

Máster en Estadística e Investigación Operativa

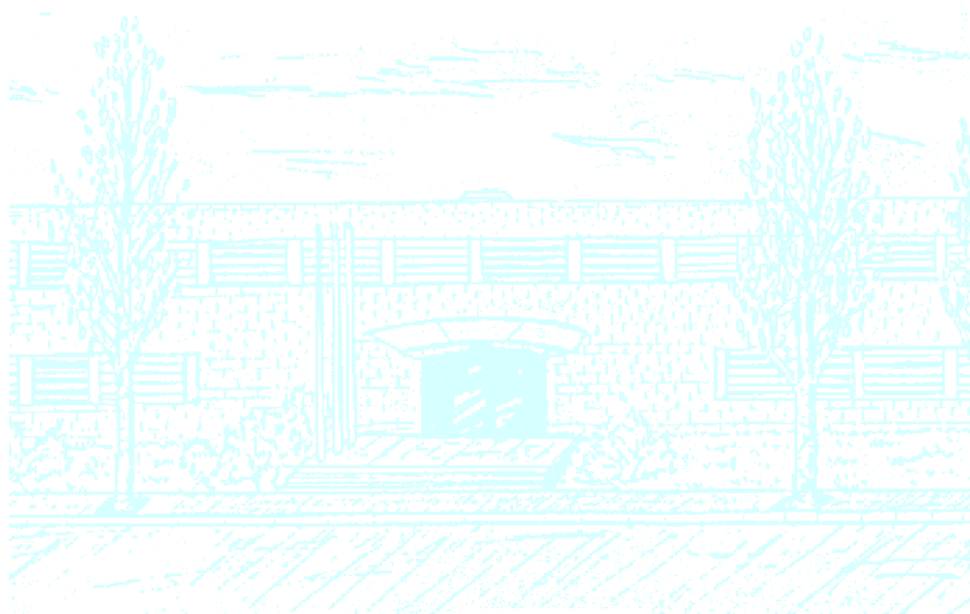
Título: Optimización de Recursos en Redes de Transporte Ópticas Multicapa

Autor: Marc Ruiz Ramírez

Director: Francisco Javier Heredia Cervera
Luis Domingo Velasco Esteban

Departamento: Estadística e Investigación Operativa

Convocatoria: Febrero 2009



Facultat de Matemàtiques
i Estadística

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO DE FIN DE MÁSTER

Índice de contenidos

	Pag.
Capítulo 1. Introducción	1
1.1 – Motivación	1
1.2- Objetivos y estructura del PFC	2
1.3 – Consideraciones	3
Capítulo 2. Introducción a las redes de transporte	5
2.1 – Panorama actual	5
2.2 – Definición de conceptos básicos	6
2.2.1 – Recursos ópticos	6
2.2.2 – Multiplexado de datos	7
2.2.3 – Arquitecturas multicapa	8
2.2.4 – Arquitectura de gestión de las redes de transporte	9
2.2.5 – Gestión de conexiones eléctricas	10
Capítulo 3. Problema de liberación de recursos ópticos	15
3.1 – Planteamiento del problema	16
3.1.1 – Utilización subóptima de recursos ópticos	16
3.1.2 – Objetivo del problema	18
3.1.3 – Consideraciones sobre el conjunto de arcos ópticos	19
3.1.4 – Consideraciones sobre los cambios de ruta de las conexiones eléctricas	20
3.2 – Formulación matemática	24
3.2.1 – Estructura de los datos	24
3.2.2 – Variables de decisión	25
3.2.3 – Restricciones del problema	26
3.2.4 – Función objetivo	29
Capítulo 4. Ejemplos de resolución exacta del problema	31
4.1 – Definición de ejemplos de red	31
4.1.1 – Topología de prueba	31
4.1.2 – Topología Europea	33
4.1.3 – Escenarios de tráfico simulados	35
4.2 – Resolución exacta sobre red de test	38
4.2.1 – Características de la implementación	38
4.2.2 – Resumen de resultados	38
4.2.3 – Análisis del tiempo de computación	40
4.2.4 – Análisis de las modificaciones producidas	41
4.2.5 – Conclusiones	43
4.3 – Resolución exacta sobre red europea	44
4.3.1 – Características de la implementación	44
4.3.2 – Resumen de resultados	44
4.3.3 – Análisis del tiempo de computación	45
4.3.4 – Análisis de las modificaciones producidas	47
4.3.5 – Conclusiones	48
Capítulo 5. Meta-heurística GRASP	49
5.1 – Definiciones básicas	50
5.1.1 – Criterio de selección <i>voraz</i>	50
5.1.2 – Aleatorización de la selección	52
5.2 – Estructura y fases de la meta-heurística GRASP	53
5.2.1 – Estructura elemental de la heurística	53
5.2.2 – Fase I o fase constructiva aleatorizada	55
5.2.3 – Fase II o fase de mejora local	57
5.2.3 – Acción combinada de las dos fases	59
5.3 – Ajuste y sensibilidad de parámetros	59
5.3.1 – Análisis de sensibilidad del parámetro α	60
5.3.2 – Estudio del criterio de parada	61
5.4 – Adaptación de la heurística al problema	63
5.4.1 – Estructura de una solución factible	63
5.4.2 – Criterio <i>greedy</i> de selección de arcos ópticos candidatos	64

5.4.3– Selección de arcos ópticos eliminables	64
5.4.4– Selección de rutas alternativas de conexiones eléctricas	64
5.4.5 – Método de mejora	65
5.4.6 – Criterio de parada	65
5.4.7 – Pseudocódigo de la fase constructiva adaptada	65
5.4.7 – Pseudocódigo de la fase de mejora adaptada	66
5.5 – Definición de versiones	67
5.5.1 – <i>Greedy</i> Básico No Aleatorizado	67
5.5.2 – GRASP óptico	68
5.5.3 – GRASP eléctrico	68
5.5.4 – GRASP doble	69
Capítulo 6. Resolución de las versiones heurísticas	71
6.1 – Heurística Básica No Aleatorizada	71
6.1.1 – Diseño de pruebas	71
6.1.2 – Resultados en red de prueba	72
6.1.3 – Resultados en red europea	74
6.1.4 – Análisis de resultados	76
6.2 – GRASP óptico	76
6.2.1 – Diseño de pruebas	76
6.2.2 – Resultados en red de prueba	77
6.2.3 – Resultados en red europea	79
6.2.4 – Análisis de resultados	80
6.3 – GRASP eléctrico	81
6.3.1 – Diseño de pruebas	81
6.3.2 – Resultados en red de prueba	81
6.3.3 – Resultados en red europea	83
6.3.4 – Análisis de resultados	84
6.4 – GRASP doble	85
6.4.1 – Diseño de pruebas	85
6.4.2 – Resultados en red de prueba	86
6.4.3 – Resultados en red europea	87
6.4.4 – Análisis de resultados	89
Capítulo 7. Conclusiones y extensión del trabajo	91
7.1 – Criterios de valoración del método de resolución	91
7.2 – Comparación Método exacto vs. Métodos heurísticos	93
7.3 – Selección de la mejor versión heurística	94
7.4 – Integración de la meta-heurística en el sistema de gestión	95
7.5 – Extensión del proyecto	97
Referencias bibliográficas	101
Anexo: Implementación de la meta-heurística GRASP en Matlab	103
Consideraciones sobre la implementación	103
Implementación Algoritmo Central	103
Implementación Fase Constructiva	104
Implementación Fase de Mejora	108

Agradecimientos

Agradezco el trabajo, la ayuda y el apoyo de mis directores de proyecto: a Luis Velasco, que me ha abierto las puertas a un mundo que hasta hace unos meses desconocía, y que voy a seguir explorando en un futuro próximo; y a Francisco Javier Heredia, quien me invitó a entrar en esta carrera y que ahora se ha preocupado por procurarme un digno final.

Agradezco especialmente a Elena Fernández la virtud que ha tenido para introducir la Investigación Operativa dentro de mis motivaciones personales.

Agradecimientos y disculpas a toda mi familia y amigos, que han consentido ausencias, desplantes, retrasos y otras faltas, con paciencia y comprensión, por el bien de esta empresa.

Y como no, un agradecimiento y un “te quiero” para Montse, que ha sabido llevar con templanza y empatía los momentos más difíciles de este proceso, sin que trascendieran más allá de lo justo y necesario.

Capítulo 1

Introducción

1.1 – Motivación

Internet se está convirtiendo en el centro neurálgico de las comunicaciones globales. El tráfico cada vez es más heterogéneo (tráfico de voz, datos, multimedia), y cada vez tiene mayor volumen y mayor amplitud. Esta enorme demanda necesita una estructura de transporte de gran capacidad y gran alcance para poder ser satisfecha. Las redes de transporte de fibra óptica, implantadas desde hace unos años para las comunicaciones, cumplen estos requisitos.

No obstante, la capacidad de estas redes, aún siendo enorme, es finita. Y por otro lado, el tráfico crece exponencialmente con el tiempo. Debido a los inconvenientes que supone ampliar estas redes, la posibilidad de albergar más tráfico se sustenta en las posibilidades de optimización que presentan. Desde su puesta en funcionamiento, el diseño de redes, el enrutamiento óptimo de las conexiones entre clientes y la gestión integral de recursos, permiten un buen uso de los elementos ópticos disponibles.

Pero con eso no es suficiente, ya que el dinamismo del tráfico y su continua evolución, hacen que la planificación de la red, que puede estar basada en métodos de optimización, quede obsoleta con el tiempo. Es necesario el desarrollo de soluciones, (que pueden estar basadas en métodos de optimización, algoritmos de predicción, etc.) que mejoren continuamente los recursos de la red, reevaluando decisiones y modificando elementos en funcionamiento. Estos métodos tienen como intención lograr una adaptación temporal óptima a las circunstancias del instante en que se aplican. Su uso continuo puede permitir una mejor utilización de los recursos y, consecuentemente, una mayor capacidad disponible para el tráfico que esté por llegar.

1.2- Objetivos y estructura del PFC

La finalidad de este proyecto es plantear un problema de liberación de recursos ópticos en escenarios de tráfico dinámicos, cuyo objetivo sea reducir el número de interfaces ópticas utilizadas para transportar ese tráfico, de forma que se abarate el coste de la red, sin cortar ni perjudicar las conexiones establecidas entre los distintos puntos de la red, y sin afectar la cantidad de tráfico que puede transportar la red.

El Capítulo 2 introduce al lector en el contexto de las redes de comunicación ópticas, con aquellos conceptos imprescindibles para entender el trabajo propuesto.

El problema de liberación de recursos ópticos se explica y formula en el Capítulo 3. También se abordará la resolución exacta mediante el motor de resolución CPLEX y se analizarán los resultados obtenidos para distintos ejemplos de red e instancias de tráfico. Estos contenidos formarán parte del Capítulo 4.

Posteriormente, se formularán una serie de heurísticas basadas en la estructura elemental de la meta-heurística GRASP. Se analizarán los resultados obtenidos a partir de la resolución de los mismos ejemplos utilizados para el procedimiento exacto. La definición de las heurísticas será el contenido del Capítulo 5, mientras que las baterías de pruebas realizadas y sus resultados se presentan en el Capítulo 6.

Finalmente se elegirá el mejor de los procedimientos propuestos y se propondrá como método de implementación para mejora de recursos ópticos en gestión de redes ópticas reales. Este contenido junto con la extensión futura del trabajo se verá en el Capítulo 7.

Finalmente, y a modo de anexo, se podrán ver los códigos implementados de los métodos heurísticos.

1.3 – Consideraciones

Parte de este proyecto ha sido incluido en un artículo de investigación publicado en el Journal of Optical Networks, de la Optical Society of America [1]. El autor de este proyecto figura como co-autor de dicho artículo.

Capítulo 2

Introducción a las redes de transporte

La incursión de la fibra óptica en el mundo de las telecomunicaciones abrió una nueva puerta al desarrollo integral de la transmisión de información a escala global. El aumento en el uso y la demanda de Internet y de las nuevas tecnologías está provocando un incremento exponencial del tráfico de datos. Esta tecnología se está convirtiendo en la base sobre la cual puede sustentarse este incremento de las comunicaciones a nivel mundial, que experimentamos día a día.

Este capítulo pretende explicar los conceptos necesarios sobre redes de transporte ópticas, proporcionando los conocimientos que permitan entender en profundidad el problema planteado a lo largo del presente documento [2].

2.1 – Panorama actual

En el momento en que empezaron a implantarse las redes de fibra óptica para el transporte principalmente de tráfico de voz, las predicciones más optimistas se quedaron lejos de lo que en un futuro cercano podría representar el volumen total a transportar. El enorme auge del tráfico de Internet está provocando una utilización cada vez mayor de

los recursos ópticos, hasta hace poco tiempo infrautilizados. En pocos años, las redes ópticas, tal y como se entienden y gestionan actualmente, estarán saturadas. Internet está sirviendo como herramienta para transportar tráfico de muchos tipos (datos, voz, video y en general multimedia) y de gran volumen, lo que hace que las redes tradicionales ya anden, a día de hoy, altamente ocupadas.

Para poder satisfacer en un futuro este tráfico creciente a través de las redes ópticas establecidas, existen dos estrategias posibles:

- 1) Desplegar más fibra óptica: es decir, ampliar los kilómetros de cable para aumentar la capacidad de la red. Esta opción es altamente costosa, por la cantidad de recursos e infraestructuras que hay que movilizar y consumir para este efecto. Por otra parte, el tiempo de despliegue de nuevas infraestructuras es largo (varios meses, incluso años). En el caso de redes de nueva instalación, tiene sentido dotarlas de mayor capacidad en relación a las que se construyeron hace años. Pero en redes ya establecidas, resulta muy caro tener que ampliar los recursos físicos de éstas.
- 2) Aprovechar al máximo los recursos existentes: una fibra óptica estándar tiene capacidad teórica para transportar 10 Tbit/s (terabytes por segundo). Esta capacidad nunca se llega a utilizar completamente, así que conviene desarrollar la posibilidad de aprovechar al máximo los recursos existentes antes de estar en la obligación de ampliar los elementos físicos. En este campo, el desarrollo de nuevas arquitecturas de gestión y la aplicación de procedimientos de investigación operativa tienen mucho que decir.

2.2 – Definición de conceptos básicos

2.2.1 – Recursos ópticos

Una red óptica está formada básicamente por dos tipos de elementos:

- 1) **Cable óptico:** las fibras ópticas se unen formando haces que a su vez forman los cables que conectan los puntos de una red entre sí.

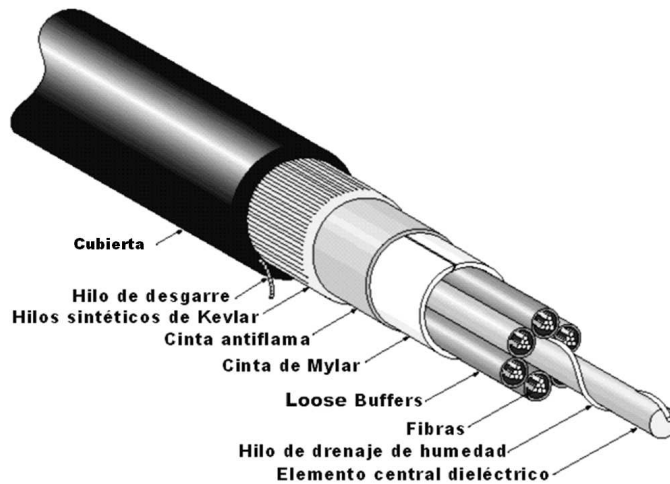


Fig. 2.1 – Cable de fibra óptica (fuente: <http://www.yio.com.ar>)

2) **Interfaz óptica:** es todo aquel elemento necesario para formar la red de comunicación. Los más comunes son:

- a. **Emisores de luz:** son dispositivos LED o láser que generan los impulsos lumínicos que transportan la información.
- b. **Convertidores E/O y O/E:** permiten pasar una señal de datos formada por impulsos eléctricos a una señal óptica y viceversa.
- c. **Amplificadores de señal:** aunque la señal óptica tiene una pérdida de intensidad menor que otras señales, debido a la longitud habitual de varios centenares de kilómetros de los cables, es necesario amplificar la señal en puntos intermedios.
- d. **Conmutadores ópticos:** son aquellos elementos que permiten conectar entre sí diversas terminaciones para formar los nodos de la red. Redirigen y distribuyen la información que llega a los nodos ópticos. Pueden ser configurados de forma que se puedan realizar y eliminar conexiones ópticas mediante simples comandos.

2.2.2 – Multiplexado de datos

El multiplexado de datos se puede definir como el envío y circulación de diferentes paquetes de datos o conexiones a través de un mismo enlace, permitiendo la transmisión simultánea de diferentes tráficos. Esta característica permite aumentar la capacidad sin aumentar la presencia de recursos físicos. Existen varios tipos de multiplexado, aunque por su relevancia podemos destacar:

- 1) **Multiplexado por división en tiempo (TDM):** diferentes tráficos se satisfacen en pequeños intervalos de tiempo correlativos. Durante un instante de tiempo concreto solo circula un único tráfico, y estos se van alternado en breves intervalos de tiempo hasta ser totalmente satisfechos.

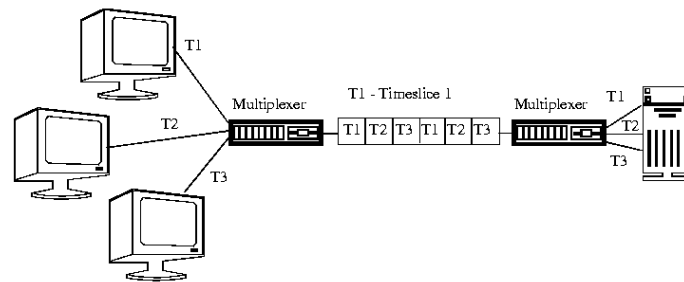


Fig. 2.2 – TDM (fuente: <http://www.rigacci.org>)

- 2) **Multiplexado por división en longitud de onda (WDM):** este tipo de multiplexación se aplica específicamente a las fibras ópticas. Cada longitud de onda en la que puede separarse el haz de luz, puede transportar una señal concreta. Eso quiere decir que podemos enviar distintas señales a la vez, aprovechando todo el ancho de banda que permite el multiplexado en longitud de onda. Para multiplexar/demultiplexar varias señales ópticas es necesario juntarles/separarlas en/del haz lumínico, utilizando dispositivos especiales denominados prismas ópticos, como se observa en la figura 2.3.

2.2.3 – Arquitecturas multicapa

El tráfico generado (datos, voz, Internet) por clientes individuales (empresas, particulares) tiene naturaleza eléctrica y requiere un ancho de banda cada vez mayor. Para hacernos una idea, en estos momentos estamos utilizando conexiones a Internet por ADSL entorno a los 10 Mbit/s. Por otra parte, las redes de fibra óptica permiten transportar tráficos de cliente a través de la transmisión de impulsos lumínicos, ofreciendo un ancho de banda en el entorno de los 10 Tbit/s. Como puede observarse es necesario organizar las redes de transporte en un conjunto de capas, de forma que los tráficos de cliente se agreguen antes de ser transportados por las redes ópticas, optimizando el uso del ancho de banda.

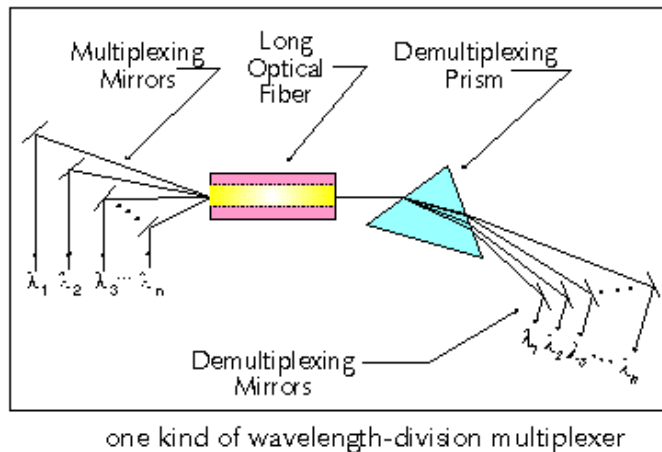


Fig. 2.3 – WDM (fuente: <http://www.its.bldrdoc.gov>)

Un ejemplo muy común de red multicapa se presenta en la figura 2.4. Aquí podemos observar una arquitectura de red formada por 4 capas: capa física de fibra, una capa de multiplexación óptica, una capa de agregación eléctrica-óptica y la capa de enrutamiento IP.

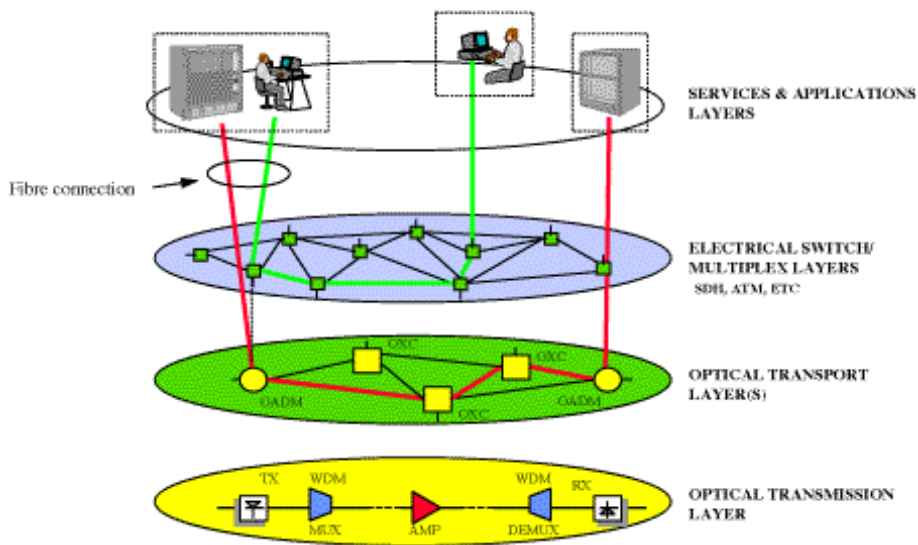


Fig. 2.4 – Red multicapa (fuente: <http://cordis.europa.eu>)

2.2.4 – Arquitectura de gestión de las redes de transporte

Como hemos comentado anteriormente, la tecnología óptica permite establecer/liberar conexiones utilizando un componente clave: el conmutador óptico. Estos dispositivos se integran en los equipos de comunicaciones ópticas. La más simple de las redes ópticas puede incluir cientos (o miles) de conmutadores ópticos. Por este motivo es

imprescindible disponer de un sistema de gestión de la red (NMS) que facilite el mantenimiento y operación de las redes ópticas.

El NMS dispone de toda la información sobre la red y su estado en cada momento; por ejemplo, permite establecer de una forma centralizada conexiones punto a punto entre dos localizaciones de la red, encadenando un conjunto de actuaciones sobre conmutadores ópticos. La figura 2.5 muestra un único sistema de gestión de red que gestiona de forma centralizada dos capas de la red multicapa anterior.

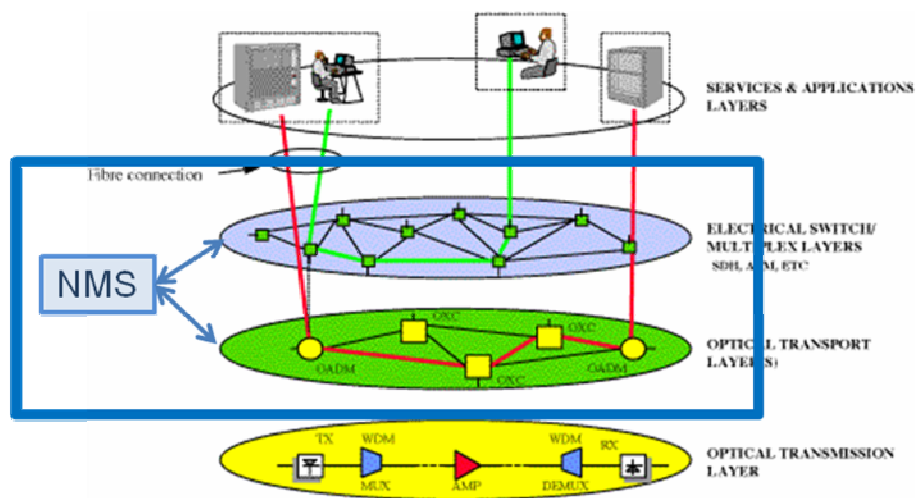


Fig. 2.5 – Gestión centralizada de la red multicapa

El método de optimización que se estudiará en este PFC se ubica en el sistema de gestión de la red. El optimizador actuará sobre la capa de agregación eléctrica, sobre la que modificará el tráfico establecido sin necesidad de actuar sobre la capa óptica. La figura 2.6 presenta la nomenclatura utilizada en este PFC para referirse a cada uno de los componentes de la arquitectura de gestión.

2.2.5 – Gestión de conexiones eléctricas

Cuando una nueva petición de conexión eléctrica llega al NMS, este se encarga de encontrar una ruta en la capa de agregación de clientes para poder dar servicio a esa conexión. Al realizar esta acción, pueden suceder dos situaciones:

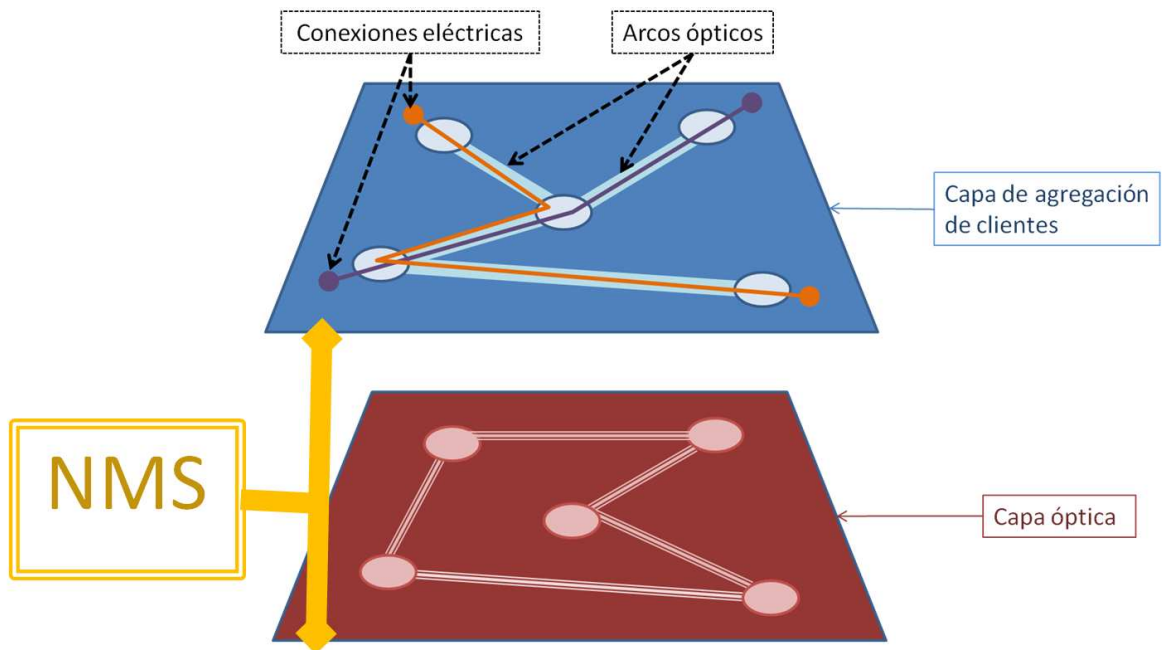


Fig. 2.6 – Nomenclatura de red y gestión

- A) Existe algún camino a través de los arcos ópticos con suficiente capacidad como para servir la conexión entre clientes. En este caso, la conexión se establece a través de la ruta de menos coste posible. Para el cálculo de la ruta se utilizan algoritmos basados en Dijkstra [3].
- B) No existe ningún camino a través de los arcos ópticos con suficiente capacidad o conectividad para satisfacer la conexión. En ese caso, puede suceder uno de los siguientes casos.
1. Existen recursos libres en la capa óptica suficientes para crear nuevos arcos ópticos que permitan satisfacer la conexión: en ese caso, se generan las nuevas estructuras ópticas en la capa de agregación eléctrica (a partir de los recursos ópticos disponibles), siguiendo una cierta función de coste asociada a los arcos ópticos.
 2. No existen recursos ópticos libres para satisfacer la conexión: en ese caso, se rechaza la petición de establecimiento de conexión.

Las figuras 2.7 y 2.8 ilustran los casos A) y B1) respectivamente.

Por otra parte, el sistema de gestión también se encarga de liberar la capacidad óptica utilizada por una conexión eléctrica cuando ésta ha finalizado su tiempo de establecimiento. En este caso, pueden ocurrir dos situaciones:

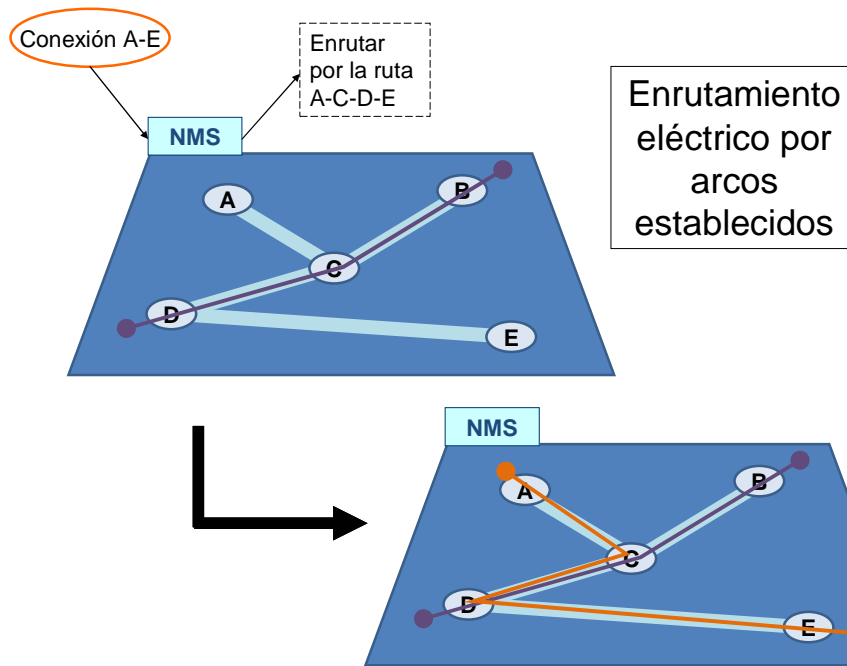


Fig 2.7 – Enrutamiento eléctrico

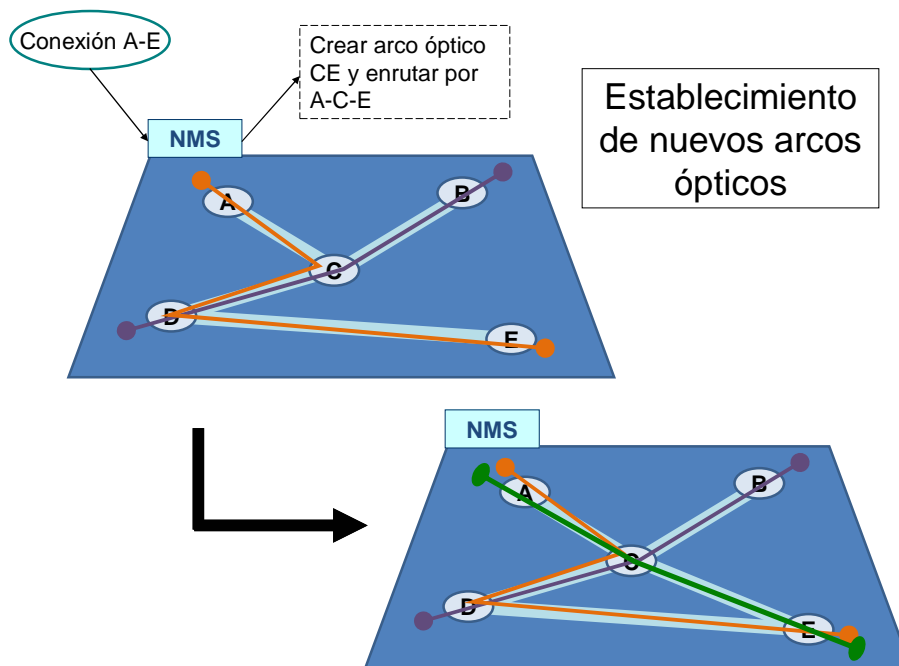


Fig 2.8 – Generación arcos ópticos

- C) La finalización del establecimiento de la conexión ha dejado sin ocupación algún o algunos arcos ópticos: en ese caso, los arcos ópticos sin ocupación se eliminan de la red de agregación eléctrica, liberando así recursos ópticos en la capa óptica.
- D) La eliminación de la conexión eléctrica no deja ningún arco sin ocupación: en ese caso, no se realiza ninguna acción sobre los arcos ópticos.

La figura 2.9 ilustra el caso C).

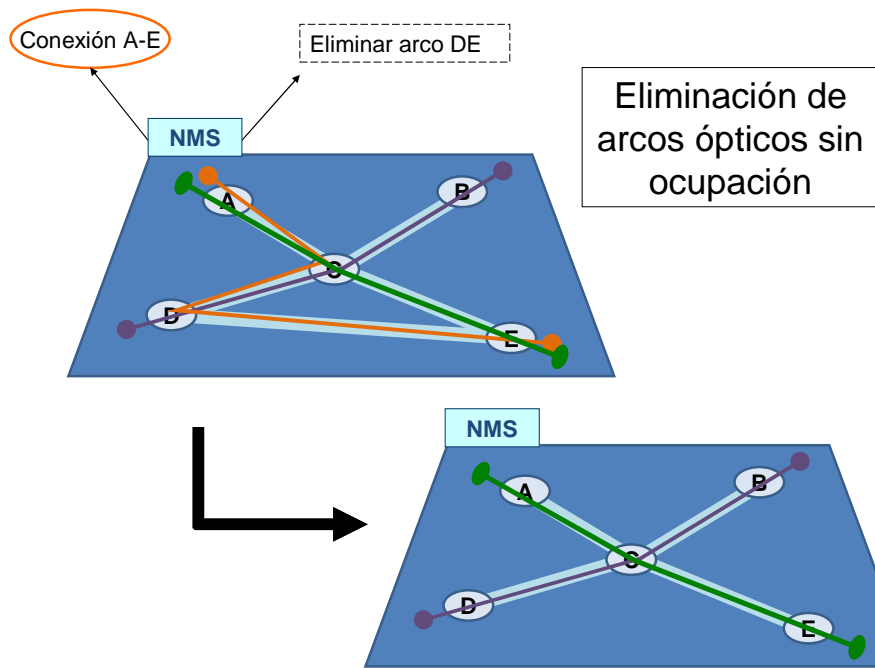


Fig 2.9 – Eliminación arcos ópticos

Capítulo 3

Problema de liberación de recursos ópticos

Como hemos visto en el capítulo anterior, el desarrollo tecnológico en el campo de las redes de comunicación de fibra óptica está permitiendo transportar un gran volumen de información heterogénea, de una manera más eficiente, segura y veloz, entre puntos que pueden estar distanciados por miles de kilómetros. Esta base tecnológica permite también una evolución del tráfico transportado, que cada vez es más intenso y heterogéneo. El futuro de las comunicaciones pasa por un óptimo transporte de datos entre los diferentes nodos de las redes

En este sentido, el diseño de redes, el enrutamiento de conexiones entre nodos, la asignación de recursos ópticos, entre otros procedimientos, permiten utilizar el entorno óptico de forma que se puedan optimizar los recursos utilizados en cada caso, sin renunciar a la cantidad y calidad del servicio. No obstante, el tránsito de información es un hecho dinámico, que constantemente se modifica, lo cual hace que sea necesario aplicar procedimientos adaptativos en el tiempo, que permitan reoptimizar estructuras y decisiones creadas en su momento con la mejor intención operativa.

En este capítulo presentaremos una situación en la que podemos aplicar un proceso de mejora que permita, dado un escenario e instante de tiempo concreto, modificar la

estructura de arcos ópticos que transporta un cierto tráfico, para poder reducir la cantidad de recursos ópticos utilizados. Esto nos debe permitir, sin pérdida de factibilidad, obtener una nueva situación en la que se pueda prestar el mismo servicio con un coste total menor.

Estos procesos de mejora están a la orden del día, y forman parte de los temas de investigación que actualmente se llevan a cabo en el mundo de las comunicaciones ópticas. El gran crecimiento de las posibilidades de conexión y transmisión de información remota puede llevar a una sobrecarga en las redes y a una probabilidad de rechazar conexiones por falta de recursos disponibles cada vez mayor. La posibilidad de ir adaptando la red a las necesidades y características del tráfico presente puede permitir compactarlo y transportarlo utilizando menos recursos y pudiendo así tener espacio físico para albergar conexiones futuras.

A continuación, describiremos el problema y la situación en la que se desarrolla. Luego plantearemos una modelización matemática que lo defina y analizaremos al detalle la solución exacta de diferentes escenarios de tráfico a partir de un motor de solución de problemas de programación lineal.

3.1 – Planteamiento del problema

3.1.1 – Utilización subóptima de recursos ópticos

En el Capítulo 2, definimos los arcos ópticos a nivel de la capa de agregación eléctrica y vimos de qué manera se generaban y se eliminaban. Por otra parte, vimos como se enrutaban a través de ellos las conexiones eléctricas que albergaban la información transportada. Como cada decisión se toma teniendo en cuenta la situación del momento, es fácil comprobar que la utilización de recursos en un instante concreto no va a ser la mejor posible. El siguiente ejemplo lo ejemplifica claramente:

Supongamos en un cierto instante de tiempo tI , la siguiente red de arcos ópticos y de conexiones eléctricas activas a nivel de la Capa de Control:

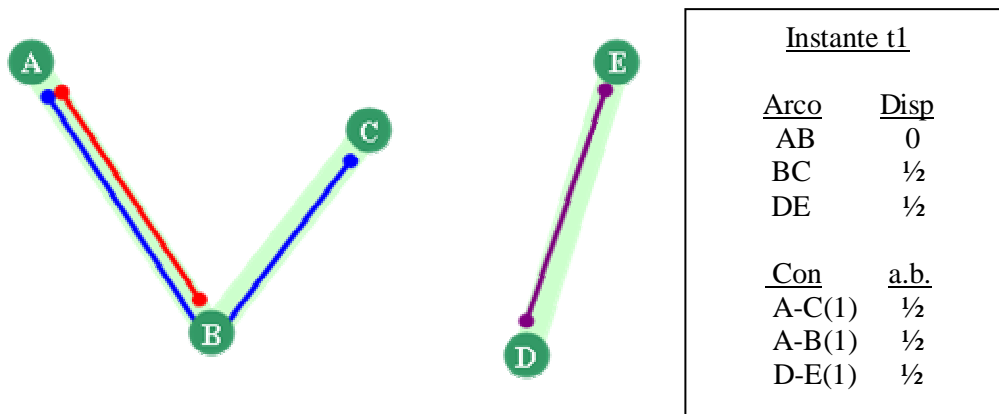


Fig 3.1- Red en el instante t1

En un instante $t2$, que cumple que $t2 > t1$, llega una petición de conexión entre los nodos A y C, y que requiere un ancho de banda de $1/2$. Pero resulta imposible enrutar la conexión por los arcos ópticos existentes (a través de los arcos AB y BC), ya que el arco AB no presenta ancho de banda disponible. Supondremos que la red óptica de base tiene suficiente capacidad como para crear un nuevo arco óptico directo entre A y C. Por lo tanto, el procedimiento a seguir será generar el arco óptico AC y enrutar directamente a través de él la conexión requerida:

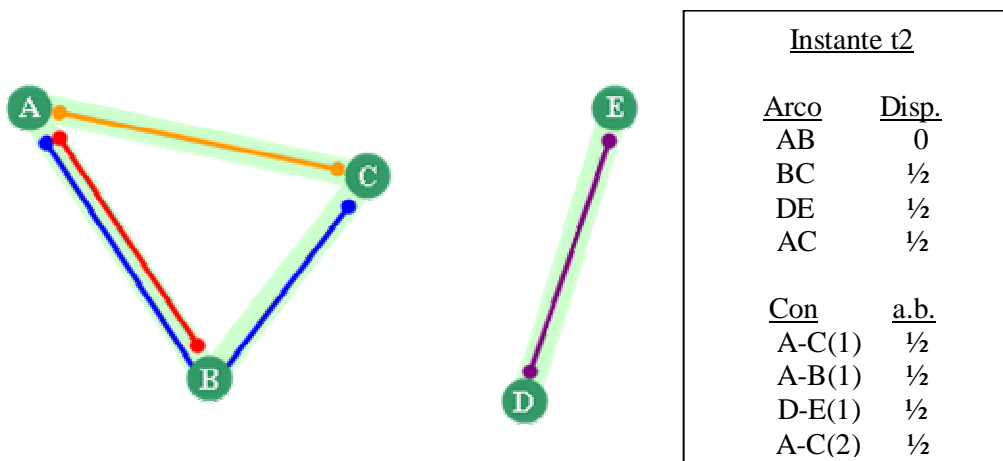


Fig 3.2 - Red en el instante t2

Un tiempo más tarde, en el instante $t3$ vence la conexión de color rojo entre los nodos A y B. Así que nos quedan dos conexiones entre A y C que están utilizando dos rutas diferentes y consumiendo más recurso óptico del que necesitan. Se puede ver como, si modificáramos la conexión azul y la enrutáramos a través del arco óptico AC, se quedarían libres de ocupación los arcos AB y BC, con lo que podríamos liberar los recursos ópticos que estuvieran utilizando ambos arcos.

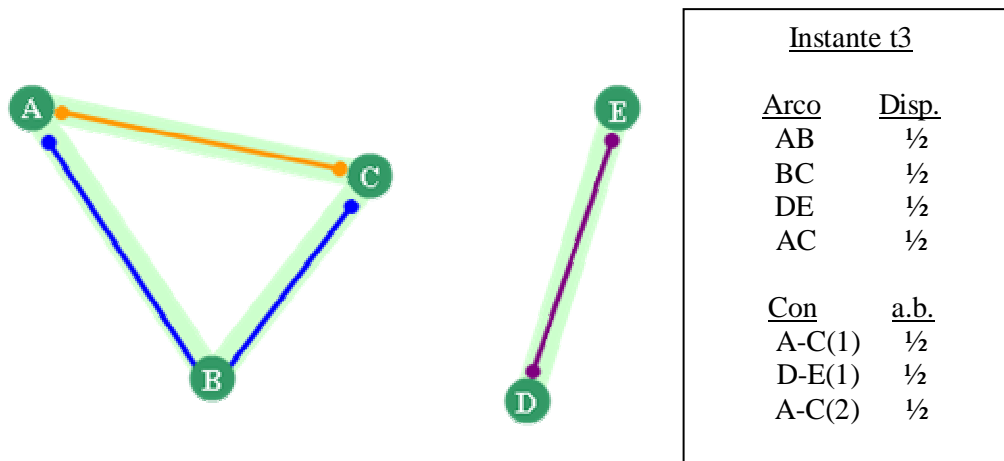


Fig. 3.3 - Capa Control en el instante t3

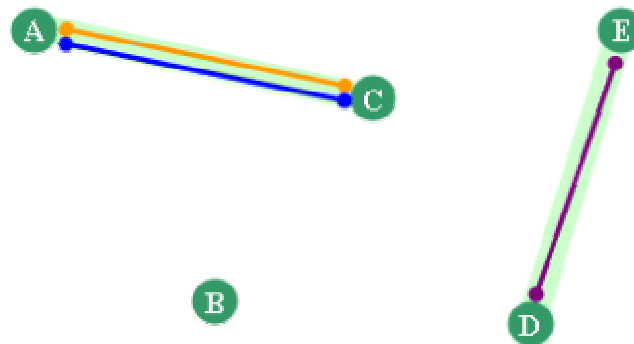


Fig. 3.4 - Situación óptima en el instante t3

Está claro pues que la utilización de recursos ópticos por parte de los arcos y el enrutamiento de las conexiones eléctricas a través de ellos, raramente se realizarán de forma óptima, ya que la continua aparición y desaparición de conexiones hace variar cualquier condición inicial tenida en cuenta. No obstante, la aplicación de un procedimiento que nos permita liberar arcos ópticos innecesarios, modificando las rutas de algunas conexiones eléctricas, nos dará una mejor y más eficaz distribución de recursos ópticos activos.

Dicho esto, definiremos el problema que queremos atacar en una serie de puntos clave.

3.1.2 – Objetivo del problema

La finalidad del planteamiento es conseguir, dada una red óptica, un instante de tiempo y distribución de tráfico concretos, liberar la mayor cantidad de recursos ópticos

utilizados, eliminando los arcos ópticos que no sean necesarios para transportar el tráfico dado. Eso conlleva necesariamente modificaciones en las rutas de algunas de las conexiones eléctricas circulantes, para poder mantener la factibilidad del tráfico.

La minimización de coste total se realizará tratando la siguiente ordenación de criterios de optimización:

- 1) Se buscará la máxima reducción posible de recursos ópticos, tratando de buscar la red derivada que consuma la menor cantidad posible de ellos.
- 2) Para soluciones alternativas que tengan el mismo coste de recursos ópticos total, se preferirá la versión que consiga un coste total de enrutamiento eléctrico menor.
- 3) Para soluciones alternativas con el mismo coste respecto a ambos criterios anteriores, se preferirá la que produzca una ocupación total menor en la red.

3.1.3 – Consideraciones sobre el conjunto de arcos ópticos

Un arco óptico tiene un coste asociado a los recursos ópticos que ocupa en la capa óptica. Para cada arco existirá en la capa óptica una serie de dispositivos y una longitud de cable ocupados. Por otra parte, un arco óptico que una dos nodos concretos de la red, puede tener diferente coste en función del conjunto de recursos que utilice. En la siguiente tienen estructuras bien diferentes en el Plano Óptico.

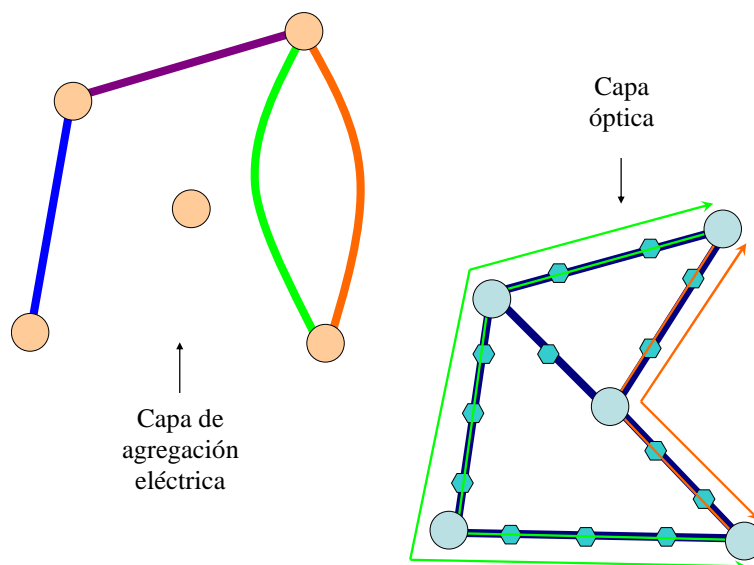


Fig. 3.5 – Arcos ópticos paralelos con coste diferente

Por ello, cabe la posibilidad de modificar esa estructura de base de los arcos ópticos para tratar de reducir su coste, sin necesidad de eliminar arcos en la red de agregación de conexiones. Es decir, nos podríamos plantear la posibilidad de añadir nuevos arcos o modificar la estructura de algunos existentes, si con ello conseguimos redistribuir el tráfico de forma factible y reducir el coste global.

No obstante, vamos a desestimar esa posibilidad. Fijaremos la restricción de escenario de no crear nuevos arcos ópticos con la finalidad de poder reducir el coste total del problema. Esta restricción la asumimos para evitar de cualquier manera que se produzca una posible interrupción de servicio. Al permitir modificar la estructura física de los arcos, tendríamos que cortar momentáneamente las conexiones eléctricas afectadas hasta que no estuviera construida la nueva estructura óptica de base. Por otra parte, la generación óptima de arcos ópticos necesitaría probablemente la liberación y reubicación de recursos ópticos en uso, con lo cual las acciones ejercidas sobre ellos obligarían en algún momento a cortar el servicio de algunas conexiones, situación que no queremos que suceda.

Al tratarse de un problema de reoptimización de un sistema de comunicaciones en funcionamiento, y no de un problema de diseño óptimo de redes, facilitaremos el proceso si aplicamos la restricción de no generar nuevos arcos, ni modificar la estructura óptica básica de éstos. Esta condición evita interrupciones de servicio, hecho que permite aplicar la optimización tantas veces como se desee sin provocar desatenciones en el servicio prestado a los clientes. La optimización de recursos ópticos no vendrá dada por la reubicación de éstos en la red, sino que se producirá al desactivar los que no sean estrictamente necesarios para conducir el tráfico eléctrico del instante en que se aplica.

3.1.4 – Consideraciones sobre los cambios de ruta de las conexiones eléctricas

Cada arco óptico que se elimina conlleva una modificación en la ruta de las conexiones eléctricas que pasan por él. Para poder eliminar un arco, las conexiones implicadas deben tener al menos una ruta alternativa disponible. Si una conexión no tiene ruta alternativa con capacidad libre, muy probablemente existirá una redistribución del tráfico que genere capacidad libre, para que sean factibles algunas de las rutas

alternativas. Por lo tanto, plantearse la posibilidad de modificar las rutas de las conexiones existentes para lograr una mejor distribución y, en consecuencia, una mayor liberación de recursos ópticos final, nos conduciría a una mayor reducción del coste de la red. La siguiente figura muestra una alternativa de reubicación de conexiones eléctricas que nos permite liberar dos arcos ópticos que estaban ocupados:

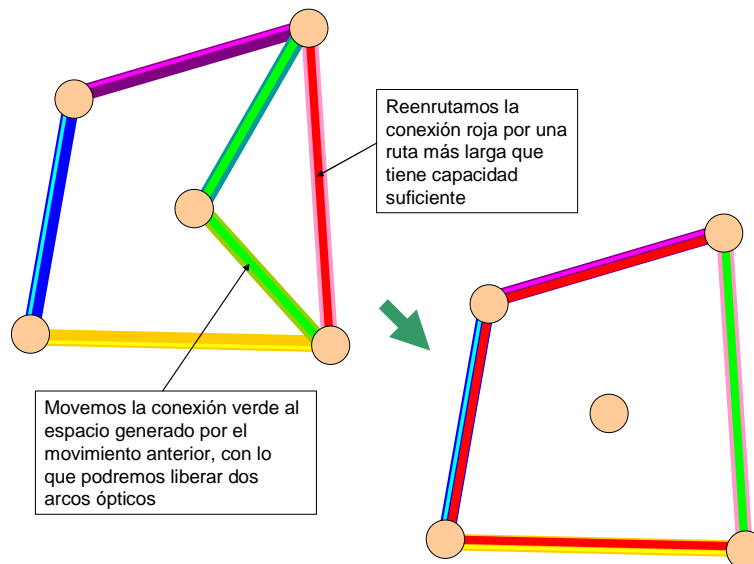


fig 3.6 – Ajuste de rutas para liberar espacio

Enfocando el problema de esta forma, la solución óptima radicaría en compactar al máximo el tráfico por el mínimo de arcos ópticos posible (a poder ser, los de menor coste) y liberar los arcos ópticos no utilizados.

No obstante, el conjunto de reubicaciones y posibles de cambios de ruta de un problema de dimensión real puede llegar a ser tan grande que, tener en cuenta todas ellas para encontrar la mejor disposición, puede resultar una tarea demasiado costosa computacionalmente como para llevarse a cabo en un tiempo asumible. La optimización depende en gran medida de la cantidad de rutas posibles que ofrezcamos como datos iniciales, y manejar todas las opciones en instancias de redes reales muchas veces carece de sentido. Por ello, fijaremos dos restricciones de escenario para reducir el número de posibilidades considerables de rutas eléctricas, con la intención de reducir el campo de exploración de los procesos de optimización y mejorar el tiempo de implementación, sin renunciar a la obtención de una solución de buena calidad.

3.1.4.1 – Limitación sobre el conjunto de conexiones eléctricas reubicadas

La condición que aplicaremos será la siguiente: limitaremos los cambios de ruta a sólo aquellas conexiones que circulen originalmente por arcos ópticos que serán eliminados en la solución final. Si una conexión pasa por un conjunto de arcos que se mantienen activos después de la optimización, ésta mantendrá exactamente la misma ruta de arcos ópticos que seguía antes de la optimización.

En el caso ejemplificado en la figura anterior, la conexión de color rojo ha sido modificada de ruta de manera que viola esta condición, porque la ruta original circula a través de un solo arco óptico que se mantiene tras la mejora. Para que se cumpliera esta restricción de movimiento de rutas y poder liberar los mismos arcos ópticos, deberíamos haber movido la conexión verde sin tocar ninguna otra conexión, tal y como queda ilustrado en la figura 7.

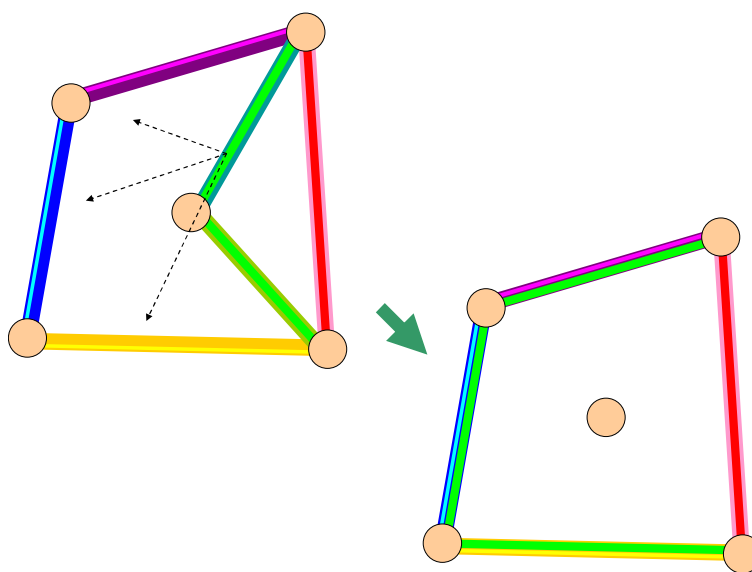


Fig. 3.7 – Reubicación de conexión eléctrica permitida

Sencillamente estamos permitiendo que solo se muevan aquellas conexiones eléctricas cuya ruta original se vea afectada por un corte provocado por la eliminación de algún arco óptico.

3.1.4.2 – Limitación sobre el máximo de rutas eléctricas consideradas

En una situación de tráfico de media ocupación sobre una red real, una conexión eléctrica entre dos puntos mínimamente distanciados (cuya ruta más corta utilice 4 o 5

arcos ópticos) puede llegar a tener cientos, hasta miles de rutas posibles. Plantear todas esas rutas para todas las conexiones presentes, nos genera una cantidad de información que en un porcentaje muy elevado no vamos a utilizar.

Generalmente, la asignación óptima de rutas para la generación de un escenario de tráfico nuevo, acostumbra a dar como ruta seleccionada para cada conexión una opción que suele estar entre las 10 primeras en la lista ordenada por coste ascendente. Es decir, al asignar el ruteo de conexiones que da un mayor beneficio global, la conexión con la peor de las rutas respecto a sus posibilidades, circula por un camino indexado en una de las primeras 10 posiciones de la lista ordenada de rutas posibles [3]. Si tenemos en cuenta el párrafo anterior, hasta un 90% de la lista guarda rutas que jamás serán utilizadas para una solución, aunque sí que serán exploradas y evaluadas.

En nuestro problema de reasignación dado un tráfico activo, probablemente las rutas seleccionadas tras la optimización ocupen posiciones algo más avanzadas en la lista ordenada por coste. A pesar de ello, encontrar rutas reasignadas a partir de la posición 20 o 30 apenas suele ocurrir. Así que, en este caso, también tenemos mucha información que nos ralentiza la obtención de la solución óptima, sin aportar mejora posterior.

Existen métodos como la generación de columnas, que permiten atacar resoluciones exactas de problema de gran dimensión sin necesidad de definir a priori todas las opciones posibles. En este proyecto, aplicaremos un criterio de reducción fijo de la cantidad de rutas consideradas.

En el terreno de las comunicaciones ópticas y en la aplicación de problemas de diseño y enrutamiento de conexiones, se suele utilizar una limitación de rutas posibles de hasta 100 por conexión eléctrica [2]. Este valor permite fijar un máximo tamaño del problema por lo que respecta a esta dimensión, con lo cual el coste computacional de resolución tanto de problemas exactos como de heurísticas se reduce drásticamente. Además, este valor es lo suficientemente grande como para asegurarnos en un porcentaje muy cercano al 100% de los casos que la solución óptima del problema con todas las rutas posibles estará contenida en la versión limitada a las 100 mejores rutas.

3.2 – Formulación matemática

A continuación vamos detallar la formulación matemática del problema de liberación de recursos ópticos. Plantearemos una modelización de los datos y variables de la situación que queremos abordar. Luego describiremos las restricciones que nos definirán el conjunto de soluciones factibles evaluables, y finalmente detallaremos la función objetivo de evaluación de esas soluciones. Las características más importantes de la formulación presentada son:

- Formulación orientada a comparación de estados: la estructura de los datos y las variables permiten compararse entre sí para poder determinar y detectar qué cambios se producen con la optimización, esto es, que arcos ópticos se eliminan y que conexiones eléctricas se reenrutan. La estructura de variables es sinónima a la estructura de datos, con lo cual se puede realizar una comparación directa y evaluar los cambios. La asignación de los valores de las variables se puede entender como la construcción de un nuevo sistema y los valores de los datos sirven para restringir que se puede modificar y que no.
- Formulación de programación lineal: las restricciones y función objetivo del problema tienen la forma de ecuaciones lineales con variables binarias y enteras. Esto permite poder aplicar motores de solución potentes como CPLEX, que solo trabajan con problemas de esta forma. Para ello, se han incorporado variables y restricciones adicionales, para poder deshacer productos de variables y funciones no lineales de evaluación en valor absoluto.

3.2.1 – Estructura de los datos

El problema de liberación de interfaces ópticas se aplica en un instante de tiempo determinado sobre una red y tráfico establecidos. La modelización de los datos permite obtener toda la información sobre el estado del sistema en el momento de aplicación de la optimización, así como tener información suficiente para poder realizar los cambios necesarios y realizar así la mejora pretendida.

Inicialmente, debemos definir los siguientes conjuntos:

- **E**: Conjunto de conexiones eléctricas que forman una instancia de tráfico = $\{1, \dots, i, \dots, I\}$.
- **R_(e)**: Conjunto de posibles rutas que darían servicio a la conexión $e = \{1, \dots, j, \dots, J\}$.
- **S**: Conjunto de arcos ópticos activos en la red = $\{1, \dots, k, \dots, K\}$.

Podemos ver como habrá tantos conjuntos $R_{(e)}$ como conexiones eléctricas haya. Definiremos un valor J_{\max} como la máxima cardinalidad de todos estos conjuntos. Esto es necesario para poder definir matrices de datos en cualquier $R_{(e)}$ y de dimensión fija, independientemente del número de rutas del conjunto. Dentro de estos conjuntos están todos los elementos que conforman nuestro sistema de red. Las siguientes estructuras guardan información sobre el tráfico circulante, así como de la relación entre conexiones, rutas y arcos:

- **N_{ij}**: matriz binaria de datos, de dimensión $I \times J_{\max}$, cuyas celdas valdrán 1 si y solo si la conexión eléctrica i utiliza la ruta indexada en j .
- **Q_{ij}^k**: matriz binaria de datos, de dimensión $I \times J_{\max} \times K$, cuyas celdas valdrán 1 si y solo si el arco óptico k forma parte de la ruta j perteneciente a la conexión i .

Finalmente, también debemos guardar información sobre algunas características de los arcos ópticos y de las conexiones eléctricas, y que no variarán en el tiempo:

- **M_k**: Capacidad del arco óptico k para transportar flujo eléctrico (en unidades de ancho de banda).
- **W_i**: Ancho de banda de la conexión eléctrica i .

3.2.2 – Variables de decisión

El planteamiento de las variables de decisión es el de modelar un estado paralelo al estado inicial del problema, que tenga una mejor valoración de coste y que cumpla las

restricciones. Veremos ahora que variables necesitamos tener en cuenta para obtener esta nueva versión del escenario:

- η_{ij} : Matriz binaria de dimensión $I \times J_{max}$, equivalente a la estructura de datos \mathbf{N}_{ij} , cuyas celdas valdrán 1 si y solo si la conexión eléctrica i utiliza la ruta j tras la optimización.
- ζ_k : Vector binario de dimensión K , con valor 1 si y solo si el arco óptico k permanece activo tras la optimización.

La optimización produce unos cambios en la estructura de los arcos activos (registrado en la variable ζ_k). Estos cambios conducen a modificaciones en las rutas de las conexiones eléctricas. Como hemos definido anteriormente, no todas las conexiones eléctricas podrán ser modificadas (sólo aquellas que se vean afectadas por un corte óptico). Es necesario crear una variable que recoja que conexiones eléctricas pueden y deben modificar su ruta para mantener la factibilidad:

- ρ_i : Vector binario de dimensión I , cuyos elementos valdrán 1 si y solo si la conexión eléctrica i debe cambiar de ruta respecto de la situación inicial.

Para una posterior factibilidad del problema y evaluación de la función objetivo, debemos determinar una variable entera que recoja la cantidad de ancho de banda circulante por los arcos ópticos tras la reasignación:

- δ_k : Vector de enteros, cuyos elementos registrarán las unidades de ancho de banda ocupadas en el arco óptico k .

3.2.3 – Restricciones del problema

Tomando las estructuras y variables definidas en el punto anterior, vamos a definir ahora las restricciones del problema, que tienen como misión cumplir las consideraciones, limitaciones y características del problema de liberación de recursos ópticos, tal y como se ha planteado en secciones anteriores.

3.2.3.1 – Restricciones sobre los arcos ópticos

Existe una única restricción que actúa exclusivamente sobre los arcos ópticos, que debe permitir la *factibilidad de la ocupación de los arcos*. Esta restricción debe permitir que la ocupación sea nula en los arcos eliminados y menor que la capacidad pero existente en los arcos activos. La expresión es la siguiente:

$$\zeta_k \leq \delta_k \leq (M_k \times \zeta_k), \quad \forall k \in S$$

3.2.3.2 – Restricciones sobre las conexiones eléctricas

Estas restricciones deben permitir que se cumplan las condiciones sobre las conexiones eléctricas y sus cambios de ruta. La primera y más elemental es que cada conexión eléctrica tenga una única ruta asignada:

$$\sum_{j \in R(i)} \eta_{ij} = 1, \quad \forall i \in E$$

La comparación de la variable de estado de las asignaciones de conexiones a rutas con la situación original, deberá dar resultados diferentes para las conexiones eléctricas que deban ser modificadas. La siguiente restricción relaciona la variable de estado de la red optimizada con la variable que fija que conexiones eléctricas deben moverse, de tal manera que no se permiten reasignaciones de rutas en conexiones que tienen que permanecer sin cambios:

$$\sum_{j \in R(i)} ((1 - N_{ij}) \times \eta_{ij}) = \rho_i, \quad \forall i \in E$$

Estas dos últimas restricciones permiten cambios en las rutas de sólo aquellas conexiones eléctricas que tengan “permiso” para ello.

3.2.3.3 – Restricciones de unión de los niveles óptico y eléctrico

Estas restricciones deben asegurar un flujo óptimo por los arcos ópticos activos y una reasignación factible de rutas. En primer lugar, debemos asegurar que la nueva

reasignación de rutas puede ser satisfecha en cuanto a capacidad disponible de los arcos se refiere:

$$\sum_{i \in E} \sum_{j \in R(i)} (W_i \times Q_{ij}^k \times \eta_{ij}) \leq \delta_k, \quad \forall k \in S$$

Finalmente, deben cumplirse una condición de limitación de conexiones modificadas. Anteriormente, se asumió que se moverían sólo las conexiones que estuvieran obligadas a hacerlo por la eliminación de un arco óptico utilizado. Esta condición la aplicaremos desglosada en dos restricciones separadas.

Por una parte, fijaremos que aquellas conexiones cuyos arcos ópticos no sufran ninguna desactivación, no puedan ser cambiadas de ruta:

$$\sum_{k \in S} \left((1 - \zeta_k) \times \sum_{j \in R(i)} (N_{ij} \times Q_{ij}^k) \right) - \rho_i \geq 0, \quad \forall i \in E$$

Esta restricción cumple que si una conexión eléctrica tiene toda la ruta original por arcos que no van a ser eliminados, esa conexión debe mantener la ruta original ($\rho_i = 0$).

En segundo lugar, obligamos que las conexiones afectadas con al menos un corte óptico sean reconducidas por un camino factible:

$$\rho_i + \frac{\sum_{k \in S} \left(\zeta_k \times \sum_{j \in R(i)} (N_{ij} \times Q_{ij}^k) \right)}{\sum_{k \in S} \sum_{j \in R(i)} (N_{ij} \times Q_{ij}^k)} \geq 1, \quad \forall i \in E$$

En esta restricción, si una conexión eléctrica presenta al menos un corte óptico, el cociente de la expresión será inferior a 1 y, por tanto, el valor de ρ_i deberá ser 1.

3.2.3.4 – Condiciones sobre las variables de decisión

Formalmente, imponemos las condiciones sobre las variables que definimos en un apartado anterior:

$$\zeta_k, \eta_{ij}, \rho_i \in \{0,1\}, \quad \delta_k \text{ int}$$

3.2.4 – Función objetivo

Los costes asociados a variables de decisión presentes en la función objetivo son los siguientes:

- C_k : coste del arco óptico k , definido por la estructura de recursos ópticos a nivel de capa óptica.
- L_{ij} : coste de la ruta j para la conexión i , definido por la suma de los costes de los arcos ópticos por los que pasa.

La expresión de la función de coste es la siguiente:

$$\min - \sum_{k \in S} (C_k \times (1 - \zeta_k)) + \frac{\sum_{i \in E} \sum_{j \in R(i)} (L_{ij} \times \eta_{ij})}{\sum_{i \in E} \sum_{j \in R(i)} (L_{ij} \times N_{ij})} + \frac{\sum_{k \in S} \delta_k}{\sum_{k \in S} M_k}$$

Esta expresión recoge el criterio ordenado de valoración propuesto en la definición del problema. La suma de coste de recursos ópticos liberados (primer sumando de la expresión) es generalmente mayor que el resto de sumandos, con lo cual será el principal objetivo de la minimización. Este valor estará comprendido entre 0 e infinito, y para una liberación de recursos moderada, estará entorno a varias decenas (en unidades de recurso óptico).

Para soluciones factibles con el mismo coste de esta expresión, se buscará la que obtenga una mejor valoración (más pequeña) de los otros dos sumandos. Así, se buscará la reubicación de conexiones eléctricas menos costosa respecto a la situación inicial. El valor de esta expresión raramente cae fuera del intervalo [0.5, 2]. Normalmente, la reubicación acostumbra a dar estados de mayor coste de enrutamiento, con lo que el valor será habitualmente mayor que 1.

Finalmente, el tercer sumando recoge la ocupación en tanto por uno de la red modificada. Este valor estará comprendido entre 0 y 1, hecho que lo convierte en el tercer criterio de minimización en orden de impacto sobre el coste total de la función objetivo.

Capítulo 4

Ejemplos de resolución exacta del problema

En este capítulo se plantean dos ejemplos de redes ópticas diferentes con diversos escenarios de tráfico, para los cuales se aplicará la solución exacta al problema de liberación de recursos ópticos. Posteriormente, se analizarán las características de las soluciones obtenidas.

4.1 – Definición de ejemplos de red

4.1.1 – Topología de prueba

La primera topología presentada es la llamada “red de prueba”. Las características principales de esta red son las siguientes:

Tabla 4.1 – Características de la red de prueba

CAPA ÓPTICA	
Nodos físicos	7
Grado Nodal Medio	1,5
CAPA DE AGREGACIÓN ELÉCTRICA	

Grado Paralelismo Medio	1,5
ESTRUCTURA RESULTANTE MAXIMAL	
Nodos reales y virtuales	19
Arcos ópticos	26
Grado Nodal Medio Combinado	2,25
Enlaces entre nodos reales	14

La red a nivel de capa de agregación eléctrica puede tener más de un arco óptico entre el mismo par de nodos. Cuando ocurre eso, decimos que los arcos son paralelos. Para poder diferenciar dos arcos paralelos que tienen el mismo nodo origen y destino, se crean nodos intermedios virtuales, para poder distinguir el paralelismo.

En la situación maximal (cuando el tráfico llega a la máxima intensidad), se llegan a generar hasta 26 arcos ópticos entre nodos reales y virtuales. Si tenemos en cuenta que las peticiones de conexión sólo se generan entre nodos reales, existen 14 enlaces distintos entre arcos de la red, 12 de los cuales son paralelos con algún otro. La figura 4.1 muestra este estado maximal de la red a nivel de la capa de agregación eléctrica:

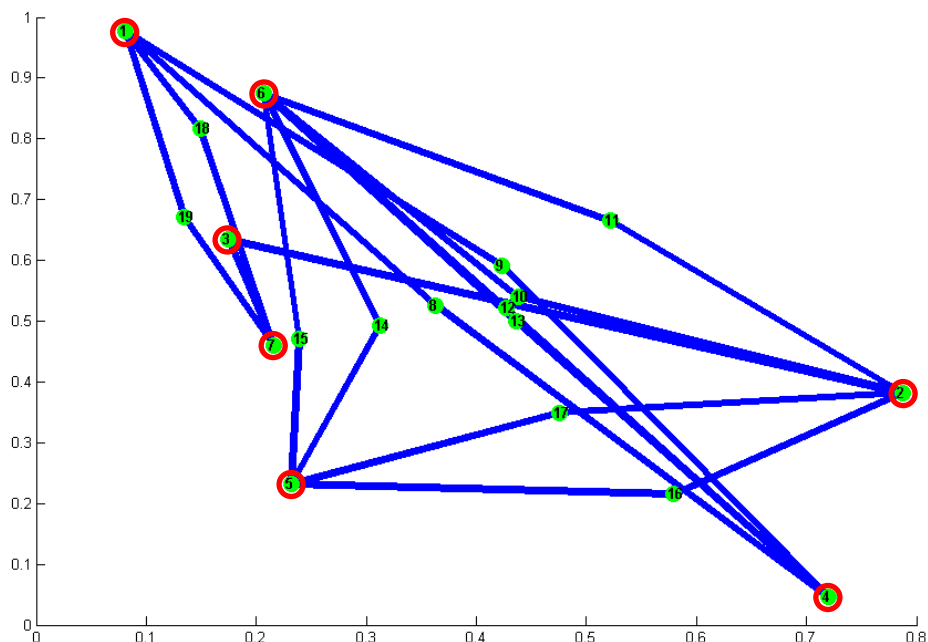


Fig. 4.1 – Red de prueba simulada

Los nodos marcados numerados del 1 al 7 son imagen de los nodos de la capa óptica, mientras que el resto son virtuales y se han creado para poder diferenciar arcos ópticos paralelos entre el mismo par de nodos.

Las características básicas de los arcos ópticos generados son:

Tabla 4.2 – Características arcos de la red de prueba

Capacidad (en unidades de ancho de banda)	16
Coste mínimo (por recursos ópticos utilizados)	1
Coste máximo (por recursos ópticos utilizados)	7

El coste de los arcos depende de los recursos ópticos que lo forman en la capa óptica. Se ha utilizado una escala de valores enteros entre 1 y 7 para caracterizar la magnitud y la complejidad de la estructura de las interfaces utilizadas.

La estructura de esta red de prueba tiene una dimensión amable para el tratamiento con metodología exacta. De todas maneras, el tamaño es lo suficientemente grande como para sacar conclusiones de análisis válidas para casos reales.

4.1.2 – Topología Europea

La segunda red definida muestra una topología de dimensión significativamente mayor respecto la anterior. Se ha tomado como modelo una topología determinada de red europea caracterizada para aplicación de escenarios en redes ópticas [4], y a partir de las especificaciones tomadas se ha simulado una red de similares características. Las más importantes son las siguientes:

Tabla 4.4 – Características de la red europea

PLANO ÓPTICO	
Nodos físicos	16
Grado Nodal Medio	2,5
PLANO DE CONTROL	
Grado Paralelismo Medio	2

ESTRUCTURA RESULTANTE MAXIMAL	
Nodos reales y virtuales	63
Arcos ópticos	107
Grado Nodal Medio Combinado	5
Enlaces entre nodos reales	60

La dimensión de esta red supera a la de prueba en todos los aspectos. El número de nodos reales es más del doble respecto al ejemplo de prueba, mientras que los enlaces distintos entre nodos se multiplican por cuatro. Esto quiere decir que la complejidad en la resolución también aumenta, ya que crece el número de variables y restricciones del problema lineal. La figura 4.2 representa esta mayor complejidad cuando estamos en el caso de máxima intensidad de tráfico:

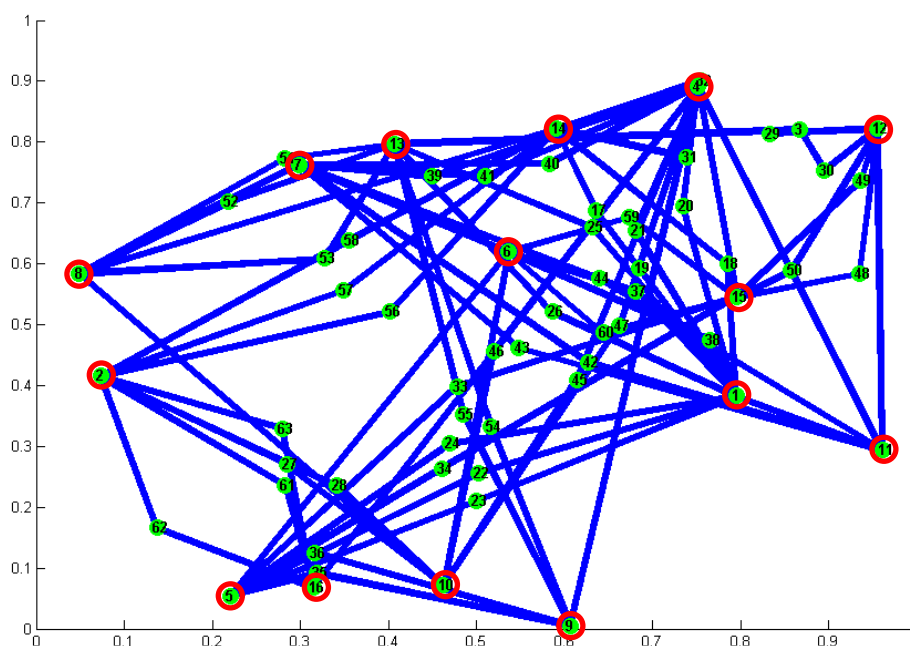


Fig. 4.2 – Red Europea Simulada

Por lo que respecta a los parámetros de los arcos ópticos, se tomarán los mismos valores que en el ejemplo de la red de prueba:

Tabla 4.5 – Características arcos de la red europea

Capacidad (en unidades de ancho de banda)	16
Coste mínimo (por recursos ópticos utilizados)	1
Coste máximo (por recursos ópticos utilizados)	7

4.1.3 – Escenarios de tráfico simulados

Como el problema de liberación de recursos se aplica sobre instancias de tráfico que usan de forma no óptima los recursos ópticos presentes, necesitamos crear un procedimiento que genere estos escenarios de tráfico mejorables, para poder analizar la potencia y la necesidad de aplicación de los métodos de optimización exactos y heurísticos.

Sin necesidad de crear un simulador y obtener estados de la red para diferentes intensidades de tráfico, vamos a generar directamente esas instancias a partir del siguiente algoritmo:

Dada una red y un número máximo de conexiones que queremos crear, aplicamos los siguientes pasos para cada petición de conexión eléctrica:

- 1) Generamos aleatoriamente las características de la conexión:
 - a. Nodos origen y destino: escogidos aleatoriamente con equiprobabilidad sobre el conjunto total de nodos reales de la red.
 - b. Ancho de banda: escogido aleatoriamente en el conjunto $B = \{1,2,3,4\}$, con equiprobabilidad.
 - 2) Elegimos la ruta óptica de la conexión eléctrica, escogida al azar entre las mejores r posibles rutas sobre la red completa.
 - 3) Insertamos la conexión en el escenario de tráfico, por la ruta escogida, siguiendo las siguientes condiciones:
 - a. Si hace falta añadir un arco óptico para satisfacer esa ruta, y si hay suficiente recurso óptico como para generarlo, se introduce el arco óptico a la red para satisfacer la conexión eléctrica
 - b. Si la conexión eléctrica no puede ser satisfecha por la ruta escogida por falta de capacidad óptica, la conexión será rechazada.
-

Con este procedimiento obtendremos conexiones eléctricas enrutadas por rutas mejorables, y arcos ópticos con posibilidad de ser eliminados, con lo que generamos escenarios dónde la aplicación del problema obtendrá resultados visibles y analizables.

Este método de generación de escenarios de tráfico nos proporciona instancias que se pueden encontrar en la práctica, y muestran los casos en los que la optimización puede ser mayor en términos de valor absoluto de la función de coste (porque serán posibles más movimientos).

Dado que los parámetros de definición de las conexiones eléctricas siguen distribuciones similares al tráfico real de datos (como el tráfico por Internet), y que el establecimiento y eliminación de conexiones eléctricas en el tiempo provoca estas situaciones subóptimas de circulación, estos escenarios se pueden considerar tan válidos como los que pudiéramos obtener utilizando técnicas de simulación más sofisticadas y complejas.

Para cada una de las redes descritas, se han generado diferentes escenarios en intensidad de tráfico. El parámetro variable para cada instancia es el número de peticiones de conexión eléctrica generadas. A un mayor número de peticiones de conexión, se habrán generado más conexiones satisfechas y se habrá construido una mayor estructura de arcos ópticos, los cuales tendrán una ocupación cada vez mayor. La tabla 4.6 muestra para cada red las características de los ejemplos simulados:

Tabla 4.6 – Características instancias de tráfico eléctrico

RED	Nº instancias	Min conexiones	Max conexiones	% Ocupación mínima	% Ocupación máxima
Prueba	50	6	35	16%	86%
Europea	100	9	145	15%	87%

La gráfica de la figura 4.3 muestra como incrementa la ocupación total de los arcos ópticos a medida que se van generando escenarios de tráfico con mayor cantidad de conexiones eléctricas. Por una parte, existe una clara tendencia lineal ascendente, que describe que al incrementar el número de conexiones eléctricas, incrementa la ocupación de los arcos ópticos. Pero también puede observarse que el incremento de

peticiones de conexión hasta valores muy altos hace que la red se sature (se aproxime al 100% de ocupación), con lo cual, la pendiente se va haciendo cada vez más pequeña, como muestra la línea de tendencia logarítmica.

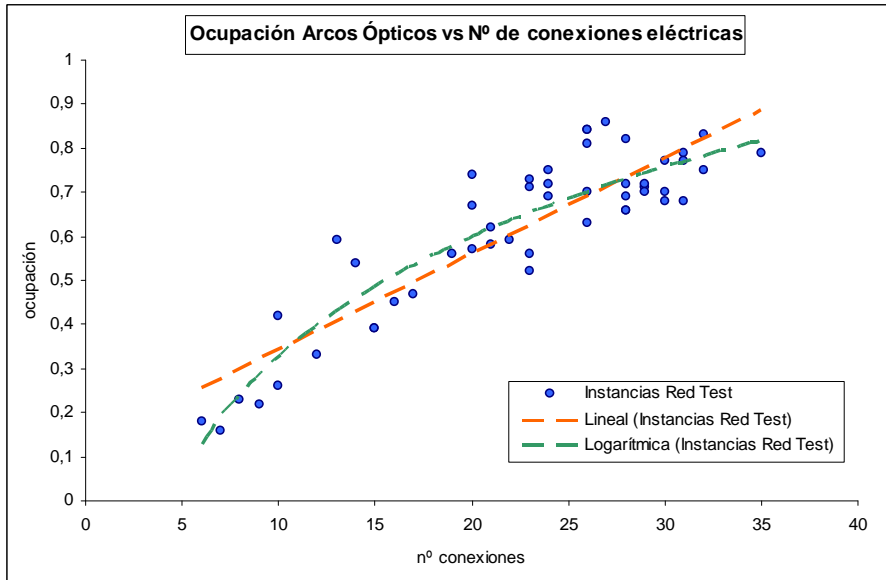


Fig. 4.3 – Incremento de la ocupación óptica respecto a la intensidad eléctrica

También podemos ver como aumenta el número de arcos ópticos activos con el incremento de conexiones eléctricas establecidas, en la figura 4.4.

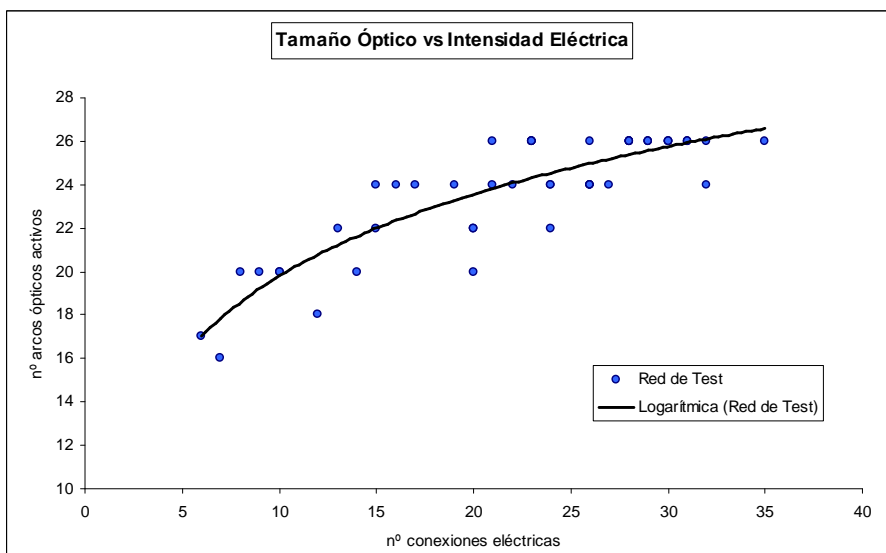


Fig. 4.4 – Dimensión óptica respecto a dimensión eléctrica

Este aumento de arcos ópticos tiene un límite, así como la cantidad de conexiones eléctricas que se pueden establecer, y vendrá determinado por la estructura física de la red de base.

4.2 – Resolución exacta sobre red de test

4.2.1 – Características de la implementación

Utilizando la modelización del Capítulo 3, se ha realizado la implementación en el lenguaje de modelización OPL, y se ha utilizado el motor de resolución CPLEX para resolver las 50 instancias de tráfico generadas para la red de prueba. Las características del equipo y del programario utilizado son las siguientes:

Tabla 4.7 – Características equipo informático

CARACTERÍSTICAS MÁQUINA	
Velocidad procesador	3 GHz
Memoria RAM	1 GB
CARACTERÍSTICAS PROGRAMA	
Plataforma de modelización	OPL IDE 5.5
Motor de solución	CPLEX v.11.0

Las instancias se han generado a partir de la implementación de funciones en lenguaje de programación MATLAB, utilizando la versión 2007 de la plataforma del mismo nombre.

4.2.2 – Resumen de resultados

La siguiente tabla muestra los resultados más relevantes para las 50 instancias de tráfico generadas:

Tabla 4.8 – Resultados CPLEX red prueba

id	Arcos ópticos	Conexiones eléctricas	Var.	Restr.	Ocupación óptica	f. obj.	Arcos eliminados	Conexiones enrutadas	Tiempo ejecución
1	17	6	364	58	0,18	-16972	6	3	0,03
2	16	7	382	60	0,16	-16981	6	2	0,001
3	20	8	440	72	0,23	-19937	10	5	0,27
4	20	9	535	76	0,22	-25973	12	5	0,11
5	20	10	540	80	0,26	-25957	9	4	0,16
6	20	10	540	80	0,42	-17939	8	6	0,33
7	18	12	696	84	0,33	-18960	8	5	0,06
8	22	13	759	96	0,59	-23939	10	8	0,78

9	20	14	810	96	0,54	-21931	8	7	0,17
10	22	15	869	104	0,39	-21972	10	6	0,44
11	24	16	848	112	0,45	-31867	14	12	0,91
12	24	17	898	116	0,47	-26871	12	12	0,64
13	24	15	873	108	0,39	-25939	12	6	0,81
14	24	19	1093	124	0,56	-24907	10	10	0,61
15	22	20	1144	124	0,67	-6953	4	5	0,48
16	26	21	1207	136	0,58	-23871	12	13	0,84
17	22	20	1144	124	0,57	-22890	10	14	0,41
18	26	23	1317	144	0,56	-29873	14	14	5,13
19	26	23	1317	144	0,52	-30860	14	16	0,81
20	20	20	1140	120	0,74	-10951	4	5	0,3
21	24	22	1258	136	0,59	-12911	6	8	0,66
22	24	21	1203	132	0,62	-18858	10	15	0,67
23	26	28	1592	164	0,72	-13894	8	13	1,09
24	26	26	1482	156	0,63	-25864	12	17	8,42
25	26	28	1592	164	0,66	-18872	10	16	5,73
26	24	24	1368	144	0,72	-11884	6	10	0,66
27	26	23	1317	144	0,71	-9842	6	13	1,39
28	24	27	1533	156	0,86	-21890	8	15	1,08
29	26	30	1702	172	0,7	-19903	10	12	5,37
30	26	23	1317	144	0,73	-19884	10	14	7,81
31	26	29	1647	168	0,71	-12826	8	16	1,42
32	26	28	1592	164	0,66	-17885	8	12	1,58
33	26	29	1647	168	0,72	-9857	6	15	1,47
34	26	31	1757	176	0,79	-10900	6	15	1,63
35	24	24	1368	144	0,75	-8925	4	11	0,62
36	26	30	1702	172	0,68	-18855	10	19	4,58
37	24	26	1478	152	0,84	-1965	2	4	1,22
38	24	26	1478	152	0,7	-16904	8	10	0,92
39	22	24	1364	140	0,69	-5886	4	11	3,69
40	26	35	1977	192	0,79	-8910	4	10	6,37
41	24	26	1348	152	0,81	-8895	6	10	0,53
42	26	30	1702	172	0,77	-10871	6	12	1,17
43	24	26	1478	152	0,84	-20845	8	16	0,88
44	26	31	1757	176	0,77	-9844	6	15	1,7
45	26	32	1812	180	0,75	-8856	6	14	2,25
46	24	32	1648	176	0,83	-9877	4	12	1,27
47	26	28	1592	164	0,82	-12847	8	17	1,41
48	26	29	1647	168	0,7	-16905	8	11	1,41
49	26	28	1592	164	0,69	-15879	7	13	1,72
50	26	31	1757	176	0,68	-12879	8	13	2,02
TOTAL						-838066			82,95

Si agrupamos los resultados por algún criterio, podemos analizar gráficamente algunas de las variables más significativas y sacar conclusiones. Aunque hayamos podido ver anteriormente que la ocupación óptica y la intensidad de tráfico eléctrico están marcadamente correlacionadas, vamos a hacer el análisis gráfico independiente para esas dos agrupaciones diferentes.

4.2.3 – Análisis del tiempo de computación

Un primer análisis es el del tiempo medio invertido por instancia para resolver el problema computacionalmente. Las gráficas obtenidas forman la figura 4.5:

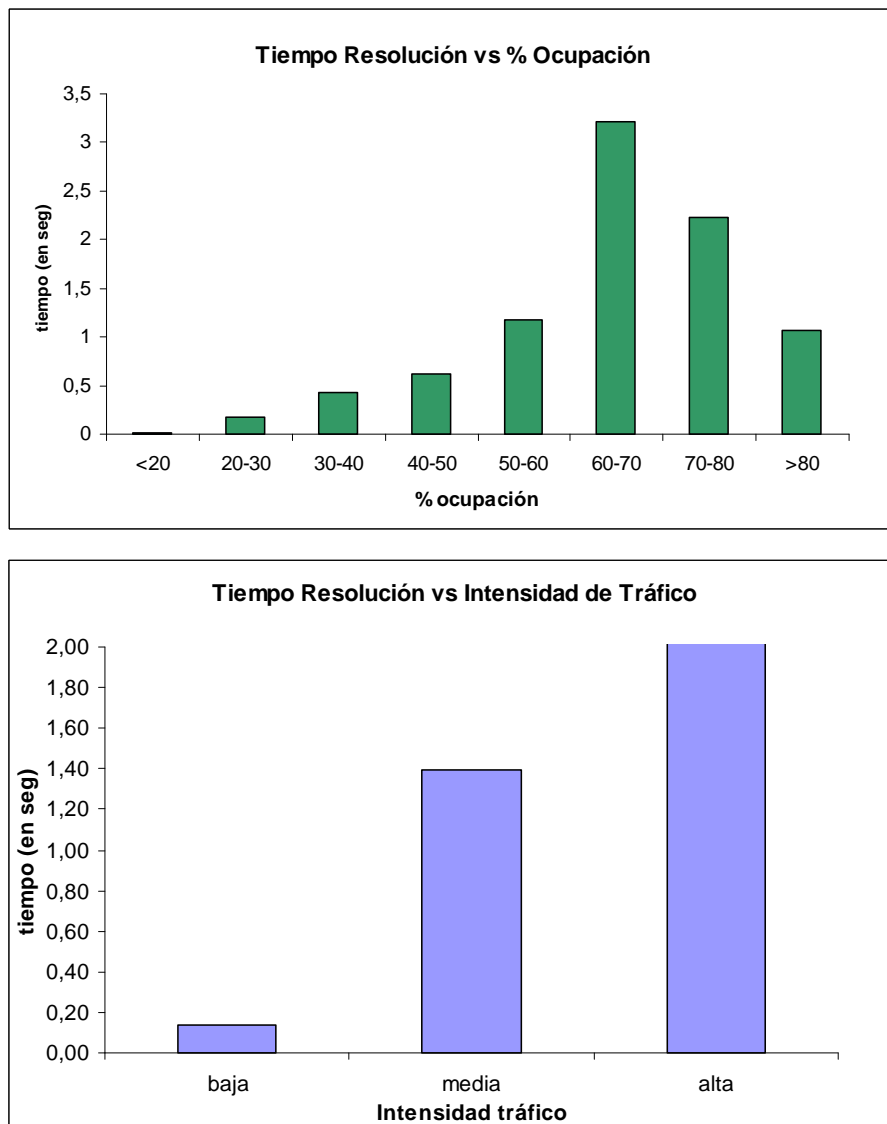


Fig. 4.5 – Tiempo de computación CPLEX para red de prueba

A simple vista de las gráficas pueden observarse dos comportamientos que tienen una justificación lógica:

- 1) El aumento de la ocupación de los arcos hace incrementar el tiempo de computación, excepto para valores muy altos de ésta, para los cuales el tiempo invertido decrece. Esto puede justificarse si analizamos lo que pasa en los

valores extremos de ocupación. Cuando hay poca ocupación, existen pocos cambios posibles a considerar que provoquen mejora, ya que habrá pocas conexiones que tener en cuenta para redistribuir. Eso reducirá bastante el entorno de exploración y, con ello, el tiempo de búsqueda de solución. Por otra parte, en casos de mucha saturación, a pesar de existir muchas conexiones, la cantidad de movimientos permitidos factibles también es baja, lo que también reduce las posibilidades de búsqueda. Con ocupaciones medias-altas, la cantidad de movimientos posibles (arcos susceptibles de eliminarse y rutas alternativas posibles) es mayor porque existe más holgura para optimizar un escenario de tráfico. De ahí que salgan los tiempos de computación más altos.

No obstante, como consideración definitiva, vamos a asegurar una clara tendencia creciente del tiempo de computación en la línea de crecimiento de la ocupación óptica. Para casos de ocupación cercanos a la saturación, situación que a la práctica trataremos de evitar, se debería hacer un estudio más detallado de la evolución de tiempos de ejecución.

- 2) El tiempo de ejecución aumenta significativamente al aumentar la cantidad de conexiones eléctricas generadas. Un mayor tráfico implica una mayor dimensión del problema (tanto en conexiones eléctricas como en arcos ópticos activos). Eso hace incrementar el tamaño de restricciones y variables del problema. A partir de la formulación del problema, podemos calcular la dimensión del problema:

Num. Variables	<i>Max Rutas x Conexiones eléctricas + 2 x Arcos ópticos</i>
Num. Restricciones	<i>4 x Conexiones eléctricas + 2 x Arcos ópticos</i>

Así, vemos como la intensidad de tráfico genera diferentes tamaños de problema, que tienen tiempos diferentes de resolución exacta.

4.2.4 – Análisis de las modificaciones producidas

Una nueva solución obtenida produce una serie de cambios respecto a la situación original, antes de aplicar el proceso de optimización. Esas modificaciones se dividen en arcos ópticos eliminados y en rutas de conexiones modificadas.

Para la agrupación por categorías de ocupación de la red óptica, tenemos los siguientes porcentajes de elementos modificados:

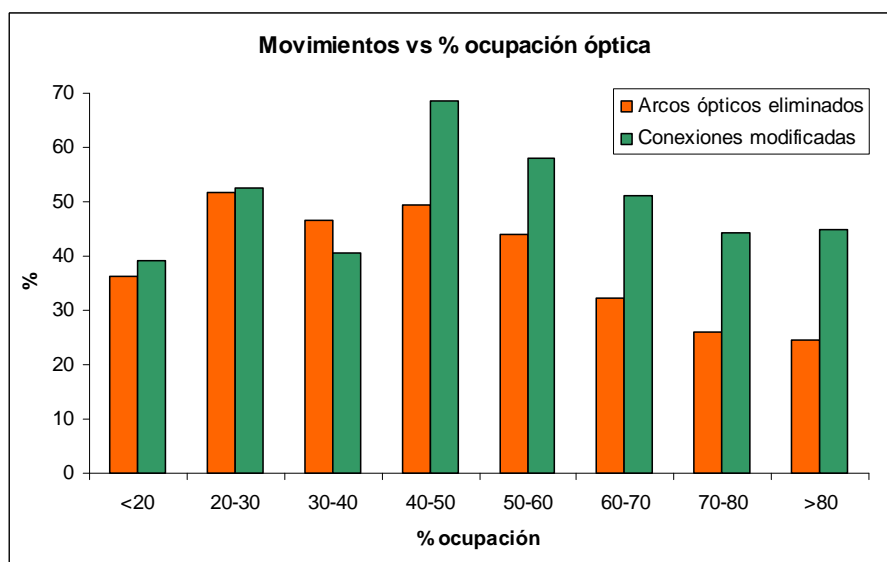


Fig. 4.6 – Movimientos realizados por ocupación

Podemos observar como para valores de ocupación medios (entre 30% y 50%) se producen los porcentajes más elevados de modificación de elementos. La mayor eliminación de arcos ópticos se produce cerca del 30% de ocupación, en la que se llegan a liberar hasta un 50% de los arcos ópticos de la red.

A medida que la ocupación se hace mayor, cada vez son menos los arcos que se pueden eliminar, puesto que cada vez cuesta más satisfacer la factibilidad de una ruta alternativa para el total de conexiones del arco a eliminar. Este incremento de conexiones circulantes por arco aumenta la probabilidad de que alguna de ellas no pueda ser reenrutada por falta de capacidad alternativa, hecho que anula la eliminación del arco.

Por lo que respecta a cambios en las rutas, en los escenarios de mayor movilidad, llegan a redistribuirse hasta un 60% de las conexiones establecidas. Los escenarios de ocupación media ofrecen la posibilidad de realizar una buena compactación de las

conexiones eléctricas, puesto que suelen ser escenarios con una topología óptica caracterizados por tener un número alto de arcos y una ocupación media.

Comparamos ahora los resultados agrupados por intensidad de tráfico:

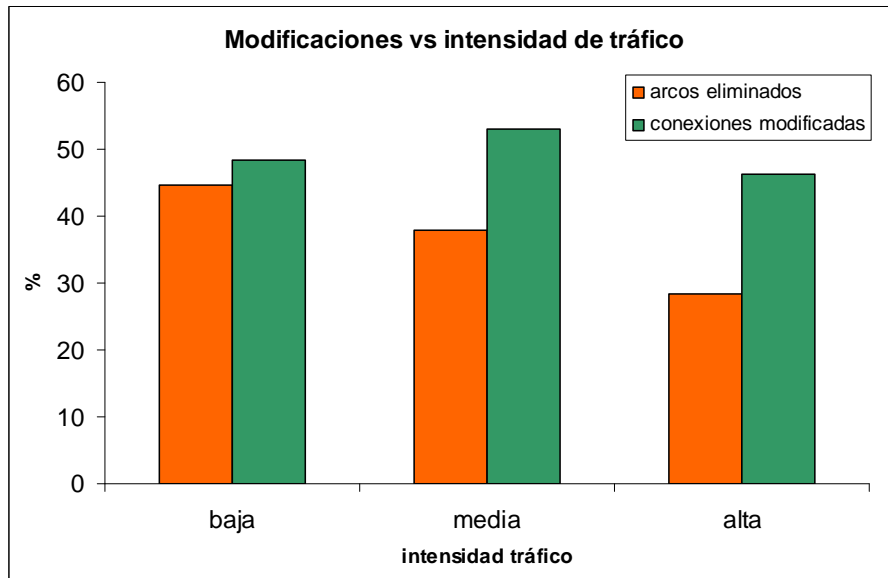


Fig. 4.7 – Movimientos realizados por intensidad

Vemos una clara tendencia lineal con pendiente negativa en intensidades crecientes de tráfico para el porcentaje de arcos ópticos eliminados. Una mayor saturación de la red proporciona una clara disminución de los arcos ópticos eliminados respecto a la estructura inicial.

Las modificaciones en las rutas siguen una interpretación similar a la realizada en la agrupación por intervalos de ocupación óptica.

4.2.5 – Conclusiones

De la aplicación del problema de optimización de interfaces ópticas en diversos escenarios de tráfico sobre una red de prueba de dimensión reducida, se pueden extraer, a la vista de los resultados, dos conclusiones evidentes:

- A) La eliminación de arcos ópticos es claramente significativa, produciendo un beneficio asociado a la liberación de los recursos ópticos que ocupan. Esta

eliminación se sitúa para escenarios de tráfico altamente dinámicos entre un 25% y un 50% dependiendo de la ocupación eléctrica de los arcos ópticos.

- B) El tiempo de resolución de estas instancias crece con el aumento de la intensidad de tráfico, debido a que esta propicia un aumento significativo de la dimensión del problema optimizado.

4.3 – Resolución exacta sobre red europea

4.3.1 – Características de la implementación

Para esta batería de pruebas se ha utilizado el mismo equipo que en el ejemplo de prueba. La configuración del motor de resolución exacta que se ha empleado se caracteriza por haber usado un valor de espacio entre la solución lineal relajada y la solución incumbente entera (*MIP gap*) del 0,01%

4.3.2 – Resumen de resultados

La siguiente tabla muestra los resultados más relevantes para las 100 instancias de tráfico generadas:

Tabla 4.9 – Resultados CPLEX red europea

id	Arcos ópticos	Conexiones eléctricas	Var.	Restr.	Ocupación óptica	f. obj.	Arcos eliminados	Conexiones enrutadas	Tiempo ejecución
1	27	9	963	90	0,28	-12976	9	4	0,06
2	41	11	1193	126	0,15	-33944	21	8	0,05
3	34	13	1381	120	0,22	-11979	10	4	0,02
4	43	15	1601	146	0,28	-25940	22	9	0,05
5	43	17	1803	154	0,27	-36952	16	9	0,41
6	47	19	2013	170	0,21	-45904	25	12	0,77
7	49	21	2219	182	0,3	-32912	21	13	0,27
8	42	23	2407	176	0,33	-39896	22	13	0,75
9	54	25	2633	208	0,3	-54890	26	15	2,02
10	59	27	2845	226	0,35	-48895	29	16	8,69
11	58	29	3045	232	0,33	-51889	28	16	57,13
12	64	31	3259	252	0,31	-59859	36	20	11,55
13	65	33	3463	262	0,28	-81827	39	25	37,56
14	67	35	3669	274	0,32	-64834	34	23	89,22
15	65	37	3867	278	0,27	-56821	37	24	27,86

16	69	39	4077	294	0,37	-57864	35	21	79,8
17	75	41	4291	314	0,27	-73837	45	26	196,17
18	76	43	4495	324	0,39	-72811	40	27	356,69
19	68	45	4681	316	0,39	-52859	31	19	622,55
20	78	47	4903	344	0,5	-74763	39	33	2452,63
21	85	49	5119	366	0,41	-77762	46	32	3491,44
22	68	51	5287	340	0,42	-52846	29	24	488,61
23	83	52	5418	374	0,38	-88778	44	36	4055,25
24	75	55	5705	370	0,51	-58815	32	28	525,33
25	83	56	5822	390	0,41	-83741	47	39	2107
26	84	58	6026	400	0,38	-80791	44	34	43273,66
27	79	61	6319	402	0,49	-63766	34	34	14020,02
28	85	68	7038	442	0,5	-81721	43	41	53796,94
...									

A partir de este punto, apenas se obtienen instancias resueltas, ya que la mayoría agotan los recursos de la máquina utilizada. Los pocos resultados que se tienen superan las 20 horas de ejecución. Por eso, haremos análisis con sólo las 28 instancias que se detallan arriba.

Los siguientes apartados tratan de analizar los mismos aspectos estudiados en la red de prueba, a pesar de no tener disponibilidad de datos en la totalidad del mismo rango

4.3.3 – Análisis del tiempo de computación

En el resumen anterior, podemos ver valores de resolución que oscilan entre centésimas de segundo y varias horas. Las gráficas de la figura 4.8 muestran el tiempo medio de resolución por ocupación de la red y por intensidad de tráfico, en los intervalos para los que se disponen datos:

Existen valores para ocupaciones inferiores al 30% y para intensidad de tráfico baja, pero son tan pequeños en comparación con el resto, que no casi no son perceptibles visualmente en la escala definida. Para intensidad de tráfico elevada, no hay registros de tiempo medio.

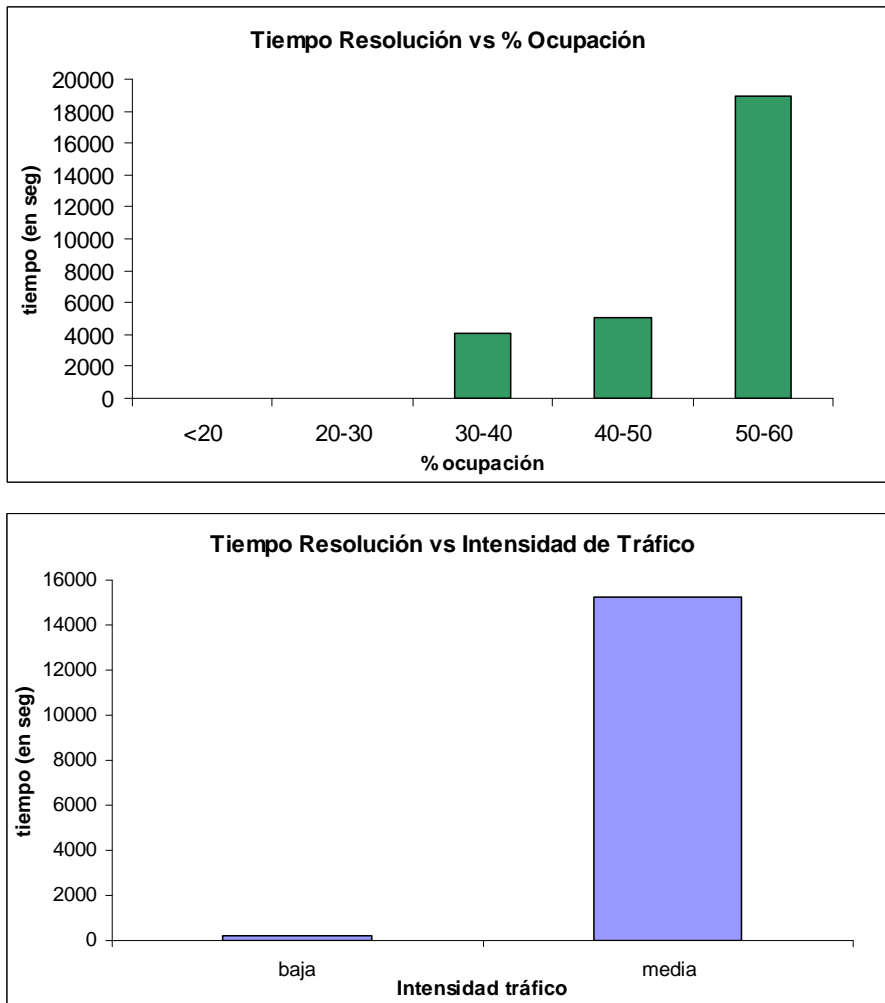


Fig. 4.8 – Tiempo de ejecución en red europea

Podemos ver que el incremento de ocupación y de tráfico provoca un elevado incremento del tiempo de computación, hasta valores que no son admisibles para el contexto en el que estamos trabajando. A priori, parece razonable considerar que el tiempo de computación crece exponencialmente con la complejidad del problema. La gráfica de la figura 4.9 muestra la nube de dispersión para el logaritmo del tiempo de resolución de cada instancia en relación al número de restricciones del problema. En ella podemos ver que existe una significativa relación lineal que nos certifica que el tiempo de resolución del problema exacto crece de forma exponencial a medida que la dimensión del problema aumenta.

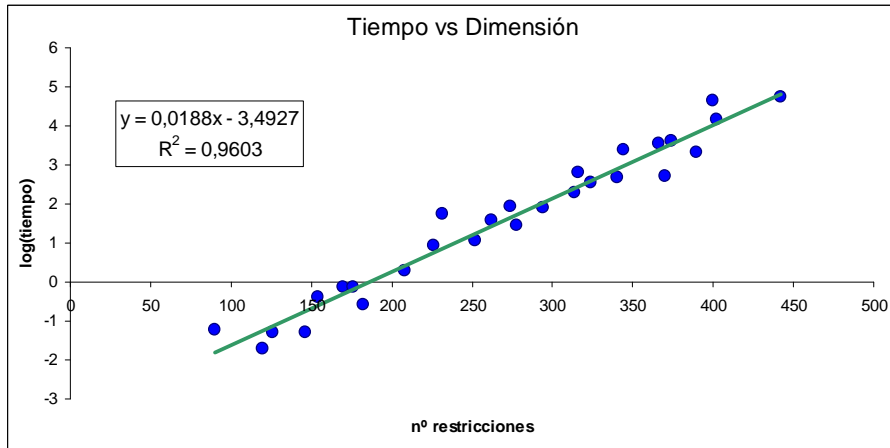


Fig. 4.9 – Tiempo de ejecución vs. nº restricciones

4.3.4 – Análisis de las modificaciones producidas

Por lo que respecta al efecto de la optimización sobre los elementos de la solución, las próximas gráficas comparativas muestran los resultados. Vemos que existe un patrón bastante claro y poco variable respecto a los criterios de agrupación de datos. Para diferentes ocupaciones de tráfico y para distintas intensidades de tráfico, el porcentaje de arcos ópticos eliminados oscila en general entre el 40% y el 50%, mientras que el porcentaje de conexiones eléctricas modificadas se sitúa mayoritariamente entre el 50% y el 60%.

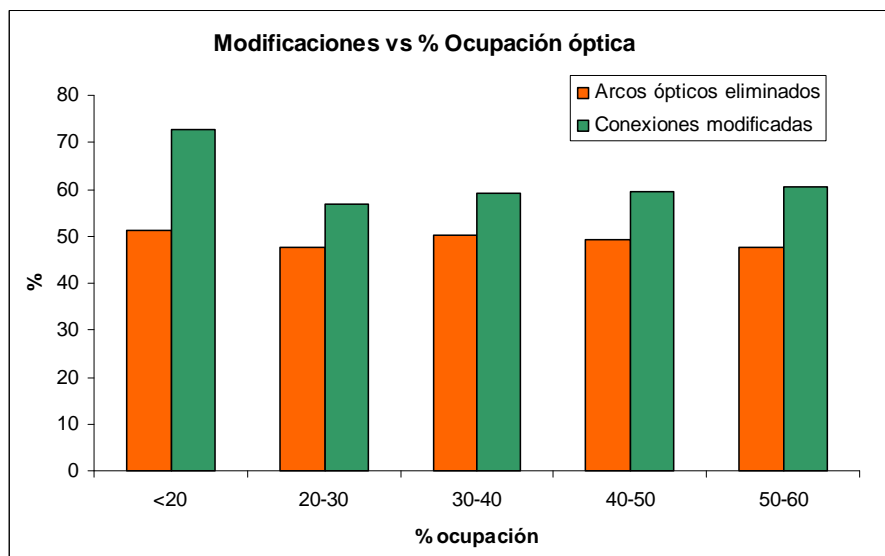


Fig. 4.10 – Modificaciones realizadas por ocupación

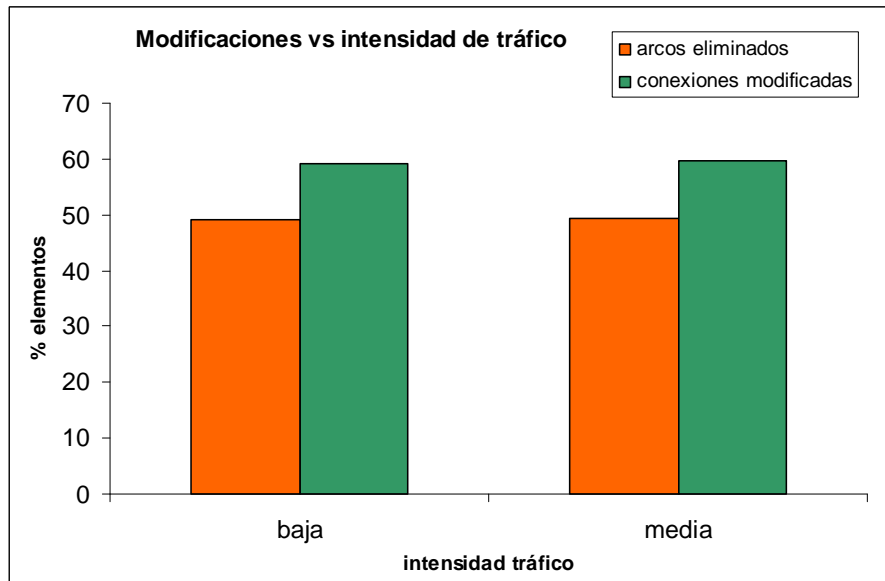


Fig. 4.11 – Modificaciones realizadas por intensidad

El porcentaje de arcos eliminados respecto a la red original es muy similar con respecto al resultado obtenido en las instancias de la red de prueba. En cuanto a las conexiones modificadas, para valores medios de ocupación, los resultados se asemejan considerablemente.

4.3.5 – Conclusiones

En este escenario también podemos sacar dos conclusiones claras:

- A) La liberación de recursos ópticos es claramente efectiva y provoca un beneficio significativo en el contexto de mejor aprovechamiento de recursos ópticos. Además, los resultados obtenidos son similares a los de la red de prueba.
- B) El método exacto para problemas de una dimensión real no es eficiente para resolver el problema en un tiempo que permita su aplicación a la práctica, sin que haya modificado el escenario de aplicación. Muchas de las instancias de máxima intensidad de tráfico no han podido ser resueltas y otras lo han sido con tiempos demasiado grandes.

Capítulo 5

Meta-heurística GRASP

Ciertos problemas de optimización tienen una estructura y una dimensión que dificultan la resolución a partir de procedimientos exactos. En el capítulo anterior hemos visto que, en determinados escenarios de tráfico, la resolución óptima del problema de liberación de recursos ópticos se obtenía con tiempos inadmisibles para poder llevarse a cabo en la realidad. En otros casos, la cantidad de conexiones y rutas posibles se hacía tan grande que un ordenador potente se quedaba sin memoria y no podía resolver el ejemplo. Para estos escenarios, que se pueden dar en situaciones reales, no se pueden aplicar procedimientos de resolución exactos porque no conseguimos el propósito de mejorar el tráfico de la red de forma rápida y efectiva.

Existen procedimientos de resolución aproximados, llamados heurísticas o meta-heurísticas, que permiten obtener buenas soluciones de los problemas planteados, con un coste computacional admisible. Si bien estos procedimientos no aseguran la obtención de la solución óptima para cada aplicación, permiten dar con soluciones que cubren las necesidades de mejora que los mecanismos exactos no pueden otorgar de forma eficiente.

En el siguiente capítulo definiremos la meta-heurística GRASP general al detalle, que será la que luego aplicaremos al problema de liberación de recursos ópticos, cuyas particularidades y resultados analizaremos en el próximo capítulo.

5.1 – Definiciones básicas

El término GRASP hace referencia a las siglas de la frase en inglés *Greedy Random Adaptive Search Procedure* [5]. Esta frase define claramente los aspectos más importantes de esta heurística: se trata de un procedimiento adaptativo y aleatorio de construcción de soluciones, aplicando un criterio voraz de selección. A continuación se pasa a definir las dos características elementales del GRASP.

5.1.1 – Criterio de selección voraz

Una solución factible a un problema como el que tratamos en este proyecto consta de una serie de elementos que la conforman y que en su conjunto forman dicha solución. Para cualquier problema nos podemos plantear esa separación, que vendrá dada por la estructura del problema. Eso quiere decir que podemos construir una solución factible a partir de ir añadiendo nuevos elementos a un conjunto de base, de forma iterativa. Este conjunto irá creciendo hasta tener la estructura propia de una solución factible de nuestro problema. A partir de una semilla inicial, que puede ser el conjunto vacío, añadimos uno a uno elementos que nos permitirán dirigirnos a un conjunto que tenga las características propias de una solución posible.

Nuestro objetivo al aplicar un procedimiento de optimización es el de encontrar la mejor solución posible dada una cierta función de evaluación. Construir una solución siguiendo un método como el descrito anteriormente puede llevarnos a una instancia que tenga un valor muy alejado del óptimo para ese problema. Por otra parte, resulta computacionalmente gigantesco y conceptualmente carente de sentido la generación y evaluación de todas las posibles soluciones factibles, para determinar la mejor evaluada. Esto nos conduce a justificar la aplicación de un criterio de selección de los elementos de una solución, que nos permita prescindir de la necesidad de formular todas las

posibilidades, y aún así podemos obtener instancias que consigan una buena evaluación en la función objetivo.

El criterio de construcción usado por esta heurística se llama criterio *greedy* o voraz. La idea es insertar en cada iteración el elemento mejor valorado, según la función de coste del problema. Cada decisión se valora con la información de que se dispone en el momento dado, y se elige la opción que consigue un mayor beneficio en la función de evaluación. Con ello logramos obtener soluciones que, a cada paso de construcción, añaden la menor cantidad posible de valor a la función de coste a minimizar. Este procedimiento es intuitivo y sensato, pero no nos conduce necesariamente a la solución óptima. En el ejemplo de la figura siguiente, podemos construir una solución añadiendo el elemento menos costoso a cada paso y obtener una buena solución, pero podemos comprobar que no daremos con la solución óptima:

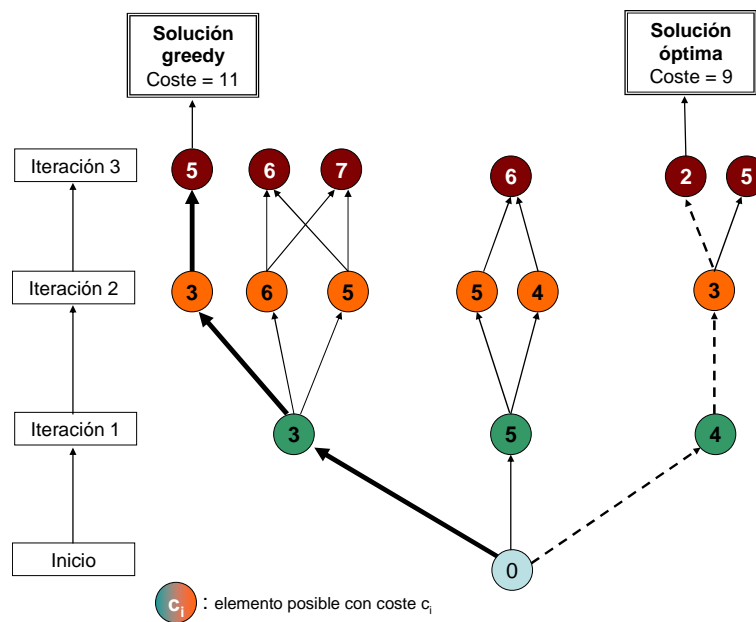


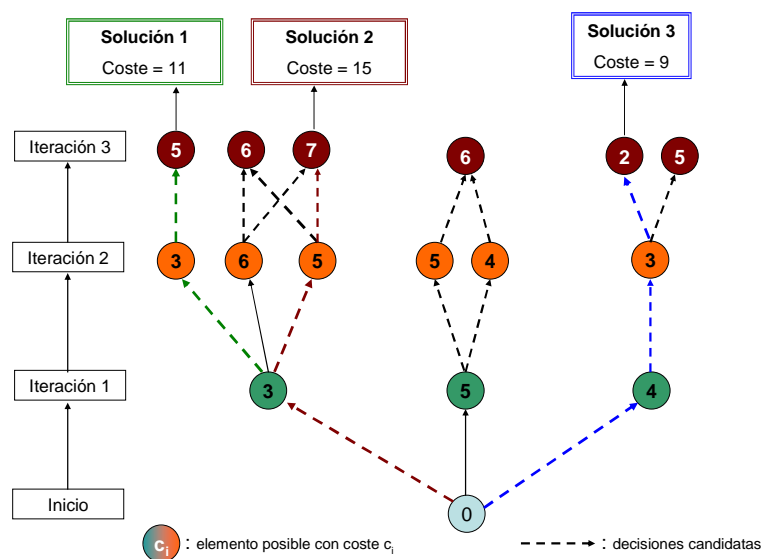
Fig. 5.1 – Selección de la mejor solución por la fase constructiva *greedy*

Así pues, la aplicación de un criterio de selección *greedy* nos permite obtener el mejor paso a cada momento, en el entorno inmediato de la toma de la decisión, sin asegurar que esa decisión sea óptima en el contexto global de la solución. Las heurísticas construyen soluciones a partir de cero y sin tener información futura de antemano que las guíe en adelante, con lo cual este criterio se plantea como un buen procedimiento para encontrar soluciones buenas a nuestro problema.

5.1.2 – Aleatorización de la selección

El criterio de selección *greedy* nos permite obtener una solución factible buena basada en un argumento de optimalidad local. Si definimos el criterio claramente y lo aplicamos exactamente, podemos ver como la obtención de la solución es única. Es decir, siempre que apliquemos el proceso en las mismas condiciones, obtendremos la misma solución. En cada iteración, siempre será la misma opción la que cumpla el mejor valor de nuestro criterio de selección. En este aspecto, el criterio *greedy* es inflexible, ya que tantas veces como lo apliquemos, tantas veces daremos con el mismo resultado.

Con la intención de dotar de cierta flexibilidad y de aumentar las posibilidades de encontrar soluciones mejores, se puede incorporar un criterio de selección que contenga una componente aleatoria que le confiera cierta holgura. Podemos a cada iteración, en lugar de tomar la mejor opción, elegir una posibilidad de entre un conjunto de opciones candidatas. La lista de elementos candidatos se obtiene de la selección de los mejores elementos según el criterio *greedy* establecido y la elección de uno de esos elementos se realiza de forma totalmente aleatoria. Sobre el ejemplo anterior podemos aplicar 3 veces el procedimiento aleatorizado a partir de las 2 mejores opciones posibles a cada momento y obtener 3 soluciones diferentes. Los resultados se ejemplifican en la figura 5.2.



Por lo tanto, a partir de un procedimiento sencillo de selección de elementos y de formación de soluciones (criterio de selección *greedy*), podemos dotarlo de mayor potencia (por lo que respecta a obtención de soluciones mejor valoradas) si definimos la toma de decisión como una selección aleatoria de elementos candidatos con buenas propiedades.

5.2 – Estructura y fases de la meta-heurística GRASP

En el punto anterior hemos visto que la heurística GRASP se basa en dos aspectos clave: la utilización de un criterio *greedy* para la construcción de soluciones factibles; y la aleatorización de este proceso constructivo para obtener soluciones diferentes en aplicaciones distintas sobre los mismos datos. A continuación detallaremos las fases de esta heurística, los parámetros ajustables, y su estructura global de aplicación.

5.2.1 – Estructura elemental de la heurística

La heurística GRASP es un algoritmo iterativo que nos permite obtener buenas soluciones a problemas de optimización de difícil resolución exacta. Debido a esa naturaleza iterativa y a la presencia de componentes aleatorios que generan soluciones variadas, el GRASP deberá ser aplicado repetidas veces hasta cumplir un criterio de parada, que nos dé como resultado una solución a nuestro problema. El esquema elemental se puede observar en la figura 5.3.

La heurística consiste en un algoritmo que se repite un cierto número de veces, y que contiene en si mismo dos fases diferenciadas: una fase constructiva donde se genera una solución mediante el procedimiento voraz y aleatorio; y una fase de post-procesamiento en la que tratamos de mejorar la solución en un entorno de búsqueda local. Tras la ejecución de estas fases, obtenemos una solución factible con la que, si resulta ser la mejor obtenida hasta el momento, pasará a quedar reservada como solución incumbente. Cuando lleguemos a la última iteración, determinada por el criterio de parada que hayamos fijado, paramos el algoritmo iterativo y recuperamos la solución incumbente, que pasará a ser la solución definitiva obtenida por la heurística.

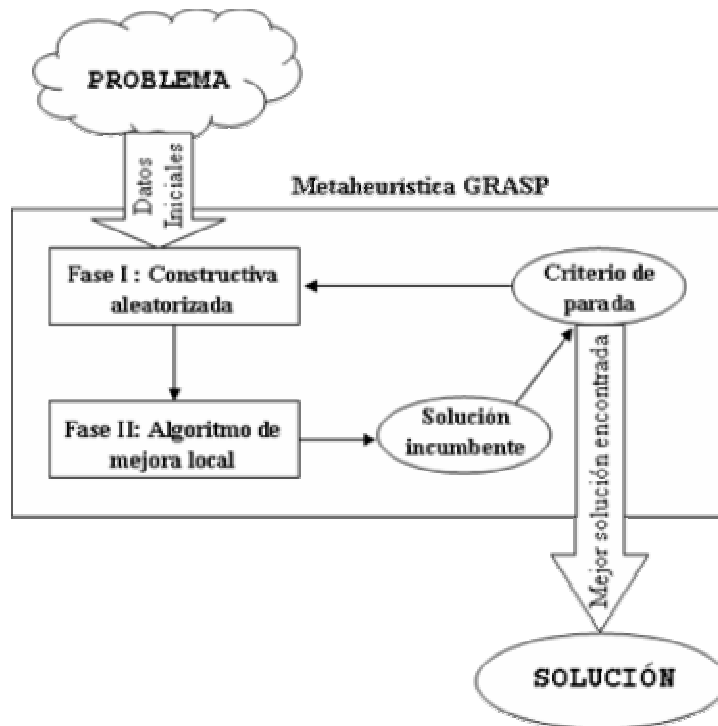


Fig5.3 – Esquema de la meta-heurística GRASP

El pseudocódigo genérico del cuerpo principal del GRASP es el siguiente:

Algoritmo 5.1 – Algoritmo Principal GRASP

Sea P el conjunto de datos iniciales del problema de minimización.
 Sea S una solución cualquiera del problema.
 Sea S_{inc} la solución incumbente del problema.
 Sea $C(S)$ la función de coste de la solución S .
 Sea C_{inc} el coste de la solución incumbente.
 Sea α el parámetro de aleatorización de la Fase I.
 Sea *metodo* el criterio de mejora utilizado.

```

alg  $S_{GRASP} = GRASP(param P, param \alpha)$ 
  var STOP := 'falso';
  var  $S_{inc} := \emptyset$ ;
  var  $C_{inc} := \infty$ 
  mientras (STOP == 'falso') hacer
     $S := FASE\ I (param P, param \alpha)$ ;
     $S' := FASE\ II (param P, param S, param metodo)$ ;
    si  $C(S') \leq C_{inc}$  entonces
       $S_{inc} := S'$ ;
       $C_{inc} := C(S')$ ;
    fsi
  si (condicion parada) entonces
  
```

```

                STOP:='verdadero' ;
        fsi
    fmientras
        SGRASP:= Sinc ;
    falg

```

5.2.2 – Fase I o fase constructiva aleatorizada

En esta fase, se genera una solución factible mediante el procedimiento iterativo de inserción de nuevos elementos, escogidos al azar de entre una lista de candidatos con mejor coste según el criterio voraz de selección. Para poder ejecutar esta fase es necesario acceder al conjunto de datos iniciales que conforman el problema. Además, tendremos que fijar previamente que grado de aleatorización deseamos tomar. Antes de detallar el pseudocódigo de esta fase, comentaremos dos aspectos esenciales:

- 1) **Parámetro α :** este parámetro que elegimos y fijamos antes de proceder a la ejecución de la fase constructiva, nos permite definir que grado de aleatorización tomaremos en la generación de cada iteración. Este parámetro oscila entre 0 y 1, y el significado que tiene es el siguiente:
 - Si $\alpha=0$, no estaremos tomando en cuenta ningún grado de aleatoriedad. Eso significa que en cada decisión constructiva, tomaremos únicamente la mejor opción posible. Podemos ver pues que el método constructivo *greedy* puro no es más que un método constructivo GRASP con $\alpha=0$.
 - Si $\alpha=1$, estaremos tomando en consideración todas las opciones posibles como posibilidades de elección. Esto significa que, en cada decisión constructiva, elegiremos puramente al azar una opción de todas las posibles
 - Si $0 \leq \alpha \leq 1$, estaremos tomando diferentes niveles de aleatorización en la decisión constructiva, tal y como se explicará más adelante.

- 2) **Lista de candidatos restringida (RCL):** una vez fijado el parámetro α , este permanecerá estático durante todas las iteraciones de la fase constructiva. Pero las posibilidades concretas que podemos tomar en cada iteración son diferentes según la situación en la que nos encontremos. Es por ello necesario, antes de tomar la decisión aleatoria, definir que opciones son las posibles de elección.

Esta lista de posibilidades se llama lista de candidatos restringida (en inglés, *Restricted Candidate List*, RCL).

Dada una iteración de la fase constructiva, definimos F como el conjunto total de elementos f_i que pueden incorporarse en esa iteración a la solución en construcción. Complementariamente, para cada elemento obtendremos el coste $C(f_i)$ como el valor que aumentaría la función de evaluación si se incorporara ese elemento a la solución. Finalmente, definimos Cf_{max} y Cf_{min} , como el máximo y mínimo valor de coste del conjunto de candidatos, respectivamente. Así pues, tenemos que la RCL queda definida de la siguiente manera:

$$RCL \equiv \left\{ f_i \mid C(f_i) \leq Cf_{min} + \alpha \times (Cf_{max} - Cf_{min}), \forall f_i \in F \right\}$$

El parámetro α nos determina la frontera de los valores que serán incorporados a la lista de candidatos. Podemos ver cual es la lista de candidatos para los valores extremos de α , con la notación presentada:

- $\alpha = 0 \rightarrow RCL \equiv \left\{ f_i \mid C(f_i) \leq Cf_{min}, \forall f_i \in F \right\} \equiv Cf_{min}$
- $\alpha = 1 \rightarrow RCL \equiv \left\{ f_i \mid C(f_i) \leq Cf_{min} + (Cf_{max} - Cf_{min}), \forall f_i \in F \right\} \equiv$
 $\left\{ f_i \mid C(f_i) \leq Cf_{max}, \forall f_i \in F \right\} \equiv F$

Visto esto, el pseudocódigo de esta fase constructiva queda detallado de la siguiente manera, en el algoritmo 5.2. El primer paso dentro del bucle iterativo es determinar el conjunto de posibles elementos que pueden entrar a formar parte de la solución sin restricciones de coste. Este conjunto lo obtenemos a partir de los datos del problema y lo llamaremos F . En segundo lugar, obtenemos la lista restringida de candidatos a partir de ese conjunto F y del parámetro fijado α . Tras la obtención de un valor aleatorio uniformemente distribuido entre 0 y 1, obtenemos el índice de la RCL del elemento que vamos a incorporar a la solución. Finalmente, concluimos la iteración añadiendo ese elemento al conjunto solución en construcción.

Algoritmo 5.2 – Fase Constructiva

```
alg S := FASE I (param P, param  $\alpha$ )
  var S= $\emptyset$ ;
  var ite=1;
  mientras (RCL  $\neq \emptyset$  || ite=1) hacer
    F:= P[ite];
    RCL:= { $f_i$  |  $C(f_i) \leq C_{\min} + \alpha(C_{\max} - C_{\min})$ ,  $f_i \in F$ };
    L:= |RCL|;
    pos:= truncar(U[0,1]*L)+1;
     $f_{\text{selec}}$ := RLC[pos];
    S:= S + { $f_{\text{selec}}$ };
  fmientras
falg
```

5.2.3 – Fase II o fase de mejora local

El objetivo de esta fase es mejorar la solución obtenida en la fase constructiva, obteniendo una solución derivada que tenga una mejor evaluación de la función objetivo. Las soluciones obtenidas en la fase constructiva no suelen ser óptimos locales, con lo que podemos intentar encontrar la solución del óptimo local más próximo a la solución constructiva. Debemos definir dos conceptos clave en esta fase:

- 1) **Entorno local:** Dada una solución S, podemos definir el entorno N(S) como el conjunto de soluciones alternativas “próximas” a la solución S inicial. Ese entorno queda definido por el criterio de elección de soluciones alternativas. Por ejemplo, podemos definir el entorno N(S) como el conjunto de soluciones que difieran de la solución S en un único elemento. Este procedimiento se conoce por método de intercambios simples, ya que se puede obtener la solución S', a partir de eliminar un elemento de S e introducir uno que no estuviera en la solución de partida.
- 2) **Criterio de avance:** En un entorno de búsqueda local, debemos movernos para poder llegar al deseado mínimo local. Para eso, deberemos elegir alguna de las soluciones del entorno local siguiendo algún criterio que nos aproxime a ese punto. El más usual es la evaluación de la función objetivo en las soluciones del entorno de búsqueda. Sea cual sea el método de generación del entorno, al

encontrar una solución alternativa, ésta se evalúa en la función objetivo del problema y si el resultado es mejor que el de la solución de partida, se toma la nueva solución como incumbente. A partir de los puntos de avance obtenidos, seguiremos calculando entornos de búsqueda y realizando movimientos, hasta llegar a un punto de equilibrio, en el que no podamos realizar ningún movimiento satisfactorio con las premisas aceptadas. Entonces, aceptaremos estar en un mínimo local.

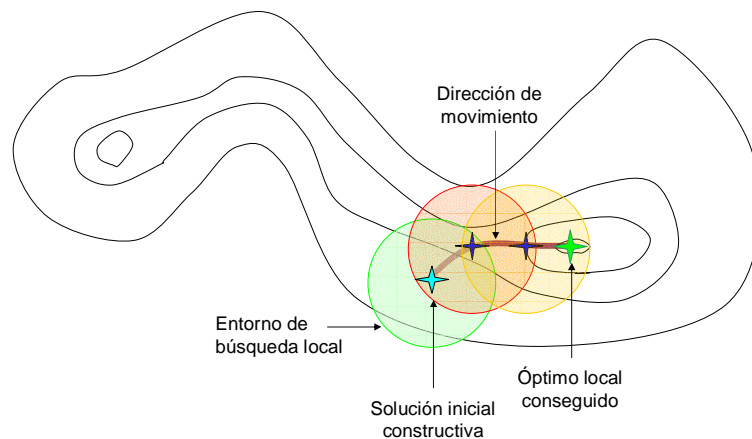


fig 5.4 – Avance en el entorno de búsqueda y llegada al óptimo local

El pseudocódigo de esta fase es el siguiente:

Algoritmo 5.3 – Fase de mejora

```

alg S' := FASE II (param P, param S, param metodo)
  var mejora='verdadero';
  var S'=S;
  mientras (mejora) hacer
    mejora:='falso';
    entorno:=N(S',P);
    S^ <- entorno(metodo);
    si C(S^)<C(S') hacer
      S':=S^;
      mejora:='verdadero';
    fsi
  fmientras
falg

```

Esta fase de mejora se realiza a partir de una solución S inicial y un conjunto de datos P del problema. Pero también hay que definir que método escoger para realizar la búsqueda de actualizaciones en el entorno vecino. Las estrategias más utilizadas son:

- **Best-improving**: una vez definido el entorno de exploración, valoramos todas las opciones posibles y seleccionamos la mejor de ellas.
- **First-improving**: al realizar la exploración y encontrar la primera de las soluciones alternativas que mejora el coste total, actualizamos la solución incumbente y procedemos a una nueva iteración de mejora.

La segunda estrategia permite obtener mejoras sin realizar una exploración exhaustiva y computacionalmente más costosa. Por otra parte, la primera permite realizar el mejor movimiento en el entorno local de búsqueda, con el precio añadido de una mayor complejidad algorítmica. Cabe decir que existe la posibilidad de prescindir de esta fase de mejora, debido al coste computacional que comporta y a la, a veces, poca mejoría que implica sobre la solución incumbente constructiva [6].

5.2.3 – Acción combinada de las dos fases

La combinación de una fase constructiva aleatoria que nos lleva a distintos puntos del espacio de soluciones factibles que pueden estar lejanos entre sí, y de una fase de mejora que traslada la solución a un óptimo local del entorno inmediato de la solución, permiten obtener un rastreo bastante amplio del espectro de soluciones factibles localmente óptimas. Esto nos da la posibilidad de obtener en una de estas iteraciones la solución óptima global. Al ser un procedimiento que memoriza la mejor solución obtenida, queda claro que si damos con la solución óptima del problema en alguna de las iteraciones intermedias, la recuperaremos al finalizar el algoritmo.

5.3 – Ajuste y sensibilidad de parámetros

Esta metaheurística tiene dos parámetros ajustables que hay que fijar previamente a cualquier ejecución: el parámetro de aleatorización α y el criterio de parada utilizado. En las siguientes líneas, se plantea la necesidad de estudiar la mejor combinación de

ambos para poder dar con resultados satisfactorios, tanto a nivel de optimización relativa como de computación.

5.3.1 – Análisis de sensibilidad del parámetro α

En el punto 4.2.2 vimos como este parámetro determina la composición de la lista restringida de candidatos de la fase constructiva, a partir de la definición de la RCL en los valores 0 y 1 de α . Si aumentamos desde 0 este valor, asumimos cada vez un mayor rango de los elementos elegibles en cada iteración. Como los elementos se ordenan según el criterio voraz definido, la selección aleatoria en una lista de poca extensión nos escogerá un elemento con buenas propiedades locales. Hay un cierto valor de α que nos proporcionará una amplitud de posibilidades lo suficientemente grande como para incluir soluciones lejanas buenas, y lo suficientemente concreta como para que llegemos a dar con esas soluciones en pocas iteraciones. Esto suele suceder en los valores de α comprendidos en el intervalo [0.4, 0.6].

A partir de esos valores, el incremento del parámetro hará que la lista de candidatos se haga demasiado grande y las posibles soluciones sean muchas y muy diversas. Eso nos amplía tanto el rango de exploración que las soluciones factibles obtenidas en cada iteración son muy diferentes, haciendo muy poco eficiente la aplicación iterativa. De todas maneras, al aumentar la cantidad de posibilidades de elección, aumentamos la probabilidad de dar con mejores soluciones, de tal manera que para un número muy grande de iteraciones (de hecho, para un valor teóricamente ilimitado), llegaríamos a dar con soluciones de menor coste.

Las figuras 5.5 y 5.6 ejemplifican lo dicho anteriormente: la primera muestra la teórica mejor solución incumbente obtenida para diferentes valores de α con un número de iteraciones muy grande; la segunda muestra un patrón clásico de evolución de la solución incumbente dados diferentes valores del parámetro α , en tantos experimentos del mismo problema, con un número máximo de iteraciones pequeño.

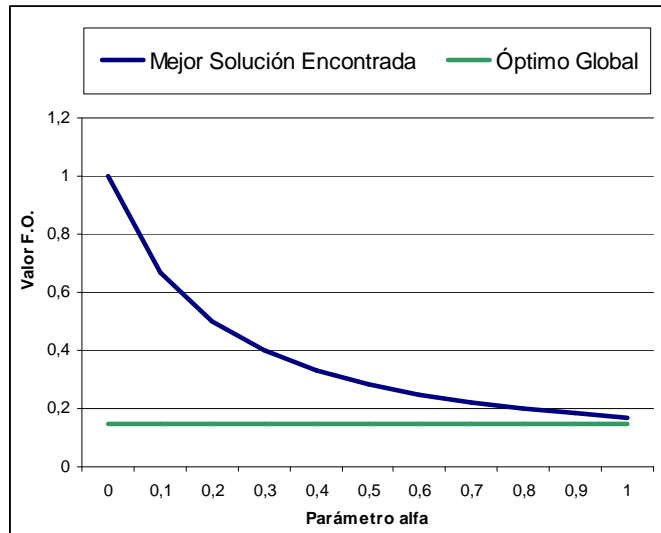


fig.5. 5 – F.O. vs. Alfa, límite iteraciones alto

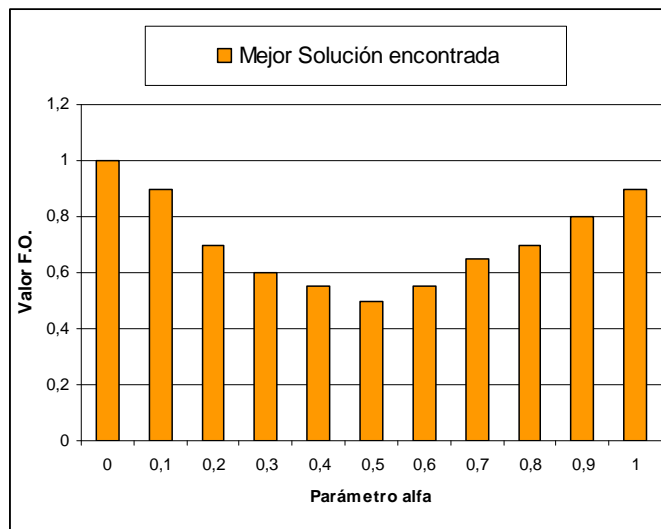


fig.5. 6 – F.O. vs. Alfa, límite iteraciones bajo

5.3.2 – Estudio del criterio de parada

La metaheurística GRASP finaliza cuando se cumple un cierto criterio de parada determinado. No es trivial decidir que criterio de parada utilizar, ya que nos condiciona la solución obtenida.

En la práctica, el criterio de parada nos limita el número de iteraciones realizadas. Un criterio que permita realizar más iteraciones nos conducirá hacia soluciones mejores, ya que cada iteración genera una solución diferente con coste diferente. Al generarse más soluciones, la probabilidad de obtener una solución mejor aumenta. No obstante, el

número de iteraciones necesarias para obtener una solución mejor que la anterior depende del instante de ejecución de la heurística. En las primeras iteraciones, se produce una mayor actualización y mejora de la solución incumbente, respecto a iteraciones posteriores. A medida que vamos realizando iteraciones y acumulamos más terreno explorado, cada vez se producen menos actualizaciones, que provocan una menor reducción relativa de coste. Así que, realizar un número muy grande de iteraciones nos puede provocar que hagamos buena parte de ellas (las últimas) sin provocar mejora efectiva. Y cada iteración cuesta lo mismo computacionalmente. Es por ello necesario encontrar un criterio de parada que permita realizar un número adecuado de iteraciones, con el que lleguemos a una solución muy buena y con un coste computacional aceptable.

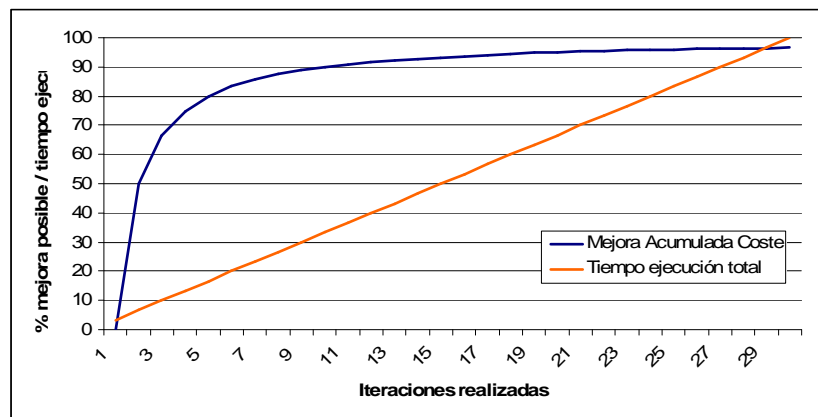


Fig. 5.7 – Tiempo de optimización invertido y mejor solución encontrada

Los dos criterios más utilizados son:

- A) **Número máximo de iteraciones sin actualización:** este criterio fija el número máximo de iteraciones que se van a realizar sin que se lleve a cabo ninguna actualización de la solución incumbente. Suponemos que, tras una realización significativa de iteraciones sin cambios, la solución incumbente será difícil de mejorar. Fijar este número máximo de iteraciones es una decisión complicada, que depende del tamaño y características del problema. La figura 5.8 muestra la evolución general del número de iteraciones sin mejora a lo largo de la aplicación de la heurística sobre un ejemplo concreto.

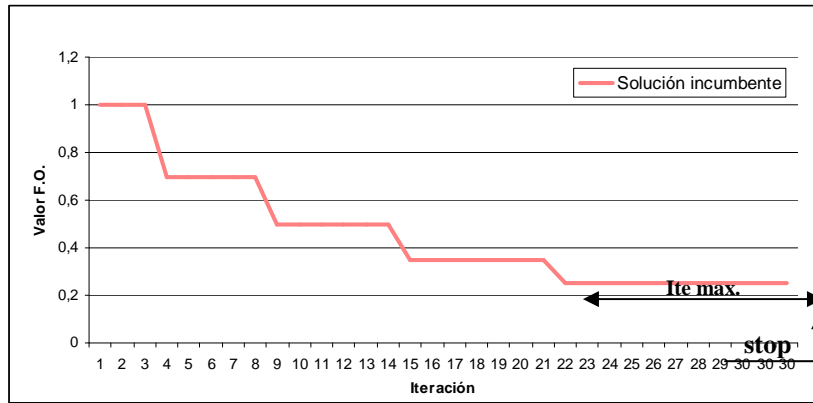


Fig. 5.8 – Evolución del número de iteraciones sin mejora

B) **Mínima pendiente de mejora:** si suavizamos la gráfica de la evolución de la solución incumbente, podemos obtener una aproximación y calcular la pendiente de descenso. La podemos medir entre el punto de la última solución incumbente obtenida y el punto de la iteración actual. Si este valor no supera una mínima pendiente exigible, la heurística GRASP finaliza. Este criterio tiene sentido ya que, en general, las primeras iteraciones producen modificaciones mayores que las siguientes, de manera que la pendiente de mejora de la función objetivo va disminuyendo a lo largo del proceso iterativo.

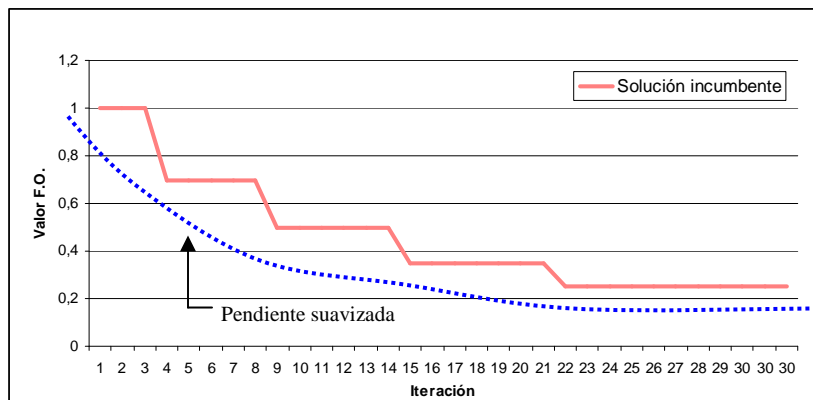


Fig. 5.9 – Suavizado de la evolución de la solución incumbente

5.4 – Adaptación de la heurística al problema

La adaptación de la meta-heurística general para el problema de adaptación de recursos ópticos se centra en los siguientes aspectos principales.

5.4.1 – Estructura de una solución factible

Al final del proceso de optimización, el resultado final debe ser un nuevo escenario de tráfico que cumpla con las condiciones de servicio y restricciones impuestas. Para llegar

a este resultado, se han tenido que producir una serie de cambios en los elementos de la estructura inicial de datos. La caracterización de esos elementos modificados va a ser el contenido de una solución factible a nuestro problema. Así pues, una solución factible va a estar formada por dos clases de elementos:

- A) **Arcos ópticos eliminados:** de la red de agregación eléctrica por tener ocupación nula.
- B) **Conexiones eléctricas modificadas:** por tener rutas originales con cortes en la ruta óptica.

5.4.2 – Criterio *greedy* de selección de arcos ópticos candidatos

Vamos a definir el criterio voraz para elegir los candidatos a ser eliminados. Nos interesa buscar los arcos ópticos que tengan mayor coste para poder obtener, con cada eliminación, un mayor beneficio en la función objetivo. Por otra parte, eliminar un arco con pocas conexiones eléctricas asociadas será más fácil, *a priori* que otro con más ocupación. Por ello, definiremos un criterio de ordenación jerárquico basado en dos valores: en primer lugar, se escogerán los arcos con menor número de conexiones eléctricas. Para los conjuntos de arcos con igual número de conexiones, y como segundo criterio de ordenación, se ordenarán de mayor a menor coste en recursos ópticos asociados.

5.4.3– Selección de arcos ópticos eliminables

Existirá una selección aleatoria que habrá que realizar para determinar que arco óptico es escogido para ser eliminado. Esta elección se hará después de crear la RCL correspondiente, a partir del coste asociado a cada arco. El parámetro que marca el grado de aleatorización de ésta selección se llamará α *óptico*.

5.4.4– Selección de rutas alternativas de conexiones eléctricas

Cuando se elige que arco óptico es el candidato para salir de la red, las conexiones presentes deben enrutarse por caminos alternativos que no contengan el arco óptico seleccionado. Estas rutas se ordenarán mediante el criterio voraz de menor a mayor coste asociado. La selección de la ruta se realizará entre un conjunto de rutas candidatas según el parámetro homólogo al α *óptico* de este punto, que llamaremos α *eléctrico*.

5.4.5 – Método de mejora

Una vez finalizada la fase constructiva aleatorizada, se procederá a una fase de mejora, en la que se aplicará un método de *first-improving*. Esto nos permite modificar la solución incumbente sin un excesivo coste computacional, un hecho que favorece la implementación y el funcionamiento en situaciones de tráfico real.

5.4.6 – Criterio de parada

Para decidir cuándo finaliza el proceso iterativo del GRASP, utilizaremos el criterio de fijar un número máximo de iteraciones sin mejora de la solución incumbente. Este número de iteraciones lo fijaremos en función de los resultados que se obtengan en las baterías de pruebas posteriores. Trataremos de encontrar para diferentes casos un valor que dé buenos resultados con un coste computacional aceptable. Los valores de prueba se van a referir a la dimensión óptica del problema, de tal manera que el máximo número de iteraciones será una fracción del máximo número de arcos ópticos posibles.

5.4.7 – Pseudocódigo de la fase constructiva adaptada

Tomando como base el pseudocódigo general de la fase I, la adaptación al problema de liberación de recursos ópticos queda así:

Algoritmo 5.4 – Fase de constructiva adaptada

```
alg S := FASE I (param P, param  $\alpha$ optico, param  $\alpha$ eléctrico )
  var S= $\emptyset$ ; // Conjunto de arcos eliminados
  var ite=1;
  mientras (RCL  $\neq \emptyset$  || ite=1) hacer
    F:= conjunto de arcos ópticos eliminables
    RCL:= { $f_i$  |  $C(f_i) \leq C_{fmin} + \alpha_{optico}(C_{fmax} - C_{fmin})$ ,  $f_i \in F$ };
    L:= |RCL|;
    pos:= truncar(U[0,1]*L)+1;
     $f_{selec}$ := RLC[pos];
    eliminararco:='verdadero';
    G := conjunto de c. eléctricas del arco  $f_{selec}$ 
    mientras (G  $\neq \emptyset$ ) hacer
      g := elemento de G
      R:= conjunto de rutas alternativas de g
      RCL:= { $r_i$  |  $C(r_i) \leq C_{rmin} + \alpha_{electrico}(C_{rmax} - C_{rmin})$ ,  $r_i \in R$ };
```

```

L:= |RCL|;
pos:= truncar(U[0,1]*L)+1;
r_selec:= RLC[pos];
si (r_selec es factible) entonces
    g(ruta):= r_selec;
    eliminar g del conjunto G;
si no eliminar r_selec de RCL y volver a
    seleccionar otra ruta;
fsi
si (g no tiene ruta factible) entonces
    eliminararco='false';
    romper bucle mientras;
fsi
fmientras
si eliminararco='verdadero' entonces
    S:= S + {f_selec};
fsi
    F:=F-{fselec};
fmientras
falg

```

En primer lugar existe una selección aleatoria del arco a eliminar, y después para cada conexión eléctrica afectada, hay una selección de la ruta eléctrica alternativa. Si la ruta elegida no es factible (no hay suficiente capacidad en los arcos ópticos como para albergar el ancho de banda durante todo el trayecto), entonces se busca por el mismo criterio aleatorio una opción diferente. Si una conexión no tiene ninguna ruta factible alternativa, esa conexión no puede ser servida si se elimina ese arco óptico, con lo cual se anulará la eliminación de ese arco.

5.4.7 – Pseudocódigo de la fase de mejora adaptada

La fase de mejora se define de la siguiente manera:

Algoritmo 5.5 – Fase de mejora adaptada

```

alg S' := FASE II (param P, param S, param metodo)
    I:= conjunto de arcos que no están en la red;
    O:= conjunto de arcos de la red
    Para cada (i ∈ I)
        Para cada (o ∈ O) tal que (C(o)>C(i))
            añadir arco i a la red;

```

```

C:=conjunto de conexiones del arco o;
Encontrar rutas factibles alternativas al arco o
para cada elemento de C;
Si (todas rutas alternativas factibles) entonces
    eliminar arco o de la red;
    pasar o al conjunto I;
    pasar i al conjunto O;
    Mover rutas de C a alternativas factibles
    Encontradas;
Si no eliminar arco i de la red;
fsi
fPara
fPara
falg

```

Esta fase intenta hacer intercambios entre un arco óptico que no está en la red y un arco de la red que tenga un coste superior. Si existen rutas alternativas que no pasen por el arco saliente sobre la red ampliada con el arco entrante, se intercambian estos dos arcos: el que estaba en la red sale fuera y el que estaba fuera pasa a forma parte de la estructura. Aunque las rutas eléctricas hayan cambiado y probablemente se haya incrementado el coste total de éstas, el intercambio provoca un beneficio en el coste óptico, que es el más influyente en la evaluación de la función objetivo.

5.5 – Definición de versiones

Con un ajuste de parámetros distinto, podemos conseguir resultados diferentes de optimización sobre los mismos ejemplos numéricos. A continuación se presentan versiones de esta heurística que se diferencian entre sí exclusivamente en los valores de los parámetros característicos del GRASP adaptado. Cada una de estas versiones será probada y analizada a partir de los ejemplos de las redes de prueba y europea en el próximo capítulo.

5.5.1 – Greedy Básico No Aleatorizado

Esta versión puede considerarse como la más elemental de todas. La combinación de parámetros que define esta heurística es la siguiente:

Tabla 5.1 – Parámetros *greedy* básico

Parámetro	Valor
Alfa óptico	0
Alfa eléctrico	0
Nº iteraciones parada	1

Es la versión sin ningún tipo de aleatorización. La aplicación para un mismo ejemplo da siempre la misma solución, puesto que no existe variabilidad en la selección *greedy*. Por eso, solo se realiza una vez, con lo que el número de iteraciones permitidas sin mejora de solución debe ser sólo 1 (la primera iteración).

Esta versión sirve para obtener resultados básicos sobre el efecto del criterio de selección *greedy*.

5.5.2 – GRASP óptico

Esta versión presenta los siguientes parámetros:

Tabla 5.2 – Parámetros GRASP óptico

Parámetro	Valor
Alfa óptico	(0,1]
Alfa eléctrico	0
Nº iteraciones parada	>1

Aquí hay una selección aleatoria del arco óptico a eliminar, pero no hay selección aleatoria de ruta alternativa eléctrica. Si la ruta alternativa de menor coste de una conexión no tiene capacidad factible, esta conexión no se va a mover y, por ello, el arco óptico seleccionado se va a mantener en la red. Obviamente, el número de iteraciones debe ser mayor que 1 para dar sentido al proceso iterativo, ya que ahora sí que hay variabilidad en las soluciones obtenidas en cada vuelta.

Esta versión sirve para hacer análisis de sensibilidad sobre el parámetro *α* óptico.

5.5.3 – GRASP eléctrico

Esta versión presenta los siguientes parámetros:

Tabla 5.3 – Parámetros GRASP eléctrico

Parámetro	Valor
Alfa óptico	0
Alfa eléctrico	(0,1]
Nº iteraciones parada	>1

Aquí se seleccionaran los arcos ópticos siguiendo el criterio *greedy* estricto de coste, pero la ruta alternativa de cada conexión eléctrica se escogerá siguiendo un patrón aleatorizado.

Esta versión sirve para sacar conclusiones respecto a los valores del parámetro α *eléctrico*.

5.5.4 – GRASP doble

Esta versión es la más completa y más potente a nivel de encontrar soluciones en el entorno de exploración:

Tabla 5.4 – Parámetros GRASP doble

Parámetro	Valor
Alfa óptico	(0,1]
Alfa eléctrico	(0,1]
Nº iteraciones parada	>1

Ahora existe variabilidad tanto en las elecciones ópticas como en las eléctricas, así que nos encontramos con un procedimiento que explora un terreno más amplio, aumentando así la posibilidad de encontrar más soluciones.

Esta versión puede partir de los valores óptimos de los parámetros de aleatorización óptico y eléctrico encontrados a partir de las versiones anteriores, y estudiar la interacción entre ellos, para encontrar la mejor combinación en cuanto a calidad de la solución obtenida. También se puede realizar un estudio sobre el número de iteraciones de la condición de parada, para fijarlo en un valor que consiga un buen equilibrio entre calidad de solución y tiempo invertido.

Capítulo 6

Resolución de las versiones heurísticas

En este capítulo se aplican las versiones heurísticas definidas anteriormente sobre los ejemplos simulados. Para cada una de ellas, se realiza la aplicación sobre los escenarios de tráfico de la red de prueba, con la intención de estudiar el efecto de cada algoritmo y encontrar la mejor configuración de parámetros. Con las conclusiones extraídas en cada conjunto de pruebas, se aplicará la resolución sobre los casos de la red europea.

6.1 – Heurística Básica No Aleatorizada

6.1.1 – Diseño de pruebas

Esta versión no tiene ningún parámetro ajustable, con lo cual se va a realizar una única aplicación para cada una de las instancias de cada ejemplo. Los resultados se presentarán para las agrupaciones de instancias utilizadas en el capítulo de resolución exacta de los problemas: por porcentajes de ocupación de los arcos ópticos, y por intensidad de tráfico eléctrico circulante en número de conexiones eléctricas establecidas.

6.1.2 – Resultados en red de prueba

La siguiente tabla muestra el resumen de resultados obtenido en las dos variables más significativas para el análisis:

Tabla 6.1 – Resultados Greedy prueba

% ocupación	Tiempo medio	Valor F.O. medio
20	0,18	-15961,46
30	0,38	-19255,92
40	0,39	-20246,55
50	0,37	-22884,86
60	0,56	-18875,71
70	0,60	-10880,54
80	0,60	-8440,29
90	0,44	-3452,36

Intensidad tráfico	Tiempo medio	Valor F.O. medio
Baja	0,31	-17212,35
Media	0,51	-15692,51
Alta	0,59	-8236,42

Estos valores se pueden comparar con los obtenidos para la solución de referencia del modelo exacto. El porcentaje de diferencia para las dos variables en la comparación de procedimientos se refleja en la siguiente tabla.

Tabla 6.2 – Comparación Greedy/CPLEX

% (CPLEX – GREEDY) / CPLEX		
% ocupación	Tiempo medio	Valor F.O. medio
20	-1029%	6%
30	-111%	20%
40	10%	9%
50	41%	10%
60	52%	21%
70	81%	31%
80	73%	33%
90	59%	73%
Intensidad tráfico	Tiempo medio	Valor F.O. medio
Baja	-126%	16%
Media	63%	20%
Alta	75%	42%

La representación gráfica del tiempo medio invertido por intensidad de tráfico nos muestra el incremento del valor de esta variable para problemas de mayor dimensión.

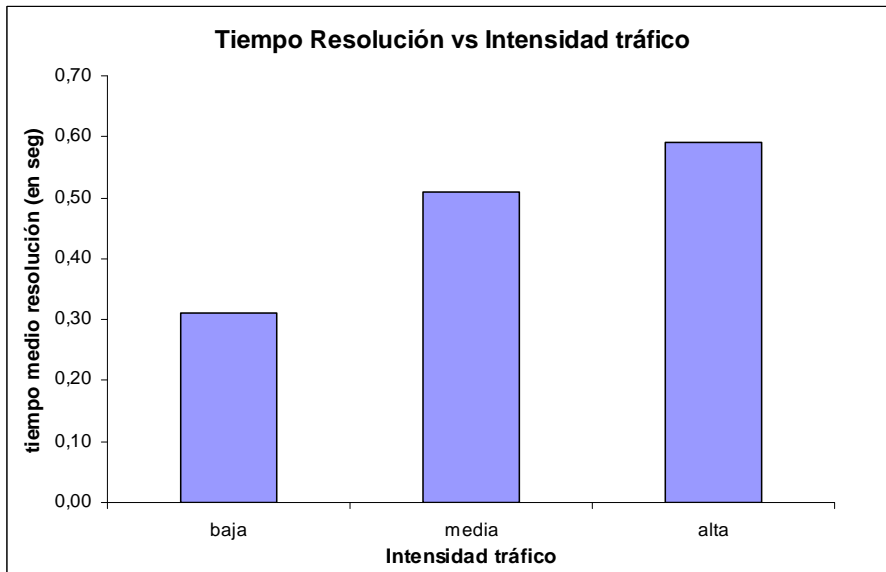


Fig 6. 1 – Tiempo de resolución para diferentes intensidades de tráfico

Las gráficas siguientes muestran el porcentaje de arcos ópticos eliminados y conexiones eléctricas afectadas por dichas eliminaciones

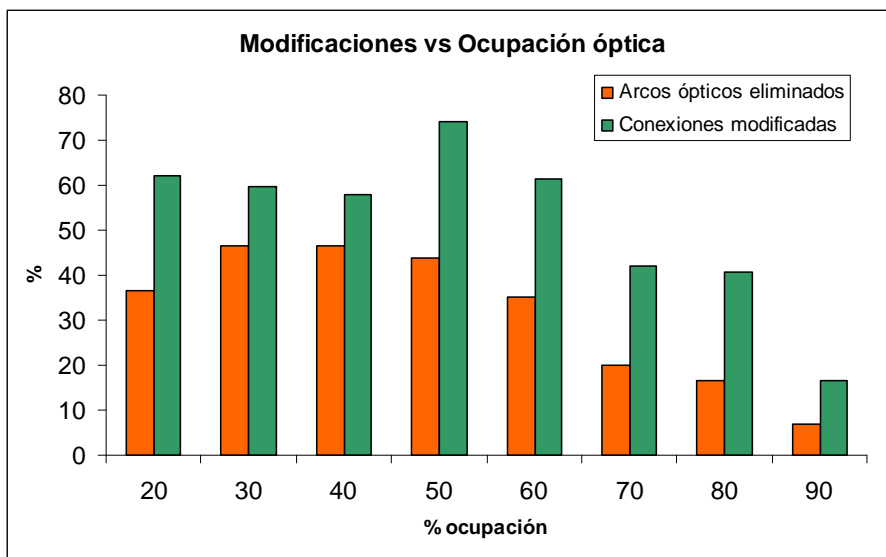


Fig. 6 2 – Movimientos realizados por diferentes ocupaciones

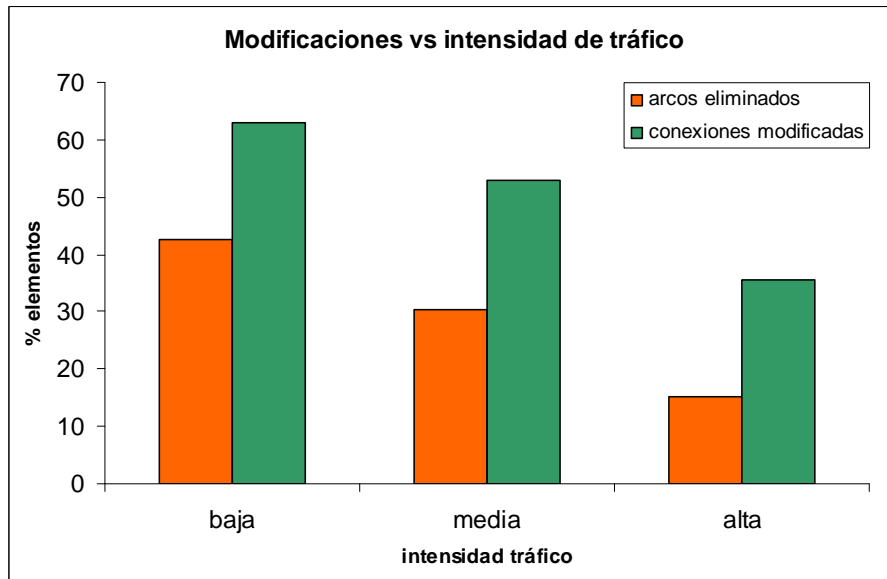


Fig. 6.3 – Movimientos realizados en diferentes intensidades

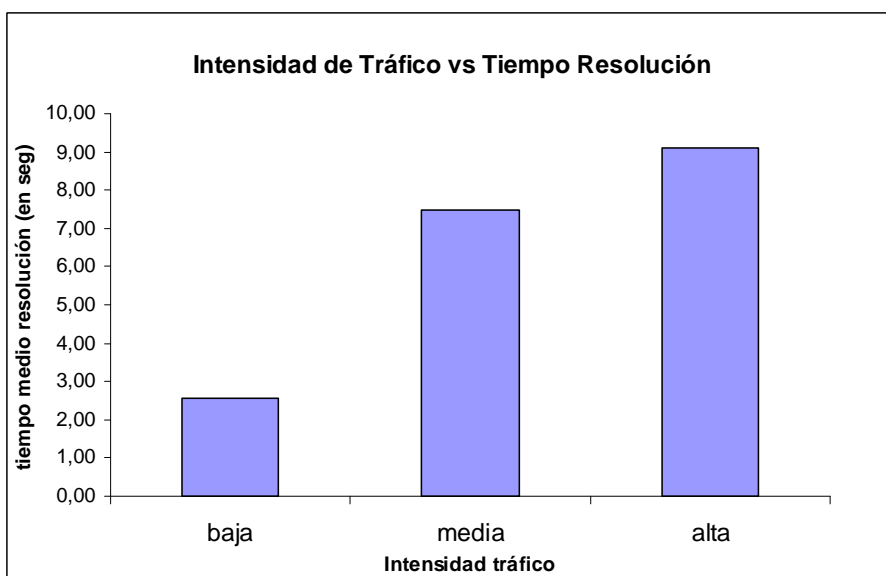
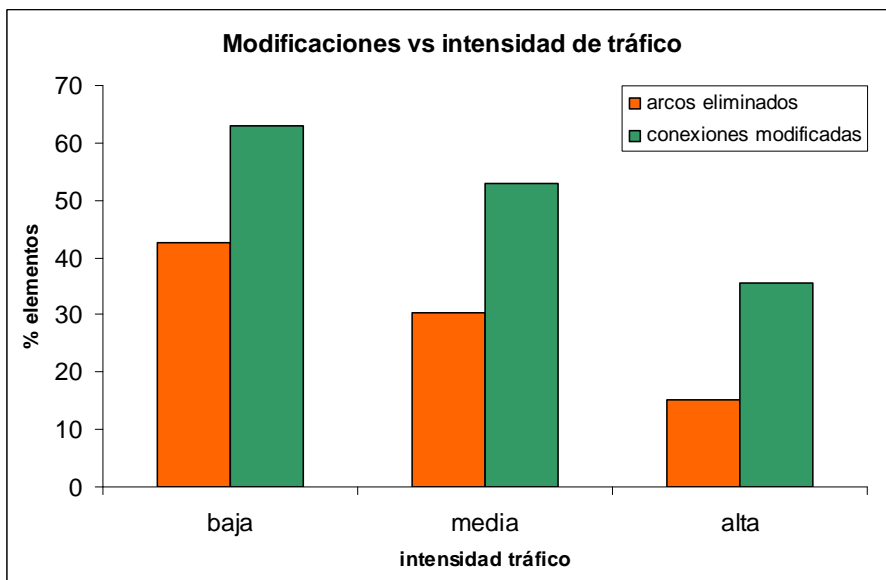
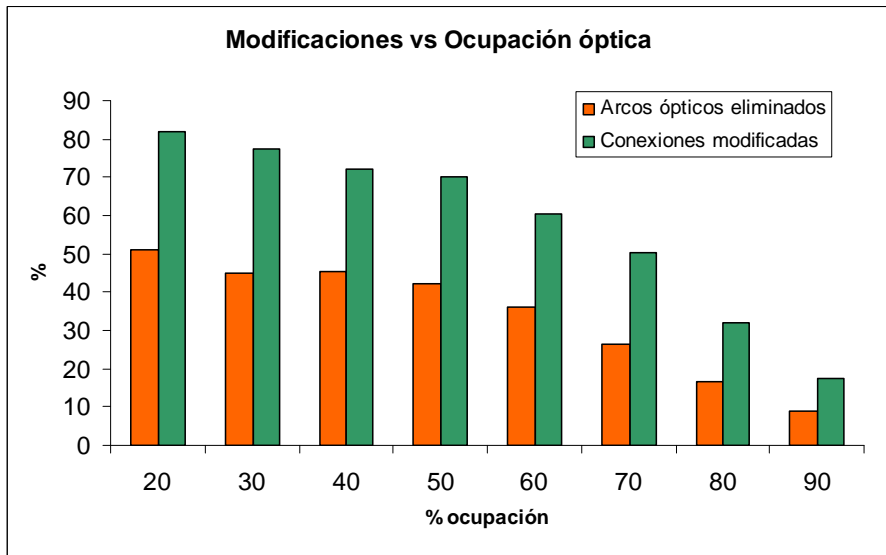
6.1.3 – Resultados en red europea

El resumen de resultados de esta batería de pruebas se presenta en la siguiente tabla:

Tabla 6.3 – Resultados Greedy Europea

% ocupación	Tiempo medio	Valor F.O. medio
20	1,81	-30932
30	3,34	-31469
40	5,05	-41025
50	8,40	-50872
60	8,97	-49910
70	10,02	-42739
80	9,15	-31223
90	8,18	-19496
Intensidad tráfico	Tiempo medio	Valor F.O. medio
baja	2,57	-79390
media	7,47	-39943
alta	9,12	-26078

Gráficamente, los elementos modificados en la solución incumbente se reflejan en las figuras 6.4 y 6.5. La evolución del tiempo de resolución por intensidad de tráfico puede verse en la figura 6.6.



figuras 6.4, 6.5 y 6.6– Evolución de los elementos modificados y del tiempo de resolución

6.1.4 – Análisis de resultados

Comparando los resultados de ambos ejemplos, podemos observar resultados similares, con lo que las conclusiones extraídas pueden ser aplicadas a cualquiera de las situaciones. Como hechos más destacados podemos decir lo siguiente:

- A) La heurística básica permite obtener una reducción de arcos ópticos entre un 15% y un 50%. Esta reducción produce una liberación de recursos ópticos de similar magnitud, hecho que valida el criterio voraz utilizado.
- B) El tiempo de ejecución del algoritmo crece de forma prácticamente lineal a medida que aumenta la complejidad del problema.
- C) La distancia entre la solución óptima y la solución obtenida por la heurística básica aumenta a medida que crece la complejidad del escenario. La diferencia media para los escenarios de la red de prueba se sitúa entorno al 22%.
- D) El tiempo invertido por el algoritmo heurístico respecto al de la solución exacta disminuye a medida que aumenta la complejidad del problema. El tiempo medio invertido por la heurística reduce en media un 60% el tiempo utilizado por el motor CPLEX.

Con todo esto podemos decir que el algoritmo básico aporta una mejora evidente que, en ocasiones, se queda bastante lejos de la solución óptima, pero en cambio tiene una evolución del tiempo de ejecución más favorable que la resolución exacta.

6.2 – GRASP óptico

6.2.1 – Diseño de pruebas

Esta versión presenta dos parámetros ajustables, cuyas combinaciones ejecutadas han sido:

Parámetro alfa	0,1 ; 0,3 ; 0,5 ; 0,7 ; 0,9
Límite de iteraciones	12 (\approx 50% arcos ópticos)

Para cada una de las 5 combinaciones resultantes, se han resuelto las instancias de prueba. Al mantener estable el número de iteraciones, podremos estudiar el impacto

neto del parámetro de aleatorización. Del análisis de sensibilidad realizado, se ha escogido la combinación con mejor resultado global respecto a la función objetivo y se ha aplicado esa configuración para la resolución del caso europeo.

6.2.2 – Resultados en red de prueba

A continuación se presenta la tabla de resumen de resultados de beneficio obtenido y tiempo invertido por esta versión heurística, en contraste con los valores obtenidos por el algoritmo sin aleatorización. En este aspecto, se presenta la diferencia porcentual respecto a la heurística básica.

Tabla 6.4 – Resultados GRASP óptico prueba

ocupación	alfa 0,1				alfa 0,3			
	F.O.	cmp FO	TMP	cmp TMP	F.O.	cmp FO	TMP	cmp TMP
20	-31922,9	0%	2,2	-529%	-33950,9	-6%	3,0	-757%
30	-69804,8	-21%	6,4	-462%	-71864,8	-24%	6,1	-432%
40	-60765,7	0%	7,8	-563%	-66870,7	-10%	7,6	-541%
50	-73646,6	-7%	8,1	-632%	-76639,6	-12%	10,6	-859%
60	-162984,6	-8%	31,6	-608%	-177108,7	-17%	39,8	-790%
70	-121938,0	-25%	48,2	-787%	-134108,0	-37%	45,1	-731%
80	-156146,6	-16%	64,5	-571%	-169426,6	-25%	76,8	-698%
90	-20714,1	0%	17,0	-544%	-25765,2	-24%	18,4	-597%
intensidad								
Baja	-119642,6	1%	10,7	-391%	-123746,6	-3%	11,0	-409%
Media	-330884,8	-5%	63,7	-527%	-357189,8	-14%	82,7	-713%
Alta	-247395,9	-31%	111,5	-720%	-274797,9	-45%	113,6	-735%
suma	-697923,3	-12%	185,9	-617%	-755734,3	-21%	207,4	-699%

ocupación	alfa 0,5				alfa 0,7			
	F.O.	cmp FO	TMP	cmp TMP	F.O.	cmp FO	TMP	cmp TMP
20	-33950,9	-6%	3,4	-874%	-33939,9	-6%	3,3	-834%
30	-71844,8	-24%	6,9	-508%	-71853,8	-24%	8,0	-604%
40	-66848,7	-10%	9,9	-741%	-66856,7	-10%	11,1	-843%
50	-76630,6	-12%	12,2	-1002%	-74650,6	-9%	9,4	-743%
60	-177168,6	-17%	53,4	-1094%	-178063,6	-18%	42,8	-857%
70	-134149,9	-37%	45,8	-744%	-134104,9	-37%	40,7	-649%
80	-166558,6	-23%	74,4	-674%	-166582,6	-23%	82,4	-756%
90	-25765,2	-24%	17,4	-558%	-25765,2	-24%	17,4	-559%
intensidad								
Baja	-123717,6	-3%	13,0	-497%	-123730,6	-3%	13,7	-529%
Media	-357241,8	-14%	89,7	-782%	-356137,8	-13%	83,3	-719%
Alta	-271957,9	-44%	120,8	-788%	-271948,9	-44%	118,1	-768%
suma	-752917,2	-21%	223,4	-761%	-751817,2	-21%	215,0	-729%

alfa 0,9				
ocupación	F.O.	cmp FO	TMP	cmp TMP
20	-33952,9	-6%	3,8	-989%
30	-71854,8	-24%	6,7	-488%
40	-66856,7	-10%	9,5	-708%
50	-71708,6	-4%	7,9	-607%
60	-176137,6	-17%	37,7	-742%
70	-134074,9	-37%	43,7	-706%
80	-169507,6	-26%	75,9	-689%
90	-25765,2	-24%	17,0	-542%
intensidad				
Baja	-123744,6	-3%	13,6	-526%
Media	-351292,8	-12%	73,8	-626%
Alta	-274820,9	-45%	114,8	-744%
suma	-749858,3	-20%	202,1	-679%

Para determinar que parámetro es el más apropiado, compararemos los resultados anteriores siguiendo un doble criterio: calidad de la solución obtenida y tiempo de ejecución del algoritmo. Nos interesará escoger el que presente una mejor valoración conjunta. Clasificaremos las instancias según un valor relativo para cada uno de los criterios, y obtendremos el producto de ambos como factor de decisión. Este criterio lo hemos escogido arbitrariamente. La gráfica 6.7 muestra los resultados obtenidos.

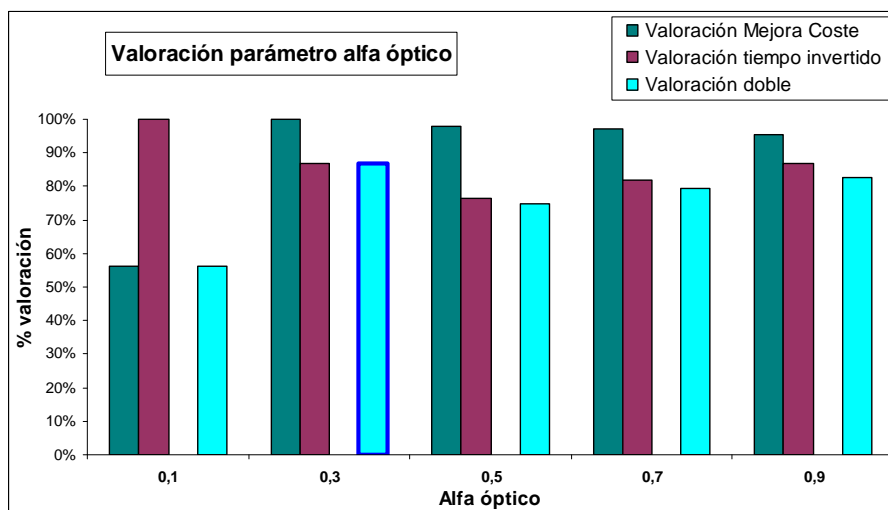


Fig. 6.7 – Selección del mejor parámetro alfa óptico

Podemos observar como un buen valor del parámetro de aleatorización de selección de arcos ópticos ronda el 30%. Tomaremos este valor como referencia para resolver las instancias de la red europea.

6.2.3 – Resultados en red europea

La configuración de la heurística para el caso europeo es la siguiente:

Parámetro alfa óptico	0,3
Límite de iteraciones	10 (\approx 10% arcos ópticos)

El límite de iteraciones es proporcionalmente inferior al realizado en el ejemplo de prueba. En una sección posterior analizaremos el efecto de éste parámetro. Por el momento, escogemos este valor para no incrementar en exceso el tiempo de ejecución de las pruebas.

Los resultados de esta batería de pruebas con el ajuste del parámetro $\alpha=0.3$, da el siguiente resumen de resultados.

Tabla 6.5 – Resultados GRASP óptico europea

GRASP OPT 0.3			% (Greedy – Grasp) / Greedy	
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio
20	10,19	-29944	-463%	3%
30	23,12	-39246	-592%	-25%
40	39,69	-49728	-686%	-21%
50	86,67	-62427	-932%	-23%
60	79,59	-60155	-787%	-21%
70	81,03	-50049	-709%	-17%
80	73,71	-35839	-706%	-15%
90	54,37	-20968	-565%	-8%
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio
baja	16,65	-96227	-547%	-21%
media	68,65	-48269	-819%	-21%
alta	69,70	-29556	-665%	-13%
Media			-716%	-17%

Las gráficas representadas en las siguientes figuras muestran la evolución del tiempo de ejecución y de los elementos modificados por la solución.

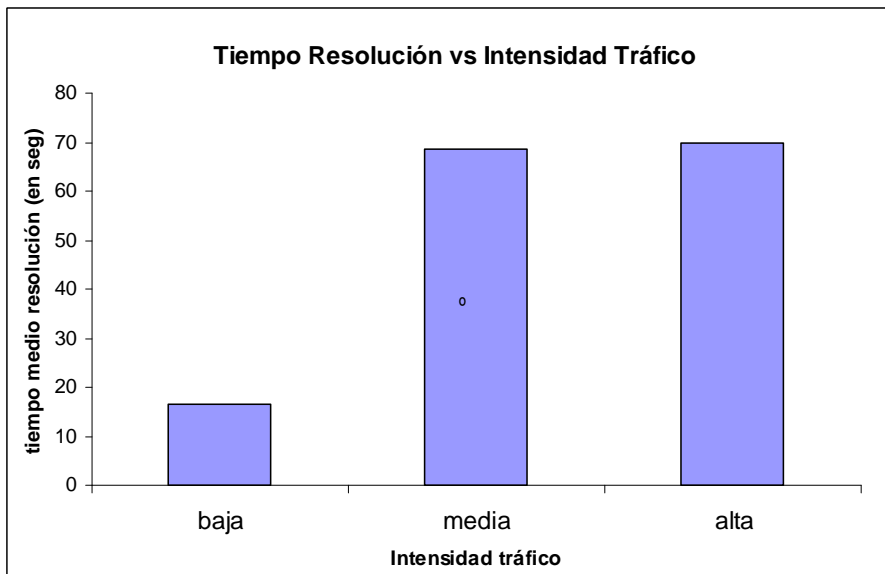
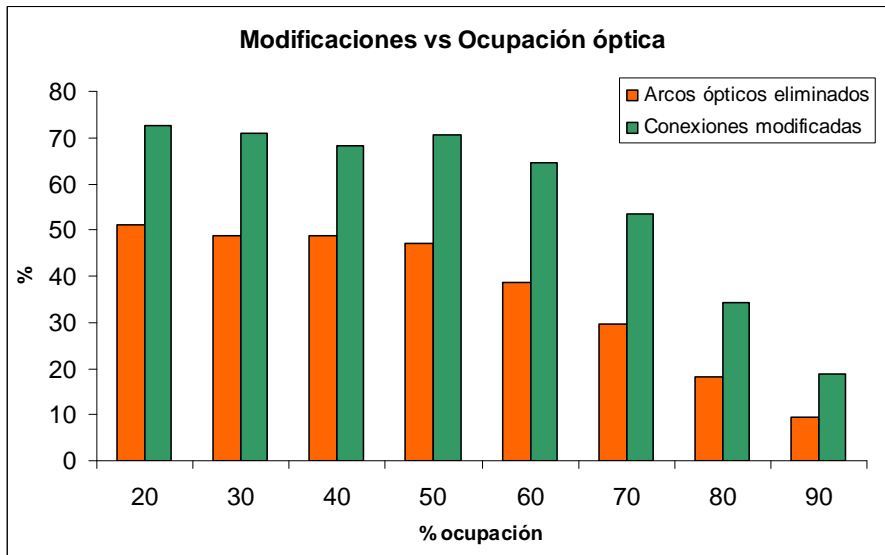


Fig. 6.8 y 6.9 – Características de las soluciones del GRASP óptico

6.2.4 – Análisis de resultados

Podemos contrastar como la inclusión de un efecto aleatorio permite mejorar claramente el resultado de la función objetivo. A partir de un valor del parámetro mayor o igual que 0.3 la mejora de coste frente a la versión greedy básica se sitúa entorno a un 20%. Para el ejemplo de la red europea, llegamos a obtener mejoras ligeramente superiores a este valor, situando la media de mejora respecto a la heurística de referencia en un 17%.

Por otra parte, al aumentar la complejidad algorítmica y al tratarse de un procedimiento que realiza varias iteraciones, los tiempos de ejecución son sustancialmente mayores

que en el caso de referencia, llegando a niveles de incremento superiores al 700% respecto al nivel de base.

El patrón de evolución de la solución obtenida es bastante similar en cuanto a tendencia al caso de la heurística básica: la cantidad de elementos modificados decrece y se separa de la solución exacta a medida que incrementa la complejidad del ejemplo, mientras que el tiempo de ejecución aumenta de forma bastante lineal (incluso vemos en el caso europeo un cierto estancamiento).

Aunque el tiempo de resolución aumente con este procedimiento aleatorizado, conviene incorporarlo al algoritmo básico para obtener soluciones con una evaluación en la función de coste significativamente mejor.

6.3 – GRASP eléctrico

6.3.1 – Diseño de pruebas

En este apartado realizaremos la misma batería de pruebas que hemos descrito para el caso anterior. No obstante, aquí haremos variaciones respecto al parámetro de aleatorización de selección de ruta eléctrica, el llamado parámetro α eléctrico. Se tomarán los mismos valores que para el experimento anterior:

Parámetro alfa	0,1 ; 0,3 ; 0,5 ; 0,7 ; 0,9
Límite de iteraciones	12 (\approx 50% arcos ópticos)

Como en el caso anterior, la mejor configuración servirá para resolver el ejemplo de red europea.

6.3.2 – Resultados en red de prueba

La tabla de resumen de resultados obtenidos comparados con el *greedy* básico se puede consultar a continuación.

Tabla 6.6 – Resultados GRASP eléctrico prueba

ocupación	alfa 0,1				alfa 0,3			
	F.O.	cmp FO	TMP	cmp TMP	F.O.	cmp FO	TMP	cmp TMP
20	-31922,9	0%	2,5	-603%	-31927,9	0%	2,4	-591%
30	-57767,8	0%	6,9	-504%	-67775,8	-17%	8,5	-643%
40	-60739,7	0%	7,6	-545%	-63784,7	-5%	9,3	-684%
50	-68658,6	0%	7,7	-591%	-73651,6	-7%	18,9	-1602%
60	-165052,7	-9%	42,0	-840%	-181041,7	-20%	57,5	-1185%
70	-108860,0	-11%	38,0	-599%	-129732,9	-32%	59,9	-1002%
80	-156143,7	-16%	73,3	-662%	-163108,6	-21%	87,1	-806%
90	-20714,1	0%	17,6	-568%	-25711,1	-24%	17,6	-566%
intensidad								
Baja	-107603,6	11%	11,4	-425%	-117616,6	2%	13,1	-502%
Media	-323037,9	-3%	75,6	-644%	-360007,9	-15%	109,4	-975%
Alta	-239217,9	-26%	108,6	-698%	-259109,8	-37%	138,6	-919%
suma	-669859,4	-7%	195,6	-654%	-736734,3	-18%	261,0	-906%

ocupación	alfa 0,5				alfa 0,7			
	F.O.	cmp FO	TMP	cmp TMP	F.O.	cmp FO	TMP	cmp TMP
20	-32932,9	-3%	2,9	-714%	-32932,9	-3%	2,8	-709%
30	-68791,8	-19%	8,9	-683%	-71816,8	-24%	7,2	-532%
40	-63813,7	-5%	11,8	-896%	-63770,7	-5%	11,6	-880%
50	-75637,6	-10%	11,6	-941%	-75643,6	-10%	9,3	-740%
60	-189059,7	-25%	60,7	-1257%	-187017,6	-24%	66,5	-1388%
70	-128760,9	-31%	54,4	-902%	-125865,0	-29%	54,5	-904%
80	-158092,6	-17%	94,3	-880%	-162068,5	-20%	106,4	-1006%
90	-25714,1	-24%	17,4	-561%	-25713,1	-24%	18,0	-582%
intensidad								
Baja	-119645,6	1%	15,1	-594%	-122676,6	-2%	12,7	-487%
Media	-364015,9	-16%	103,2	-914%	-367923,8	-17%	116,4	-1044%
Alta	-259141,9	-37%	143,7	-956%	-254227,8	-34%	147,2	-983%
suma	-742803,3	-19%	261,9	-910%	-744828,2	-19%	276,4	-965%

ocupación	alfa 0,9			
	F.O.	cmp FO	TMP	cmp TMP
20	-32932,9	-3%	2,6	-643%
30	-71829,8	-24%	7,1	-526%
40	-63752,7	-5%	11,7	-889%
50	-75638,6	-10%	12,0	-977%
60	-186026,6	-23%	70,5	-1478%
70	-122853,9	-25%	59,1	-989%
80	-154170,6	-14%	91,5	-851%
90	-25713,1	-24%	18,2	-589%
intensidad				
Baja	-122675,6	-2%	12,7	-484%
Media	-360918,8	-15%	119,4	-1074%

Alta	-249323,8	-32%	140,6	-934%
suma	-732918,2	-17%	272,7	-951%

Si aplicamos el mismo criterio de selección doble del caso óptico, dónde valorábamos conjuntamente calidad relativa de la solución y coste computacional, obtenemos la siguiente gráfica para obtener el más adecuado valor del parámetro.

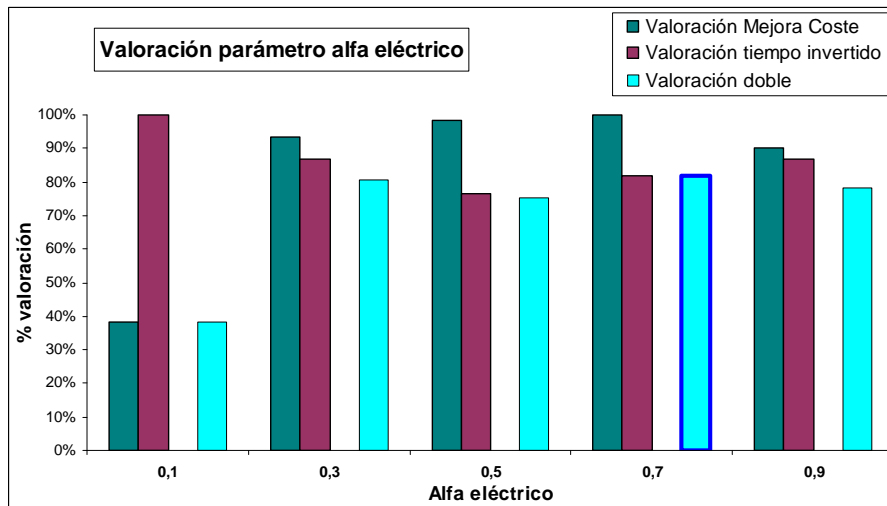


Fig. 6.10 – Selección del mejor alfa eléctrico

Ahora conseguimos un máximo de valoración en el valor del 70% del parámetro, aunque podemos encontrar un resultado alternativo en el 30%. Podríamos pues, a la vista de la gráfica, concluir que los valores entorno al 50% se pueden considerar buenos para éste caso.

6.3.3 – Resultados en red europea

Para esta prueba, los parámetros ajustables toman estos valores:

Parámetro alfa eléctrico	0,5
Límite de iteraciones	10 (\approx 10% arcos ópticos)

Del párrafo anterior, escogemos el valor de $\alpha=0.5$ como referencia para la resolución del caso europeo. El resumen de resultados obtenido es el que sigue.

Tabla 6.7 – Resultados GRASP eléctrico europea

GRASP ELEC 0.5			% (Greedy – Grasp) / Greedy	
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio
20	17,91	-30934	-890%	0%
30	27,21	-36097	-715%	-15%
40	40,22	-46766	-696%	-14%
50	96,47	-59005	-1049%	-16%
60	77,71	-56949	-766%	-14%
70	88,64	-47386	-785%	-11%
80	85,86	-35806	-838%	-15%
90	66,70	-21460	-715%	-10%
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio
baja	22,56	-90607	-776%	-14%
media	71,47	-45898	-856%	-15%
alta	80,40	-29125	-782%	-12%
Media			-812%	-12%

Gráficamente, los cambios respecto a las soluciones originales y los tiempos medios de ejecución para esta aplicación, evolucionan tal y como muestran las figuras 6.11 y 6.12.

6.3.4 – Análisis de resultados

Esta versión heurística presenta valores similares al GRASP óptico, mejorando en media un 12% el valor de la función objetivo. Aunque este valor es 5 puntos menor que en la heurística predecesora, la mejora por grupos de ocupación e intensidad eléctrica es más uniforme (entre el 10% y el 15% en la mayoría de los casos).

El tiempo de ejecución también se dispara, llegando a ser 10 veces mayor que el necesario para ejecutar el sencillo *greedy*. Aunque los valores son algo más elevados que en el GRASP óptico, estos se pueden considerar similares entre sí.

Dado que los resultados son significativamente mejores que los de la heurística de referencia, podemos aceptar esta versión aleatorizada y considerarla tan válida como la anterior.

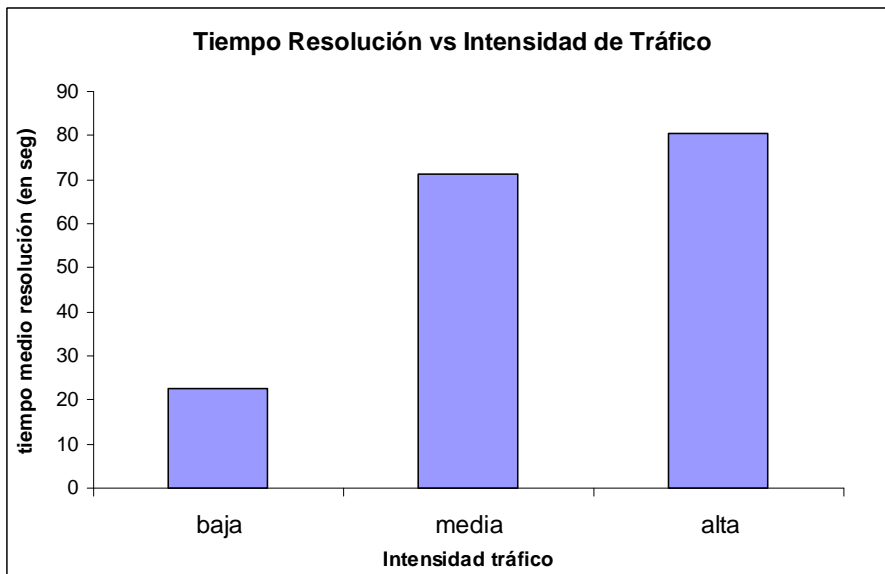
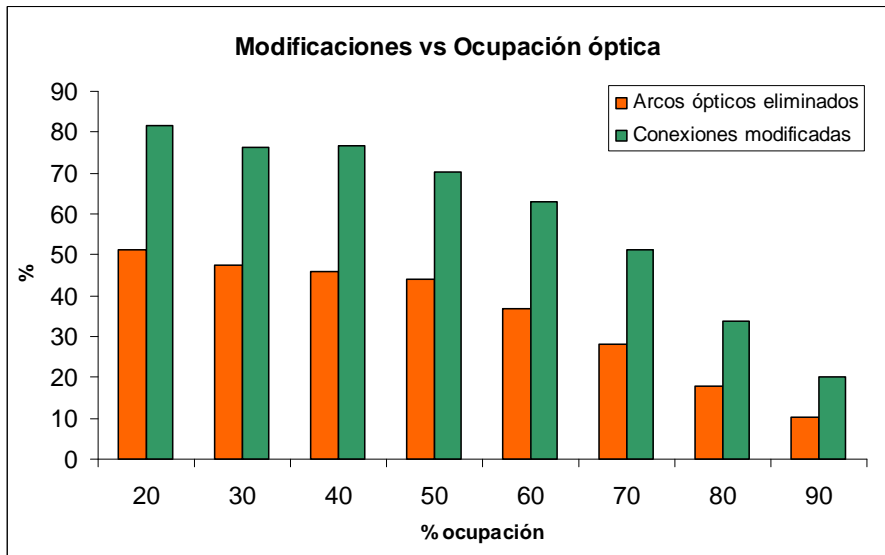


Fig.6.11 y6. 12 – Características de las resoluciones del GRASP eléctrico

6.4 – GRASP doble

6.4.1 – Diseño de pruebas

En esta nueva aplicación, tomaremos los mejores valores obtenidos de forma independiente en los dos experimentos anteriores y evaluaremos su interacción conjunta. Realizaremos la siguiente batería de pruebas:

Parámetro alfa óptico	0,3
Parámetro alfa eléctrico	0,5

Límite de iteraciones	12 (\approx 50% arcos ópticos)
	25 (\approx 100% arcos ópticos)
	50 (\approx 200% arcos ópticos)

Esta batería de pruebas servirá para interpretar el efecto del número límite de iteraciones sin mejorar la solución incumbente.

6.4.2 – Resultados en red de prueba

Presentamos los resultados resumidos para la batería de pruebas especificada.

Tabla 6.8 – Resultados GRASP doble prueba

ocupación	Iteraciones 12				Iteraciones 25			
	F.O.	cmp FO	TMP	cmp TMP	F.O.	cmp FO	TMP	cmp TMP
20	-33950,9	-6%	2,9	-717%	-33952,9	-6%	5,7	-1520%
30	-71833,8	-24%	8,5	-649%	-71866,8	-24%	20,1	-1666%
40	-66823,7	-10%	13,9	-1074%	-66870,7	-10%	17,8	-1411%
50	-75656,6	-10%	12,4	-1013%	-75642,6	-10%	19,5	-1654%
60	-186050,6	-23%	59,1	-1223%	-189059,7	-25%	106,2	-2276%
70	-132902,9	-36%	64,4	-1087%	-133941,9	-37%	110,2	-1930%
80	-163452,6	-21%	95,9	-897%	-163558,6	-21%	184,7	-1820%
90	-25765,2	-24%	21,2	-703%	-25765,2	-24%	35,6	-1249%
intensidad								
Baja	-123721,6	-3%	13,6	-525%	-123756,6	-3%	33,1	-1424%
Media	-365130,8	-16%	109,3	-975%	-369168,8	-18%	193,2	-1800%
Alta	-267583,9	-41%	155,4	-1043%	-267732,8	-41%	273,6	-1912%
suma	-756436,2	-21%	278,3	-973%	-760658,2	-22%	499,8	-1827%

ocupación	Iteraciones 50			
	F.O.	cmp FO	TMP	cmp TMP
20	-33952,9	-6%	11,0	-3051%
30	-71866,8	-24%	30,1	-2539%
40	-66853,7	-10%	49,6	-4107%
50	-76658,6	-12%	55,9	-4934%
60	-191042,7	-27%	202,8	-4436%
70	-136971,9	-40%	223,0	-4008%
80	-170525,6	-26%	399,6	-4054%
90	-25765,2	-24%	75,1	-2744%
intensidad				
Baja	-123758,6	-3%	62,4	-2777%
Media	-376133,8	-20%	359,3	-3433%
Alta	-273744,9	-45%	625,4	-4498%
suma	-773637,3	-24%	1047,1	-3937%

El criterio de selección utilizado para decidir el mejor parámetro α óptico y α eléctrico se repite en este caso para encontrar un buen valor para el número máximo de iteraciones sin solución incumbente mejorada.

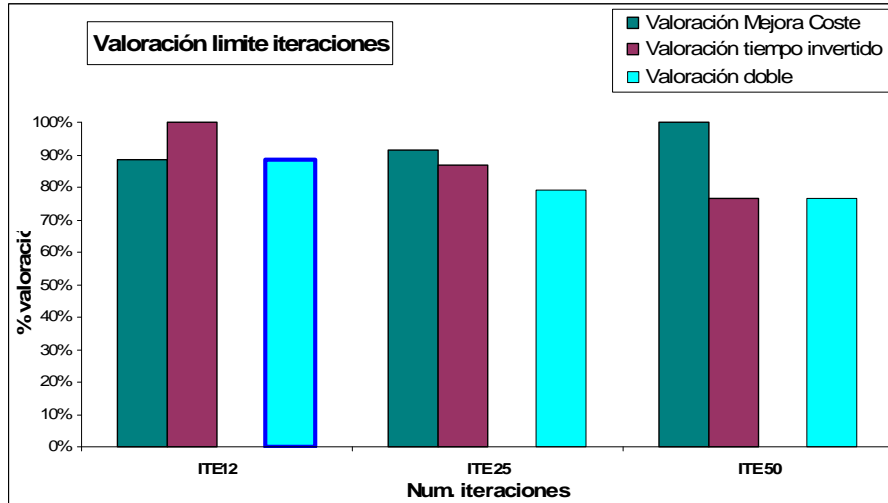


Fig. 6.13 – Selección del mejor valor límite de iteraciones

A la vista de estos resultados, no aumentaremos significativamente el número de iteraciones realizadas con la heurística de doble aleatorización. En todo caso, debido a la mayor complejidad de esta última versión, aumentaremos de 10 a 15 iteraciones este parámetro de parada.

6.4.3 – Resultados en red europea

Los parámetros para esta prueba toman los siguientes valores:

Parámetro alfa óptico	0,3
Parámetro alfa eléctrico	0,5
Límite de iteraciones	15(≈ 15% arcos ópticos)

El resumen de resultados comparados con la referencia básica, y las gráficas de elementos modificados y tiempo de ejecución invertido se muestran a continuación.

Tabla 6.9 – Resultados GRASP doble europea

GRASP OPT 0.3/ELEC 0.5			% (Greedy – Grasp) / Greedy	
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio

20	58,67	-33943	-3141%	-10%
30	38,18	-39122	-1043%	-24%
40	65,09	-49625	-1189%	-21%
50	84,10	-62518	-901%	-23%
60	103,76	-60647	-1057%	-22%
70	126,09	-48092	-1159%	-13%
80	114,05	-37377	-1147%	-20%
90	117,40	-22703	-1335%	-16%
% ocupación	Tiempo medio	Valor F.O. medio	Tiempo medio	Valor F.O. medio
baja	48,42	-96423	-1781%	-21%
media	84,32	-48424	-1028%	-21%
alta	119,18	-30417	-1207%	-17%
Media			-1188%	-19%

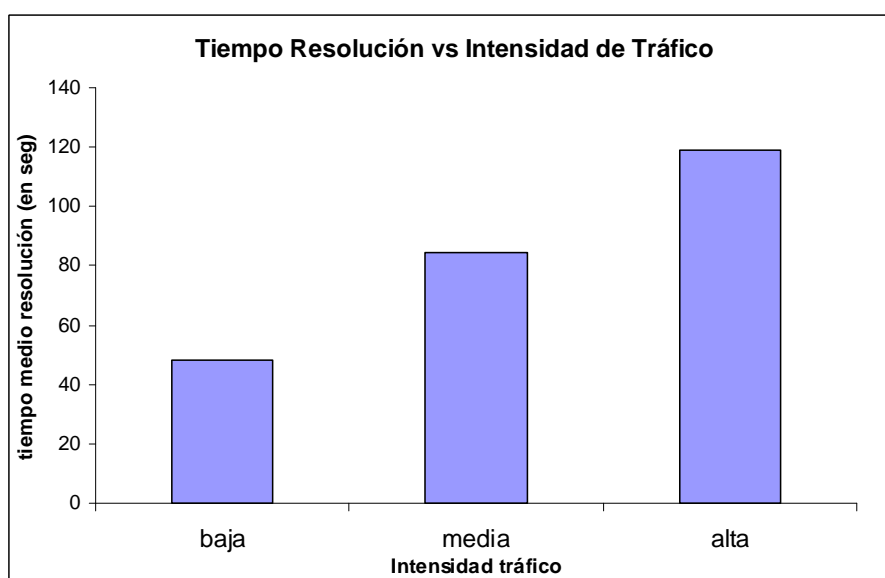
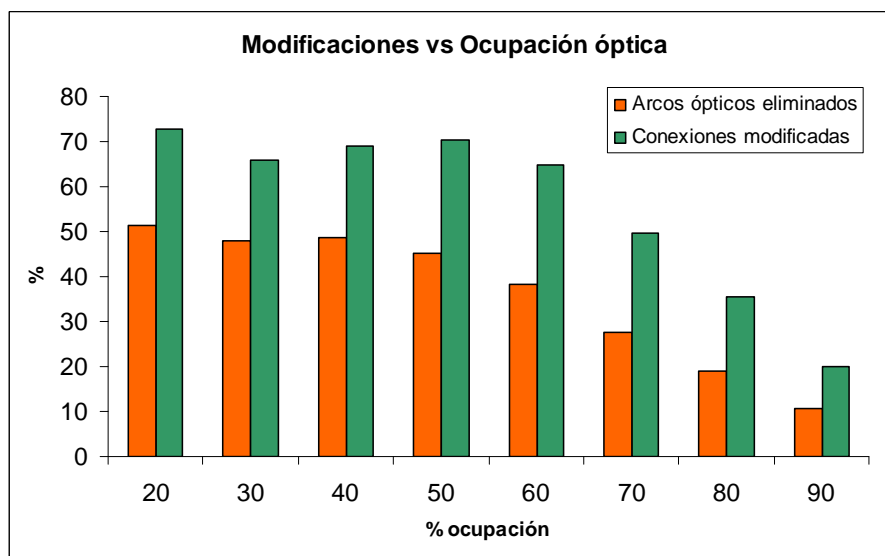


Fig. 6.14 y6.15 – Características de la versión doble

6.4.4 – Análisis de resultados

En este apartado podemos ver como la inclusión conjunta de los dos parámetros en la heurística, añadido a un ligero aumento en el máximo de iteraciones del criterio de parada, logra mejorar los resultados respecto la heurística de base en un 19% de media. Este valor es el mejor que hemos obtenido en todas las versiones.

El efecto del aumento de las iteraciones es determinante y, a medida que este valor aumenta, la probabilidad de encontrar soluciones de mayor reducción de coste en todos los escenarios aumenta. En contra de este aumento en la calidad de la solución encontrada, aumenta el tiempo de ejecución del procedimiento, con lo que perdemos valoración en el criterio por tiempo invertido.

Centrándonos en el tiempo de ejecución medio, podemos tratar de encontrar relación entre esta variable y el número de iteraciones realizadas, para cada versión de la heurística en la aplicación de la red europea.

Tabla 6.10 – Comparación tiempo medio por versión

Versión	nº iteraciones	tmp medio	tmp / nº ite
Greedy	1	7,95	7,95
GRASP óptico	10	63,64	6,36
GRASP eléctrico	10	71,14	7,11
GRASP doble	15	100,09	6,67

Podemos ver que el número de segundos medio por iteración fijada se mantiene entre 6 y 8 segundos, un intervalo considerablemente pequeño, independientemente del método utilizado. De ahí podemos extraer que el tiempo de ejecución del procedimiento aplicado depende prácticamente en exclusividad del número de iteraciones fijadas en el criterio de parada. Si realizamos el mismo análisis para la batería de pruebas del efecto del límite de iteraciones sobre la red de prueba, obtenemos resultados que nos llevan a concluir que el tiempo medio por iteración no es significativamente distinto entre pruebas con diferente número de iteraciones fijado.

Finalmente, podemos comprobar como esta heurística da patrones similares en cuanto a evolución lineal del tiempo de resolución y características de los elementos modificados, al resto de versiones heurísticas aplicadas.

Tabla 6.11 – Comparación tiempo medio por iteración parada

Versión	n° iteraciones	tmp medio	tmp / n° ite
ITE 12	12	5,57	0,46
ITE 25	25	10,00	0,40
ITE 50	50	20,94	0,42

Capítulo 7

Conclusiones y extensión del trabajo

En los capítulos de resolución exacta y resolución heurística de los escenarios, hemos obtenido y analizado distintas soluciones para los mismos ejemplos. En este capítulo se realiza una comparación global de todos los métodos, con la intención de encontrar el más adecuado para resolver el problema de liberación de recursos ópticos en un entorno de gestión de redes ópticas reales.

7.1 – Criterios de valoración del método de resolución

El motivo de este estudio es el de encontrar un método de mejora que pueda utilizarse en la gestión real de redes ópticas.

Los criterios más importantes son los siguientes:

- A) **Calidad de la solución encontrada:** se valora obtener la mejor solución posible, a poder ser la óptima, según las condiciones y restricciones del problema.

B) Tiempo de ejecución necesario: se valora positivamente que el procedimiento tarde poco tiempo, ya que se realiza a partir de un escenario dinámico que puede cambiar entre la captura de datos inicial y el retorno de la solución optimizada.

C) Integración en el sistema de gestión: se valora la sencillez con la que el sistema de gestión se comunica con el optimizador para intercambiar información.

Los métodos evaluados hasta ahora cumplen positivamente alguno de los criterios, en detrimento de tener como punto débil alguno de los otros. En líneas generales podemos comparar el conjunto de métodos heurísticos con la referencia del procedimiento exacto, para poder asignar valoraciones a los criterios definidos.

Criterio	Método Exacto	Métodos Heurísticos
Calidad de la solución	Mejor solución posible. Este método encuentra el óptimo global del problema.	Solución buena, con tendencia a alejarse de la solución óptima a medida que aumenta la dimensión del problema.
Tiempo invertido	Tiempo exponencialmente creciente con el tamaño del problema. En ejemplos de dimensiones reales, este método necesita un equipo de grandes prestaciones para devolver soluciones, y aún así, las genera en tiempos inadmisibles para los escenarios considerados.	Tiempo creciente linealmente con el tamaño del problema. Tiempo ajustable a partir del parámetro del criterio de parada. Posibilidad de equilibrar el tiempo invertido con la calidad de la solución obtenida.

<p style="text-align: center;">Integración</p>	<p>Integración compleja. Es necesaria una plataforma externa y un motor de resolución de licencia restringida integrado en esa plataforma. El sistema de gestión debe conectarse con esa plataforma externa para resolver el problema.</p>	<p>Integración sencilla, como función o extensión de la implementación del sistema de gestión.</p>
---	--	--

7.2 – Comparación Método exacto vs. Métodos heurísticos

Las figuras 7.1 y 7.2 muestran los criterios de calidad de solución y tiempo invertido para la comparación de todos los métodos aplicados sobre la red de prueba y la red europea (en este último caso, comparando sólo las instancias de las que se disponen datos de la resolución exacta). Los resultados se muestran referidos a la solución óptima devuelta por el motor de resolución CPLEX.

Podemos sacar varias conclusiones a la vista de las gráficas:

- 1) El valor óptimo de la función objetivo queda, en media, bastante cerca del alcance de las versiones heurísticas que incorporan procedimientos de selección de elementos aleatoria. A partir de la versión heurística básica, que se sitúa entorno a un 75% de la solución óptima, las versiones aleatorizadas llegan a situarse a tan sólo un 10% de distancia respecto a la mejor solución.
- 2) El tiempo de ejecución de las heurísticas aumenta a medida que incorporamos elementos de aleatorización y a medida que aumentamos el número de iteraciones del criterio de parada.
- 3) El tiempo de ejecución de las heurísticas en el caso de la red de tipo europea son más de 10 veces inferiores a los obtenidos por la resolución exacta.

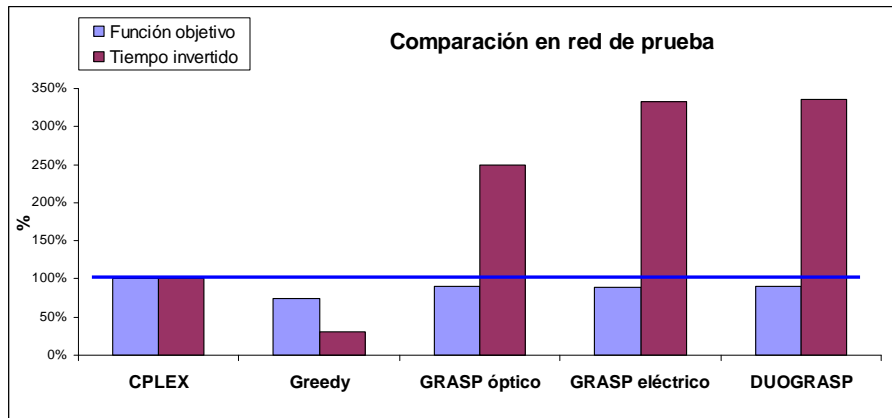


fig 7.1 – Comparación de metodologías en red de prueba

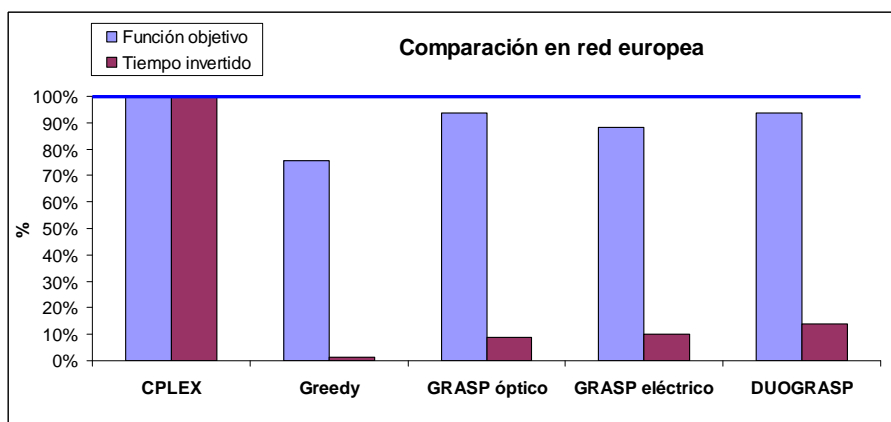


fig 7.2 – Comparación de metodologías en red europea

Si a estos comentarios, añadimos que la integración de las heurísticas es más sencilla que la del método exacto, podemos ver claramente que cualquier versión heurística aleatorizada que usemos nos dará resultados efectivos, tanto por calidad de solución como por tiempo de generación. Dada la poca utilidad del método exacto para problemas de cierta dimensión, descartaremos este procedimiento como candidato a ser escogido para resolver este problema.

7.3 – Selección de la mejor versión heurística

Dados los resultados anteriores, vamos a escoger una versión heurística como la más apropiada para resolver el problema de liberación de recursos ópticos en una situación real.

La tabla 7.1 muestra la cantidad de veces que una heurística ha conseguido la mejor solución entre los cuatro métodos heurísticos, en grupos de instancias caracterizadas por el número de restricciones del problema. Los datos se presentan en frecuencias relativas en tanto por uno.

Tabla 7.1- Comparación entre métodos heurísticos

Restricciones	Greedy	GRASP Óptico	GRASP Eléctrico	GRASP Doble	TOTAL
100-200	0,00	0,50	0,00	0,50	0,08
200-300	0,00	0,50	0,25	0,25	0,08
300-400	0,00	0,67	0,11	0,22	0,09
400-500	0,00	0,33	0,08	0,58	0,12
500-600	0,00	0,50	0,21	0,29	0,14
600-700	0,00	0,50	0,18	0,32	0,22
700-800	0,07	0,30	0,22	0,41	0,27
TOTAL	0,02	0,44	0,17	0,37	1

Podemos ver como en un 44% de las instancias el método GRASP óptico es el mejor, aunque le sigue de cerca el método GRASP doble con un 37% de instancias mejor resueltas. Vemos como el GRASP eléctrico consigue solo mejorar el resto de heurísticas en un 17% de los casos, mientras que el algoritmo básico *greedy*, apenas resuelve mejor 2 de las 100 instancias. Si miramos los resultados por grupo de restricciones, vemos como las dos heurísticas dominantes presentan buenos resultados en diferentes niveles de dimensión del problema.

Visto que el tiempo por iteración realizada es bastante constante para todas las pruebas, este criterio de calidad de la solución se muestra determinante para escoger el mejor método. Dado que el algoritmo GRASP óptico presenta los mejores resultados globales y supone una implementación algo más sencilla que la versión GRASP doble, tomamos esta versión como la mejor de todas cuantas hemos propuesto.

7.4 – Integración de la meta-heurística en el sistema de gestión

Una vez escogida la metodología para resolver el problema de liberación de arcos ópticos, procederemos a su implementación en un sistema de gestión de una red real. Este proyecto ha sido realizado como parte de la investigación del Grupo de Comunicaciones Ópticas (GCO) de la UPC, así que la integración se realizará sobre la red de pruebas CARISMA.

El proyecto CARISMA comenzó en 2003 con el objetivo de construir una red óptica WDM para ser utilizada como plataforma experimental para la integración y evaluación de las tecnologías ópticas más innovadoras. La red CARISMA ha sido implementada como una red de transporte óptico basada en el estándar ASON [7] de la Unión Internacional de Telecomunicaciones (ITU-T) y está provista de un plano de control GMPLS [8]. El plano de transporte está formado por un anillo óptico de dos fibras y tres nodos ópticos con capacidad de extracción e inserción de canales ópticos. Cada nodo está equipado con cuatro láseres WDM a 10Gbit/s.

Los nodos están localizados en los campus de la UPC en Tarrasa y Barcelona y en TVC en Sant Joan Despi. La distancia total del anillo es de alrededor de 100km. Además de los nodos físicos, se dispone de emuladores de nodo (componente software), con los que es posible realizar pruebas sobre cualquier topología de red.

La red CARISMA cuenta además con un sistema de gestión de red que es responsable de la gestión completa del sistema y se encarga de la generación de peticiones de conexión ópticas.

Hoy en día, la red CARISMA se utiliza como base experimental dentro de varios proyectos de investigación tanto nacionales como europeos.

La heurística seleccionada como mejor método en este capítulo, ha sido probada sobre la red CARISMA configurada con la topología que se muestra en la figura 7.3. Se han generado distintas intensidades de tráfico, para las cuales se han obtenido resultados en línea con los expuestos anteriormente. Como ejemplo, en la figura 7.4 se muestra el porcentaje de modificaciones realizadas sobre la red original para distintos grados de ocupación de la red.

Puede observarse que el patrón es muy similar a los obtenidos con los escenarios generados aleatoriamente. Por una parte, obtenemos una eliminación de recursos y una modificación de rutas eléctricas que decrecen a medida que aumenta la ocupación de la red. La cantidad de arcos eliminados se sitúa en el intervalo del 15% al 35% respecto a la red original.

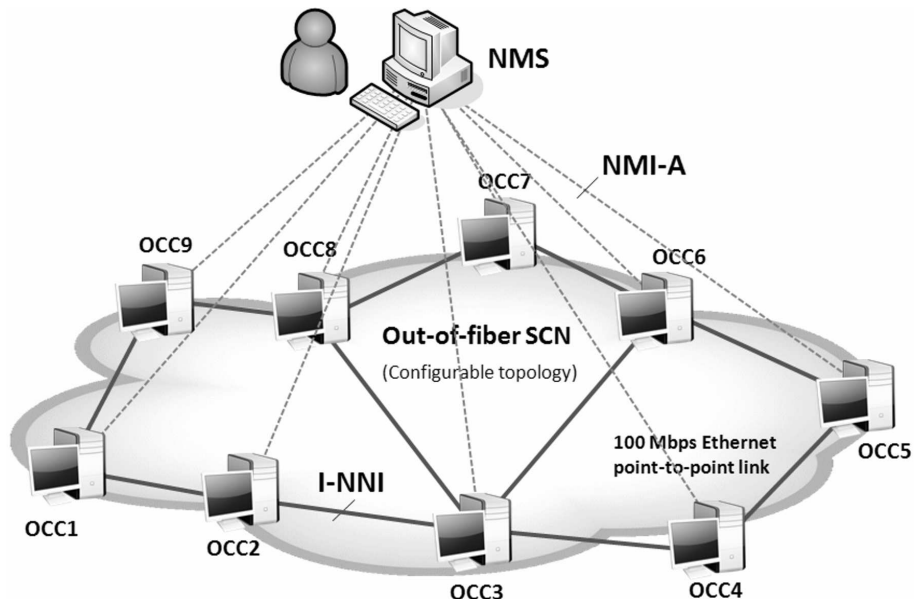


Fig. 7.3 – Red de prueba CARISMA

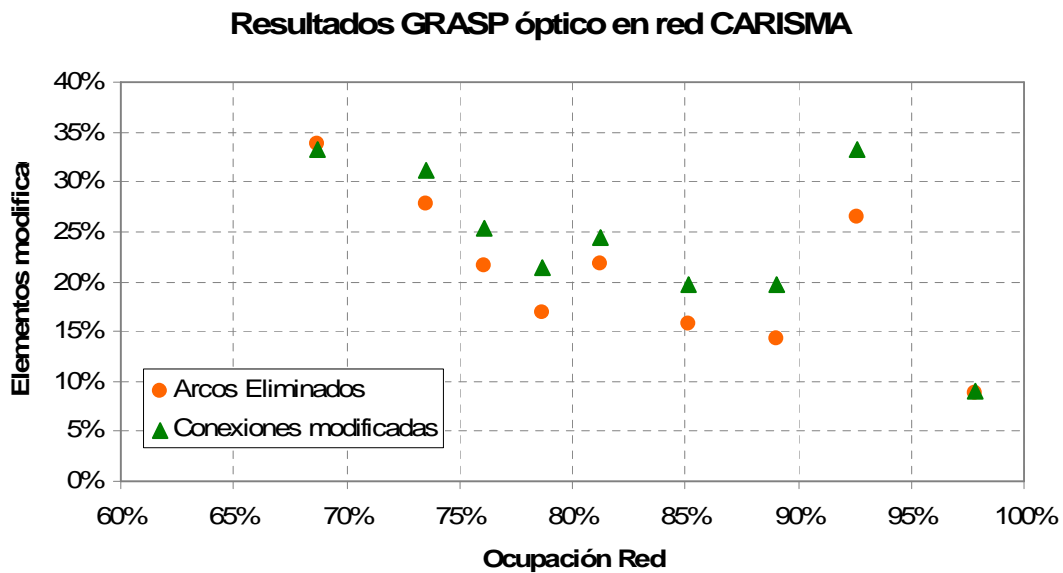


Fig. 7.4 – Heurística sobre red CARISMA

7.5 – Extensión del proyecto

Las consideraciones y definiciones tomadas inicialmente al presentar el problema han sido determinantes para la calidad de las soluciones obtenidas. La cantidad de mejora que hemos podido realizar se ha visto limitada por una serie de restricciones que hemos aplicado, como condición indispensable para el problema propuesto. No obstante, podríamos relajar alguna de estas condiciones para intentar aumentar la cantidad de

recursos ópticos liberados, sin perder factibilidad en las conexiones eléctricas establecidas. Algunas de estas modificaciones podrían ser:

- 1) Permitir el establecimiento de nuevos arcos ópticos, sin permitir interrupción de tráfico: a partir de los recursos ópticos disponibles en la red óptica, se podrían generar nuevos arcos ópticos en la capa de agregación de clientes, que permitieran redistribuir el tráfico eléctrico de manera que se redujera el coste final.
- 2) Permitir el establecimiento de nuevos arcos ópticos, permitiendo interrupción de tráfico: esta versión del problema permite cambios sobre los recursos ópticos utilizados, con lo cual debe interrumpirse el servicio eléctrico. Esta aplicación se podría alternar con la que no permite cortes de señal, para ofrecer un procedimiento de optimización que corte el tráfico solo en situaciones en la que se puede producir una reoptimización significativamente mejor con este método que con el que mantiene el tráfico estable.
- 3) Permitir la modificación del tráfico a nivel de las fibras ópticas: este método pretende optimizar las asignaciones de arcos ópticos a canales de longitud de onda. Este método afecta tanto a la capa de agregación eléctrica como a la capa óptica de base.
- 4) Estudiar un método de resolución exacta para diferentes versiones del problema original, basado en algún método iterativo como la generación de columnas, que permita encontrar la solución óptima en un tiempo sensiblemente menor a las resoluciones exactas directas.

Referencias bibliográficas

- [1] Fernando Agraz, Luis Velasco, Jordi Perelló, Marc Ruiz Salvatore Spadaro, Gabriel Junyent and Jaime Comellas, “Design and Implementation of a GMPLS-Controlled Grooming-capable Optical Transport Network”, aceptado para su publicación en Journal of Optical Networking
- [2] Wayne D. Grover, “Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking”, Ed. Prentice Hall, 2004
- [3] Deepankar Medhi, Karthikeyan Ramasamy, “Network Routing: Algorithms, Protocols, and Architectures”, Ed. Elsevier – Morgan Kaufmann, 2007
- [4] Sophie De Maesschalck, Didier Colle, Ilse Lievens, Mario Pickavet, Piet Demeester “Pan-European Optical Transport Networks: An Availability-based Comparison”, Photonic Network Communications, Vol.5, pp.203-225, 2003
- [5] T. Feo, M. Resende, “Greedy randomized adaptative search procedures”, Journal of Global Optimization, pp.109-133, Jun. 1995.
- [6] Martínez García, Alfredo Gabriel, “Creación de un algoritmo GRASP para la empresa Junghanns”, http://catarina.udlap.mx/u_dl_a/tales/documentos/lii/martinez_g_ag.html, Cap. 3, 2003.
- [7] ITU-T G.8080, “Architecture for the automatically switched optical network (ASON)”, 2001.
- [8] E. Mannie, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, RFC-3945 (2004).

Anexo: Implementación de la meta-heurística GRASP en Matlab

Consideraciones sobre la implementación

En este anexo se muestran las funciones principales de la meta-heurística GRASP. Para la realización de este PFC, se han utilizado funciones de generación de escenarios, almacenamiento de datos, cálculo de rutas posibles, evaluación de soluciones, etc. Este anexo no recoge estas funciones secundarias, que han sido implementadas siguiendo las definiciones y características descritas a lo largo de este documento.

Implementación Algoritmo Central

```
function
[OPT_INC,ELEC_INC,EO_INC,res,FO_INC,AE_INC,PM_INC,tiempo1,tiempo2,lap_
inc]=OROGRASP(OPT,ELEC,EO,alfaopt,alfaelec,mejora,MAXITE,order)
tiempo1=0;
tiempo2=0;
FO_INC=inf;
ITE=0;
laps=0;
while(ITE<MAXITE)
    laps=laps+1;
    inicio=cputime;

[OPT_1,ELEC_1,EO_1,res,FO_1,AE_1,PM_1]=ORO_FASE1(OPT,ELEC,EO,alfaopt,a
lfaelec,order);
    tiempo1=tiempo1+cputime-inicio;
    inicio=cputime;
    FO_2=inf;
    if mejora==1
        [OPT_2,ELEC_2,EO_2]=ORO_FASE2(OPT,OPT_1,ELEC_1,EO_1);
```

```

        [FO_2,AE_2,PM_2]=fobjeval(OPT,OPT_2,ELEC,ELEC_2,1000,1,0.001);
    end
    tiempo2=tiempo2+cputime-inicio;

    if FO_2<FO_1
        OPT_ITE=OPT_2;
        ELEC_ITE=ELEC_2;
        EO_ITE=EO_2;
        FO_ITE=FO_2;
        AE_ITE=AE_2;
        PM_ITE=PM_2;
    else
        OPT_ITE=OPT_1;
        ELEC_ITE=ELEC_1;
        EO_ITE=EO_1;
        FO_ITE=FO_1;
        AE_ITE=AE_1;
        PM_ITE=PM_1;
    end

    if FO_ITE<FO_INC
        OPT_INC=OPT_ITE;
        ELEC_INC=ELEC_ITE;
        EO_INC=EO_ITE;
        FO_INC=FO_ITE;
        AE_INC=AE_ITE;
        PM_INC=PM_ITE;
        ITE=0;
        lap_inc=laps;
    else
        ITE=ITE+1;
    end
    end
    fobj_vct(laps)=FO_INC;
end

%figure();

%plot(1:laps,fobj_vct,'b-');

```

Implementación Fase Constructiva

```

function
[OPT_INC,ELEC_INC,EO_INC,res,FO_INC,AE_INC,PM_INC]=ORO_FASE1(OPT,ELEC,
EO,alfaopt,alfaelec,order)
m=size(OPT,1);%Número de arcos ópticos
%n=length(FIELDNAMES(ELEC)); Número de conexiones eléctricas
ELEC_ORI=ELEC;
%Paso 0: Inicializaciones
res=fopen('ORO_GRASP_RESULT.txt','w');
BLOQ=zeros(m,1);
AE=zeros(m,1);
PM=zeros(length(fieldnames(ELEC)),1);

%Paso 1: Obtengo la lista de candidatos ordenada según el parámetro
alfa
OPT_GRASP=grasppear(OPT,alfaopt,order); %Lista de arcos ópticos por
eliminar
OPT_REF=sortrows(OPT,1); %Lista de arcos ópticos ordenados por id

```



```

%Paso 2: Intento eliminar arcos ópticos
pos=1; %Posición de exploración de arco óptico

while pos<=m %mientras haya arcos que puedan ser liberados
    %2.0 - Obtengo id del arco óptico a liberar
    idopt=OPT_GRASP(pos,1);
    fprintf(res,'%i) Arco optico: %i - Ocupación:
%i\n',pos,idopt,OPT_REF(idopt,4));
    if (OPT_REF(idopt,4)>0 && BLOQ(idopt)==0) %si el arco tiene
ocupación
        %2.1 - Obtengo los id de las conexiones eléctricas que pasan
por arco
        %idopt
        CON=EO(:,idopt);
        ncon=sum(CON); %Número de conexiones eléctricas que pasan por
el arco idopt
        rcon=0; %Número de conexiones modificadas
        %Me guardo la información antes de cualquier cambio
        OPT_INI=OPT_REF;
        ELEC_AUX=ELEC;
        welec=zeros(ncon,1);
        for i=1:length(CON)
            if CON(i)==1 %Si la conexión i pasa por el arco optico
idopt
                % obtengo información de la conexión electrica
                idelec=['elec' int2str(i)];
                PATHS=getfield(ELEC,idelec,'paths');
                bw=getfield(ELEC,idelec,'bandwidth');
                ruta=getfield(ELEC,idelec,'ruta');
                [c,p]=size(PATHS);
                fprintf(res,'%i) Reenrutando conexión %i de BW %i ...
',pos,i,bw);
                % elimino bandwidth de los arcos por los que pasa la
                % conexión
                for j=4:p
                    if PATHS(ruta,j)>0
OPT_REF(PATHS(ruta,j),4)=OPT_REF(PATHS(ruta,j),4)-bw;
OPT_REF(PATHS(ruta,j),8)=OPT_REF(PATHS(ruta,j),8)-1;
                    else
                        break;
                    end
                end
                PATHS=graspearpath(PATHS,alfaelec);
                OPT_VOID=OPT_REF; %Estructura con el arco idopt vacio
                % reenruto por la más corta disponible
                for k=1:c
                    r=PATHS(k,1);
                    RUTABUENA=0;
                    if r~=ruta
                        for j=4:p
                            if PATHS(k,j)>0
                                if (PATHS(k,j)~=idopt &&
OPT_REF(PATHS(k,j),4)+bw<=OPT_REF(PATHS(k,j),5) &&
OPT_REF(PATHS(k,j),5)>0)
OPT_REF(PATHS(k,j),4)=OPT_REF(PATHS(k,j),4)+bw;
OPT_REF(PATHS(k,j),8)=OPT_REF(PATHS(k,j),8)+1;
                                else

```

```

                                RUTABUENA=0;
                                break;
                                end
                                else
                                RUTABUENA=1;
                                k_inc=r;
                                break;
                                end
                                end
                                if RUTABUENA==1
                                %Habremos encontrado una buena ruta
                                break;
                                else
                                OPT_REF=OPT_VOID;
                                end
                                end
                                end

                                if RUTABUENA==0 % Hemos intentado encontrar una ruta
alternativa y no la hemos encontrado
                                OPT_REF=OPT_INI;
                                ELEC=ELEC_AUX;
                                BLOQ(idopt)=1;
                                fprintf(res,'%i) KO conexion %i\n',pos, i);
                                break;
                                else
                                rcon=rcon+1; % Aumentamos el contador de
conexiones reenrutadas
                                % Modificamos la base de datos de las conexiones
                                ELEC=setfield(ELEC,idelec,'ruta',k_inc);
                                welec(rcon)=i;
                                fprintf(res,'%i) OK conexion %i\n',pos, i);
                                end
                                end
                                end

                                if (ncon==rcon && rcon~=0)%Hemos reenrutado todas las
conexiones
                                AE(idopt)=1;
                                for b=1:length(welec)
                                PM(welec(b))=1;
                                end
                                %Eliminamos la capacidad del arco optico idopt
                                OPT_REF(idopt,5)=0;
                                BLOQ(idopt)=1;
                                %Controlamos si tiene un arco paralelo y lo eliminamos
                                if OPT_REF(idopt,7)~=0
                                OPT_REF(OPT_REF(idopt,7),4:5)=0;
                                end
                                fprintf(res,'%i) Arco eliminado!\n',pos);

                                % Reordenamos otra vez
                                OPT_GRASP=grasppear(OPT_REF,alfaopt,order);
                                %Modificamos matriz EO
                                EO(:,idopt)=0;
                                if OPT_REF(idopt,7)~=0
                                EO(:,OPT_REF(idopt,7))=0;
                                end

                                %Ponemos el marcador de posición a 1
                                pos=1;

```

```

        fprintf(res,'%i) Arco óptico eliminado: %i \n',pos,idopt);
        fprintf(res,'%i) Conexiones reenrutadas: ',pos);
        for o=1:ncon
            if welec(o)~=0
                fprintf(res,'%i ',welec(o));
                idelec=['elec' int2str(welec(o))];
                ruta=getfield(ELEC,idelec,'ruta');
                paths=getfield(ELEC,idelec,'paths');
                c=size(paths,1);
                EO(welec(o),:)=0;
                for i=4:c
                    if paths(ruta,i)~=0
                        EO(welec(o),paths(ruta,i))=1;
                    else
                        break;
                    end
                end
            else
                break;
            end
        end
        fprintf(res,'\n\n');
    else
        % Aumentamos el marcador de posición
        pos=pos+1;
        OPT_REF=OPT_INI;
        ELEC=ELEC_AUX;
    end

else
    %Aseguramos que la capacidad sea 0
    if OPT_GRASP(pos,5)>0 && OPT_GRASP(pos,4)==0
        OPT_GRASP(pos,5)=0;
        OPT_REF(idopt,5)=0;
    end
    % Aumentamos el marcador de posición
    fprintf(res,'%i) Arco optico sin ocupación\n',pos) ;
    pos=pos+1;
end
end

OPT_INC=sortrows(OPT_GRASP,1);
ELEC_INC=ELEC;
fclose(res);
EO_INC=EO;

[FO_INC,AE_INC,PM_INC]=fobjeval(OPT,OPT_INC,ELEC_ORI,ELEC_INC,1000,1,0
.001);

```

Implementación Fase de Mejora

```
function [OPT_INC,ELEC_INC,EO]=ORO_FASE2(OPT,OPT_INC,ELEC_INC,EO)

OPT_ORI=OPT_INC;
ELEC_ORI=ELEC_INC;

%1) Creamos los conjuntos INSET y OUTSET
m=size(OPT,1);
h=0;
p=0;

for k=1:m
    if (OPT(k,6)>0 && OPT(k,5)>0 && OPT_INC(k,4)==0)
        h=h+1;
        INSET(h,:)=OPT_INC(k,:);
        if (OPT(k,7)~=0)
            h=h+1;
            i=OPT_INC(k,7);
            INSET(h,:)=OPT_INC(i,:);
        end
    end

    if (OPT_INC(k,6)>0 && OPT_INC(k,5)>0)
        p=p+1;
        OUTSET(p,:)=OPT(k,:);
        if (OPT(k,7)~=0)
            p=p+1;
            i=OPT(k,7);
            OUTSET(p,:)=OPT(i,:);
        end
    end
end

%2) Los ordenamos segun criterio

if (h~=0 && p~=0)
    INSET=sortrows(INSET,6);
    OUTSET=sortrows(OUTSET,-6);
    numout=size(OUTSET,1);
    numin=size(INSET,1);

%3) Para cada uno de los arcos candidatos a salir

for u=1:numout
    if OUTSET(u,6)>0
        costelim=OUTSET(u,6);
        idout=OUTSET(u,1);
        OPT_INI=OPT_INC;
        % Quitamos el arco de la solución ORIGINAL, eliminando las
        conexiones
        % que pasen por el:
        CON=EO(:,idout);
        for i=1:length(CON)
            if CON(i)==1 %Si la conexión i pasa por el arco optico
                idopt
                    % obtengo información de la conexión electrica
                    idelec=['elec' int2str(i)];
                    PATHS=getfield(ELEC_INC,idelec,'paths');
```

```

        bw=getfield(ELEC_INC,idelec,'bandwidth');
        ruta=getfield(ELEC_INC,idelec,'ruta');
        [c,p]=size(PATHS);
        % elimino bandwidth de los arcos por los que pasa la
conexión
        for j=4:p
            if PATHS(ruta,j)>0
OPT_INC(PATHS(ruta,j),4)=OPT_INC(PATHS(ruta,j),4)-bw;
OPT_INC(PATHS(ruta,j),8)=OPT_INC(PATHS(ruta,j),8)-1;
                else
                    break;
                end
            end
        end
        end
        OPT_INC(idout,5)=0;
        g=OPT_INC(idout,7);
        if g~=0
            OPT_INC(g,5)=0;
        end
        OPT_AUX=OPT_INC;
        OPT_BK=OPT_INC;
        ELEC_AUX=ELEC_INC;
        hecho=0;
        for h=1:numin
            if (INSET(h,6)<costelim && INSET(h,6)>0)

                idin=INSET(h,1);
                % Activamos arco
                OPT_INC(idin,5)=OPT(idin,5);
                g=OPT_INC(idin,7);
                if g~=0
                    OPT_INC(g,5)=OPT(g,5);
                end

                % Tratamos de meter las conexiones
                rcon=0;
                for i=1:length(CON)
                    if CON(i)==1 %Si la conexión i pasa por el arco optico
idopt
                        % obtengo información de la conexión eléctrica
                        idelec=['elec' int2str(i)];
                        PATHS=getfield(ELEC_INC,idelec,'paths');
                        bw=getfield(ELEC_INC,idelec,'bandwidth');
                        [c,p]=size(PATHS);
                        for k=1:c
                            RUTABUENA=0;
                            for j=4:p
                                if PATHS(k,j)>0
                                    if
(OPT_INC(PATHS(k,j),4)+bw<=OPT_INC(PATHS(k,j),5))
OPT_INC(PATHS(k,j),4)=OPT_INC(PATHS(k,j),4)+bw;
OPT_INC(PATHS(k,j),8)=OPT_INC(PATHS(k,j),8)+1;
                                            else
                                                RUTABUENA=0;
                                                break;
                                            end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

else
    RUTABUENA=1;
    k_inc=k;
    break;
end
end
if RUTABUENA==1
    %Habremos encontrado una buena ruta
    break;
else
    OPT_INC=OPT_AUX;
end
end
if RUTABUENA==0 % Hemos intentado encontrar una
ruta alternativa y no la hemos encontrado
    OPT_INC=OPT_AUX;
    ELEC_INC=ELEC_AUX;
    break;
else
    rcon=rcon+1; % Aumentamos el contador de
conexiones reenrutadas
    % Modificamos la base de datos de las
conexiones
ELEC_INC=setfield(ELEC_INC,idelec,'ruta',k_inc);
    welec(rcon)=i;
    OPT_AUX=OPT_INC;
end
end
end
if (rcon==sum(CON) && rcon~=0)

    OPT_INC(idout,5)=0;

    %Controlamos si tiene un arco paralelo y lo
eliminamos

    if OPT_INC(idout,7)~=0
        OPT_INC(OPT_INC(idout,7),4:5)=0;
    end
    EO(:,idout)=0;
    if OPT_INC(idout,7)~=0
        EO(:,OPT_INC(idout,7))=0;
    end
    for o=1:sum(CON)
        if welec(o)~=0
            idelec=['elec' int2str(welec(o))];
            ruta=getfield(ELEC_INC,idelec,'ruta');
            paths=getfield(ELEC_INC,idelec,'paths');
            c=size(paths,1);
            EO(welec(o),:)=0;
            for i=4:c
                if paths(ruta,i)~=0
                    EO(welec(o),paths(ruta,i))=1;
                else
                    break;
                end
            end
        end
    end
else

```

```
                break;
            end
            end
            INSET(h,6)=inf;
            OUTSET(u,6)=0;
            hecho=1
            break;

        else
            OPT_INC=OPT_BK;
            ELEC_INC=ELEC_AUX;
        end
    end
end
if hecho==0
    OPT_INC=OPT_INI;
end
end
end
end
```