = sqrt((x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2); % Distance between satellites
158             sp_km = (r1_km + r2_km + d_km)/2; % Triangle semiperimeter
159             h_km = (2/d_km)*sqrt(sp_km*(sp_km-d_km)*(sp_km-r1_km)*(sp_km-r2_km)); % Triangle
                            height formula
160
161             % If triangle's height is higher than Earth's radius (+60 km extra) satellites are
                            visible [7]
162             if h_km >= R_Earth + dR
163                 Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 1;
164             else
165                 Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 0;
166             end
167         end
168     end
169 end
170
171 %% Sat_Network completion and Sat_Multiplex
172
173 for it=1:it_max
174     %% Sat_Network structure
175
176     % Changes for each iteration
177     if it>1
178         % Changes in NORAD names list
179         Sat_Multiplex(it).Missed_Satellite = NORAD_Network(s,:); % Missed satellite
180         NORAD_Network(s,:) = [];
    
```

```

181
182     % Changes in indexes
183     n_Sat_Network = n_Sat_Network-1;
184
185     % Changes in adjacency matrixes for each date
186     for i=1:n_d_grid_Network
187         Sat_Network(i).Iteration(it).Adjacency_Matrix = Sat_Network(i).Iteration(it-1).
            Adjacency_Matrix;
188         Sat_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
189         Sat_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
190     end
191 end
192
193 % Graph properties of the network for for each date
194 parfor i=1:n_d_grid_Network
195     % Adjacency matrix
196     A = Sat_Network(i).Iteration(it).Adjacency_Matrix;
197
198     % Save satellite network as a graph
199     G = graph(A,NORAD_Network');
200
201     % Node degree
202     nd = degree(G);
203
204     % Number of nodes and edges
205     nn = size(G.Nodes,1);
206     ne = size(G.Edges,1);
207
208     % Shortest path length between nodes matrix
209     D = distances(G);
210
211     % Path length of each node
212     pl = sum(D)'/(nn-1);
213
214     % Characteristic path length of the network
215     cpl = sum(pl)/nn;
216
217     % Efficiency of each node
218     D_Inv = zeros(nn);
219     for j=1:nn
220         for k=1:nn
221             if D(j,k) == 0
222                 D_Inv(j,k) = 0;
223             else
224                 D_Inv(j,k) = 1/D(j,k);
225             end
226         end
227     end
228     e = sum(D_Inv)'/(nn-1);
229
230     % Efficiency of the network
231     ef = sum(e)/nn;
232
233     % Local clustering coefficient
234     % Reference: Complex network measures of brain connectivity: Uses and interpretations.
            Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
            % Files: https://sites.google.com/site/bctnet/
235     lcc = clustering_coef_bu(A);
236
237     % Global clustering coefficient
238     gcl = sum(lcc)/nn;
239
240     % Normalized betweenness centrality
241     b = 2*centrality(G,'betweenness'); % 2*b because network is undirected
242     if nn>2
243         Norm_b = b/((nn-1)*(nn-2));
244     elseif nn==2
245         Norm_b = [0;0]; % If there are only 2 satellites left it assigns centrality to
            both
246     else
247         Norm_b = 0;
248     end
249 end
250
251 % Placement of elements in the structure
252 for j=1:nn

```

```

253     Sat_Network(i).Iteration(it).Satellites(j).NORAD = NORAD_Network(j,1);
254     Sat_Network(i).Iteration(it).Satellites(j).Node_Degree = nd(j,1);
255     Sat_Network(i).Iteration(it).Satellites(j).Path_Length = pl(j,1);
256     Sat_Network(i).Iteration(it).Satellites(j).Efficiency = e(j,1);
257     Sat_Network(i).Iteration(it).Satellites(j).Local_Clustering_Coefficient = lcc(j,1)
        ;
258     Sat_Network(i).Iteration(it).Satellites(j).Betweenness_Centrality = b(j,1);
259     Sat_Network(i).Iteration(it).Satellites(j).Norm_Betweenness_Centrality = Norm_b(j
        ,1);
260     end
261     Sat_Network(i).Iteration(it).Network = G;
262     Sat_Network(i).Iteration(it).Distances = D;
263     Sat_Network(i).Iteration(it).n_Edges = ne;
264     Sat_Network(i).Iteration(it).Characteristic_Path_Length = cpl;
265     Sat_Network(i).Iteration(it).Efficiency = ef;
266     Sat_Network(i).Iteration(it).Global_Clustering_Coefficient = gcl;
267 end
268
269 %% Sat_Multiplex structure
270
271 % Degree, path length, efficiency, local clustering coefficient and betweenness centrality
    matrixes for each satellite and each date
272 Degree = zeros(n_Sat_Network,n_d_grid_Network);
273 PL = zeros(n_Sat_Network,n_d_grid_Network);
274 E = zeros(n_Sat_Network,n_d_grid_Network);
275 LCC = zeros(n_Sat_Network,n_d_grid_Network);
276 Norm_Betweenness_Centrality = zeros(n_Sat_Network,n_d_grid_Network);
277 for i=1:n_d_grid_Network
278     for j=1:n_Sat_Network
279         Degree(j,i) = sum(Sat_Network(i).Iteration(it).Adjacency_Matrix(j,:));
280         PL(j,i) = Sat_Network(i).Iteration(it).Satellites(j).Path_Length;
281         E(j,i) = Sat_Network(i).Iteration(it).Satellites(j).Efficiency;
282         LCC(j,i) = Sat_Network(i).Iteration(it).Satellites(j).Local_Clustering_Coefficient
            ;
283         Norm_Betweenness_Centrality(j,i) = Sat_Network(i).Iteration(it).Satellites(j).
            Norm_Betweenness_Centrality;
284     end
285 end
286
287 % Mean and standar deviation degree, path length, efficiency, local clustering coefficient
    and betweenness centrality for each satellite
288 Av_Degree = zeros(n_Sat_Network,1);
289 Av_PL = zeros(n_Sat_Network,1);
290 Av_E = zeros(n_Sat_Network,1);
291 Av_LCC = zeros(n_Sat_Network,1);
292 Av_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
293 Std_Degree = zeros(n_Sat_Network,1);
294 Std_PL = zeros(n_Sat_Network,1);
295 Std_E = zeros(n_Sat_Network,1);
296 Std_LCC = zeros(n_Sat_Network,1);
297 Std_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
298 for i=1:n_Sat_Network
299     Av_Degree(i,1) = mean(Degree(i,:));
300     Av_PL(i,1) = mean(PL(i,:));
301     Av_E(i,1) = mean(E(i,:));
302     Av_LCC(i,1) = mean(LCC(i,:));
303     Av_Norm_Betweenness_Centrality(i,1) = mean(Norm_Betweenness_Centrality(i,:));
304     Std_Degree(i,1) = std(Degree(i,:));
305     Std_PL(i,1) = std(PL(i,:));
306     Std_E(i,1) = std(E(i,:));
307     Std_LCC(i,1) = std(LCC(i,:));
308     Std_Norm_Betweenness_Centrality(i,1) = std(Norm_Betweenness_Centrality(i,:));
309 end
310
311 % Generate NORAD names column for the whole period
312 for i=1:n_Sat_Network
313     Sat_Multiplex(it).Satellites(i).NORAD = NORAD_Network(i,1); % NORAD names column (
        double)
314 end
315
316 % Place mean and standar deviation degree, path length, efficiency, local clustering
    coefficient and betweenness centrality for each satellite
317 for i=1:n_Sat_Network
318     Sat_Multiplex(it).Satellites(i).Av_Degree = Av_Degree(i,1);
319     Sat_Multiplex(it).Satellites(i).Std_Degree = Std_Degree(i,1);

```

```

320     Sat_Multiplex(it).Satellites(i).Av_Path_Length = Av_PL(i,1);
321     Sat_Multiplex(it).Satellites(i).Std_Path_Length = Std_PL(i,1);
322     Sat_Multiplex(it).Satellites(i).Av_Efficiency = Av_E(i,1);
323     Sat_Multiplex(it).Satellites(i).Std_Efficiency = Std_E(i,1);
324     Sat_Multiplex(it).Satellites(i).Av_LCC = Av_LCC(i,1);
325     Sat_Multiplex(it).Satellites(i).Std_LCC = Std_LCC(i,1);
326     Sat_Multiplex(it).Satellites(i).Av_Norm_Betweenness_Centrality =
        Av_Norm_Betweenness_Centrality(i,1);
327     Sat_Multiplex(it).Satellites(i).Std_Norm_Betweenness_Centrality =
        Std_Norm_Betweenness_Centrality(i,1);
328 end
329
330 % Number of edges, characteristic path length, efficiency and global clustering
    coefficient for each date
331 n_Edges = zeros(n_d_grid_Network,1);
332 CPL = zeros(n_d_grid_Network,1);
333 Efficiency = zeros(n_d_grid_Network,1);
334 GCC = zeros(n_d_grid_Network,1);
335 for i=1:n_d_grid_Network
336     n_Edges(i,1) = Sat_Network(i).Iteration(it).n_Edges;
337     CPL(i,1) = Sat_Network(i).Iteration(it).Characteristic_Path_Length;
338     Efficiency(i,1) = Sat_Network(i).Iteration(it).Efficiency;
339     GCC(i,1) = Sat_Network(i).Iteration(it).Global_Clustering_Coefficient;
340 end
341
342 % Mean and standar deviation number of edges, characteristic path length, efficiency and
    global clustering coefficient for each date
343 Sat_Multiplex(it).Av_n_Edges = mean(n_Edges);
344 Sat_Multiplex(it).Std_n_Edges = std(n_Edges);
345 Sat_Multiplex(it).Av_Characteristic_Path_Length = mean(CPL);
346 Sat_Multiplex(it).Std_Characteristic_Path_Length = std(CPL);
347 Sat_Multiplex(it).Av_Efficiency = mean(Efficiency);
348 Sat_Multiplex(it).Std_Efficiency = std(Efficiency);
349 Sat_Multiplex(it).Av_Global_Clustering_Coefficient = mean(GCC);
350 Sat_Multiplex(it).Std_Global_Clustering_Coefficient = std(GCC);
351
352 % Select method to identify most vulnerable satellite [6]
353 switch(method)
354     case 1 % Satellite with higher centrality
355         s = find(Av_Norm_Betweenness_Centrality==max(Av_Norm_Betweenness_Centrality)); %
            Allows to find more than one index
356         % If 2 satellites have the same centrality the one with higher degree is chosen
357         if length(s) > 1
358             Degree_aux = zeros(1,length(s));
359             for i=1:length(s)
360                 Degree_aux(i) = Av_Degree(s(i));
361             end
362             s = find(Av_Degree==max(Degree_aux));
363             % If 2 satellites have the same degree the one with lower LCC is chosen
364             if length(s) > 1
365                 LCC_aux = zeros(1,length(s));
366                 for i=1:length(s)
367                     LCC_aux(i) = Av_LCC(s(i));
368                 end
369                 s = find(Av_LCC==min(LCC_aux));
370                 % If 2 satellites have the same LCC the first one is chosen
371                 if length(s) > 1
372                     s = s(1);
373                 end
374             end
375         end
376     case 2 % Satellite with higher degree
377         s = find(Av_Degree==max(Av_Degree));
378         % If 2 satellites have the same degree the one with lower LCC is chosen
379         if length(s) > 1
380             LCC_aux = zeros(1,length(s));
381             for i=1:length(s)
382                 LCC_aux(i) = Av_LCC(s(i));
383             end
384             s = find(Av_LCC==min(LCC_aux));
385             % If 2 satellites have the same LCC the first one is chosen
386             if length(s) > 1
387                 s = s(1);
388             end
389         end
390     end

```



```

390 case 3 % Satellite with lower local clustering coefficient
391     s = find(Av_LCC==min(Av_LCC));
392     % If 2 satellites have the same LCC the one with higher degree is chosen
393     if length(s) > 1
394         nd_aux = zeros(1,length(s));
395         for i=1:length(s)
396             nd_aux(i) = Av_Degree(s(i));
397         end
398         s = find(Av_Degree==max(nd_aux));
399         % If 2 satellites have the same degree the first one is chosen
400         if length(s) > 1
401             s = s(1);
402         end
403     end
404 case 4 % Satellite with higher efficiency
405     s = find(Av_Efficiency==max(Av_Efficiency)); % Allows to find more than one index
406     % If 2 satellites have the same efficiency the one with higher degree is chosen
407     if length(s) > 1
408         Degree_aux = zeros(1,length(s));
409         for i=1:length(s)
410             Degree_aux(i) = Av_Degree(s(i));
411         end
412         s = find(Av_Degree==max(Degree_aux));
413         % If 2 satellites have the same degree the one with lower LCC is chosen
414         if length(s) > 1
415             LCC_aux = zeros(1,length(s));
416             for i=1:length(s)
417                 LCC_aux(i) = Av_LCC(s(i));
418             end
419             s = find(Av_LCC==min(LCC_aux));
420             % If 2 satellites have the same LCC the first one is chosen
421             if length(s) > 1
422                 s = s(1);
423             end
424         end
425     end
426 end
427 end
428
429 %% Sat_Monoplex
430
431 % Weight matrix for the whole period
432 W = zeros(n_Sat,n_Sat);
433 for i=1:n_d_grid_Network
434     W = W + Sat_Network(i).Iteration(1).Adjacency_Matrix;
435 end
436 W = W/n_d_grid_Network;
437 Sat_Monoplex(1).Weight_Matrix = W;
438
439 % Sat_Monoplex structure
440 for it=1:it_max
441     % Changes for each iteration
442     if it>1
443         % Changes in NORAD names list
444         Sat_Monoplex(it).Missed_Satellite = NORAD_Monoplex(s,:); % Missed satellite
445         NORAD_Monoplex(s,:) = [];
446
447         % Changes in adjacency matrixes for each date
448         Sat_Monoplex(it).Weight_Matrix = Sat_Monoplex(it-1).Weight_Matrix;
449         Sat_Monoplex(it).Weight_Matrix(s,:) = [];
450         Sat_Monoplex(it).Weight_Matrix(:,s) = [];
451     end
452
453     % Weight matrix
454     W = Sat_Monoplex(it).Weight_Matrix;
455
456     % Save weight matrix as a graph
457     G = graph(W,NORAD_Monoplex');
458
459     % Node degree
460     nd = degree(G);
461     nd_Wei = sum(W,2);
462
463     % Number of nodes and edges
464     nn = size(G.Nodes,1);

```

```

465 ne = size(G.Edges,1);
466
467 % Shortest path length between nodes (distances) matrix
468 W_Inv = zeros(nn);
469 for i=1:nn
470     for j=1:nn
471         if W(i,j) == 0
472             W_Inv(i,j) = 0;
473         else
474             W_Inv(i,j) = 1/W(i,j); % If the graph is Wei those weights are used as the
                                     distances along the edges in the graph; because of that the inverse is used
475         end
476     end
477 end
478 G_Inv = graph(W_Inv,NORAD_Monoplex');
479 D = distances(G_Inv);
480
481 % Path length of each node
482 pl_Wei = sum(D)'/ (nn-1);
483
484 % Characteristic path length of the network
485 cpl_Wei = sum(pl_Wei)/nn;
486
487 % Efficiency of each node
488 D_Inv = zeros(nn);
489 for i=1:nn
490     for j=1:nn
491         if D(i,j) == 0
492             D_Inv(i,j) = 0;
493         else
494             D_Inv(i,j) = 1/D(i,j);
495         end
496     end
497 end
498 e_Wei = sum(D_Inv)'/ (nn-1);
499
500 % Efficiency of the network
501 ef_Wei = sum(e_Wei)/nn;
502
503 % Weighted local clustering coefficient
504 % Reference: Complex network measures of brain connectivity: Uses and interpretations.
                    Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
505 % Files: https://sites.google.com/site/bctnet/
506 lcc_Wei = clustering_coef_wu(W);
507
508 % Weighted global clustering coefficient
509 gcl_Wei = sum(lcc_Wei)/nn;
510
511 % Weighted normalized betweenness centrality
512 b_Wei = 2*centrality(G_Inv,'betweenness','Cost',G_Inv.Edges.Weight); % 2*B because network
                    is undirected
513 if nn>2
514     Norm_b_Wei = b_Wei/((nn-1)*(nn-2));
515 elseif nn==2
516     Norm_b_Wei = [0;0]; % If there are only 2 satellites left it assigns centrality to
                    both
517 else
518     Norm_b_Wei = 0;
519 end
520
521 % Placement of elements in the structure
522 for j=1:nn
523     Sat_Monoplex(it).Satellites(j).NORAD = NORAD_Monoplex(j,1);
524     Sat_Monoplex(it).Satellites(j).Node_Degree = nd(j,1);
525     Sat_Monoplex(it).Satellites(j).Node_Degree_Wei = nd_Wei(j,1);
526     Sat_Monoplex(it).Satellites(j).Path_Length_Wei = pl_Wei(j,1);
527     Sat_Monoplex(it).Satellites(j).Efficiency_Wei = e_Wei(j,1);
528     Sat_Monoplex(it).Satellites(j).Local_Clustering_Coefficient_Wei = lcc_Wei(j,1);
529     Sat_Monoplex(it).Satellites(j).Betweenness_Centrality_Wei = b_Wei(j,1);
530     Sat_Monoplex(it).Satellites(j).Norm_Betweenness_Centrality_Wei = Norm_b_Wei(j,1);
531 end
532 Sat_Monoplex(it).Network = G;
533 Sat_Monoplex(it).Network_Inv = G_Inv;
534 Sat_Monoplex(it).Distances = D;
535 Sat_Monoplex(it).Distances_Inv = D_Inv;

```

```

536 Sat_Monoplex(it).n_Edges = ne;
537 Sat_Monoplex(it).Characteristic_Path_Length_Wei = cpl_Wei;
538 Sat_Monoplex(it).Efficiency_Wei = ef_Wei;
539 Sat_Monoplex(it).Global_Clustering_Coefficient_Wei = gcl_Wei;
540
541 % Select method to identify most vulnerable satellite [6]
542 switch(method)
543     case 1 % Satellite with higher weighted centrality
544         s = find(Norm_b_Wei==max(Norm_b_Wei)); % Allows to find more than one index
545         % If 2 satellites have the same weighted centrality the one with higher weighted
           degree is chosen
546         if length(s) > 1
547             Degree_aux = zeros(1,length(s));
548             for i=1:length(s)
549                 Degree_aux(i) = nd_Wei(s(i));
550             end
551             s = find(nd_Wei==max(Degree_aux));
552             % If 2 satellites have the same weighted degree the one with lower weighted
           LCC is chosen
553             if length(s) > 1
554                 LCC_aux = zeros(1,length(s));
555                 for i=1:length(s)
556                     LCC_aux(i) = lcc_Wei(s(i));
557                 end
558                 s = find(lcc_Wei==min(LCC_aux));
559                 % If 2 satellites have the same weighted LCC the first one is chosen
560                 if length(s) > 1
561                     s = s(1);
562                 end
563             end
564         end
565     case 2 % Satellite with higher weighted degree
566         s = find(nd_Wei==max(nd_Wei));
567         % If 2 satellites have the same weighted degree the one with lower weighted LCC is
           chosen
568         if length(s) > 1
569             LCC_aux = zeros(1,length(s));
570             for i=1:length(s)
571                 LCC_aux(i) = lcc_Wei(s(i));
572             end
573             s = find(lcc_Wei==min(LCC_aux));
574             % If 2 satellites have the same weighted LCC the first one is chosen
575             if length(s) > 1
576                 s = s(1);
577             end
578         end
579     case 3 % Satellite with lower weighted local clustering coefficient
580         s = find(lcc_Wei==min(lcc_Wei));
581         % If 2 satellites have the same weighted LCC the one with higher weighted degree
           is chosen
582         if length(s) > 1
583             nd_aux = zeros(1,length(s));
584             for i=1:length(s)
585                 nd_aux(i) = nd_Wei(s(i));
586             end
587             s = find(nd_Wei==max(nd_aux));
588             % If 2 satellites have the same weighted LCC the first one is chosen
589             if length(s) > 1
590                 s = s(1);
591             end
592         end
593     case 4 % Satellite with higher weighted efficiency
594         s = find(e_Wei==max(e_Wei)); % Allows to find more than one index
595         % If 2 satellites have the same weighted efficiency the one with higher weighted
           degree is chosen
596         if length(s) > 1
597             Degree_aux = zeros(1,length(s));
598             for i=1:length(s)
599                 Degree_aux(i) = nd_Wei(s(i));
600             end
601             s = find(nd_Wei==max(Degree_aux));
602             % If 2 satellites have the same weighted degree the one with lower weighted
           LCC is chosen
603             if length(s) > 1
604                 LCC_aux = zeros(1,length(s));

```

```

605         for i=1:length(s)
606             LCC_aux(i) = lcc_Wei(s(i));
607         end
608         s = find(lcc_Wei==min(LCC_aux));
609         % If 2 satellites have the same weighted LCC the first one is chosen
610         if length(s) > 1
611             s = s(1);
612         end
613     end
614 end
615 end
616 end
617
618 %% Missed satellites lists
619
620 % NORAD ids
621 Missed_Satellites_Monoplex = zeros(it_max-1,1);
622 Missed_Satellites_Multiplex = zeros(it_max-1,1);
623 for i=1:it_max-1
624     Missed_Satellites_Monoplex(i) = str2double(Sat_Monoplex(i+1).Missed_Satellite);
625     Missed_Satellites_Multiplex(i) = str2double(Sat_Multiplex(i+1).Missed_Satellite);
626 end
627
628 % Level of coincidence
629 Coincidence = zeros(it_max-1,1);
630 for i=1:it_max-1
631     for j=1:i
632         for k=1:i
633             Coincidence(i) = Coincidence(i) + isequal(Missed_Satellites_Monoplex(j),
634                 Missed_Satellites_Multiplex(k));
635         end
636     end
637 end
638 %% Recalculate structure elements
639
640 % Sat_Rec_Network and Sat_Rec_Multiplex initialization
641 NORAD_Network_Rec = NORAD;
642 n_Sat_Network = n_Sat;
643 Sat_Rec_Network = Sat_Network;
644 Sat_Rec_Multiplex(1) = Sat_Multiplex(1);
645
646 for it=2:it_max
647     %% Sat_Rec_Network structure
648
649     % Changes in NORAD names list
650     s = find(str2double(NORAD_Network_Rec)==Missed_Satellites_Monoplex(it-1));
651     Sat_Rec_Multiplex(it).Missed_Satellite = NORAD_Network_Rec(s,:);
652     NORAD_Network_Rec(s,:) = [];
653
654     % Changes in indexes
655     n_Sat_Network = n_Sat_Network - 1;
656
657     % Changes in adjacency matrix
658     for i=1:n_d_grid_Network
659         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix = Sat_Rec_Network(i).Iteration(it-1)
660             .Adjacency_Matrix;
661         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
662         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
663     end
664
665     % Graph properties of the network for for each date
666     for i=1:n_d_grid_Network
667         % Adjacency matrix
668         A = Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix;
669
670         % Save satellite network as a graph
671         G = graph(A,NORAD_Network_Rec');
672
673         % Node degree
674         nd = degree(G);
675
676         % Number of nodes and edges
677         nn = size(G.Nodes,1);
678         ne = size(G.Edges,1);

```

```

678
679 % Shortest path length between nodes matrix
680 D = distances(G);
681
682 % Path length of each node
683 pl = sum(D)'/(nn-1);
684
685 % Characteristic path length of the network
686 cpl = sum(pl)/nn;
687
688 % Efficiency of each node
689 D_Inv = zeros(nn);
690 for j=1:nn
691     for k=1:nn
692         if D(j,k) == 0
693             D_Inv(j,k) = 0;
694         else
695             D_Inv(j,k) = 1/D(j,k);
696         end
697     end
698 end
699 e = sum(D_Inv)'/(nn-1);
700
701 % Efficiency of the network
702 ef = sum(e)/nn;
703
704 % Local clustering coefficient
705 % Reference: Complex network measures of brain connectivity: Uses and interpretations.
706 % Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
707 % Files: https://sites.google.com/site/bctnet/
708 lcc = clustering_coef_bu(A);
709
710 % Global clustering coefficient
711 gcl = sum(lcc)/nn;
712
713 % Normalized betweenness centrality
714 b = 2*centrality(G,'betweenness'); % 2*b because network is undirected
715 if nn>2
716     Norm_b = b/((nn-1)*(nn-2));
717 elseif nn==2
718     Norm_b = [0;0]; % If there are only 2 satellites left it assigns centrality to
719     both
720 else
721     Norm_b = 0;
722 end
723
724 % Placement of elements in the structure
725 for j=1:nn
726     Sat_Rec_Network(i).Iteration(it).Satellites(j).NORAD = NORAD_Network_Rec(j,1);
727     Sat_Rec_Network(i).Iteration(it).Satellites(j).Node_Degree = nd(j,1);
728     Sat_Rec_Network(i).Iteration(it).Satellites(j).Path_Length = pl(j,1);
729     Sat_Rec_Network(i).Iteration(it).Satellites(j).Efficiency = e(j,1);
730     Sat_Rec_Network(i).Iteration(it).Satellites(j).Local_Clustering_Coefficient = lcc(
731     j,1);
732     Sat_Rec_Network(i).Iteration(it).Satellites(j).Betweenness_Centrality = b(j,1);
733     Sat_Rec_Network(i).Iteration(it).Satellites(j).Norm_Betweenness_Centrality =
734     Norm_b(j,1);
735 end
736 Sat_Rec_Network(i).Iteration(it).Network = G;
737 Sat_Rec_Network(i).Iteration(it).Distances = D;
738 Sat_Rec_Network(i).Iteration(it).n_Edges = ne;
739 Sat_Rec_Network(i).Iteration(it).Characteristic_Path_Length = cpl;
740 Sat_Rec_Network(i).Iteration(it).Efficiency = ef;
741 Sat_Rec_Network(i).Iteration(it).Global_Clustering_Coefficient = gcl;
742 end
743
744 %% Sat_Multiplex structure
745
746 % Degree, path length, efficiency, local clustering coefficient and betweenness centrality
747 % matrixes for each satellite and each date
748 Degree = zeros(n_Sat_Network,n_d_grid_Network);
749 PL = zeros(n_Sat_Network,n_d_grid_Network);
750 E = zeros(n_Sat_Network,n_d_grid_Network);
751 LCC = zeros(n_Sat_Network,n_d_grid_Network);
752 Norm_Betweenness_Centrality = zeros(n_Sat_Network,n_d_grid_Network);

```

```

748     for i=1:n_d_grid_Network
749         for j=1:n_Sat_Network
750             Degree(j,i) = sum(Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(j,:));
751             PL(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).Path_Length;
752             E(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).Efficiency;
753             LCC(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).
                Local_Clustering_Coefficient;
754             Norm_Betweenness_Centrality(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).
                Norm_Betweenness_Centrality;
755         end
756     end
757
758     % Mean and standar deviation degree, path length, efficiency, local clustering coefficient
        and betweenness centrality for each satellite
759     Av_Degree = zeros(n_Sat_Network,1);
760     Av_PL = zeros(n_Sat_Network,1);
761     Av_E = zeros(n_Sat_Network,1);
762     Av_LCC = zeros(n_Sat_Network,1);
763     Av_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
764     Std_Degree = zeros(n_Sat_Network,1);
765     Std_PL = zeros(n_Sat_Network,1);
766     Std_E = zeros(n_Sat_Network,1);
767     Std_LCC = zeros(n_Sat_Network,1);
768     Std_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
769     for i=1:n_Sat_Network
770         Av_Degree(i,1) = mean(Degree(i,:));
771         Av_PL(i,1) = mean(PL(i,:));
772         Av_E(i,1) = mean(E(i,:));
773         Av_LCC(i,1) = mean(LCC(i,:));
774         Av_Norm_Betweenness_Centrality(i,1) = mean(Norm_Betweenness_Centrality(i,:));
775         Std_Degree(i,1) = std(Degree(i,:));
776         Std_PL(i,1) = std(PL(i,:));
777         Std_E(i,1) = std(E(i,:));
778         Std_LCC(i,1) = std(LCC(i,:));
779         Std_Norm_Betweenness_Centrality(i,1) = std(Norm_Betweenness_Centrality(i,:));
780     end
781
782     % Generate NORAD names column for the whole period
783     for i=1:n_Sat_Network
784         Sat_Rec_Multiplex(it).Satellites(i).NORAD = NORAD_Network_Rec(i,1); % NORAD names
                column (double)
785     end
786
787     % Place mean and standar deviation degree, path length, efficiency, local clustering
        coefficient and betweenness centrality for each satellite
788     for i=1:n_Sat_Network
789         Sat_Rec_Multiplex(it).Satellites(i).Av_Degree = Av_Degree(i,1);
790         Sat_Rec_Multiplex(it).Satellites(i).Std_Degree = Std_Degree(i,1);
791         Sat_Rec_Multiplex(it).Satellites(i).Av_Path_Length = Av_PL(i,1);
792         Sat_Rec_Multiplex(it).Satellites(i).Std_Path_Length = Std_PL(i,1);
793         Sat_Rec_Multiplex(it).Satellites(i).Av_Efficiency = Av_E(i,1);
794         Sat_Rec_Multiplex(it).Satellites(i).Std_Efficiency = Std_E(i,1);
795         Sat_Rec_Multiplex(it).Satellites(i).Av_LCC = Av_LCC(i,1);
796         Sat_Rec_Multiplex(it).Satellites(i).Std_LCC = Std_LCC(i,1);
797         Sat_Rec_Multiplex(it).Satellites(i).Av_Norm_Betweenness_Centrality =
                Av_Norm_Betweenness_Centrality(i,1);
798         Sat_Rec_Multiplex(it).Satellites(i).Std_Norm_Betweenness_Centrality =
                Std_Norm_Betweenness_Centrality(i,1);
799     end
800
801     % Number of edges, characteristic path length, efficiency and global clustering
        coefficient for each date
802     n_Edges = zeros(n_d_grid_Network,1);
803     CPL = zeros(n_d_grid_Network,1);
804     Efficiency = zeros(n_d_grid_Network,1);
805     GCC = zeros(n_d_grid_Network,1);
806     for i=1:n_d_grid_Network
807         n_Edges(i,1) = Sat_Rec_Network(i).Iteration(it).n_Edges;
808         CPL(i,1) = Sat_Rec_Network(i).Iteration(it).Characteristic_Path_Length;
809         Efficiency(i,1) = Sat_Rec_Network(i).Iteration(it).Efficiency;
810         GCC(i,1) = Sat_Rec_Network(i).Iteration(it).Global_Clustering_Coefficient;
811     end
812
813     % Mean and standar deviation number of edges, characteristic path length, efficiency and
        global clustering coefficient for each date

```

```

814 Sat_Rec_Multiplex(it).Av_n_Edges = mean(n_Edges);
815 Sat_Rec_Multiplex(it).Std_n_Edges = std(n_Edges);
816 Sat_Rec_Multiplex(it).Av_Characteristic_Path_Length = mean(CPL);
817 Sat_Rec_Multiplex(it).Std_Characteristic_Path_Length = std(CPL);
818 Sat_Rec_Multiplex(it).Av_Efficiency = mean(Efficiency);
819 Sat_Rec_Multiplex(it).Std_Efficiency = std(Efficiency);
820 Sat_Rec_Multiplex(it).Av_Global_Clustering_Coefficient = mean(GCC);
821 Sat_Rec_Multiplex(it).Std_Global_Clustering_Coefficient = std(GCC);
822 end
823
824 % Sat_Rec_Monoplex initialization
825 NORAD_Monoplex_Rec = NORAD;
826 Sat_Rec_Monoplex(1) = Sat_Monoplex(1);
827
828 % Sat_Rec_Monoplex structure
829 for it=2:it_max
830     % Changes in NORAD names list
831     s = find(str2double(NORAD_Monoplex_Rec)==Missed_Satellites_Multiplex(it-1));
832     Sat_Rec_Monoplex(it).Missed_Satellite = NORAD_Monoplex_Rec(s,:);
833     NORAD_Monoplex_Rec(s,:) = [];
834
835     % Changes in weight matrixes
836     Sat_Rec_Monoplex(it).Weight_Matrix = Sat_Rec_Monoplex(it-1).Weight_Matrix;
837     Sat_Rec_Monoplex(it).Weight_Matrix(s,:) = [];
838     Sat_Rec_Monoplex(it).Weight_Matrix(:,s) = [];
839
840     % Weight matrix
841     W = Sat_Rec_Monoplex(it).Weight_Matrix;
842
843     % Save weight matrix as a graph
844     G = graph(W,NORAD_Monoplex_Rec');
845
846     % Node degree
847     nd = degree(G);
848     nd_Wei = sum(W,2);
849
850     % Number of nodes and edges
851     nn = size(G.Nodes,1);
852     ne = size(G.Edges,1);
853
854     % Shortest path length between nodes (distances) matrix
855     W_Inv = zeros(nn);
856     for i=1:nn
857         for j=1:nn
858             if W(i,j) == 0
859                 W_Inv(i,j) = 0;
860             else
861                 W_Inv(i,j) = 1/W(i,j); % If the graph is Wei those weights are used as the
                        distances along the edges in the graph; because of that the inverse is used
862             end
863         end
864     end
865     G_Inv = graph(W_Inv,NORAD_Monoplex_Rec');
866     D = distances(G_Inv);
867
868     % Path length of each node
869     pl_Wei = sum(D)'/(nn-1);
870
871     % Characteristic path length of the network
872     cpl_Wei = sum(pl_Wei)/nn;
873
874     % Efficiency of each node
875     D_Inv = zeros(nn);
876     for i=1:nn
877         for j=1:nn
878             if D(i,j) == 0
879                 D_Inv(i,j) = 0;
880             else
881                 D_Inv(i,j) = 1/D(i,j);
882             end
883         end
884     end
885     e_Wei = sum(D_Inv)'/(nn-1);
886
887     % Efficiency of the network

```

```

888     ef_Wei = sum(e_Wei)/nn;
889
890     % Weighted local clustering coefficient
891     % Reference: Complex network measures of brain connectivity: Uses and interpretations.
892     % Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
893     % Files: https://sites.google.com/site/bctnet/
894     lcc_Wei = clustering_coef_wu(W);
895
896     % Weighted global clustering coefficient
897     gcl_Wei = sum(lcc_Wei)/nn;
898
899     % Weighted normalized betweenness centrality
900     b_Wei = 2*centrality(G_Inv,'betweenness','Cost',G_Inv.Edges.Weight); % 2*B because network
901     % is undirected
902     if nn>2
903         Norm_b_Wei = b_Wei/((nn-1)*(nn-2));
904     elseif nn==2
905         Norm_b_Wei = [0;0]; % If there are only 2 satellites left it assigns centrality to
906         % both
907     else
908         Norm_b_Wei = 0;
909     end
910
911     % Placement of elements in the structure
912     for j=1:nn
913         Sat_Rec_Monoplex(it).Satellites(j).NORAD = NORAD_Monoplex_Rec(j,1);
914         Sat_Rec_Monoplex(it).Satellites(j).Node_Degree = nd(j,1);
915         Sat_Rec_Monoplex(it).Satellites(j).Node_Degree_Wei = nd_Wei(j,1);
916         Sat_Rec_Monoplex(it).Satellites(j).Path_Length_Wei = pl_Wei(j,1);
917         Sat_Rec_Monoplex(it).Satellites(j).Efficiency_Wei = e_Wei(j,1);
918         Sat_Rec_Monoplex(it).Satellites(j).Local_Clustering_Coefficient_Wei = lcc_Wei(j,1);
919         Sat_Rec_Monoplex(it).Satellites(j).Betweenness_Centrality_Wei = b_Wei(j,1);
920         Sat_Rec_Monoplex(it).Satellites(j).Norm_Betweenness_Centrality_Wei = Norm_b_Wei(j,1);
921     end
922     Sat_Rec_Monoplex(it).Network = G;
923     Sat_Rec_Monoplex(it).Network_Inv = G_Inv;
924     Sat_Rec_Monoplex(it).Distances = D;
925     Sat_Rec_Monoplex(it).Distances_Inv = D_Inv;
926     Sat_Rec_Monoplex(it).n_Edges = ne;
927     Sat_Rec_Monoplex(it).Characteristic_Path_Length_Wei = cpl_Wei;
928     Sat_Rec_Monoplex(it).Efficiency_Wei = ef_Wei;
929     Sat_Rec_Monoplex(it).Global_Clustering_Coefficient_Wei = gcl_Wei;
930 end
931
932 %% Method comparission
933
934 % Efficiencies of original methods and recalculations
935 Efficiency_Multiplex = zeros(1,it_max);
936 Efficiency_Multiplex_Rec = zeros(1,it_max);
937 Efficiency_Multiplex_Dif = zeros(1,it_max);
938 Efficiency_Monoplex = zeros(1,it_max);
939 Efficiency_Monoplex_Rec = zeros(1,it_max);
940 Efficiency_Monoplex_Dif = zeros(1,it_max);
941 for it=1:it_max
942     Efficiency_Multiplex(it) = Sat_Multiplex(it).Av_Efficiency;
943     Efficiency_Multiplex_Rec(it) = Sat_Rec_Multiplex(it).Av_Efficiency;
944     Efficiency_Multiplex_Dif(it) = Efficiency_Multiplex_Rec(it) - Efficiency_Multiplex(it);
945     Efficiency_Monoplex(it) = Sat_Monoplex(it).Efficiency_Wei;
946     Efficiency_Monoplex_Rec(it) = Sat_Rec_Monoplex(it).Efficiency_Wei;
947     Efficiency_Monoplex_Dif(it) = Efficiency_Monoplex_Rec(it) - Efficiency_Monoplex(it);
948 end
949
950 % x axis
951 it_array = 1:it_max;
952
953 % Multiplex efficiency comparission plot
954 figure
955 plot(it_array,Efficiency_Multiplex,'-o',it_array,Efficiency_Multiplex_Rec,'-o')
956 title('Efficiency comparison between the multiplex networks of the Globalstar constellation
957 during day 01/09/2021 (00:00-01:54 ~ 1 orbit)')
958 xlabel('Iteration')
959 ylabel('Network efficiency')
960 legend('Multiplex','Recalculated multiplex','FontSize',12)
961
962 % Multiplex efficiency diferencial plot

```



```

959 figure
960 bar(it_array,Efficiency_Multiplex_Dif)
961 title('Difference in efficiencies between the multiplex networks of the Globalstar
          constellation during day 01/09/2021 (00:00-01:54 ~ 1 orbit)')
962 xlabel('Iteration')
963 ylabel('Efficiency difference between methods')
964
965 % Monoplex efficiency comparission plot
966 figure
967 plot(it_array,Efficiency_Monoplex,'-o',it_array,Efficiency_Monoplex_Rec,'-o')
968 title('Efficiency comparison between the monoplex networks of the Globalstar constellation
          during day 01/09/2021 (00:00-01:54 ~ 1 orbit)')
969 xlabel('Iteration')
970 ylabel('Network efficiency')
971 legend('Monoplex','Recalculated monoplex','FontSize',12)
972
973 % Monoplex efficiency differencial plot
974 figure
975 bar(it_array,Efficiency_Monoplex_Dif)
976 title('Difference in efficiencies between the monoplex networks of the Globalstar
          constellation during day 01/09/2021 (00:00-01:54 ~ 1 orbit)')
977 xlabel('Iteration')
978 ylabel('Efficiency difference between methods')
979
980 %% Multiplex graph plots
981
982 % Functioning parameters
983 it = 1; % Iteration index (number of eliminated satellites+1)
984 i = 1; % Date index
985 g = Sat_Network(i); % Structure name (it is needed to indicate date index "i")
986 c = zeros(1,n_Sat+1-it);
987 for j=1:(n_Sat+1-it)
988     c(j) = g.Iteration(it).Satellites(j).Norm_Betweenness_Centrality; % Colorbar metric
989 end
990
991 % Force graph
992 figure
993 p = plot(g.Iteration(it).Network,'MarkerSize',8);
994 p.NodeCData = c;
995 colormap jet
996 cb = colorbar;
997 cb.Label.String = 'Betweenness centrality';
998 cb.FontSize = 12;
999 title('Force graph of the Globalstar constellation during day 01/09/2021 (00:00)')
1000
1001 % Circular graph
1002 figure
1003 pc = plot(g.Iteration(it).Network,'Layout','circle','MarkerSize',8);
1004 pc.NodeCData = c;
1005 colormap jet
1006 cb = colorbar;
1007 cb.Label.String = 'Betweenness centrality';
1008 cb.FontSize = 12;
1009 title('Circular graph of the Globalstar constellation during day 01/09/2021 (00:00)')
1010
1011 % Layered graph
1012 figure
1013 pl = plot(g.Iteration(it).Network,'Layout','layered','MarkerSize',8);
1014 pl.NodeCData = c;
1015 colormap jet
1016 cb = colorbar;
1017 cb.Label.String = 'Betweenness centrality';
1018 cb.FontSize = 12;
1019 title('Layered graph of the Globalstar constellation during day 01/09/2021 (00:00)')
1020
1021 % 3D graph
1022 x = zeros(1,n_Sat+1-it);
1023 y = zeros(1,n_Sat+1-it);
1024 z = zeros(1,n_Sat+1-it);
1025 for j=1:(n_Sat+1-it)
1026     x(j) = g.Nodes(j).Position_Mean(1);
1027     y(j) = g.Nodes(j).Position_Mean(2);
1028     z(j) = g.Nodes(j).Position_Mean(3);
1029 end
1030 figure
    
```

```

1031 p3d = plot(g.Iteration(it).Network,'XData',x,'YData',y,'ZData',z,'MarkerSize',8);
1032 p3d.NodeCData = c;
1033 colormap jet
1034 cb = colorbar;
1035 cb.Label.String = 'Betweenness centrality';
1036 cb.FontSize = 12;
1037 title('Positions and edges of the Globalstar constellation during day 01/09/2021 (00:00)')
1038 xlabel('x [km]')
1039 ylabel('y [km]')
1040 zlabel('z [km]')
1041
1042 %% Monoplex graph plots
1043
1044 % Functioning parameters
1045 it = 1; % Iteration index (number of eliminated satellites+1)
1046 g = Sat_Monoplex(it); % Structure name (it is needed to indicate iteration index "it")
1047 c = zeros(1,n_Sat+1-it);
1048 for j=1:(n_Sat+1-it)
1049     c(j) = g.Satellites(j).Norm_Betweenness_Centrality_Wei; % Colorbar metric
1050 end
1051 LWidths = (5*g.Network.Edges.Weight)';
1052
1053 % Force graph
1054 figure
1055 p_wei = plot(g.Network,'EdgeLabel',g.Network.Edges.Weight,'LineWidth',LWidths,'MarkerSize',10)
1056 ;
1057 p_wei.NodeCData = c;
1058 colormap jet
1059 cb = colorbar;
1060 cb.Label.String = 'Betweenness centrality';
1061 cb.FontSize = 12;
1062 title('Monoplex force graph of the Globalstar constellation during day 01/09/2021')
1063
1064 % Circular graph
1065 figure
1066 p_wei = plot(g.Network,'Layout','circle','EdgeLabel',g.Network.Edges.Weight,'LineWidth',
1067     LWidths,'MarkerSize',10);
1068 p_wei.NodeCData = c;
1069 colormap jet
1070 cb = colorbar;
1071 cb.Label.String = 'Betweenness centrality';
1072 cb.FontSize = 12;
1073 title('Monoplex circular graph of the Globalstar constellation during day 01/09/2021')
1074
1075 % Layered graph
1076 figure
1077 p_wei = plot(g.Network,'Layout','layered','EdgeLabel',g.Network.Edges.Weight,'LineWidth',
1078     LWidths,'MarkerSize',10);
1079 p_wei.NodeCData = c;
1080 colormap jet
1081 cb = colorbar;
1082 cb.Label.String = 'Betweenness centrality';
1083 cb.FontSize = 12;
1084 title('Monoplex layered graph of the Globalstar constellation during day 01/09/2021')
1085
1086 %% Stops clock to measure time of execution
1087 toc

```

Capítulo 3

Comparación de métodos de análisis de vulnerabilidades de la red de satélites

3.1. Programa para el almacenamiento de métricas de la red de satélites para distintos intervalos periódicos (Analisis.m)

```
1 %% Analisis.m
2 % Antonio Serda Mena. (2022). Application of Complex Network Theory to the Vulnerability
   Analysis of a Satellite Network
3 % Analyzes the satellite network using Complex Network Theory and performing a certain number
   of experiments
4
5 %% Considerations
6 % Parallel Computing Toolbox needed for faster execution: parfor loops used
7 % Functions from the Brain Connectivity Toolbox needed:
8 % clustering_coef_bu.m
9 % clustering_coef_wu.m
10 % Reference: Complex network measures of brain connectivity: Uses and interpretations. Rubinov
    M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
11 % Files: https://sites.google.com/site/bctnet/
12 % It is needed to have satellite position first, using the program Nube.mat
13
14 %% Initialization
15 close all force
16 clear all
17 clc
18 tic; % Starts clock to measure time of execution
19
20 %% Functioning parameters
21
22 % Loading of satellite positions and related variables
23 load('Cloud_Sept');
24
25 % List of wanted satellites [1]
26 list_title = 'Globalstar.txt';
27
28 % Loop adjustments
29 % Initial date [2]
30 s0 = 0;
31 m0 = 0;
32 h0 = 0;
33 d0 = 1;
34 mo0 = 9;
35 y0 = 2021;
36 % Date increment (orbital period) [3]
37 sd = 0;
38 md = 54;
39 hd = 1;
40 dd = 0;
41 mod = 0;
```

III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```
42 yd = 0;
43 % Date index (depends on how many measures are wanted) [4]
44 si = 0;
45 mi = 0;
46 hi = 8;
47 di = 0;
48 moi = 0;
49 yi = 0;
50
51 % Time period between positions multiplum [5]
52 mult_period = 1; % Time between measures must be a multiplum of the time grid interval of the
    cloud (must be an integer > 1)
53
54 % Maximum number of iterations [6]
55 p = 1; % Portion of the total number of satellites to eliminate (from 0 to 1)
56
57 % Method for determining vulnerability [7]
58 method = 1;
59 % 1 -> Max betweenes centrality
60 % 2 -> Max node degree
61 % 3 -> Min local clustering coefficient
62 % 4 -> Max node efficiency
63
64 %Astronomical constants [8]
65 R_Earth = 6378.137; % Earth's radius [km] (WGS-84)
66 dR = 60; % Extra radius [km]
67
68 % Maximum number of intervals to perform the analysis [9]
69 i_point_max = 90;
70
71 %% Initial adjustments
72
73 % List of wanted satellites [1]
74 list = fopen(list_title); % Open text file
75 NORAD_Network_Struct = textscan(list, '%s'); % Text scan data
76 fclose(list); % Close file
77 n_Sat = size(NORAD_Network_Struct{1,1},1); % Number of satellites wanted
78 NORAD = sort(NORAD_Network_Struct{1,1}); % Sort NORAD numbers of wanted satellites
79 NORAD_Double = str2double(NORAD);
80
81 % Set maximum number of iterations [6]
82 it_max = round(n_Sat*p); % Translates the percentage of eliminated satellites into an integer
83
84 %% Analysis variables initialization
85
86 An_Coincidence = zeros(it_max-1,i_point_max);
87 An_Efficiency_Multiplex = zeros(it_max,i_point_max);
88 An_Efficiency_Multiplex_Rec = zeros(it_max,i_point_max);
89 An_Efficiency_Multiplex_Dif = zeros(it_max,i_point_max);
90 An_Efficiency_Monoplex = zeros(it_max,i_point_max);
91 An_Efficiency_Monoplex_Rec = zeros(it_max,i_point_max);
92 An_Efficiency_Monoplex_Dif = zeros(it_max,i_point_max);
93
94 %% Analysis loop
95
96 for i_point=1:i_point_max % [9]
97     %% Initial adjustments
98
99     % List of wanted satellites [1]
100     NORAD_Network = NORAD;
101     NORAD_Monoplex = NORAD; % NORAD numbers of wanted satellites for a graph representing the
        whole period
102     n_Sat_Network = n_Sat;
103
104     % Date interval selection
105     % Initial [2]
106     Date_Network.s_0 = s0; % Second 0-59
107     Date_Network.m_0 = m0; % Minute 0-59
108     Date_Network.h_0 = h0; % Hour 0-23
109     Date_Network.d_0 = d0; % Day 1-28/29/30/31 depending on the month
110     Date_Network.mo_0 = mo0; % Month 1-12
111     Date_Network.y_0 = y0; % Year 1957-2056
112     Date_Network.Epoch_Day_0 = round(datumum(Date_Network.y_0,Date_Network.mo_0,Date_Network.
        d_0,Date_Network.h_0,Date_Network.m_0,Date_Network.s_0) - datumum(Date_Network.y_0
        ,0,0,0,0,0),8); % Day of the year epoch
```

III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```
113 Date_Network.Full_Epoch_0 = Date_Network.y_0*1000 + Date_Network.Epoch_Day_0;
114 Date_Network.Date_0 = datetime(Date_Network.y_0,Date_Network.mo_0,Date_Network.d_0,
    Date_Network.h_0,Date_Network.m_0,Date_Network.s_0);
115 % Final [2] [3] (114 minutes (Globalstar orbital period) after initial date)
116 Date_Network.s_f = s0 + sd; % Second 0-59
117 Date_Network.m_f = m0 + md; % Minute 0-59
118 Date_Network.h_f = h0 + hd; % Hour 0-23
119 Date_Network.d_f = d0 + dd; % Day 1-28/29/30/31 depending on the month
120 Date_Network.mo_f = mo0 + mod; % Month 1-12
121 Date_Network.y_f = y0 + yd; % Year 1957-2056
122 Date_Network.Epoch_Day_f = round(datenum(Date_Network.y_f,Date_Network.mo_f,Date_Network.
    d_f,Date_Network.h_f,Date_Network.m_f,Date_Network.s_f) - datenum(Date_Network.y_f
    ,0,0,0,0,0),8); % Day of the year epoch
123 Date_Network.Full_Epoch_f = Date_Network.y_f*1000 + Date_Network.Epoch_Day_f;
124 Date_Network.Date_f = datetime(Date_Network.y_f,Date_Network.mo_f,Date_Network.d_f,
    Date_Network.h_f,Date_Network.m_f,Date_Network.s_f);
125 % Time interval
126 Date_Network.t_max = split(between(Date_Network.Date_0,Date_Network.Date_f,'time'),'time')
    ;
127
128 % Date comprobation
129 if Date_Network.t_max <= 0 || Date_Network.Date_0 < Date_Cloud.Date_0 || Date_Network.
    Date_f > Date_Cloud.Date_f
130     disp('Incorrect date interval')
131     pause
132 end
133
134 % Time period between positions [minutes] [5]
135 if mult_period < 1
136     disp('Incorrect multiplum')
137     pause
138 end
139 T_Network_minutes = T_Cloud_minutes*round(mult_period); % [minutes]
140
141 % Date grid array
142 t_dif_minutes = minutes(minutes(split(between(Date_Cloud.Date_0,Date_Network.Date_0,'Time'
    ),'Time')));
143 t_dif_adjusted_minutes = T_Network_minutes*ceil(datenum(t_dif_minutes)/datenum(
    T_Network_minutes));
144 d_Network_0 = Date_Cloud.Date_0 + t_dif_adjusted_minutes;
145 d_grid_Network = d_Network_0:T_Network_minutes:Date_Network.Date_f;
146 d_grid_Network = d_grid_Network';
147 n_d_grid_Network = size(d_grid_Network,1);
148
149 %% Sat_Network initialization
150
151 % Date collumn
152 for i=1:n_d_grid_Network
153     Sat_Network(i).Date = d_grid_Network(i);
154 end
155
156 % Convert date to text to simplify data view
157 for i=1:n_d_grid_Network
158     Sat_Network(i).Date_Char = datestr(Sat_Network(i).Date);
159 end
160
161 % Generate NORAD names column for each date
162 for i=1:n_d_grid_Network
163     for j=1:n_Sat_Network
164         Sat_Network(i).Nodes(j).NORAD = NORAD_Double(j,1); % NORAD names column (double)
165     end
166 end
167
168 % Position of each satellite for each date
169 for i=1:n_d_grid_Network
170     for j=1:n_Sat_Network
171         for k=1:n_Sat_Cloud
172             if Sat_Cloud(k).NORAD == Sat_Network(i).Nodes(j).NORAD
173                 for m=1:round(mult_period):n_d_grid_Cloud
174                     if Sat_Cloud(k).Position(m).Date == Sat_Network(i).Date
175                         Sat_Network(i).Nodes(j).Position_Mean = Sat_Cloud(k).Position(m).
                            Position_Mean;
176                     end
177                 end
178             end
179         end
180     end
181 end
```

III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```

179         end
180     end
181 end
182
183 % Adjacency matrix for each date for the initial state
184 parfor i=1:n_d_grid_Network
185     for j=1:n_Sat_Network
186         x1 = Sat_Network(i).Nodes(j).Position_Mean(1,1);
187         y1 = Sat_Network(i).Nodes(j).Position_Mean(2,1);
188         z1 = Sat_Network(i).Nodes(j).Position_Mean(3,1);
189         r1_km = sqrt(x1^2 + y1^2 + z1^2); % Satellite height
190         for k=1:n_Sat_Network
191             % Triangle between Earth's center and 2 satellites
192             x2 = Sat_Network(i).Nodes(k).Position_Mean(1,1);
193             y2 = Sat_Network(i).Nodes(k).Position_Mean(2,1);
194             z2 = Sat_Network(i).Nodes(k).Position_Mean(3,1);
195             r2_km = sqrt(x2^2 + y2^2 + z2^2); % Height of the rest of the satellites
196             d_km = sqrt((x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2); % Distance between satellites
197             sp_km = (r1_km + r2_km + d_km)/2; % Triangle semiperimeter
198             h_km = (2/d_km)*sqrt(sp_km*(sp_km-d_km)*(sp_km-r1_km)*(sp_km-r2_km)); %
                Triangle height formula
199
200             % If triangle's height is higher than Earth's radius (+60 km extra) satellites
                are visible [8]
201             if h_km >= R_Earth + dR
202                 Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 1;
203             else
204                 Sat_Network(i).Iteration(1).Adjacency_Matrix(j,k) = 0;
205             end
206         end
207     end
208 end
209
210 %% Sat_Network completion and Sat_Multiplex
211
212 for it=1:it_max
213     %% Sat_Network structure
214
215     % Changes for each iteration
216     if it>1
217         % Changes in NORAD names list
218         Sat_Multiplex(it).Missed_Satellite = NORAD_Network(s,:); % Missed satellite
219         NORAD_Network(s,:) = [];
220
221         % Changes in indexes
222         n_Sat_Network = n_Sat_Network-1;
223
224         % Changes in adjacency matrixes for each date
225         for i=1:n_d_grid_Network
226             Sat_Network(i).Iteration(it).Adjacency_Matrix = Sat_Network(i).Iteration(it-1)
                .Adjacency_Matrix;
227             Sat_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
228             Sat_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
229         end
230     end
231
232     % Graph properties of the network for for each date
233     parfor i=1:n_d_grid_Network
234         % Adjacency matrix
235         A = Sat_Network(i).Iteration(it).Adjacency_Matrix;
236
237         % Save satellite network as a graph
238         G = graph(A,NORAD_Network');
239
240         % Node degree
241         nd = degree(G);
242
243         % Number of nodes and edges
244         nn = size(G.Nodes,1);
245         ne = size(G.Edges,1);
246
247         % Shortest path length between nodes matrix
248         D = distances(G);
249
250         % Path length of each node

```

```

251     pl = sum(D)/(nn-1);
252
253     % Characteristic path length of the network
254     cpl = sum(pl)/nn;
255
256     % Efficiency of each node
257     D_Inv = zeros(nn);
258     for j=1:nn
259         for k=1:nn
260             if D(j,k) == 0
261                 D_Inv(j,k) = 0;
262             else
263                 D_Inv(j,k) = 1/D(j,k);
264             end
265         end
266     end
267     e = sum(D_Inv)/(nn-1);
268
269     % Efficiency of the network
270     ef = sum(e)/nn;
271
272     % Local clustering coefficient
273     % Reference: Complex network measures of brain connectivity: Uses and
                interpretations. Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.
                neuroimage.2009.10.003
274     % Files: https://sites.google.com/site/bctnet/
275     lcc = clustering_coef_bu(A);
276
277     % Global clustering coefficient
278     gcl = sum(lcc)/nn;
279
280     % Normalized betweenness centrality
281     b = 2*centrality(G,'betweenness'); % 2*b because network is undirected
282     if nn>2
283         Norm_b = b/((nn-1)*(nn-2));
284     elseif nn==2
285         Norm_b = [0;0]; % If there are only 2 satellites left it assigns centrality to
                both
286     else
287         Norm_b = 0;
288     end
289
290     % Placement of elements in the structure
291     for j=1:nn
292         Sat_Network(i).Iteration(it).Satellites(j).NORAD = NORAD_Network(j,1);
293         Sat_Network(i).Iteration(it).Satellites(j).Node_Degree = nd(j,1);
294         Sat_Network(i).Iteration(it).Satellites(j).Path_Length = pl(j,1);
295         Sat_Network(i).Iteration(it).Satellites(j).Efficiency = e(j,1);
296         Sat_Network(i).Iteration(it).Satellites(j).Local_Clustering_Coefficient = lcc(
                j,1);
297         Sat_Network(i).Iteration(it).Satellites(j).Betweenness_Centrality = b(j,1);
298         Sat_Network(i).Iteration(it).Satellites(j).Norm_Betweenness_Centrality =
                Norm_b(j,1);
299     end
300     Sat_Network(i).Iteration(it).Network = G;
301     Sat_Network(i).Iteration(it).Distances = D;
302     Sat_Network(i).Iteration(it).n_Edges = ne;
303     Sat_Network(i).Iteration(it).Characteristic_Path_Length = cpl;
304     Sat_Network(i).Iteration(it).Efficiency = ef;
305     Sat_Network(i).Iteration(it).Global_Clustering_Coefficient = gcl;
306 end
307
308 %% Sat_Multiplex structure
309
310 % Degree, path length, efficiency, local clustering coefficient and betweenness
                centrality matrixes for each satellite and each date
311 Degree = zeros(n_Sat_Network,n_d_grid_Network);
312 PL = zeros(n_Sat_Network,n_d_grid_Network);
313 E = zeros(n_Sat_Network,n_d_grid_Network);
314 LCC = zeros(n_Sat_Network,n_d_grid_Network);
315 Norm_Betweenness_Centrality = zeros(n_Sat_Network,n_d_grid_Network);
316 for i=1:n_d_grid_Network
317     for j=1:n_Sat_Network
318         Degree(j,i) = sum(Sat_Network(i).Iteration(it).Adjacency_Matrix(j,:));
319         PL(j,i) = Sat_Network(i).Iteration(it).Satellites(j).Path_Length;

```

III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```

320         E(j,i) = Sat_Network(i).Iteration(it).Satellites(j).Efficiency;
321         LCC(j,i) = Sat_Network(i).Iteration(it).Satellites(j).
            Local_Clustering_Coefficient;
322         Norm_Betweenness_Centrality(j,i) = Sat_Network(i).Iteration(it).Satellites(j).
            Norm_Betweenness_Centrality;
323     end
324 end
325
326 % Mean and standar deviation degree, path length, efficiency, local clustering
    coefficient and betweenness centrality for each satellite
327 Av_Degree = zeros(n_Sat_Network,1);
328 Av_PL = zeros(n_Sat_Network,1);
329 Av_E = zeros(n_Sat_Network,1);
330 Av_LCC = zeros(n_Sat_Network,1);
331 Av_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
332 Std_Degree = zeros(n_Sat_Network,1);
333 Std_PL = zeros(n_Sat_Network,1);
334 Std_E = zeros(n_Sat_Network,1);
335 Std_LCC = zeros(n_Sat_Network,1);
336 Std_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
337 for i=1:n_Sat_Network
338     Av_Degree(i,1) = mean(Degree(i,:));
339     Av_PL(i,1) = mean(PL(i,:));
340     Av_E(i,1) = mean(E(i,:));
341     Av_LCC(i,1) = mean(LCC(i,:));
342     Av_Norm_Betweenness_Centrality(i,1) = mean(Norm_Betweenness_Centrality(i,:));
343     Std_Degree(i,1) = std(Degree(i,:));
344     Std_PL(i,1) = std(PL(i,:));
345     Std_E(i,1) = std(E(i,:));
346     Std_LCC(i,1) = std(LCC(i,:));
347     Std_Norm_Betweenness_Centrality(i,1) = std(Norm_Betweenness_Centrality(i,:));
348 end
349
350 % Generate NORAD names column for the whole period
351 for i=1:n_Sat_Network
352     Sat_Multiplex(it).Satellites(i).NORAD = NORAD_Network(i,1); % NORAD names column (
        double)
353 end
354
355 % Place mean and standar deviation degree, path length, efficiency, local clustering
    coefficient and betweenness centrality for each satellite
356 for i=1:n_Sat_Network
357     Sat_Multiplex(it).Satellites(i).Av_Degree = Av_Degree(i,1);
358     Sat_Multiplex(it).Satellites(i).Std_Degree = Std_Degree(i,1);
359     Sat_Multiplex(it).Satellites(i).Av_Path_Length = Av_PL(i,1);
360     Sat_Multiplex(it).Satellites(i).Std_Path_Length = Std_PL(i,1);
361     Sat_Multiplex(it).Satellites(i).Av_Efficiency = Av_E(i,1);
362     Sat_Multiplex(it).Satellites(i).Std_Efficiency = Std_E(i,1);
363     Sat_Multiplex(it).Satellites(i).Av_LCC = Av_LCC(i,1);
364     Sat_Multiplex(it).Satellites(i).Std_LCC = Std_LCC(i,1);
365     Sat_Multiplex(it).Satellites(i).Av_Norm_Betweenness_Centrality =
        Av_Norm_Betweenness_Centrality(i,1);
366     Sat_Multiplex(it).Satellites(i).Std_Norm_Betweenness_Centrality =
        Std_Norm_Betweenness_Centrality(i,1);
367 end
368
369 % Number of edges, characteristic path length, efficiency and global clustering
    coefficient for each date
370 n_Edges = zeros(n_d_grid_Network,1);
371 CPL = zeros(n_d_grid_Network,1);
372 Efficiency = zeros(n_d_grid_Network,1);
373 GCC = zeros(n_d_grid_Network,1);
374 for i=1:n_d_grid_Network
375     n_Edges(i,1) = Sat_Network(i).Iteration(it).n_Edges;
376     CPL(i,1) = Sat_Network(i).Iteration(it).Characteristic_Path_Length;
377     Efficiency(i,1) = Sat_Network(i).Iteration(it).Efficiency;
378     GCC(i,1) = Sat_Network(i).Iteration(it).Global_Clustering_Coefficient;
379 end
380
381 % Mean and standar deviation number of edges, characteristic path length, efficiency
    and global clustering coefficient for each date
382 Sat_Multiplex(it).Av_n_Edges = mean(n_Edges);
383 Sat_Multiplex(it).Std_n_Edges = std(n_Edges);
384 Sat_Multiplex(it).Av_Characteristic_Path_Length = mean(CPL);
385 Sat_Multiplex(it).Std_Characteristic_Path_Length = std(CPL);

```



```

386 Sat_Multiplex(it).Av_Efficiency = mean(Efficiency);
387 Sat_Multiplex(it).Std_Efficiency = std(Efficiency);
388 Sat_Multiplex(it).Av_Global_Clustering_Coefficient = mean(GCC);
389 Sat_Multiplex(it).Std_Global_Clustering_Coefficient = std(GCC);
390
391 % Select method to identify most vulnerable satellite [7]
392 switch(method)
393     case 1 % Satellite with higher centrality
394         s = find(Av_Norm_Betweenness_Centrality==max(Av_Norm_Betweenness_Centrality));
395             % Allows to find more than one index
396             % If 2 satellites have the same centrality the one with higher degree is
397             chosen
398             if length(s) > 1
399                 Degree_aux = zeros(1,length(s));
400                 for i=1:length(s)
401                     Degree_aux(i) = Av_Degree(s(i));
402                 end
403                 s = find(Av_Degree==max(Degree_aux));
404                 % If 2 satellites have the same degree the one with lower LCC is chosen
405                 if length(s) > 1
406                     LCC_aux = zeros(1,length(s));
407                     for i=1:length(s)
408                         LCC_aux(i) = Av_LCC(s(i));
409                     end
410                     s = find(Av_LCC==min(LCC_aux));
411                     % If 2 satellites have the same LCC the first one is chosen
412                     if length(s) > 1
413                         s = s(1);
414                     end
415                 end
416             end
417         case 2 % Satellite with higher degree
418             s = find(Av_Degree==max(Av_Degree));
419             % If 2 satellites have the same degree the one with lower LCC is chosen
420             if length(s) > 1
421                 LCC_aux = zeros(1,length(s));
422                 for i=1:length(s)
423                     LCC_aux(i) = Av_LCC(s(i));
424                 end
425                 s = find(Av_LCC==min(LCC_aux));
426                 % If 2 satellites have the same LCC the first one is chosen
427                 if length(s) > 1
428                     s = s(1);
429                 end
430             end
431         case 3 % Satellite with lower local clustering coefficient
432             s = find(Av_LCC==min(Av_LCC));
433             % If 2 satellites have the same LCC the one with higher degree is chosen
434             if length(s) > 1
435                 nd_aux = zeros(1,length(s));
436                 for i=1:length(s)
437                     nd_aux(i) = Av_Degree(s(i));
438                 end
439                 s = find(Av_Degree==max(nd_aux));
440                 % If 2 satellites have the same degree the first one is chosen
441                 if length(s) > 1
442                     s = s(1);
443                 end
444             end
445         case 4 % Satellite with higher efficiency
446             s = find(Av_Efficiency==max(Av_Efficiency)); % Allows to find more than one
447             index
448             % If 2 satellites have the same efficiency the one with higher degree is chosen
449             if length(s) > 1
450                 Degree_aux = zeros(1,length(s));
451                 for i=1:length(s)
452                     Degree_aux(i) = Av_Degree(s(i));
453                 end
454                 s = find(Av_Degree==max(Degree_aux));
455                 % If 2 satellites have the same degree the one with lower LCC is chosen
456                 if length(s) > 1
457                     LCC_aux = zeros(1,length(s));
458                     for i=1:length(s)
459                         LCC_aux(i) = Av_LCC(s(i));
460                     end
461                 end
462             end
463     end

```

```

458         s = find(Av_LCC==min(LCC_aux));
459         % If 2 satellites have the same LCC the first one is chosen
460         if length(s) > 1
461             s = s(1);
462         end
463     end
464 end
465 end
466 end
467
468 %% Sat_Monoplex
469
470 % Weight matrix for the whole period
471 W = zeros(n_Sat,n_Sat);
472 for i=1:n_d_grid_Network
473     W = W + Sat_Network(i).Iteration(1).Adjacency_Matrix;
474 end
475 W = W/n_d_grid_Network;
476 Sat_Monoplex(1).Weight_Matrix = W;
477
478 % Sat_Monoplex structure
479 for it=1:it_max
480     % Changes for each iteration
481     if it>1
482         % Changes in NORAD names list
483         Sat_Monoplex(it).Missed_Satellite = NORAD_Monoplex(s,:); % Missed satellite
484         NORAD_Monoplex(s,:) = [];
485
486         % Changes in adjacency matrixes for each date
487         Sat_Monoplex(it).Weight_Matrix = Sat_Monoplex(it-1).Weight_Matrix;
488         Sat_Monoplex(it).Weight_Matrix(s,:) = [];
489         Sat_Monoplex(it).Weight_Matrix(:,s) = [];
490     end
491
492     % Weight matrix
493     W = Sat_Monoplex(it).Weight_Matrix;
494
495     % Save weight matrix as a graph
496     G = graph(W,NORAD_Monoplex');
497
498     % Node degree
499     nd = degree(G);
500     nd_Wei = sum(W,2);
501
502     % Number of nodes and edges
503     nn = size(G.Nodes,1);
504     ne = size(G.Edges,1);
505
506     % Shortest path length between nodes (distances) matrix
507     W_Inv = zeros(nn);
508     for i=1:nn
509         for j=1:nn
510             if W(i,j) == 0
511                 W_Inv(i,j) = 0;
512             else
513                 W_Inv(i,j) = 1/W(i,j); % If the graph is Wei those weights are used as the
                    distances along the edges in the graph; because of that the inverse is
                    used
514             end
515         end
516     end
517     G_Inv = graph(W_Inv,NORAD_Monoplex');
518     D = distances(G_Inv);
519
520     % Path length of each node
521     pl_Wei = sum(D)/(nn-1);
522
523     % Characteristic path length of the network
524     cpl_Wei = sum(pl_Wei)/nn;
525
526     % Efficiency of each node
527     D_Inv = zeros(nn);
528     for i=1:nn
529         for j=1:nn
530             if D(i,j) == 0

```

```

531         D_Inv(i,j) = 0;
532     else
533         D_Inv(i,j) = 1/D(i,j);
534     end
535 end
536 end
537 e_Wei = sum(D_Inv)/(nn-1);
538
539 % Efficiency of the network
540 ef_Wei = sum(e_Wei)/nn;
541
542 % Weighted local clustering coefficient
543 % Reference: Complex network measures of brain connectivity: Uses and interpretations.
544 % Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
545 % Files: https://sites.google.com/site/bctnet/
546 lcc_Wei = clustering_coef_wu(W);
547
548 % Weighted global clustering coefficient
549 gcl_Wei = sum(lcc_Wei)/nn;
550
551 % Weighted normalized betweenness centrality
552 b_Wei = 2*centrality(G_Inv,'betweenness','Cost',G_Inv.Edges.Weight); % 2*B because
553 % network is undirected
554 if nn>2
555     Norm_b_Wei = b_Wei/((nn-1)*(nn-2));
556 elseif nn==2
557     Norm_b_Wei = [0;0]; % If there are only 2 satellites left it assigns centrality to
558 % both
559 else
560     Norm_b_Wei = 0;
561 end
562
563 % Placement of elements in the structure
564 for j=1:nn
565     Sat_Monoplex(it).Satellites(j).NORAD = NORAD_Monoplex(j,1);
566     Sat_Monoplex(it).Satellites(j).Node_Degree = nd(j,1);
567     Sat_Monoplex(it).Satellites(j).Node_Degree_Wei = nd_Wei(j,1);
568     Sat_Monoplex(it).Satellites(j).Path_Length_Wei = pl_Wei(j,1);
569     Sat_Monoplex(it).Satellites(j).Efficiency_Wei = e_Wei(j,1);
570     Sat_Monoplex(it).Satellites(j).Local_Clustering_Coefficient_Wei = lcc_Wei(j,1);
571     Sat_Monoplex(it).Satellites(j).Betweenness_Centrality_Wei = b_Wei(j,1);
572     Sat_Monoplex(it).Satellites(j).Norm_Betweenness_Centrality_Wei = Norm_b_Wei(j,1);
573 end
574 Sat_Monoplex(it).Network = G;
575 Sat_Monoplex(it).Network_Inv = G_Inv;
576 Sat_Monoplex(it).Distances = D;
577 Sat_Monoplex(it).Distances_Inv = D_Inv;
578 Sat_Monoplex(it).n_Edges = ne;
579 Sat_Monoplex(it).Characteristic_Path_Length_Wei = cpl_Wei;
580 Sat_Monoplex(it).Efficiency_Wei = ef_Wei;
581 Sat_Monoplex(it).Global_Clustering_Coefficient_Wei = gcl_Wei;
582
583 % Select method to identify most vulnerable satellite [7]
584 switch(method)
585     case 1 % Satellite with higher weighted centrality
586         s = find(Norm_b_Wei==max(Norm_b_Wei)); % Allows to find more than one index
587         % If 2 satellites have the same weighted centrality the one with higher
588         % weighted degree is chosen
589         if length(s) > 1
590             Degree_aux = zeros(1,length(s));
591             for i=1:length(s)
592                 Degree_aux(i) = nd_Wei(s(i));
593             end
594             s = find(nd_Wei==max(Degree_aux));
595         % If 2 satellites have the same weighted degree the one with lower
596         % weighted LCC is chosen
597         if length(s) > 1
598             LCC_aux = zeros(1,length(s));
599             for i=1:length(s)
600                 LCC_aux(i) = lcc_Wei(s(i));
601             end
602             s = find(lcc_Wei==min(LCC_aux));
603         % If 2 satellites have the same weighted LCC the first one is chosen
604         if length(s) > 1
605             s = s(1);

```

```

601         end
602     end
603 end
604 case 2 % Satellite with higher weighted degree
605     s = find(nd_Wei==max(nd_Wei));
606     % If 2 satellites have the same weighted degree the one with lower weighted
        LCC is chosen
607     if length(s) > 1
608         LCC_aux = zeros(1,length(s));
609         for i=1:length(s)
610             LCC_aux(i) = lcc_Wei(s(i));
611         end
612         s = find(lcc_Wei==min(LCC_aux));
613         % If 2 satellites have the same weighted LCC the first one is chosen
614         if length(s) > 1
615             s = s(1);
616         end
617     end
618 case 3 % Satellite with lower weighted local clustering coefficient
619     s = find(lcc_Wei==min(lcc_Wei));
620     % If 2 satellites have the same weighted LCC the one with higher weighted
        degree is chosen
621     if length(s) > 1
622         nd_aux = zeros(1,length(s));
623         for i=1:length(s)
624             nd_aux(i) = nd_Wei(s(i));
625         end
626         s = find(nd_Wei==max(nd_aux));
627         % If 2 satellites have the same weighted LCC the first one is chosen
628         if length(s) > 1
629             s = s(1);
630         end
631     end
632 case 4 % Satellite with higher weighted efficiency
633     s = find(e_Wei==max(e_Wei)); % Allows to find more than one index
634     % If 2 satellites have the same weighted efficiency the one with higher
        weighted degree is chosen
635     if length(s) > 1
636         Degree_aux = zeros(1,length(s));
637         for i=1:length(s)
638             Degree_aux(i) = nd_Wei(s(i));
639         end
640         s = find(nd_Wei==max(Degree_aux));
641         % If 2 satellites have the same weighted degree the one with lower
        weighted LCC is chosen
642     if length(s) > 1
643         LCC_aux = zeros(1,length(s));
644         for i=1:length(s)
645             LCC_aux(i) = lcc_Wei(s(i));
646         end
647         s = find(lcc_Wei==min(LCC_aux));
648         % If 2 satellites have the same weighted LCC the first one is chosen
649         if length(s) > 1
650             s = s(1);
651         end
652     end
653 end
654 end
655 end
656
657 %% Missed satellites lists
658
659 % NORAD ids
660 Missed_Satellites_Monoplex = zeros(it_max-1,1);
661 Missed_Satellites_Multiplex = zeros(it_max-1,1);
662 for i=1:it_max-1
663     Missed_Satellites_Monoplex(i) = str2double(Sat_Monoplex(i+1).Missed_Satellite);
664     Missed_Satellites_Multiplex(i) = str2double(Sat_Multiplex(i+1).Missed_Satellite);
665 end
666
667 % Level of coincidence
668 Coincidence = zeros(it_max-1,1);
669 for i=1:it_max-1
670     for j=1:i
671         for k=1:i

```

```

672         Coincidence(i) = Coincidence(i) + isequal(Missed_Satellites_Monoplex(j),
673             Missed_Satellites_Multiplex(k));
674     end
675 end
676
677 %% Recalculate structure elements
678
679 % Sat_Rec_Network and Sat_Rec_Multiplex initialization
680 NORAD_Network_Rec = NORAD;
681 n_Sat_Network = n_Sat;
682 Sat_Rec_Network = Sat_Network;
683 Sat_Rec_Multiplex(1) = Sat_Multiplex(1);
684
685 for it=2:it_max
686     %% Sat_Rec_Network structure
687
688     % Changes in NORAD names list
689     s = find(str2double(NORAD_Network_Rec)==Missed_Satellites_Monoplex(it-1));
690     Sat_Rec_Multiplex(it).Missed_Satellite = NORAD_Network_Rec(s,:);
691     NORAD_Network_Rec(s,:) = [];
692
693     % Changes in indexes
694     n_Sat_Network = n_Sat_Network - 1;
695
696     % Changes in adjacency matrix
697     for i=1:n_d_grid_Network
698         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix = Sat_Rec_Network(i).Iteration(
699             it-1).Adjacency_Matrix;
700         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(s,:) = [];
701         Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(:,s) = [];
702     end
703
704     % Graph properties of the network for for each date
705     for i=1:n_d_grid_Network
706         % Adjacency matrix
707         A = Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix;
708
709         % Save satellite network as a graph
710         G = graph(A,NORAD_Network_Rec');
711
712         % Node degree
713         nd = degree(G);
714
715         % Number of nodes and edges
716         nn = size(G.Nodes,1);
717         ne = size(G.Edges,1);
718
719         % Shortest path length between nodes matrix
720         D = distances(G);
721
722         % Path length of each node
723         pl = sum(D)/(nn-1);
724
725         % Characteristic path length of the network
726         cpl = sum(pl)/nn;
727
728         % Efficiency of each node
729         D_Inv = zeros(nn);
730         for j=1:nn
731             for k=1:nn
732                 if D(j,k) == 0
733                     D_Inv(j,k) = 0;
734                 else
735                     D_Inv(j,k) = 1/D(j,k);
736                 end
737             end
738         end
739         e = sum(D_Inv)/(nn-1);
740
741         % Efficiency of the network
742         ef = sum(e)/nn;
743
744         % Local clustering coefficient

```

```

744 % Reference: Complex network measures of brain connectivity: Uses and
      interpretations. Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.
      neuroimage.2009.10.003
745 % Files: https://sites.google.com/site/bctnet/
746 lcc = clustering_coef_bu(A);
747
748 % Global clustering coefficient
749 gcl = sum(lcc)/nn;
750
751 % Normalized betweenness centrality
752 b = 2*centrality(G,'betweenness'); % 2*b because network is undirected
753 if nn>2
754     Norm_b = b/((nn-1)*(nn-2));
755 elseif nn==2
756     Norm_b = [0;0]; % If there are only 2 satellites left it assigns centrality to
      both
757 else
758     Norm_b = 0;
759 end
760
761 % Placement of elements in the structure
762 for j=1:nn
763     Sat_Rec_Network(i).Iteration(it).Satellites(j).NORAD = NORAD_Network_Rec(j,1);
764     Sat_Rec_Network(i).Iteration(it).Satellites(j).Node_Degree = nd(j,1);
765     Sat_Rec_Network(i).Iteration(it).Satellites(j).Path_Length = pl(j,1);
766     Sat_Rec_Network(i).Iteration(it).Satellites(j).Efficiency = e(j,1);
767     Sat_Rec_Network(i).Iteration(it).Satellites(j).Local_Clustering_Coefficient =
      lcc(j,1);
768     Sat_Rec_Network(i).Iteration(it).Satellites(j).Betweenness_Centrality = b(j,1)
      ;
769     Sat_Rec_Network(i).Iteration(it).Satellites(j).Norm_Betweenness_Centrality =
      Norm_b(j,1);
770 end
771 Sat_Rec_Network(i).Iteration(it).Network = G;
772 Sat_Rec_Network(i).Iteration(it).Distances = D;
773 Sat_Rec_Network(i).Iteration(it).n_Edges = ne;
774 Sat_Rec_Network(i).Iteration(it).Characteristic_Path_Length = cpl;
775 Sat_Rec_Network(i).Iteration(it).Efficiency = ef;
776 Sat_Rec_Network(i).Iteration(it).Global_Clustering_Coefficient = gcl;
777 end
778
779 %% Sat_Multiplex structure
780
781 % Degree, path length, efficiency, local clustering coefficient and betweenness
      centrality matrixes for each satellite and each date
782 Degree = zeros(n_Sat_Network,n_d_grid_Network);
783 PL = zeros(n_Sat_Network,n_d_grid_Network);
784 E = zeros(n_Sat_Network,n_d_grid_Network);
785 LCC = zeros(n_Sat_Network,n_d_grid_Network);
786 Norm_Betweenness_Centrality = zeros(n_Sat_Network,n_d_grid_Network);
787 for i=1:n_d_grid_Network
788     for j=1:n_Sat_Network
789         Degree(j,i) = sum(Sat_Rec_Network(i).Iteration(it).Adjacency_Matrix(j,:));
790         PL(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).Path_Length;
791         E(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).Efficiency;
792         LCC(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites(j).
      Local_Clustering_Coefficient;
793         Norm_Betweenness_Centrality(j,i) = Sat_Rec_Network(i).Iteration(it).Satellites
      (j).Norm_Betweenness_Centrality;
794     end
795 end
796
797 % Mean and standar deviation degree, path length, efficiency, local clustering
      coefficient and betweenness centrality for each satellite
798 Av_Degree = zeros(n_Sat_Network,1);
799 Av_PL = zeros(n_Sat_Network,1);
800 Av_E = zeros(n_Sat_Network,1);
801 Av_LCC = zeros(n_Sat_Network,1);
802 Av_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
803 Std_Degree = zeros(n_Sat_Network,1);
804 Std_PL = zeros(n_Sat_Network,1);
805 Std_E = zeros(n_Sat_Network,1);
806 Std_LCC = zeros(n_Sat_Network,1);
807 Std_Norm_Betweenness_Centrality = zeros(n_Sat_Network,1);
808 for i=1:n_Sat_Network

```

```

809     Av_Degree(i,1) = mean(Degree(i,:));
810     Av_PL(i,1) = mean(PL(i,:));
811     Av_E(i,1) = mean(E(i,:));
812     Av_LCC(i,1) = mean(LCC(i,:));
813     Av_Norm_Betweenness_Centrality(i,1) = mean(Norm_Betweenness_Centrality(i,:));
814     Std_Degree(i,1) = std(Degree(i,:));
815     Std_PL(i,1) = std(PL(i,:));
816     Std_E(i,1) = std(E(i,:));
817     Std_LCC(i,1) = std(LCC(i,:));
818     Std_Norm_Betweenness_Centrality(i,1) = std(Norm_Betweenness_Centrality(i,:));
819 end
820
821 % Generate NORAD names column for the whole period
822 for i=1:n_Sat_Network
823     Sat_Rec_Multiplex(it).Satellites(i).NORAD = NORAD_Network_Rec(i,1); % NORAD names
824     column (double)
825 end
826
827 % Place mean and standar deviation degree, path length, efficiency, local clustering
828 % coefficient and betweenness centrality for each satellite
829 for i=1:n_Sat_Network
830     Sat_Rec_Multiplex(it).Satellites(i).Av_Degree = Av_Degree(i,1);
831     Sat_Rec_Multiplex(it).Satellites(i).Std_Degree = Std_Degree(i,1);
832     Sat_Rec_Multiplex(it).Satellites(i).Av_Path_Length = Av_PL(i,1);
833     Sat_Rec_Multiplex(it).Satellites(i).Std_Path_Length = Std_PL(i,1);
834     Sat_Rec_Multiplex(it).Satellites(i).Av_Efficiency = Av_E(i,1);
835     Sat_Rec_Multiplex(it).Satellites(i).Std_Efficiency = Std_E(i,1);
836     Sat_Rec_Multiplex(it).Satellites(i).Av_LCC = Av_LCC(i,1);
837     Sat_Rec_Multiplex(it).Satellites(i).Std_LCC = Std_LCC(i,1);
838     Sat_Rec_Multiplex(it).Satellites(i).Av_Norm_Betweenness_Centrality =
839     Av_Norm_Betweenness_Centrality(i,1);
840     Sat_Rec_Multiplex(it).Satellites(i).Std_Norm_Betweenness_Centrality =
841     Std_Norm_Betweenness_Centrality(i,1);
842 end
843
844 % Number of edges, characteristic path length, efficiency and global clustering
845 % coefficient for each date
846 n_Edges = zeros(n_d_grid_Network,1);
847 CPL = zeros(n_d_grid_Network,1);
848 Efficiency = zeros(n_d_grid_Network,1);
849 GCC = zeros(n_d_grid_Network,1);
850 for i=1:n_d_grid_Network
851     n_Edges(i,1) = Sat_Rec_Network(i).Iteration(it).n_Edges;
852     CPL(i,1) = Sat_Rec_Network(i).Iteration(it).Characteristic_Path_Length;
853     Efficiency(i,1) = Sat_Rec_Network(i).Iteration(it).Efficiency;
854     GCC(i,1) = Sat_Rec_Network(i).Iteration(it).Global_Clustering_Coefficient;
855 end
856
857 % Mean and standar deviation number of edges, characteristic path length, efficiency
858 % and global clustering coefficient for each date
859 Sat_Rec_Multiplex(it).Av_n_Edges = mean(n_Edges);
860 Sat_Rec_Multiplex(it).Std_n_Edges = std(n_Edges);
861 Sat_Rec_Multiplex(it).Av_Characteristic_Path_Length = mean(CPL);
862 Sat_Rec_Multiplex(it).Std_Characteristic_Path_Length = std(CPL);
863 Sat_Rec_Multiplex(it).Av_Efficiency = mean(Efficiency);
864 Sat_Rec_Multiplex(it).Std_Efficiency = std(Efficiency);
865 Sat_Rec_Multiplex(it).Av_Global_Clustering_Coefficient = mean(GCC);
866 Sat_Rec_Multiplex(it).Std_Global_Clustering_Coefficient = std(GCC);
867 end
868
869 % Sat_Rec_Monoplex initialization
870 NORAD_Monoplex_Rec = NORAD;
871 Sat_Rec_Monoplex(1) = Sat_Monoplex(1);
872
873 % Sat_Rec_Monoplex structure
874 for it=2:it_max
875     % Changes in NORAD names list
876     s = find(str2double(NORAD_Monoplex_Rec)==Missed_Satellites_Multiplex(it-1));
877     Sat_Rec_Monoplex(it).Missed_Satellite = NORAD_Monoplex_Rec(s,:);
878     NORAD_Monoplex_Rec(s,:) = [];
879
880     % Changes in weight matrixes
881     Sat_Rec_Monoplex(it).Weight_Matrix = Sat_Rec_Monoplex(it-1).Weight_Matrix;
882     Sat_Rec_Monoplex(it).Weight_Matrix(s,:) = [];
883     Sat_Rec_Monoplex(it).Weight_Matrix(:,s) = [];

```

```

878
879     % Weight matrix
880     W = Sat_Rec_Monoplex(it).Weight_Matrix;
881
882     % Save weight matrix as a graph
883     G = graph(W,NORAD_Monoplex_Rec');
884
885     % Node degree
886     nd = degree(G);
887     nd_Wei = sum(W,2);
888
889     % Number of nodes and edges
890     nn = size(G.Nodes,1);
891     ne = size(G.Edges,1);
892
893     % Shortest path length between nodes (distances) matrix
894     W_Inv = zeros(nn);
895     for i=1:nn
896         for j=1:nn
897             if W(i,j) == 0
898                 W_Inv(i,j) = 0;
899             else
900                 W_Inv(i,j) = 1/W(i,j); % If the graph is Wei those weights are used as the
                                     % distances along the edges in the graph; because of that the inverse is
                                     % used
901             end
902         end
903     end
904     G_Inv = graph(W_Inv,NORAD_Monoplex_Rec');
905     D = distances(G_Inv);
906
907     % Path length of each node
908     pl_Wei = sum(D)'/(nn-1);
909
910     % Characteristic path length of the network
911     cpl_Wei = sum(pl_Wei)/nn;
912
913     % Efficiency of each node
914     D_Inv = zeros(nn);
915     for i=1:nn
916         for j=1:nn
917             if D(i,j) == 0
918                 D_Inv(i,j) = 0;
919             else
920                 D_Inv(i,j) = 1/D(i,j);
921             end
922         end
923     end
924     e_Wei = sum(D_Inv)'/(nn-1);
925
926     % Efficiency of the network
927     ef_Wei = sum(e_Wei)/nn;
928
929     % Weighted local clustering coefficient
930     % Reference: Complex network measures of brain connectivity: Uses and interpretations.
931     % Rubinov M, Sporns O (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
932     % Files: https://sites.google.com/site/bctnet/
933     lcc_Wei = clustering_coef_wu(W);
934
935     % Weighted global clustering coefficient
936     gcl_Wei = sum(lcc_Wei)/nn;
937
938     % Weighted normalized betweenness centrality
939     b_Wei = 2*centrality(G_Inv,'betweenness','Cost',G_Inv.Edges.Weight); % 2*B because
940     % network is undirected
941     if nn>2
942         Norm_b_Wei = b_Wei/((nn-1)*(nn-2));
943     elseif nn==2
944         Norm_b_Wei = [0;0]; % If there are only 2 satellites left it assigns centrality to
945         % both
946     else
947         Norm_b_Wei = 0;
948     end
949
950     % Placement of elements in the structure

```


III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```

948     for j=1:nn
949         Sat_Rec_Monoplex(it).Satellites(j).NORAD = NORAD_Monoplex_Rec(j,1);
950         Sat_Rec_Monoplex(it).Satellites(j).Node_Degree = nd(j,1);
951         Sat_Rec_Monoplex(it).Satellites(j).Node_Degree_Wei = nd_Wei(j,1);
952         Sat_Rec_Monoplex(it).Satellites(j).Path_Length_Wei = pl_Wei(j,1);
953         Sat_Rec_Monoplex(it).Satellites(j).Efficiency_Wei = e_Wei(j,1);
954         Sat_Rec_Monoplex(it).Satellites(j).Local_Clustering_Coefficient_Wei = lcc_Wei(j
,1);
955         Sat_Rec_Monoplex(it).Satellites(j).Betweenness_Centrality_Wei = b_Wei(j,1);
956         Sat_Rec_Monoplex(it).Satellites(j).Norm_Betweenness_Centrality_Wei = Norm_b_Wei(j
,1);
957     end
958     Sat_Rec_Monoplex(it).Network = G;
959     Sat_Rec_Monoplex(it).Network_Inv = G_Inv;
960     Sat_Rec_Monoplex(it).Distances = D;
961     Sat_Rec_Monoplex(it).Distances_Inv = D_Inv;
962     Sat_Rec_Monoplex(it).n_Edges = ne;
963     Sat_Rec_Monoplex(it).Characteristic_Path_Length_Wei = cpl_Wei;
964     Sat_Rec_Monoplex(it).Efficiency_Wei = ef_Wei;
965     Sat_Rec_Monoplex(it).Global_Clustering_Coefficient_Wei = gcl_Wei;
966 end
967
968 %% Method comparission
969
970 % Efficiencies of original methods and recalculations
971 Efficiency_Multiplex = zeros(1,it_max);
972 Efficiency_Multiplex_Rec = zeros(1,it_max);
973 Efficiency_Multiplex_Dif = zeros(1,it_max);
974 Efficiency_Monoplex = zeros(1,it_max);
975 Efficiency_Monoplex_Rec = zeros(1,it_max);
976 Efficiency_Monoplex_Dif = zeros(1,it_max);
977 for it=1:it_max
978     Efficiency_Multiplex(it) = Sat_Multiplex(it).Av_Efficiency;
979     Efficiency_Multiplex_Rec(it) = Sat_Rec_Multiplex(it).Av_Efficiency;
980     Efficiency_Multiplex_Dif(it) = Efficiency_Multiplex_Rec(it) - Efficiency_Multiplex(it)
;
981     Efficiency_Monoplex(it) = Sat_Monoplex(it).Efficiency_Wei;
982     Efficiency_Monoplex_Rec(it) = Sat_Rec_Monoplex(it).Efficiency_Wei;
983     Efficiency_Monoplex_Dif(it) = Efficiency_Monoplex_Rec(it) - Efficiency_Monoplex(it);
984 end
985
986 %% Variable saving
987
988 % Level of coincidence and efficiency
989 An_Coincidence(:,i_point) = Coincidence;
990 An_Efficiency_Multiplex(:,i_point) = Efficiency_Multiplex';
991 An_Efficiency_Multiplex_Rec(:,i_point) = Efficiency_Multiplex_Rec';
992 An_Efficiency_Multiplex_Dif(:,i_point) = Efficiency_Multiplex_Dif';
993 An_Efficiency_Monoplex(:,i_point) = Efficiency_Monoplex';
994 An_Efficiency_Monoplex_Rec(:,i_point) = Efficiency_Monoplex_Rec';
995 An_Efficiency_Monoplex_Dif(:,i_point) = Efficiency_Monoplex_Dif';
996
997 %% Loop actualization
998
999 % Initial date update [4]
1000 s0 = s0 + si;
1001 m0 = m0 + mi;
1002 h0 = h0 + hi;
1003 d0 = d0 + di;
1004 mo0 = mo0 + moi;
1005 y0 = y0 + yi;
1006
1007 end
1008
1009 %% Saving of workspace variables
1010 save('Analysis_Variables','An_Coincidence','An_Efficiency_Multiplex','
An_Efficiency_Multiplex_Rec','An_Efficiency_Multiplex_Dif','An_Efficiency_Monoplex','
An_Efficiency_Monoplex_Rec','An_Efficiency_Monoplex_Dif');
1011
1012 %% Stops clock to measure time of execution
1013 toc

```

3.2. Programa para el cálculo de la probabilidad de coincidencia aleatoria entre satélites eliminados por cada uno de los métodos de análisis (Probabilidad.m)

```
1 %% Probabilidad.m
2 % Antonio Serda Mena. (2022). Application of Complex Network Theory to the Vulnerability
   Analysis of a Satellite Network
3 % Obtains random probability distribution of matching eliminated satellites using multiplex
   and monoplex method
4
5 %% Initial adjustments
6
7 % Number of satellites
8 s = 32;
9
10 % Variable initialization
11 Probability = zeros(s+1,s);
12 F_O = zeros(s+1,s); % Favorable outcomes
13 P_O = zeros(1,s); % Possible outcomes
14
15 %% Probability calculation loop
16 for it = 1:s
17     for n = 1:it+1 % One additional initial row for representing 0 coincidences
18         if it-(n-1) <= s-it % Only calculates combinations if the number of elements in each
               group is inferior to the total number of elements
19             F_O(n,it) = nchoosek(s, (n-1))*nchoosek(s-(n-1),it-(n-1))*nchoosek(s-it,it-(n-1));
20         else
21             F_O(n,it) = 0;
22         end
23         P_O(it) = nchoosek(s,it)*nchoosek(s,it);
24         Probability(n,it) = F_O(n,it)/P_O(it);
25     end
26 end
27
28 %% Saving of workspace variables
29 save('Probability_Variables','Probability');
```

3.3. Programa para la comparación de la eficacia de los métodos multiplex y monoplex (Comparacion.m)

```
1 %% Comparacion.m
2 % Antonio Serda Mena. (2022). Application of Complex Network Theory to the Vulnerability
   Analysis of a Satellite Network
3 % Compares multiplex and monoplex method effectiveness
4
5 %% Considerations
6 % It is needed to have experiments data first, using the program Analisis.mat
7 % It is needed to have probability distribution first, using the program Probabilidad.mat
8 % discretisample.m function needed
9 % Reference: Dahua Lin (2022). Sampling from a discrete distribution (https://www.mathworks.com/matlabcentral/fileexchange/21912-sampling-from-a-discrete-distribution), MATLAB
   Central File Exchange. Retrieved July 16, 2022.
10 % chi2test2.m function needed
11 % Reference: James Meldrum (2022). CHI2TEST2 (https://www.mathworks.com/matlabcentral/fileexchange/41025-chi2test2), MATLAB Central File Exchange. Retrieved July 16, 2022.
12
13 %% Initialization
14 close all force
15 clear
16 clc
17
18 %% Functioning parameters
19
20 % Loading of experiment variables
21 load('Analysis_Variables');
```

III. CÓDIGO FUENTE 3. COMPARACIÓN DE MÉTODOS DE ANÁLISIS DE VULNERABILIDADES DE LA RED DE SATÉLITES

```

22
23 % Loading of random probability distribution
24 load('Probability_Variables');
25
26 % Iteration interval
27 it0 = 1; % Initial iteration
28 itf = 31; % Final iteration
29
30 %% Initial adjustments
31
32 % Number of rows and columns of variables
33 n_it = size(An_Coincidence,1); % Number of iterations
34 n_Experiments = size(An_Coincidence,2); % Number of experiments
35
36 % Grid array of coincidences (starts at 0)
37 n_Coincidence_grid = 0:1:n_it;
38
39 % Grid array of iterations
40 n_it_grid = 1:1:n_it;
41
42 % Average values of efficiency
43 Av_Efficiency_Multiplex = zeros(n_it,1);
44 Av_Efficiency_Multiplex_Rec = zeros(n_it,1);
45 Av_Efficiency_Monoplex = zeros(n_it,1);
46 Av_Efficiency_Monoplex_Rec = zeros(n_it,1);
47 for i=1:n_it
48     Av_Efficiency_Multiplex(i,1) = mean(An_Efficiency_Multiplex(i,:));
49     Av_Efficiency_Multiplex_Rec(i,1) = mean(An_Efficiency_Multiplex_Rec(i,:));
50     Av_Efficiency_Monoplex(i,1) = mean(An_Efficiency_Monoplex(i,:));
51     Av_Efficiency_Monoplex_Rec(i,1) = mean(An_Efficiency_Monoplex_Rec(i,:));
52 end
53
54 % Elimination of last rows (empty)
55 An_Efficiency_Multiplex(n_it+1,:) = [];
56 An_Efficiency_Multiplex_Rec(n_it+1,:) = [];
57 An_Efficiency_Multiplex_Dif(n_it+1,:) = [];
58 An_Efficiency_Monoplex(n_it+1,:) = [];
59 An_Efficiency_Monoplex_Rec(n_it+1,:) = [];
60 An_Efficiency_Monoplex_Dif(n_it+1,:) = [];
61
62 % Elimination of last row and column (empty)
63 Probability(n_it+2,:) = [];
64 Probability(:,n_it+1) = [];
65
66 %% Comparisson of number of coincident eliminated satellites per iteration with random
    probability function
67
68 % Pearson's chi-squared test
69 Probability_Sample = zeros(n_it,n_Experiments);
70 h = zeros(n_it,1);
71 p = zeros(n_it,1);
72 for i=1:n_it
73     Probability_Sample(i,:) = discretessample(Probability(:,i),n_Experiments)-1;
74     [h(i,1),p(i,1)] = chi2test2(An_Coincidence(i,:)',Probability_Sample(i,:)',0.05);
75 end
76
77 % Coincidence distribution
78 Coincidence = zeros(n_it+1,n_it);
79 for i=1:n_it+1 % Coincidence number (i=1 -> 0 coincidences)
80     for j=1:n_it % Iteration index
81         for k=1:n_Experiments % Coincidence index
82             Coincidence(i,j) = Coincidence(i,j) + isequal(An_Coincidence(j,k),i-1);
83         end
84     end
85 end
86 Coincidence = Coincidence/n_Experiments; % Normalization
87
88 % Mean values of random and experimental distributions
89 Av_Coincidence = zeros(n_it,1);
90 Std_Coincidence = zeros(n_it,1);
91 Av_Probability = zeros(n_it,1);
92 Prop_Coincidence = zeros(n_it,1);
93 for i=1:n_it
94     Av_Coincidence(i,1) = mean(An_Coincidence(i,:));
95     Std_Coincidence(i,1) = std(An_Coincidence(i,:));

```

```

96     for j=2:n_it+1
97         Av_Probability(i,1) = Av_Probability(i,1) + (j-1)*Probability(j,i);
98     end
99     Prop_Coincidence(i,1) = (Av_Coincidence(i,1) - Av_Probability(i,1))/(i - Av_Probability(i
    ,1));
100 end
101
102 %% Probability distribution and coincidence distribuion plots
103
104 % Distributions per iteration
105 for i=1:n_it
106     figure
107     bar(n_Coincidence_grid,Probability(:,i),'FaceAlpha',.75)
108     hold on
109     bar(n_Coincidence_grid,Coincidence(:,i),'FaceAlpha',.75)
110     hold off
111     title('Comparison between the proportion of random and experiment matches for the
        Globalstar constellation during month 09/2021. Iteration: ',num2str(i))
112     xlabel('Number of coincidences')
113     ylabel('Probability distribution')
114     legend('Random','Experiments','FontSize',12)
115     xticks(n_Coincidence_grid)
116 end
117
118 % Mean values
119 figure
120 plot(n_it_grid,n_it_grid,'-o',n_it_grid,Av_Probability,'-o',n_it_grid,Av_Coincidence,'-o')
121 title('Mean values of the distributions of random matches and of the experiments for the
        Globalstar constellation during month 09/2021')
122 xlabel('Iteration')
123 ylabel('Number of coincidences')
124 legend('Maximum','Random','Experiments','FontSize',12)
125 xticks(n_it_grid)
126 yticks(n_Coincidence_grid)
127
128 %% Multiplex plots
129
130 figure
131 plot(n_it_grid(it0:itf),Av_Efficiency_Multiplex(it0:itf,1),'-o',n_it_grid(it0:itf),
        Av_Efficiency_Multiplex_Rec(it0:itf,1),'-o')
132 title('Evolution of the efficiency with each iteration of the multiplex method for the
        Globalstar constellation during month 09/2021')
133 xlabel('Iteration')
134 ylabel('Network efficiency')
135 legend('Multiplex','Recalculated multiplex','FontSize',12)
136 xticks(n_it_grid(it0:itf))
137
138 figure
139 boxplot(An_Efficiency_Multiplex_Dif(it0:itf,:),n_it_grid(it0:itf))
140 title('Evolution of the differences in efficiency with each iteration of the multiplex method
        for the Globalstar constellation during month 09/2021')
141 xlabel('Iteration')
142 ylabel('Network efficiency')
143
144 figure
145 boxplot(An_Efficiency_Multiplex(it0:itf,:),n_it_grid(it0:itf))
146 title('Evolution of efficiency with each iteration of the multiplex method for the Globalstar
        constellation during month 09/2021')
147 xlabel('Iteration')
148 ylabel('Network efficiency')
149
150 figure
151 boxplot(An_Efficiency_Multiplex_Rec(it0:itf,:),n_it_grid(it0:itf))
152 title('Evolution of the efficiency with each iteration of the recalculated multiplex method
        for the Globalstar constellation during month 09/2021')
153 xlabel('Iteration')
154 ylabel('Network efficiency')
155
156 figure
157 subplot(1,2,1)
158 boxplot(An_Efficiency_Multiplex(it0:itf,:),n_it_grid(it0:itf))
159 title('Multiplex')
160 xlabel('Iteration')
161 ylabel('Network efficiency')
162 subplot(1,2,2)

```

```

163 | boxplot(An_Efficiency_Multiplex_Rec(it0:itf,:)',n_it_grid(it0:itf))
164 | title('Recalculated multiplex')
165 | xlabel('Iteration')
166 | ylabel('Network efficiency')
167 |
168 | figure
169 | parallelplot(An_Efficiency_Multiplex', 'DataNormalization', 'none')
170 | title('Evolution of the efficiency with each iteration of the recalculated multiplex method
      | for the Globalstar constellation during month 09/2021')
171 | xlabel('Iteration')
172 | ylabel('Network efficiency')
173 |
174 | figure
175 | parallelplot(An_Efficiency_Multiplex_Rec', 'DataNormalization', 'none')
176 | title('Evolution of the efficiency with each iteration of the recalculated multiplex method
      | for the Globalstar constellation during month 09/2021')
177 | xlabel('Iteration')
178 | ylabel('Network efficiency')
179 |
180 | %% Monoplex plot
181 |
182 | figure
183 | plot(n_it_grid(it0:itf),Av_Efficiency_Monoplex(it0:itf,1)', '-o',n_it_grid(it0:itf),
      | Av_Efficiency_Monoplex_Rec(it0:itf,1)', '-o')
184 | title('Evolution of the efficiency with each iteration of the monoplex method for the
      | Globalstar constellation during the month 09/2021')
185 | xlabel('Iteration')
186 | ylabel('Network efficiency')
187 | legend('Monoplex', 'Recalculated monoplex', 'FontSize', 12)
188 | xticks(n_it_grid(it0:itf))
189 |
190 | figure
191 | boxplot(An_Efficiency_Monoplex_Dif(it0:itf,:)',n_it_grid(it0:itf))
192 | title('Evolution of the efficiency differences with each iteration of the monoplex method for
      | the Globalstar constellation during month 09/2021')
193 | xlabel('Iteration')
194 | ylabel('Network efficiency')
195 |
196 | figure
197 | boxplot(An_Efficiency_Monoplex(it0:itf,:)',n_it_grid(it0:itf))
198 | title('Evolution of the efficiency with each iteration of the monoplex method for the
      | Globalstar constellation during the month 09/2021')
199 | xlabel('Iteration')
200 | ylabel('Network efficiency')
201 |
202 | figure
203 | boxplot(An_Efficiency_Monoplex_Rec(it0:itf,:)',n_it_grid(it0:itf))
204 | title('Efficiency evolution with each iteration of the recalculated monoplex method for the
      | Globalstar constellation during month 09/2021')
205 | xlabel('Iteration')
206 | ylabel('Network efficiency')
207 |
208 | figure
209 | subplot(1,2,1)
210 | boxplot(An_Efficiency_Monoplex(it0:itf,:)',n_it_grid(it0:itf))
211 | title('Monoplex')
212 | xlabel('Iteration')
213 | ylabel('Network efficiency')
214 | subplot(1,2,2)
215 | boxplot(An_Efficiency_Monoplex_Rec(it0:itf,:)',n_it_grid(it0:itf))
216 | title('Recalculated monoplex')
217 | xlabel('Iteration')
218 | ylabel('Network efficiency')
219 |
220 | figure
221 | parallelplot(An_Efficiency_Monoplex', 'DataNormalization', 'none')
222 | title('Efficiency evolution with each iteration of the recalculated monoplex method for the
      | Globalstar constellation during month 09/2021')
223 | xlabel('Iteration')
224 | ylabel('Network efficiency')
225 |
226 | figure
227 | parallelplot(An_Efficiency_Monoplex_Rec', 'DataNormalization', 'none')
228 | title('Efficiency evolution with each iteration of the recalculated monoplex method for the
      | Globalstar constellation during month 09/2021')

```

```
229 xlabel('Iteration')  
230 ylabel('Network efficiency')
```