



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Organización Industrial

ASIGNACIÓN DE FRANJAS HORARIAS A VUELOS. AJUSTE Y COMPARATIVA DE ALGORITMOS.

Autor:

Gómez Miguel, Patricia

Tutor:

Araúzo Araúzo, José Alberto
Departamento de Organización de
Empresas y CIM

Valladolid, Junio 2019.

RESUMEN

El gran incremento de la demanda de transporte aéreo en los últimos años ha provocado que aumente el desequilibrio entre la capacidad de los aeropuertos y dicha demanda, dando lugar a congestiones y retrasos.

En el presente documento se llevará a cabo el ajuste y comparación de diferentes métodos de resolución del problema de asignación de franjas horarias. Se utilizará la programación lineal para modelar, y posteriormente se resolverá computacionalmente mediante dos algoritmos: el branch and bound y la relajación lagrangiana con el método del subgradiente (interpretable también como una subasta combinatoria). Esto permitirá realizar varios experimentos para contrastar ambos métodos en diferentes condiciones.

PALABRAS CLAVE

Asignación de franjas horarias, programación lineal, branch and bound, subastas combinatorias, relajación lagrangiana.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1. Introducción.....	13
1.1. Justificación.....	15
1.2. Objetivos	15
1.3. Contenidos	16
CAPÍTULO 2. Asignación de franjas horarias.....	17
2.1. Situación actual del tráfico aéreo mundial	19
2.2. El problema de la capacidad aeroportuaria	20
2.2.1. Principales problemas.....	20
2.2.2. Posibles soluciones	22
2.3. Procedimiento para la asignación de franjas horarias	23
2.3.1. Comercialización primaria de slots.....	23
2.3.2. Comercialización secundaria de slots	24
2.4. Conclusiones	25
CAPÍTULO 3. Programación lineal entera mixta	27
3.1. Introducción.....	29
3.2. Programación lineal	29
3.2.1. Descripción	29
3.2.2. Formulación de un problema general	30
3.2.3. Métodos de resolución	31
3.3. Método de branch and bound.....	33
3.3.1. Descripción del método.....	33
3.3.2. Algoritmo	34
3.4. Modelo de programación lineal para asignación de slots aeroportuarios.35	
3.4.1. Parámetros del modelo	35
3.4.2. Variables	37
3.4.3. Modelo de programación lineal	38
3.5. Conclusiones	40

CAPÍTULO 4. Subastas combinatorias.....	41
4.1. Introducción.....	43
4.2. Definición.....	43
4.3. Tipos de subastas.....	44
4.4. Subastas combinatorias.....	45
4.4.1. ¿Por qué es el más apropiado para el reparto de slots?.....	45
4.4.2. Pasos de resolución de una subasta combinatoria.....	45
4.5. Aplicación de subasta combinatoria a problema de asignación de franjas horarias.....	47
4.5.1. Producto subastado.....	47
4.5.2. Cantidad de productos subastados.....	48
4.5.3. Distribución de los ingresos.....	49
4.5.4. Restricciones para la subasta.....	49
4.5.5. Frecuencia de realización de la subasta.....	50
4.5.6. Mecanismo de subasta. Subasta combinatoria con lotes.....	50
4.6. Conclusiones.....	53
CAPÍTULO 5. Relajación Lagrangiana.....	55
5.1. Introducción.....	57
5.2. Descripción del método.....	57
5.3. Formulación básica.....	57
5.3.1. Relajación lagrangiana.....	58
5.3.2. Función dual asociada.....	59
5.3.3. Teorema de Dualidad débil.....	59
5.3.4. Problema dual.....	60
5.3.5. Gap de dualidad.....	60
5.4. Resolución del problema dual. Método del subgradiente.....	60
5.5. Relajación Lagrangiana aplicada al problema de programación lineal de Asignación de Slots en aeropuertos.....	62
5.5.1. Relajación Lagrangiana de las restricciones de capacidad.....	62
5.5.2. Método del subgradiente.....	66
5.6. Conclusiones.....	73

CAPÍTULO 6. Ajuste y comparación de métodos	75
6.1. Introducción	77
6.2. Generación de problemas.....	77
6.3. Resolución mediante Relajación Lagrangiana	80
6.3.1. Parámetros a modificar	80
6.3.2. ¿Qué obtenemos?.....	81
6.3.3. Estudio 1: ajuste de parámetros.....	82
6.4. Comparación de los dos métodos	86
6.4.1. Estudio 2: ¿Cuál es el mejor método de resolución?	86
6.4.2. Estudio 2. Solución óptima RL vs GRB	95
6.4.3. Estudio 3. Solución óptima en función de problema tipo a, b o c.....	96
6.4.4. Estudio 4: Resolución con horizonte de planificación periódico.....	99
6.5. Conclusiones	104
CAPÍTULO 7. Conclusiones	105
BIBLIOGRAFÍA.....	109
ANEXO.....	113

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Evolución porcentual de pasajeros por kilómetro de los últimos seis años. Fuente: IATA (2018)	19
Ilustración 2. Mapa del sistema FACENT, que muestra la congestión del tráfico aéreo mundial. Fuente: NASA.....	21
Ilustración 3. Algoritmo de Branch and Bound. Fuente: Adaptación de (Alvarado Ortega, 2016).....	34
Ilustración 4. Proceso de subasta combinatoria. Fuente: adaptación de (Forde & Doyle, 2008).....	46
Ilustración 5. Plantilla para cada participante en la subasta. Fuente: (Ball, Berardino, & Hansen, 2018)	51
Ilustración 6. Muestra de una puja de un participante de la subasta. Fuente: (Ball, Berardino, & Hansen, 2018)	52
Ilustración 7. Ejemplo de ofertas de todos los participantes. Fuente: (Ball, Berardino, & Hansen, 2018)	52
Ilustración 8. Convergencia de la resolución de un problema de optimización mediante RL.	81
Ilustración 9. Representación del valor objetivo y del nº de iteraciones de 1 problema, para cada una de las combinaciones de parámetros.	82
Ilustración 10. Combinación de parámetros óptima (mayor valor objetivo) para cada tamaño de problema.....	84
Ilustración 11. Combinación de parámetros óptima (menor número de iteraciones) para cada tamaño de problema.	85
Ilustración 12. Representación gráfica del valor objetivo para cada problema pequeño.....	90
Ilustración 13. Representación gráfica del valor objetivo para cada problema mediano.....	90
Ilustración 14. Representación gráfica del valor objetivo para cada problema grande.....	91

Ilustración 15. Representación gráfica del número de iteraciones para cada problema pequeño.	92
Ilustración 16 Representación gráfica del número de iteraciones para cada problema mediano.	92
Ilustración 17. Representación gráfica del número de iteraciones para cada problema grande.	93
Ilustración 18 Representación gráfica del tiempo de ejecución para cada problema pequeño.	94
Ilustración 19. Representación gráfica del tiempo de ejecución para cada problema mediano.	94
Ilustración 20. Representación gráfica del tiempo de ejecución para cada problema grande.	95
Ilustración 21. Porcentaje de diferencia del valor objetivo entre método LR1 y GRB.....	96
Ilustración 22. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla A.....	97
Ilustración 23. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla B.....	98
Ilustración 24. Diferencia (% del valor objetivo) entre método GRB con t de ejecución reducido y LR1 con semilla B.....	98
Ilustración 25. Valor objetivo para horizonte periódico métodos RL y GRB para problemas pequeños.	101
Ilustración 26. Valor objetivo para horizonte periódico métodos RL y GRB para problemas medianos.	102
Ilustración 27. Valor objetivo para horizonte periódico métodos RL y GRB para problemas grandes.	102
Ilustración 28. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla A para horizonte de planificación periódico.	103

ÍNDICE DE TABLAS

Tabla 1. Constantes de capacidad del aeropuerto. Fuente: (de la Fuente, 2016)	77
Tabla 2. Tipos de problemas aleatorios generados para cada semilla.....	78
Tabla 3: Combinaciones utilizadas para resolver cada uno de los problemas.	80
Tabla 4. Combinación óptima de parámetros para cada tamaño (maximiza valor objetivo).....	83
Tabla 5. Combinación óptima de parámetros para cada tamaño (minimiza número de iteraciones).	84
Tabla 6. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas pequeños con todos los métodos de resolución. .	87
Tabla 7. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas medianos con todos los métodos de resolución...	88
Tabla 8. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas grandes con todos los métodos de resolución.....	89
Tabla 9. Diferencia (% del valor objetivo) entre método GRB y LR1 con semillas A y B.....	97
Tabla 10. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas pequeños.....	100
Tabla 11. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas medianos.....	100
Tabla 12. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas grandes.	101

CAPÍTULO 1. Introducción

1.1. Justificación

El transporte aéreo es uno de los medios de transporte más complejos del momento y de los que cada vez están más solicitados. Su demanda continúa creciendo a un ritmo elevado y se ha estimado que en 2030 el número de pasajeros por kilómetro transportado (RPK) duplicará al actual.

En los últimos años se han incrementado notablemente los problemas en los aeropuertos, con un aumento en retrasos y congestiones que tienen un alto impacto negativo en la economía. La complejidad e importancia del tema ha provocado que se intensifique el estudio sobre cómo mejorar estos sistemas y se han propuesto nuevas técnicas para llevar a cabo.

El mecanismo que se utiliza para el ajuste entre capacidad y demanda es un sistema de asignación de franjas horarias (slots), que consiste en definir un conjunto de intervalos de tiempo de uso de las instalaciones y asignarlos a los diferentes operadores de vuelo a través de diferentes métodos.

El curso pasado, en nuestro grupo de trabajo, se elaboraron dos programas que permiten la resolución del problema de asignación de franjas horarias: uno mediante programación lineal a través de Gurobi (Pascual Pulido, 2018) y otro a partir del método del subgradiente de la relajación lagrangiana (interpretable también como una subasta combinatoria) (de la Fuente, 2016). Ambos métodos están basados en el modelo del problema de (Araújo, 2018). En este proyecto se va a generar una batería de problemas aleatoriamente con los que comparar ambos procedimientos.

1.2. Objetivos

Debido a la importancia y a la actualidad del problema de asignación de franjas horarias, en el presente documento se va a realizar un estudio de sus características y de dos de los métodos más referenciados para su resolución.

Para ello, se plantean los siguientes objetivos parciales:

- Comprender cómo se encuentra la situación actual del tráfico aéreo y sus principales problemas. Este campo de investigación recibe el nombre de gestión del flujo del tráfico aéreo (ATFM, *Air Traffic Flow Management*).

- Estudiar los diferentes métodos que existen actualmente para la asignación de slots, comprendiendo su funcionamiento y viendo sus principales ventajas e inconvenientes.
- Describir más en profundidad dos de los métodos más actuales para el reparto de franjas horarias, como son el método de relajación lagrangiana y la programación lineal entera mixta.
- Ajustar los parámetros de convergencia del método de relajación lagrangiana.
- Analizar y comparar los resultados obtenidos a partir de ambos métodos, con el fin de encontrar la mejor forma de optimizar este tipo de problemas.

1.3. Contenidos

En el presente documento se han explicado los conceptos e ideas más importantes sobre el problema de asignación de franjas horarias a vuelos y sus principales métodos de resolución, para posteriormente comprender el análisis computacional realizado con dichos métodos.

La parte teórica comienza en el CAPÍTULO 2, en el que se describe la situación actual del tráfico aéreo y todos los problemas que conlleva. También se explica cómo se realiza la asignación de slots en los aeropuertos. Después, en el CAPÍTULO 3, se describe en qué consiste la programación lineal entera mixta y cómo se aplica al modelado del problema de asignación de franjas horarias. A continuación, en el CAPÍTULO 4, se muestra el concepto de subasta junto a los principales tipos, centrándonos en la subasta combinatoria, ya que es la más apropiada para la asignación de slots, para posteriormente, en el CAPÍTULO 5 desarrollar el método de Relajación Lagrangiana, que nos permite simplificar el problema y resolverlo mediante el método del subgradiente basado en las subastas combinatorias.

La parte práctica se encuentra en el CAPÍTULO 6, en el que se explica cómo obtenemos los datos y los resultados utilizados para ajustar los métodos de resolución y a continuación realizar la comparación entre ambos.

Para finalizar, en el CAPÍTULO 7, se recogen todas las conclusiones del contenido del documento.

CAPÍTULO 2. Asignación de franjas horarias

2.1. Situación actual del tráfico aéreo mundial

El tráfico aéreo sufre un crecimiento continuo cada año, y cada vez de mayor magnitud. Según IATA, en 2017 volaron 4.081 millones de pasajeros, más de la mitad de la población mundial. (Luciano Potenze, 2018)

A continuación, en Ilustración 1, se puede ver que con el paso de los años Asia/Pacífico ha quitado el liderazgo a Medio Oriente y que una vez superada la crisis de Brasil, América Latina volvió al entorno del 6%. Es importante ver, que aunque las evoluciones por región son variables, el tráfico mundial de pasajeros siempre crece.

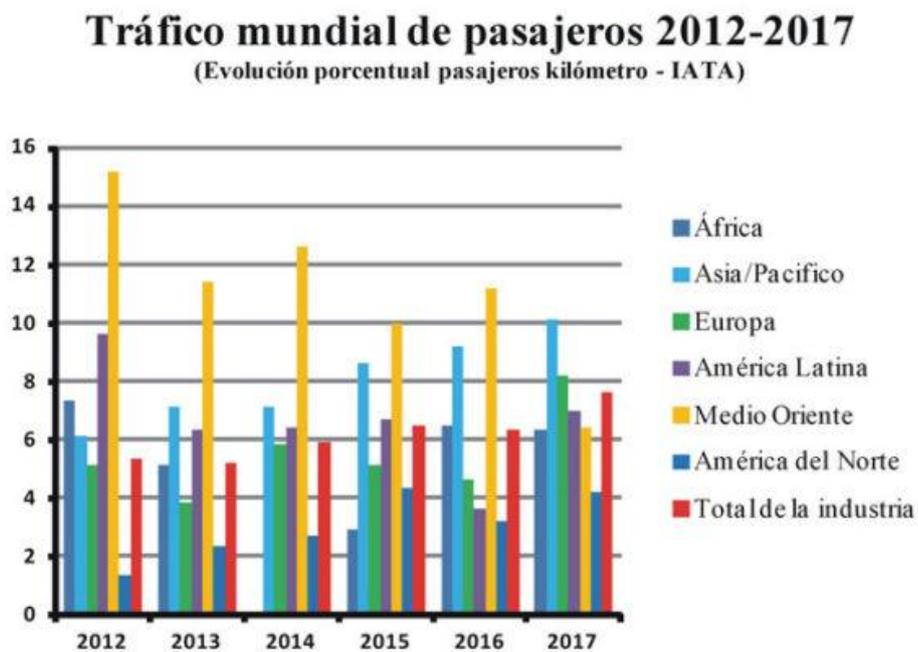


Ilustración 1. Evolución porcentual de pasajeros por kilómetro de los últimos seis años. Fuente: IATA (2018)

En el último año, también se ha visto un crecimiento, que según la Organización Internacional de Aviación Civil (OACI), es debido a las compañías Low Cost (de bajo coste). Esta afirmación se debe a que estas aerolíneas tuvieron un crecimiento de sus volúmenes de tráfico más alto que la media mundial, con unos 1.200 millones de pasajeros.

Según Olumuyiwa Benard Aliu, presidente de la OACI, el motivo del crecimiento es la continua mejora de la seguridad, protección, eficiencia e impacto tecnológico.

Además, los datos publicados por la Aviation Safety Network, manifiestan que el año 2017 fue el más seguro de todos los tiempos tanto por el número de víctimas como el de accidentes. En 2017 hubo 10 accidentes con 44 víctimas, y por ejemplo, en 2016 hubo 16 accidentes y 303 víctimas mortales. Esto demuestra el gran avance en seguridad que se ha logrado realizar año tras año.

Todos estos factores hacen que la demanda de vuelos aumente a un ritmo más alto en todas las regiones. Las compañías aéreas de todo el mundo transportaron 4.100 millones de pasajeros en 2017, aumentando un 7,1% respecto al año anterior, en el que también creció 6% en relación a 2015, según datos de la agencia de la ONU especializada en transporte aéreo. (Noticias El Universo, 2018)

En cuanto al futuro, según un estudio realizado por el fabricante aeronáutico Airbus, la demanda de tráfico aéreo crecerá a una media anual de un 4,7%, provocando que en el año 2032 se dispare el tráfico mundial de pasajeros, llegando a los 6.700 millones. La demanda aumentará en cada país de acuerdo al crecimiento de sus economías, y llegaremos a tener a Asia/Pacífico a la cabeza del mundo. (Ramón Vilarasán, 2014)

Esta situación de continuo crecimiento va a generar una serie de problemas que se verán a continuación.

2.2. El problema de la capacidad aeroportuaria

2.2.1. Principales problemas

Todo este crecimiento hace que exista un significativo exceso de demanda. La demanda de acceso al aeropuerto viene dada por los vuelos que han programado las líneas aéreas y no la aviación general, y cuando esta demanda excede la capacidad de la pista, aparecerán colas y retrasos.

Las operaciones aeroportuarias implican una serie de procesos, desde las operaciones de pasajeros y carga en los edificios de la terminal, hasta las operaciones de las aeronaves en las puertas, plataformas, calles de rodaje, pistas y espacio aéreo de la terminal. El sistema de pista actúa como principal cuello de botella en los aeropuertos más ocupados, por lo que el rendimiento de un aeropuerto está limitado principalmente por la capacidad del sistema en pista.

Este rendimiento muestra una variabilidad significativa, ya que algunos factores de los que depende (como las condiciones climatológicas), no pueden ser

controlados y están sujetos a incertidumbre. Por ello, cualquier valor de capacidad es solo una estimación, un valor esperado.

Los motivos más frecuentes por los que pueden producirse colapsos son: la propagación de perturbaciones operativas en una red de aeropuertos, es decir, si hay una demora en una aeronave, puede provocar cambios en las horas de los vuelos de la misma aerolínea y cambia la dinámica de todo el día de operaciones. También se puede producir por interrupciones imprevistas en las operaciones de la compañía, como embarques tardíos o problemas mecánicos. Pero sin duda, el motivo más frecuente e importante de la aparición de congestión es el exceso de demanda. (Jacquillat, 2018)

Este problema se traduce en un coste que engloba: los costes directos a las líneas aéreas (mantenimiento, combustible...), costes a los pasajeros (cancelaciones de vuelos, conexiones perdidas...) y los costes sociales (pérdidas de demanda de vuelos o impactos en el PIB del país).

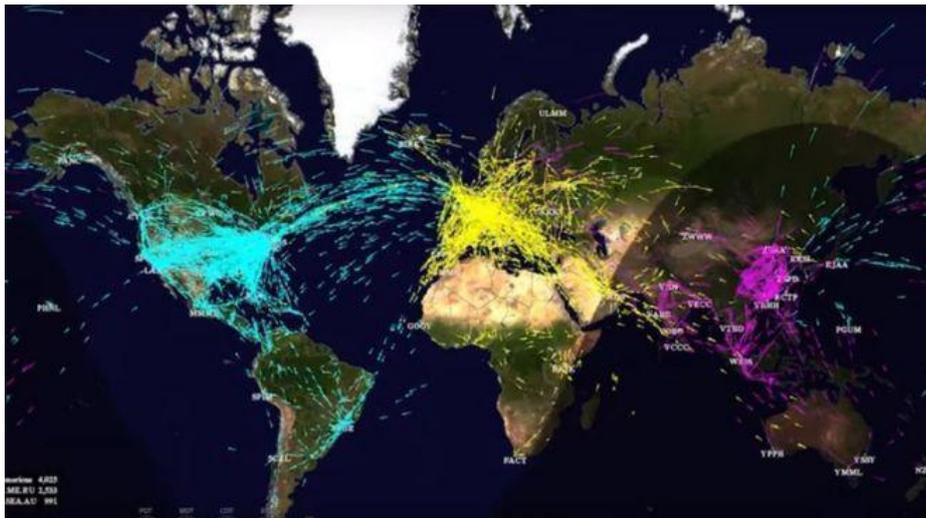


Ilustración 2. Mapa del sistema FACENT, que muestra la congestión del tráfico aéreo mundial. Fuente: NASA

Las líneas de colores que aparecen en Ilustración 2, muestran la gran cantidad de aviones que en un solo día cruzan el mundo. Se puede apreciar que hay alta congestión en el tráfico aéreo mundial.

Considerando el casi seguro crecimiento a medio y largo plazo del transporte aéreo, se pronostica que la situación va a ser muy complicada, hasta tal punto que se estima que alrededor del 70% de los primeros cincuenta aeropuertos europeos tendrán su capacidad totalmente saturada para el año 2025 (T&E, 2005).

Se agravarán de forma significativa todas estas consecuencias, por lo que es totalmente necesario hacer cambios para mejorar el modelo.

2.2.2. Posibles soluciones

La gestión del tráfico aéreo (Air Traffic Management, ATM), ha evolucionado a ATFM, una gestión del flujo cuyo objetivo es emparejar las capacidades de los aeropuertos con la demanda de los mismos, de forma que las aeronaves vuelen por el espacio aéreo de forma segura y eficiente (Odoni, 1987).

Todos los problemas que surgen con las operaciones aeroportuarias afectan a muchas partes interesadas, como son las Autoridades de Aviación Civil, operadores de aeropuertos, aerolíneas comerciales, pasajeros, comunidades locales, etc. Por lo tanto, controlar la demanda y la capacidad de los aeropuertos crea complejos desafíos para tomar decisiones en el transcurso de los ciclos de vida de los aeropuertos (Zografos, Andreatta, & Odoni, 2013).

Para poder eliminar o reducir el desequilibrio entre capacidad y demanda, se puede aplicar alguna de estas opciones:

1. Aumentar la capacidad: consiste en llevar a cabo un proyecto a corto/medio plazo, haciendo una ampliación de los aeropuertos existentes, o bien una propuesta a medio/largo plazo en la que se construirían nuevos aeropuertos, pero aparecerían una gran cantidad de restricciones a nivel social, económico o medioambiental.
2. Reducir la demanda: la forma más común de hacerlo es fomentando el uso de otros medios de transporte, que también satisfagan la ventaja de la rapidez, como los trenes de alta velocidad.
3. Innovar en tecnología: se llevan a cabo diferentes medidas como mejorar la capacidad de las aeronaves, redistribuir las terminales para disminuir el tiempo desde la entrada hasta el acceso al avión o la automatización de la tarjeta de embarque. Todas ellas con una inversión más baja y un menor número de problemas legales que en el caso 1.
4. Gestionar: se adoptan diferentes procedimientos administrativos o económicos, con el objetivo de equilibrar la demanda y la capacidad aeroportuaria.

Si comparamos las cuatro posibles actuaciones, según (Herrera García, 2006) la opción más asequible es la gestión. Hay que conseguir una mejora en la

asignación de franjas horarias, para una mejor utilización de los recursos existentes. A continuación, se explicará con detalle en qué consiste la asignación de slots y cómo se lleva a cabo.

2.3. Procedimiento para la asignación de franjas horarias

Según (Ball, Berardino, & Hansen, 2018), las franjas horarias o slots son espacios de tiempo, que se forman dividiendo el día en una serie de ventanas de tiempo, cuyo ancho es de entre 15 minutos y 1 hora, y para cada ventana se limitan un determinado número de slots.

Un slot le permite a su propietario el derecho de realizar una operación de vuelo de entrada o salida durante un período de tiempo determinado. Las directrices de programación de IATA en todo el mundo, indican que el uso de franjas horarias deben cumplir las siguientes características:

1. Derechos de abuelo: se asignan slots a aerolíneas según prácticas históricas de programación, es decir, una regla antigua continúa aplicándose a situaciones existentes.
2. Regla de uso o pérdida: las titulares de los slots mantendrán sus derechos siempre que la den uso, cumpliendo las normas requeridas.

El método de asignación de slots se realiza en base a unos parámetros técnicos, operativos y medioambientales, y se lleva a cabo mediante los siguientes subprocesos, según (Delgado Díez, 2014) y (Díaz Olariaga & Benito Ruiz de Villa, 2010).

2.3.1. Comercialización primaria de slots

Según este proceso, los slots son asignados a las compañías aéreas por algún miembro del organismo de gestión del aeropuerto, como puede ser el gobierno o el operador de ese mismo aeropuerto.

Si las personas encargadas de esa función, pertenecen al aeropuerto o al entorno de las líneas aéreas, hay riesgo de que la decisión no sea imparcial.

La principal función de este tipo de comercialización es detectar las líneas aéreas que tienen alto interés y disponibilidad de pagar por un slot determinado y poder garantizar que esa franja horaria sea asignada a esa misma compañía

(Czerny & Tegner, 2002). Para ello definen unos precios para los slots que equilibren la demanda procedente de las líneas aéreas y la capacidad utilizable en los aeropuertos.

Según cómo se determinen los precios de los slots, tenemos varios tipos de asignación primaria, que pueden ser:

- Fijado por la parte vendedora, en función de la relación oferta-demanda de ese slot. Su principal dificultad es la complejidad de fijar un precio óptimo de mercado.
- La parte vendedora saca a subasta el slot para ver qué precio se pagaría por él. También es un proceso complicado, ya que hay que estudiar cuál es el tipo de subasta más apropiado para obtener los resultados más favorables.

Hay una gran diferencia entre los dos mecanismos anteriores. Si el precio es fijado por la parte vendedora, y no es el valor óptimo (más alto) ajustado al mercado, tendrá que sufrir el efecto de un gran aumento de la demanda de dichos slots, mínimo hasta la temporada siguiente cuando ese valor ya pueda ser modificado.

En cambio, si el precio se fija tras pujar en diversas rondas, el valor del slot aumenta hasta que se compense la demanda por dicho slot.

2.3.2. Comercialización secundaria de slots

Todos los slots que hayan sido ya asignados con un mecanismo primario (excepto los concedidos a nuevas compañías), posteriormente van a poder formar parte de este tipo de comercialización, ya que sus propietarios podrán realizar transacciones con ellos, como:

- Intercambio directo entre compañías
- Cambios de rutas de slots dentro de la aerolínea
- Transferencia de slots a otras líneas aéreas

Estos intercambios y transferencias de slots deben notificarse al coordinador responsable. Si no se ha cumplido alguno de los parámetros técnicos, operativos, medioambientales, o la normativa de la propia compañía, éste puede desautorizar la modificación que se ha producido en esta segunda fase.

Las formas más comunes de implementar este tipo de mercado son las siguientes:

- Ofrecer una libertad completa a las aerolíneas para la compra y venta de las franjas horarias. Es el método más sencillo, ya que se reduciría a negociaciones directas entre las compañías a las que las interesa comprar y a las que las interesa vender. (Matthews & Menaz, 2003)
- Crear un 'mercado de slots', un lugar en donde los slots pueden ser comprados y vendidos libremente, con un comportamiento similar al del mercado de valores.

Para la asignación secundaria de slots hay que tener en cuenta:

1. Una aerolínea no puede recibir dinero por el intercambiar o transferir ningún slot.
2. Para poder participar es imprescindible haber tenido una asignación primaria con coste. Esto es para evitar que si el encargado del mecanismo de asignación primaria regala un slot a una aerolínea, ésta posteriormente podrá alquilar o vendérselo a otra y obtendría un beneficio sin coste.
3. No pueden participar los slots asignados a compañías nuevas, durante los 2 primeros periodos de programación.
4. Si ha pasado el 20% del periodo de programación y un slot está sin usar, se devuelve al "fondo de slots".

2.4. Conclusiones

Tras el notable aumento de la demanda en el sector de la aviación de los últimos años, se han incrementado los problemas de colapsos y retrasos, por lo que es necesario actuar ante tal hecho.

La forma actual de asignación de slots, tanto primaria como secundaria, mediante una buena gestión puede ser mejorada, en lugar de invertir en tecnología o ampliaciones de aeropuertos.

En resumen, con una gestión óptima de los recursos existentes en un aeropuerto es posible reducir e incluso solucionar el problema de desequilibrio entre capacidad y demanda.

CAPÍTULO 3. Programación lineal entera mixta

3.1. Introducción

La programación matemática es una de las herramientas más utilizadas para la asignación de recursos, por lo que vamos a estudiar su aplicación al problema descrito en el CAPÍTULO 2.

Según (Boirivant, 2009), “cualquier fenómeno en que interviene un número determinado de variables no negativas, que se pueden ligar entre sí mediante relaciones de desigualdad o igualdad y que reflejen las limitaciones o restricciones que el fenómeno presenta con miras a optimizar un objetivo, puede ser formulado como un modelo de programación matemática. Si tanto las restricciones como la función objetivo se pueden enunciar mediante expresiones lineales, estamos frente a un campo particular de la programación matemática denominada programación lineal”.

Debido a esta afirmación, el problema que nos ocupa se podrá modelar como función objetivo a optimizar y unas restricciones que acoten los posibles valores de las variables. Este modelo será lineal.

Por ello, en este capítulo vamos a ver cómo aplicamos la programación lineal a la formulación y resolución del problema de asignación de franjas horarias en un aeropuerto.

3.2. Programación lineal

La programación lineal es una potente herramienta matemática, que se aplica en numerosos problemas. Vamos a explicar los conceptos más importantes describiendo un problema general y los principales métodos para su resolución.

3.2.1. Descripción

Para que un problema pueda modelarse y resolverse mediante programación lineal (PL: linear programming), debe cumplir:

1. Existe una función objetivo (de maximización o de minimización)
2. Existen restricciones, es decir, los recursos son limitados.

3. Hay una relación lineal entre la función objetivo y las restricciones.
4. Todas las funciones del modelo son lineales.

En función de los valores que pueden tomar las variables del modelo, el problema puede ser de:

1. Programación lineal continua: el valor de las variables puede ser fraccionaria.
2. Programación entera (PE: integer Programming Model): el valor de todas las variables debe ser entero.
3. Programación Entera Mixta (PEM: Mixed integer Programming Model): el valor de sólo alguna de las variables debe ser entero.

Los modelos de Programación Entera Pura no son muy habituales, los problemas prácticos se resuelven en la mayoría de los casos con programación entera mixta. En este tipo de programación no existe un algoritmo que converja, por lo que los tiempos de resolución de los problemas en algunos casos son muy elevados.

Un tipo de problema en el que se utiliza la programación entera mixta, es un problema combinatorio de asignación de productos, por lo que es un modelo apropiado para el problema de reparto de slots. (López, 2003).

3.2.2. Formulación de un problema general

A continuación se va a describir un problema general de programación lineal, basado en las ideas de (Castillo, Conejo, Pedregal, Garcia, & Alguacil, 2002).

Todo problema de programación lineal tiene una función objetivo a maximizar o minimizar de la manera:

$$Z = f(x) = \sum_{j=1}^n c_j x_j \quad (1.1)$$

Sujeto a unas restricciones del tipo:

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, 2, \dots, p-1 \quad (1.2)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = p, \dots, q-1 \quad (1.3)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = q, \dots, m \quad (1.4)$$

En las que los parámetros p , q y m toman valores enteros positivos, cumpliendo:

$$1 \leq p \leq q \leq m \quad (1.5)$$

Se puede observar que todas las funciones son lineales. En el caso de que una de ellas no lo fuera, pasaría directamente a ser un problema de programación no lineal.

Algunos de los conceptos más importantes son:

- Solución factible: cualquier punto $x = (x_1, x_2, \dots, x_n)$ que satisface las restricciones (1.2), (1.3), (1.4).
- Región de factibilidad: conjunto de todas las soluciones factibles.
- Solución óptima: un punto factible x_a , que cumple $f(x) \geq f(x_a)$ para cualquier otro punto factible x .
- Si la región factible está acotada, el problema siempre tendrá solución.
- El óptimo de un problema de programación lineal, es siempre un óptimo global.
- Si x e y son soluciones óptimas, también es una solución óptima su combinación lineal convexa.
- La solución óptima se consigue siempre, al menos, en un punto extremo de la región factible.

3.2.3. Métodos de resolución

Una vez modelado el problema, el siguiente paso es resolverlo. Para ello, pueden utilizarse distintos métodos en función de las condiciones del problema formulado. (Bazaraa, 1998)

Si las variables del problema son continuas, podemos utilizar los siguientes métodos, entre otros:

1. Método gráfico:

Las restricciones se dibujan gráficamente en unos ejes coordenados y la región delimitada por el conjunto de rectas nos muestra las soluciones factibles.

Las soluciones óptimas se encuentran en el perímetro de esa región.

En este método existe la limitación para problemas con más de dos variables de decisión y de más de 3 incógnitas.

2. Método Simplex:

Este método se utiliza para problemas cuyas restricciones sean del tipo menor o igual (\leq), cuyos coeficientes independientes no sean negativos. En el caso de que no lo sean se pueden estandarizar o usar otro método.

Consiste en un método iterativo, en el que en cada paso se mejora la función objetivo. Cuando en una ronda no se mejora ese valor, se para y ese valor es la solución óptima del problema.

3. Método de las 2 fases:

Este método se usa en problemas cuyas restricciones son del tipo mayor o igual (\geq), o igual (=).

Se transforman estas restricciones a otra forma, que sí que permiten la resolución mediante el método Simplex. Se realiza añadiendo una nueva variable (Y_k), llamada holgura, para convertir la desigualdad en igualdad. A continuación se muestra el cambio, siendo (1.6) la restricción inicial y (1.7) la restricción que ya puede ser utilizada en el método Simplex.

$$\sum a_{ij} \cdot X_j \geq b_i \quad (1.6)$$

$$\sum a_{ij} \cdot X_j - Y_k \geq b_i \quad (1.7)$$

4. Cambio de variable:

Este método se aplica a problemas que tienen variables acotadas inferiormente por alguna restricción.

Tras ciertos cambios en dichas variables, para que queden de la forma $X_i \geq 0$, ya es posible resolverlo mediante el método Simplex.

En el caso de que no todas las variables del problema sean continuas, se necesita utilizar otro tipo de métodos para su resolución. El problema de asignación de franjas horarias, debido a sus características, es considerado de programación entera mixta, por lo que a continuación explicamos el método de branch and bound, un algoritmo apropiado para su resolución.

3.3. Método de branch and bound

Una vez vistos los modelos más comunes de resolución de un problema de programación lineal continua, vamos a explicar detalladamente el algoritmo de branch and bound (en español, Ramificación y Acotamiento). Este método se basa en los explicados anteriormente para encontrar la solución.

El motivo de su descripción más detallada es que es el método en el que se basa Gurobi, la herramienta de resolución que vamos a utilizar para los experimentos que realizaremos en el CAPÍTULO 6.

3.3.1. Descripción del método

El método de branch and bound, o método de ramificación y acotación, se va a usar para resolver problemas modelados con programación entera.

Su procedimiento consiste en relajar el problema de programación entera y en el caso de que alguna de las variables de decisión sea fraccionaria, se generan cotas (restricciones adicionales), que permitirán la obtención de números enteros para las variables de decisión. (Gestión de Operaciones, 2011).

Para empezar a resolverlo se utiliza el problema relajado, es decir, no se tiene en cuenta que las variables de decisión tienen que ser enteras porque eliminamos las condiciones de integridad. Cada problema resuelto lo dividimos en 2 subproblemas, y cada ramificación va a añadir una restricción extra al problema inicial, por lo que en cada una de ellas, el valor de la función objetivo va a aumentar en el caso de una minimización y va a disminuir en el caso de una maximización.

Si solo utilizáramos la ramificación, el problema va creciendo exponencialmente, por ello también se aplica la acotación. Consiste en utilizar como cota inferior (en el caso de maximización) el mayor valor entero que se ha obtenido como solución de la función objetivo. Estas cotas nos permiten no ramificar los subproblemas que tengan un valor de solución óptima inferior a la cota fijada. De este modo se reduce el tiempo de resolución del método. (Alvarado Ortega, 2016)

Una vez explicado en qué consiste el método, vamos a ver con más detalle los pasos para su resolución.

3.3.2. Algoritmo

El algoritmo de resolución del método branch and bound está formado por los siguientes pasos (Medrano, 2015):

1. Obtener la solución del problema relajado utilizando el Método Simplex. Si la solución está formada por variables enteras paramos, en caso contrario pasamos al punto 2.
2. Elegimos una de las variables enteras del problema (x_i), cuyo valor en la resolución ha sido fraccionaria.
3. Se crean dos ramificaciones, excluyentes entre sí, con las que obtenemos 2 nuevos problemas de programación lineal. Las restricciones que hay que añadir a cada uno de ellos son: $x_i < [x_i]$ en uno y $x_i > [x_i] + 1$ en el otro.
4. Resolvemos ambos problemas con las nuevas restricciones. Si la solución es entera, paramos. Si no lo es, seguimos ramificando, creando nuevos problemas y repitiendo todos los pasos hasta encontrar la solución entera óptima.

A continuación, en la Ilustración 3, se muestra gráficamente el algoritmo:

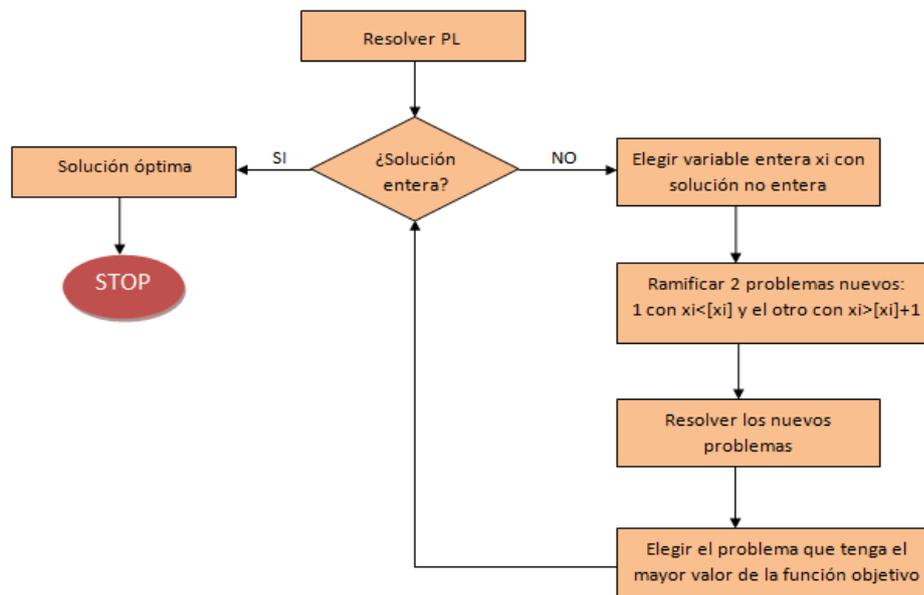


Ilustración 3. Algoritmo de Branch and Bound. Fuente: Adaptación de (Alvarado Ortega, 2016).

3.4. Modelo de programación lineal para asignación de slots aeroportuarios.

Una vez que se ha descrito la programación lineal, se va a utilizar para la creación de un modelo que representa el problema de asignación de franjas horarias en aeropuertos.

Se van a describir, según (Araúzo, 2018), los parámetros, las variables, la función objetivo y las restricciones de un modelo de programación lineal para un solo aeropuerto, y con el horizonte de planificación cerrado.

3.4.1. Parámetros del modelo

Parámetros generales:

T : cantidad de slots disponibles en un aeropuerto. Cada uno de ellos es denotado por $t = \{0, 1, 2, \dots, T-1\}$.

Parámetros sobre los vuelos:

M : cantidad de vuelos que se van a realizar en un aeropuerto. Cada uno se escribe de la forma $m = \{1, 2, \dots, M\}$, e incluye una franja horaria para la llegada y otra para la salida.

gu_m : utilidad bruta para el vuelo m , es decir, la utilidad de la compañía si el vuelo se programa en su franja deseada.

$lart_m$: mínima franja horaria deseada para la llegada del vuelo m .

$rart_m$: máxima franja horaria deseada para la llegada del vuelo m .

$ldrt_m$: mínima franja horaria deseada para la salida del vuelo m .

$rdrt_m$: máxima franja horaria deseada para la salida del vuelo m .

mtb_m : tiempo mínimo (en número de franjas horarias) que debe transcurrir entre las operaciones de llegada y salida.

cab_m : penalización del vuelo m por programar la llegada antes de $lart_m$.

caa_m : penalización del vuelo m por programar la llegada después de $rart_m$.

cdb_m : penalización de m por programar la salida antes de $ldrt_m$.

dca_m : penalización de m por programar la salida después de $rdrt_m$.

fk_m : tipo de vuelo.

Si $fk_m = 1$ el vuelo tiene la base en otro aeropuerto (llegan, luego salen).

Si $fk_m = -1$ el vuelo tiene la base en este aeropuerto (salen, luego llegan).

dca_{mt} : coste de desviación de programar la llegada en otra franja horaria t .

$$dca_{mt} = \begin{cases} cab_m \cdot (lart_m - t) & \text{si } t < lart_m \\ 0 & \text{si } lart_m \leq t \leq rart_m \\ caa_m \cdot (t - rart_m) & \text{si } t > rart_m \end{cases} \quad (1.8)$$

dcd_{mt} : coste de desviación de programar la salida en otra franja horaria t .

$$dcd_{mt} := \begin{cases} cdb_m \cdot (ldrt_m - t) & \text{si } t < ldrt_m \\ 0 & \text{si } ldrt_m \leq t \leq rdrt_m \\ cda_m \cdot (t - rdrt_m) & \text{si } t > rdrt_m \end{cases} \quad (1.9)$$

Parámetros sobre el aeropuerto:

H : cantidad de restricciones de capacidad rodantes aplicables sobre las franjas horarias. Cada una de ellas $h = \{1, 2, \dots, H\}$ afecta a T_h franjas consecutivas. Si $T_h = 1$ la restricción será equivalente a una restricción de capacidad normal.

ac_{ht} : capacidad de llegadas de la restricción h , es decir, cantidad de llegadas permitidas en el conjunto de intervalos que van desde t hasta $(t+T_h-1)$. Si $T_h=1$, la restricción se referirá a un solo intervalo (número de llegadas permitidas en el intervalo t)

dc_{ht} : capacidad de salidas de la restricción h , es decir, número de salidas permitidas en el conjunto de intervalos que van desde t hasta $(t+T_h-1)$.

tc_{ht} : capacidad de movimientos totales de la restricción h , es decir, cantidad de movimientos (salidas + llegadas) permitidas en el conjunto de intervalos que van desde t hasta $(t+T_h-1)$.

3.4.2. Variables

Las variables que conforman el modelo son las siguientes:

ta_m : franja horaria asignada a la llegada del vuelo m .

td_m : franja horaria asignada a la salida del vuelo m .

También es necesaria la existencia de las siguientes variables binarias:

δa_{tm} :

$\delta a_{tm} = 1$, si llegada del vuelo m se programa en franja horaria t .

$\delta a_{tm} = 0$, si llegada del vuelo m se programa en una franja no programada, es decir, distinta a t .

δd_{tm} :

$\delta d_{tm} = 1$, si la salida del vuelo m se programa en franja horaria t .

$\delta d_{tm} = 0$, si la salida del vuelo m se programa en una franja no programada, es decir, distinta a t .

Para nuestro modelo, la demanda de slots no está limitada y puede superar la capacidad del aeropuerto, por lo que no todos los vuelos van a ser aceptados. Para saber la aceptación de un vuelo m , se utiliza la variable:

δ_m :

$\delta_m = 1$, si el vuelo m es programado.

$\delta_m = 0$, si el vuelo m es rechazado.

La relación entre las variables temporales y las binarias es la siguiente:

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} = ta_m \quad (1.10)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} = td_m \quad (1.11)$$

3.4.3. Modelo de programación lineal

Función objetivo:

Se desea obtener la mayor utilidad total U de los vuelos programados en el aeropuerto, por lo que será un problema de maximización. Esta utilidad total será la suma de las utilidades brutas de cada vuelo programado menos las penalizaciones.

Va a existir la penalización por desviación y se puede incluir también una penalización (ctb_m) por el tiempo que transcurre entre las dos operaciones. Este tiempo será considerado como una variable (tb_m), y para cada vuelo m cumplirá:

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - fk_m \cdot tb_m = 0 \quad (1.12)$$

Por lo que si consideramos un coste por franja horaria ctg_m de permanencia en el aeropuerto del vuelo m , nuestra función objetivo será:

$$U = \sum_{i=1}^M \left(gu_m \cdot \delta_m - \sum_{t=0}^{T-1} (dca_{mt} \cdot \delta a_{mt} + dcd_{mt} \cdot \delta d_{mt}) - ctb_m \cdot tb_m \right) \quad (1.13)$$

Restricciones entre variables:

Todo vuelo asignado al aeropuerto debe tener un slot de llegada y uno de despegue.

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} = \sum_{t=0}^{T-1} \delta d_{mt} \quad m = \{1, 2, \dots, M\} \quad (1.14)$$

El tiempo entre llegadas y salidas viene dado por:

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - fk_m \cdot tb_m = 0 \quad m = \{1, 2, \dots, M\} \quad (1.15)$$

Restricciones de programación:

El número de franjas horarias entre las dos operaciones debe ser mayor o igual que el número de franjas horarias mínimo establecido para ello mtb_m .

$$tb_m - mtb_m \cdot \delta_m \geq 0 \quad m = \{1, 2, \dots, M\} \quad (1.16)$$

Cuando $\delta_m = 0$, el vuelo no se programa y por lo tanto la restricción se cumple siempre.

Restricciones de capacidad:

Restricciones sobre llegadas: el número de llegadas en el conjunto de intervalos que van desde (t) hasta $(t+T_h-1)$ no puede superar la capacidad ac_{ht} .

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} \leq ac_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.17)$$

Restricciones sobre salidas: el número de salidas en el conjunto de intervalos que van desde (t) hasta $(t+T_h-1)$ no puede superar la capacidad dc_{ht} .

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} \leq dc_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.18)$$

Restricciones sobre los movimientos totales: el número salidas y llegadas en el conjunto de intervalos que van desde (t) hasta $(t+T_h-1)$ no puede superar la capacidad tc_{ht} .

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} (\delta a_{m\tau} + \delta d_{m\tau}) \leq tc_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.19)$$

Modelo completo

Agrupando la función objetivo y todas las restricciones formadas anteriormente, obtenemos el modelo total:

$$\max_{\delta_m, \delta a_m, \delta d_m} U = \sum_{i=1}^M \left(gu_m \cdot \delta_m - \sum_{t=0}^{T-1} (dca_{mt} \cdot \delta a_{mt} + dcd_{mt} \cdot \delta d_{mt}) - ctb_m \cdot tb_m \right) \quad (1.20)$$

Sujeto a:

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} \leq ac_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.21)$$

Programación lineal entera mixta

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} \leq dc_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.22)$$

$$\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} (\delta a_{m\tau} + \delta d_{m\tau}) \leq tc_{ht} \quad \begin{array}{l} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{array} \quad (1.23)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} \quad m = \{1, 2, \dots, M\} \quad (1.24)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta d_{mt} \quad m = \{1, 2, \dots, M\} \quad (1.25)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - fk_m \cdot tb_m = 0 \quad m = \{1, 2, \dots, M\} \quad (1.26)$$

$$tb_m - mtb_m \cdot \delta_m \geq 0 \quad m = \{1, 2, \dots, M\} \quad (1.27)$$

$$\delta a_{mt}, \delta d_{mt}, \delta_m \in \{0, 1\} \quad \begin{array}{l} m = \{1, 2, \dots, M\} \\ t = \{1, 2, \dots, T - 1\} \end{array} \quad (1.28)$$

3.5. Conclusiones

La programación lineal es un método apropiado para modelar el problema de asignación de franjas horarias, ya que tanto sus restricciones como su función objetivo pueden enunciarse mediante expresiones lineales.

En los problemas prácticos, como es el caso, el valor de sólo alguna de las variables debe ser entero, por lo que estamos ante un problema de programación entera mixta.

Tras la explicación detallada del modelo completo del problema de asignación de franjas horarias, en el CAPÍTULO 6 vamos a poder resolverlo computacionalmente mediante un procedimiento basado en el algoritmo de branch and bound (ramificación y acotación).

CAPÍTULO 4. Subastas combinatorias

4.1. Introducción

El mecanismo de las subastas combinatorias es uno de los métodos más utilizados en los últimos años para resolver problemas de asignación de recursos. Algunos de los más importantes han sido: los problemas de flujos de mercancías en (Bertsekas, 1990), la organización de los tramos de ferrocarriles en (Parkes & Ungar, 2001) o la programación de los vuelos en un aeropuerto en (Rassenti, Smith, & Bulfin, 1982).

Según autores como (Dewan & Joshi, 2002), también es un método muy útil para coordinar la producción de forma distribuida.

En este capítulo se va a explicar en qué consiste el método de la subasta, cuáles son los principales tipos, según (Laviós Villahoz, 2013) y cómo se va a aplicar al problema descrito en el CAPÍTULO 2, a la asignación de franjas horarias en un aeropuerto.

4.2. Definición

Una subasta es un proceso de venta organizado, en el que los demandantes o compradores compiten entre ellos para conseguir el bien o servicio que está siendo ofertado en esa subasta. Este tipo de adjudicación de bienes y servicios es una forma de determinar un precio de equilibrio, de manera alternativa a los mercados. (Roldán)

Durante el proceso, los interesados ofrecen cantidades de dinero, que se comparan con las del resto para poder elegir al vencedor.

El objetivo principal de una subasta es maximizar las ganancias de la venta, consiguiendo que todas las personas interesadas desvelen el valor que le dan a ese bien o servicio subastado. Aunque la valoración es algo privado, y no siempre se consigue saber.

Antes de empezar la subasta se deben fijar unas normas, que deben ser conocidas por todos. Dependiendo de esas reglas, hay varios tipos de subastas.

4.3. Tipos de subastas

Este procedimiento, puede ser de muchos tipos dependiendo del parámetro que utilicemos para clasificarlo. Vamos a ver la clasificación en función de:

1. Número de participantes:

- Directa: un vendedor y varios compradores.
- Inversa: un comprador y varios vendedores.
- Doble: varios compradores y varios vendedores.

2. Tipo de producto subastado:

- Divisible: puede repartirse en partes proporcionales.
- Discreto: no puede ser dividido.

3. Número de artículos subastados:

- Uno: un único bien subastado.
- Varios: si los bienes subastados son independientes entre sí pueden realizarse subastas de forma simultánea, pero si van en paquete, el valor de cada uno de los elementos que lo conforman está determinado por el resto de productos que forman el lote, usando para ello una subasta combinatoria.

4. Método de realización:

- Abiertas: los participantes pujan de forma continua y conociendo las valoraciones del resto de ofertantes.
- Cerradas: las pujas de los participantes no son conocidas por el resto a la hora de realizar su valoración. Se realiza en una sola ronda.

5. Precio al que se intercambia:

- Al primer precio: la persona a la que se le adjudica el producto, es la que ha hecho la oferta más alta.
- Al segundo: la persona a la que se le adjudica el producto, es la que ha hecho la segunda oferta más alta.

Para el problema de reparto de slots, el tipo de subasta que vamos a utilizar es la subasta combinatoria. Vamos a ver en qué consiste y cómo se va a aplicar al problema citado anteriormente.

4.4. Subastas combinatorias

A continuación, se va a describir el funcionamiento de una subasta combinatoria, ya que es el proceso más adecuado para resolver el problema de asignación de franjas horarias.

4.4.1. ¿Por qué es el más apropiado para el reparto de slots?

Un sistema efectivo de subasta de slots debería permitir:

1. Sacar a la venta todas las franjas horarias simultáneamente.
2. Los participantes pueden ofertar pujas y valoraciones en paquetes de franjas horarias (conjunto de todas las ranuras por las que están interesados, ya que están relacionadas entre sí para sus intereses).

Los sistemas que hacen posibles estas condiciones son llamados subastas combinatorias, y son las que vamos a explicar aplicado al problema de asignación de slots.

En este tipo de subastas, las pujas se realizarán sobre lotes de productos entre los que existen complementariedades, como por ejemplo, una relación de precedencia.

Son las más apropiadas porque al ser un proceso iterativo, los participantes van conociendo las prioridades del resto de los ofertantes, y se van ajustando las pujas de forma más efectiva a causa de adquirir esta información.

4.4.2. Pasos de resolución de una subasta combinatoria

Una vez que se ha definido el tipo de subasta que vamos a utilizar, se va a explicar cómo funciona.

Es una subasta iterativa, es decir, hay un conjunto de pasos que se repiten en cada iteración (actualizando los parámetros) hasta que se active la condición definida de parada.

Los pasos para cada iteración son los siguientes:

1. Inicialización de variables: el responsable de la subasta fija el precio inicial de los productos subastados. Lo normal es empezar por cero.

2. Envío de los precios a todos los participantes.
3. Cálculo de la puja: cada uno de los demandantes establece su oferta. La puja de cada participante consiste en la combinación de elementos que quiere adquirir, con los precios fijados en el paso 1.
4. Valoración de las pujas de los demandantes: se combinan las propuestas de los demandantes y se valoran en cada instante de tiempo, las necesidades de capacidad de cada recurso.
5. Creación de un programa factible: se detectan las restricciones no cumplidas y se construye una solución factible, aumentando los precios de los productos que tengan exceso de demanda.

Estos pasos se realizan iterativamente, hasta que se alcanza el criterio de parada. En este caso se termina cuando el sistema llegue al equilibrio, o los precios se estabilicen. La solución será la establecida en el paso 4.

A continuación, en la Ilustración 4, se describen estos pasos de forma gráfica.

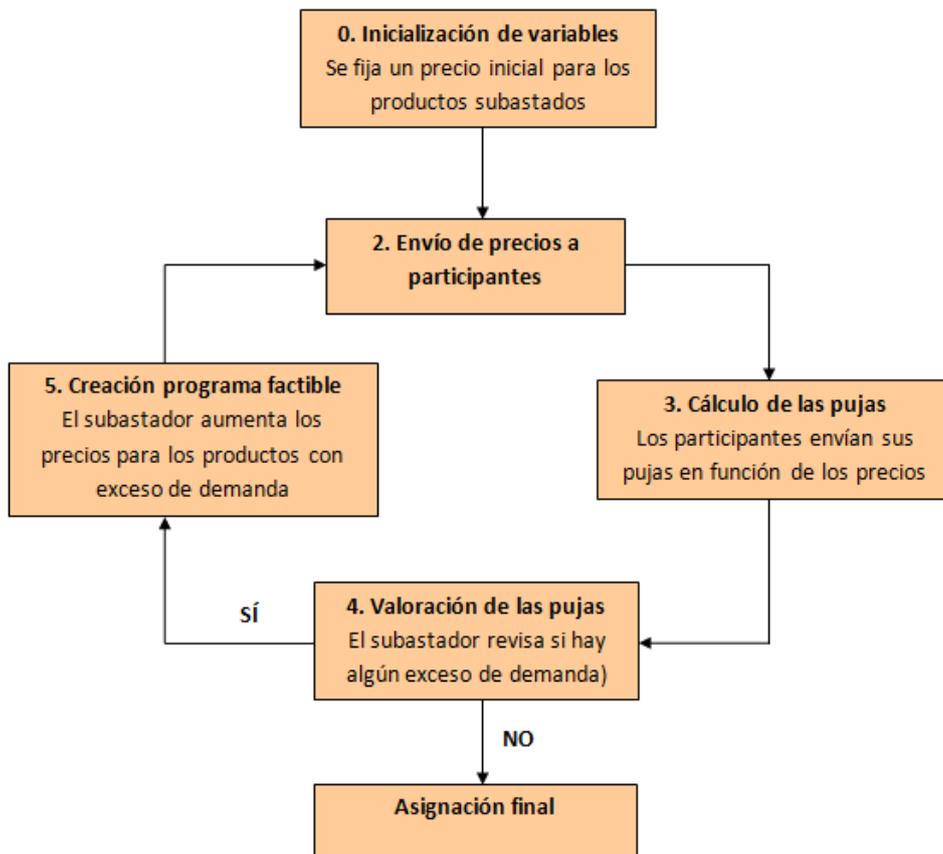


Ilustración 4. Proceso de subasta combinatoria. Fuente: adaptación de (Forde & Doyle, 2008).

4.5. Aplicación de subasta combinatoria a problema de asignación de franjas horarias

Una motivación para utilizar este método de repartición es permitir que la asignación sea guiada por las fuerzas del mercado, para que sea mucho más eficiente. Este efecto no es fácil de conseguir, debido a que las ranuras tienen restricciones como el tipo de aeronave o la categoría del vuelo.

Los mecanismos de subasta combinatoria, pueden utilizarse para cualquier número de productos y restricciones laterales. Pero hay que tener en cuenta que a medida que la cantidad de productos crece y las restricciones se vuelven más complejas, este método se vuelve menos efectivo.

Según (Ball, Berardino, & Hansen, 2018), los principales conceptos sobre subastas combinatorias para el reparto de franjas horarias son los que se describen a continuación.

4.5.1. Producto subastado

Cuando aplicamos el método de subasta al problema del tráfico aéreo, lo primero es definir cuál va a ser el producto subastado. Como se vio en el Asignación de franjas horarias, un día se reparte en un conjunto de ranuras de tiempo, en las que el propietario de cada ranura tiene el derecho de realizar una operación durante el período de su duración, por lo que el producto subastado va a ser cada una de estas franjas, que podrán ser sólo de llegadas, sólo de despegues, o que abarquen ambas opciones.

Para cada compañía, un vuelo que parta en una franja horaria tendrá que volver, y por lo tanto necesita una ranura de llegada (también aplicable al vuelo que llega y que posteriormente necesita una ranura de salida). Además, para cada conexión entre dos ciudades intentan proporcionar múltiples vuelos por día.

Por estos dos motivos, los propietarios que pujen en esta subasta no encontrarán mucho valor en disponer de una sola ranura. Lo que les interesa es poder hacer una oferta por múltiples ranuras y, además, el valor colocado en una sola ranura dependerá del paquete de ranuras obtenido.

Cada aerolínea tiene unas preferencias de franjas horarias. Quieren adquirir aquellos slots con los que les sea posible realizar todos sus vuelos con el mínimo

coste posible. Este coste es el resultante de comprarlos y de las penalizaciones por retrasos.

4.5.2. Cantidad de productos subastados

En el caso de las subastas de lotes, debe especificarse antes el número de slots que hay disponibles en cada ventana de tiempo. La cantidad de slots está directamente relacionada con la capacidad física de la pista del aeropuerto, pero no es el único condicionante. Hay que tener en cuenta las limitaciones de otros recursos de ese aeropuerto como las puertas o el manejo de equipaje.

Ajustar el tamaño óptimo ofertado de franjas horarias es complicado porque se ve afectado por la variabilidad del número de operaciones que un aeropuerto es capaz de abarcar en un determinado slot.

Variaciones climáticas o presencia de daños en la aeronave, entre otros, son factores que aportan esa variabilidad y que transforman el establecimiento de la oferta del número de slots en un problema de gestión de los retrasos de los vuelos, que necesita la ayuda de simulaciones para la estimación de estos retrasos en distintos escenarios.

A medida que se aumentan los límites del intervalo de tiempo, los operadores aéreos pueden aumentar la capacidad general del servicio programado. Pero, a la vez, este beneficio se ve limitado por un aumento en los retrasos asociados a estos servicios.

Actualmente, las negociaciones se debaten muy arriba, y puede ser una de las causas del mal ajuste de los niveles de slots. Los operadores de los aeropuertos están motivados para maximizar el rendimiento de los pasajeros en un aeropuerto, lo que provoca una reducción de los niveles y operaciones de las franjas horarias. Aunque también puede darse el caso de que se resistan al aumento de los niveles de ranura para mantener un ambiente de trabajo menos intenso.

En resumen, el mayor impedimento para establecer niveles de ranura socialmente óptimos es que las personas encargadas de tomar esta decisión lo hacen en función al interés personal. Esta decisión la deben tomar personas que tenga a su cargo objetivos bien definidos, y sea inmune a cualquier influencia política. También está la posibilidad del diseño de un mecanismo de subasta que determine el número de slots.

4.5.3. Distribución de los ingresos

Los ingresos obtenidos en la subasta de slots, en la mayoría de aeropuertos del mundo, se limitan a cubrir todos los costes o a generar un beneficio limitado. Estos ingresos no se transfieren directamente a los operadores de aeropuertos, a menos que puedan reemplazar otras fuentes de ingresos.

Para mejorar la eficiencia económica, en cuanto a los ingresos, se deberían llevar a cabo las siguientes consideraciones:

1. Los ingresos de la subasta deben utilizarse para establecer tarifas de aterrizaje asociadas.
2. Cualquier exceso de fondos debe invertirse en el sistema de transporte aéreo local o regional.
3. El reparto de ingresos de la subasta deberá describirse en un documento legalmente vinculante antes de la ejecución de ésta.

4.5.4. Restricciones para la subasta

Para llevar a cabo este proceso de la manera más eficiente, es necesario tener en cuenta algunas consideraciones.

1. Restricciones a los propietarios:

Los lotes de slots obtenidos en las subastas no siempre son utilizados como se espera. Es posible que personas o corporaciones adineradas puedan comprar lotes para su exclusivo viaje privado, o que posibles especuladores compren, vendan o acumulen espacios.

Durante las deliberaciones sobre el uso de subastas en el aeropuerto estadounidense de La Guardia (Ball, y otros, 2007), se sugirió que este problema podría evitarse mediante restricciones en el uso de espacios.

Una posible restricción es exigir un mínimo nivel de uso para cada ranura por parte de cada compañía aérea.

2. Restricciones a la concentración del mercado:

El uso de subastas para asignar slots no evita que aparezca una concentración en el mercado. Según (Gale & O'Brien, 2013), una aerolínea dominante asume que otras aerolíneas operarán en la mayoría

o en todas sus franjas horarias, sin embargo, la aerolínea dominante no. Lo que hace es optimizar sus propios beneficios programando los vuelos justos hasta el punto en el que los beneficios netos de los retrasos sean los máximos.

Tiene que existir un control para que haya un nivel de competencia siempre que los espacios sean escasos. Para ello se puede aplicar una restricción explícita, que regule la concentración del mercado, asumiendo así un entorno más dinámico.

Hay que tener en cuenta que estas restricciones son muy importantes para lograr esos objetivos, pero a su vez, una gran cantidad de restricciones puede reducir la efectividad del uso de las subastas para lograr objetivos más grandes de eficiencia económica.

4.5.5. Frecuencia de realización de la subasta

El producto subastado, las franjas horarias, son un arrendamiento con plazo limitado. Por este motivo, es necesario realizar subastas a medida que expiren los arrendamientos.

La cantidad de slots que expiran deben escalonarse. Cada aeropuerto fija la duración de los arrendamientos, que estará entre 5 y 25 años. Suponiendo que el contrato dura 20 años, cada año, aproximadamente el 5% de los contratos expiraría. Por este motivo, en una subasta celebrada anualmente, se ponen a disposición de los participantes un 5% de los slots disponibles en cada aeropuerto.

Con esta medida, no cambiarían los propietarios de los slots todos al mismo tiempo, así que los aeropuertos tendrán menos dificultades a la hora de gestionarse.

4.5.6. Mecanismo de subasta. Subasta combinatoria con lotes.

Para la venta de bienes privados a través de una subasta, el objetivo más frecuente es maximizar los ingresos obtenidos de la venta. Lo que se quiere conseguir en el caso de una subasta de slots aeroportuarios es asignar a cada compañía las franjas horarias que permitan maximizar la eficiencia y el bienestar social.

Este procedimiento es una subasta combinatoria iterativa. Para cada iteración se llevan a cabo los siguientes pasos:

1. Para poder comenzar la subasta, hay que definir los slots ofertados y el precio de cada uno de ellos. Una vez fijados, se les informa a los participantes.

En la Ilustración 5, se puede ver una muestra de la plantilla que se le entrega a cada uno de los participantes.

Slots	Price (x1,000,000)	Numb	Total (x1,000,000)
7:00-8:00	\$0.80		
8:00-9:00	\$1.40		
9:00-10:00	\$1.30		
10:00-11:00	\$1.10		
11:00-12:00	\$0.90		
12:00-13:00	\$0.90		
13:00-14:00	\$0.80		
14:00-15:00	\$0.90		
15:00-16:00	\$1.00		
16:00-17:00	\$1.30		
17:00-18:00	\$1.40		
19:00-20:00	\$1.00		
20:00-21:00	\$0.80		
TOTAL:			

Ilustración 5. Plantilla para cada participante en la subasta. Fuente: (Ball, Berardino, & Hansen, 2018)

2. Los participantes, en cada iteración, especifican el número de espacios que quieren adquirir en cada ventana de tiempo, conociendo el precio de cada uno de ellos anteriormente fijado.

En la Ilustración 6 se observa una muestra en la que el licitante ha rellenado el vector de cantidades. Este paso se puede comparar con una persona que va al supermercado a hacer la compra. Cada producto tiene un precio asignado y el comprador sólo tiene que indicar la cantidad de cada producto que desea.

Subastas combinatorias

Slots	Price (x1,000,000)	Numb	Total (x1,000,000)
7:00-8:00	\$0.80	2	\$1.60
8:00-9:00	\$1.40	3	\$4.20
9:00-10:00	\$1.30	2	\$2.60
10:00-11:00	\$1.10	2	\$2.20
11:00-12:00	\$0.90	0	\$0.00
12:00-13:00	\$0.90	2	\$1.80
13:00-14:00	\$0.80	1	\$0.80
14:00-15:00	\$0.90	1	\$0.90
15:00-16:00	\$1.00	3	\$3.00
16:00-17:00	\$1.30	3	\$3.90
17:00-18:00	\$1.40	1	\$1.40
19:00-20:00	\$1.00	0	\$0.00
20:00-21:00	\$0.80	2	\$1.60
TOTAL (x1,000,000):			\$24.00

Ilustración 6. Muestra de una puja de un participante de la subasta. Fuente: (Ball, Berardino, & Hansen, 2018)

A partir de este dato, en la siguiente columna se calcula el precio de la oferta para el paquete de lotes, que se calcula multiplicando cada ranura por la cantidad especificada y luego sumando estos resultados para obtener un total que aparece en la celda inferior derecha.

3. A continuación, el subastador recopila las ofertas de todos los participantes y obtiene la información de sus pujas, como se muestra en la Ilustración 7.

Slots	Price (x1,000,000)	A1 TOT	A2 TOT	A3 TOT	A4 TOT	A5 TOT	A6 TOT	TOT BID	SLOT LIMIT	PRICE CHANGE
7:00-8:00	\$0.80	2	3	8	2	0	2	17	30	
8:00-9:00	\$1.40	3	10	4	8	2	8	35	30	+
9:00-10:00	\$1.30	2	12	0	8	3	14	39	30	+
10:00-11:00	\$1.10	2	12	0	6	3	12	35	30	+
11:00-12:00	\$0.90	0	8	0	3	3	6	20	30	
12:00-13:00	\$0.90	2	4	0	4	0	4	14	30	
13:00-14:00	\$0.80	1	6	4	8	0	6	25	30	
14:00-15:00	\$0.90	1	8	4	9	2	12	36	30	+
15:00-16:00	\$1.00	3	8	4	7	2	12	36	30	+
16:00-17:00	\$1.30	3	10	4	6	4	10	37	30	+
17:00-18:00	\$1.40	1	4	0	5	4	12	26	30	
19:00-20:00	\$1.00	0	2	0	5	2	2	11	30	
20:00-21:00	\$0.80	2	3	0	1	0	2	8	30	

Ilustración 7. Ejemplo de ofertas de todos los participantes. Fuente: (Ball, Berardino, & Hansen, 2018)

La columna Slot Limit, indica la capacidad máxima de slots que se ofertan en esa ventana de tiempo. Si la suma de las peticiones de todas las compañías en cada franja horaria es superior a esa capacidad, se marca

con una cruz en la columna de la derecha. Eso significa que esos precios van a ser modificados en una nueva iteración.

4. Los precios de las franjas horarias en los que la oferta supere a la demanda, se van a modificar para reducir la demanda en estos slots y conducir hacia una solución en la que la demanda total de la franja horaria esté dentro de la capacidad. Esa modificación puede ser de dos tipos:

- Un aumento del precio de las franjas horarias sobresaturadas.
- Reducción del precio en las ventanas de tiempo de baja demanda

5. El subastador anuncia el número total de ofertas para todas las ventanas de tiempo y los nuevos precios para las mismas.

Después del paso 5, volverá a repetirse el ciclo hasta que se llegue a la condición de parada. Este proceso termina en el momento en el que la subasta se cierre de forma natural, es decir, cuando no exista un exceso de demanda de espacios en cualquier periodo de tiempo.

Para un buen funcionamiento, se debe cumplir una regla de actividad, que requiere que los licitadores ofrezcan de manera consistente durante todo el proceso. Sirve para evitar el “recorte de la oferta”, en la que el participante oferta cantidades relativamente pequeñas durante varias iteraciones.

4.6. Conclusiones

Tras ver qué es una subasta combinatoria y los tipos que hay, la mejor opción para utilizar en el reparto de franjas horarias es la subasta combinatoria iterativa. Su funcionamiento y fijación de precios es el más apropiado para llevar a cabo en dicho problema.

Este tipo de subasta correctamente diseñada, es capaz de lograr un equilibrio eficiente entre la cantidad de vuelos ofrecidos y las acciones llevadas a cabo el día de las operaciones.

Si se tienen en cuenta todas las consideraciones, reglas y restricciones citadas en el capítulo, el reparto de slots y sus precios serán los óptimos para el funcionamiento del aeropuerto.

CAPÍTULO 5. Relajación Lagrangiana

5.1. Introducción

Algunos problemas de optimización son demasiado complicados para resolverlos, por lo que se utilizan algunos métodos para simplificarlos, como la relajación lagrangiana, que se va a explicar a continuación. Se han utilizado las ideas de la tesis de (Laviós Villahoz, 2013), para dicha explicación.

En este capítulo se explica la utilización de este método para el problema de asignación de slots, que como se verá a continuación, su aplicación podrá interpretarse como una subasta combinatoria.

5.2. Descripción del método

La relajación lagrangeana o lagrangiana, es un método que consiste en descomponer un problema con restricciones muy complejo, para que sea más simple resolverlo.

Aquellas restricciones que sean más complicadas para la resolución, se eliminan y se añaden cada una de ellas a la función objetivo, pero asociadas a un multiplicador de Lagrange (denominado con la letra griega lambda, λ). La función de éste, es penalizar de forma iterativa el incumplimiento de cada restricción eliminada.

Éste nuevo problema se denomina Problema Relajado y cumple que su resolución es más sencilla que en su versión original. Estas soluciones no suelen ser factibles en el problema original, por lo que posteriormente deben ser transformadas para cumplir todas las restricciones del problema inicial.

Éste método es barato, tiene corto tiempo de resolución y obtiene buenas soluciones, aunque no se puede asegurar que sea la óptima.

5.3. Formulación básica

A continuación se describirá el proceso de forma más detallada.

5.3.1. Relajación lagrangiana

Para ver la aplicación de la técnica de relajación lagrangiana vamos a partir de un problema básico de optimización.

Dicho problema de partida (P) será de minimización y es el siguiente:

$$(P) \quad \min_x f(x) \quad (5.1)$$

Sujeto a las restricciones:

$$Ax \leq b \quad (5.2)$$

$$Cx \leq d \quad (5.3)$$

$$x \in X \quad (5.4)$$

Una vez definido el problema, vamos a buscar los valores de x que minimicen la función $f(x)$ (5.1) y que cumplan las condiciones (5.2), (5.3) y las restricciones de signo e integridad (5.4). Suponemos que la restricción (5.2) es la que nos está complicando gravemente el problema, por lo que eliminarla del modelo nos haría mucho más sencilla la resolución. Ese método no es posible, ya que sería un problema diferente que no satisface las restricciones pedidas.

Por este motivo, vamos a transformar el problema inicial en un problema relajado. La nueva función a minimizar sigue dependiendo de la variable x , pero ahora a mayores, de un multiplicador de Lagrange ($\lambda \in \mathbb{R}^n$), que va a aparecer multiplicado por la restricción que queremos eliminar, así penaliza su no cumplimiento. Esta nueva función (5.5) se denomina función lagrangiana, $L(x, \lambda)$.

$$L(x, \lambda) = f(x) + \lambda(Ax - b) \quad (5.5)$$

Con esta nueva función (5.5), el problema a minimizar es el siguiente:

$$\begin{aligned} (LR\lambda) \quad \min_x L(x, \lambda) &= \min_x \{f(x) + \lambda(Ax - b)\} \\ \text{s.a.} \quad Cx &\leq d \\ x &\in X \end{aligned} \quad (5.6)$$

Con este nuevo problema, ya hemos relajado la restricción problemática (con su correspondiente penalización λ) y sólo nos quedan las otras sencillas. El vector

no negativo de penalizaciones ($\lambda \geq 0$) se denomina vector de multiplicadores de Lagrange.

En un problema relajado siempre se cumple:

1. El conjunto factible del problema relajado contiene al del problema original.
2. La función objetivo del problema relajado sobre el conjunto factible del problema inicial, es mejor que la del propio problema inicial.

5.3.2. Función dual asociada

Al resolver el problema relajado (LR_λ) (5.6), se obtiene un valor dependiente de λ , denominado función dual asociada. Su representación es:

$$\theta(\lambda) = v(LR_\lambda) \quad (5.7)$$

5.3.3. Teorema de Dualidad débil

Según (Guignard, 2003), el teorema de la dualidad débil nos indica que el valor de la función dual $\theta(\lambda)$, es una cota inferior de la solución al problema original (P) (5.1). Es decir, el valor que obtengamos en el problema inicial (más restrictivo), no podrá ser menor que el que obtengamos en el problema relajado, independientemente del valor que tome λ .

El teorema es el siguiente:

1. Sea $x(\lambda)$ una solución de (LR_λ), puede ser factible en el problema original, pero no la óptima en la mayoría de los casos.
2. El valor de la función dual $\theta(\lambda)$ es una cota inferior del óptimo del problema (LR_λ).

$$\theta(\lambda) = f(x(\lambda)) + \lambda(Ax(\lambda) - b) \leq v(P) \quad (5.8)$$

3. Si $x(\lambda)$ es factible para el problema original (P) (5.1), el valor óptimo de (P) es mayor que el valor de la función dual asociada en λ y es menor que el valor de la función objetivo de (P) en $x(\lambda)$:

$$\theta(\lambda) = f(x(\lambda)) + \lambda(Ax(\lambda) - b) \leq v(P) \leq f(x(\lambda)) \quad (5.9)$$

4. Si $\lambda(Ax(\lambda) - b)$ es igual a cero, $x(\lambda)$ es la solución óptima del problema inicial sin relajar (P) (5.1).

$$v(P) = f(x(\lambda)) \quad (5.10)$$

5.3.4. Problema dual

El Problema Dual Lagrangiano (LD) es un problema de los multiplicadores de Lagrange, que consiste en maximizar la función dual (5.7). Su objetivo es hallar el valor de dichos multiplicadores, para conseguir la mayor cota inferior de la solución del problema inicial (P).

$$(LD) \max_{\lambda \geq 0} v(LR_{\lambda}) = \max_{\lambda \geq 0} \theta(\lambda) \quad (5.11)$$

Si λ^* es la solución del problema dual, su valor óptimo es denominado cota lagrangiana.

$$v(LD) = \theta(\lambda^*) \quad (5.12)$$

5.3.5. Gap de dualidad

El gap de dualidad nos indica la distancia que hay entre la solución obtenida en el problema inicial (P) (5.1), y la obtenida en el problema dual (LD) (5.11). Cuanto menor sea esa diferencia, más parecidos son los dos valores obtenidos, es decir, mejor resultado hemos obtenido mediante relajación lagrangiana.

$$GAP = v(P) - v(LD) \quad (5.13)$$

5.4. Resolución del problema dual. Método del subgradiente.

El método del subgradiente es uno de los procedimientos para la resolución del problema dual. Es el que se va a describir en este apartado debido a que es sencillo de implantar y converge muy bien.

El proceso de resolución es iterativo y se divide en dos niveles: el primero es la resolución del problema de relajación lagrangiana (LR_{λ}) (5.6), y el segundo el de problema dual (LD) (5.11). Se van alimentando datos entre los dos niveles, que

son los valores de los multiplicadores de Lagrange y los de las soluciones del problema relajado.

El valor que tomen inicialmente los multiplicadores de Lagrange, determinan la solución del modelo relajado, debido a que determinan el tiempo de cálculo que es necesario para encontrar una solución óptima y el valor de ésta. Por este motivo, es importante fijar un valor inicial a los multiplicadores que vaya a dar lugar a la máxima cota inferior, es decir, el límite inferior que es lo más cercano posible al valor de la solución óptima. (Dekrajangpetch, 1999)

El algoritmo de resolución es el siguiente (considerando r a cada iteración):

1. Inicialización de los multiplicadores de Lagrange (λ^0): se les asigna un valor inicial relacionado con el problema que estamos resolviendo, aunque normalmente es cero.
2. Resolución del problema relajado: se resuelve el problema (LR_λ) (5.6), utilizando los valores de λ^r , y obtenemos el resultado óptimo $x(\lambda^r)$.
3. Cálculo del subgradiente: calculamos el vector que representa el incumplimiento de las restricciones relajadas, mediante la siguiente fórmula:

$$g^r = Ax(\lambda^r) - b \quad (5.14)$$

4. Generación de solución factible: la solución obtenida en el problema relajado no cumple con las restricciones del problema original, por lo que hay que factibilizarlo para que satisfaga todas esas restricciones y sea una solución del problema inicial.
5. Con el fin de mejorar el valor de la función dual, se realizan varias iteraciones y en cada una de ellas se actualiza el contador r para volver a hacer los pasos 2, 3 y 4.

Hay que establecer un criterio de parada. Al ser un proceso iterativo, se necesitan unas pautas para saber en qué iteración detenerse. Si la solución del problema original y relajado es la misma o la diferencia entre dos iteraciones es muy pequeña, el algoritmo finaliza.

5.5. Relajación Lagrangiana aplicada al problema de programación lineal de Asignación de Slots en aeropuertos.

Una vez explicado el método de relajación Lagrangiana, vamos a utilizarlo para realizar el reparto de slots mediante subastas combinatorias, aplicándolo al problema de asignación de franjas horarias descrito en el CAPÍTULO 2.

El modelo del que vamos a partir es el problema de programación lineal construido por (Araúzo, 2018), para un aeropuerto con un horizonte de planificación cerrado, según está descrito en el Programación lineal entera mixt.

Las técnicas de relajación lagrangiana simplificarán el modelo eliminando alguna de sus restricciones más complejas y así obtener una solución buena, aunque no tiene por qué ser la óptima. A continuación se explica la aplicación de este proceso al modelo citado anteriormente con más detalle.

5.5.1. Relajación Lagrangiana de las restricciones de capacidad.

Relajamos las restricciones de capacidad incluyéndolas en la función objetivo multiplicadas por un escalar positivo $(\lambda a_{ht}, \lambda d_{ht}, \lambda t_{ht})$, llamado multiplicador de Lagrange, para que el incumplimiento de la restricción penalice la función objetivo.

Nueva función objetivo:

$$\begin{aligned}
 \max_{\delta_m, \delta a_m, \delta d_m} L = & \sum_{i=1}^M \left(g u_m \cdot \delta_m - \sum_{t=0}^{T-1} (d c a_{mt} \cdot \delta a_{mt} + d c d_{mt} \cdot \delta d_{mt}) - c t b_m \right) \\
 & \cdot t b_m \\
 & - \sum_{h=1}^H \sum_{t=0}^{T-T_h} \left(\lambda a_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} - a c_{ht} \right) \right. \\
 & + \lambda d_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} - d c_{ht} \right) \\
 & \left. + \lambda t_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} (\delta a_{m\tau} + \delta d_{m\tau}) - t c_{ht} \right) \right)
 \end{aligned} \tag{5.15}$$

Sujeta a las restricciones:

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} \quad m = \{1, 2, \dots, M\} \quad (5.16)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta d_{mt} \quad m = \{1, 2, \dots, M\} \quad (5.17)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - f k_m \cdot t b_m = 0 \quad m = \{1, 2, \dots, M\} \quad (5.18)$$

$$t b_m - m t b_m \cdot \delta_m \geq 0 \quad m = \{1, 2, \dots, M\} \quad (5.19)$$

$$\delta a_{mt}, \delta d_{mt}, \delta_m \in \{0, 1\} \quad m = \{1, 2, \dots, M\} \\ t = \{1, 2, \dots, T-1\} \quad (5.20)$$

Reagrupamos la función objetivo:

$$\begin{aligned} L = \sum_{m=1}^M \left(g u_m \cdot \delta_m - \sum_{t=0}^{T-1} (d c a_{mt} \cdot \delta a_{mt} + d c d_{mt} \cdot \delta d_{mt}) - c t b_m \cdot t b_m \right) \\ - \sum_{h=1}^H \sum_{t=0}^{T-T_h} \left((\lambda a_{ht} + \lambda t_{ht}) \sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} \right. \\ \left. + (\lambda d_{ht} + \lambda t_{ht}) \sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} \right) \\ + \sum_{h=1}^H \sum_{t=0}^{T-T_h} (\lambda a_{ht} \cdot a c_{ht} + \lambda d_{ht} \cdot d c_{ht} + \lambda t_{ht} \cdot t c_{ht}) \end{aligned} \quad (5.21)$$

$$\begin{aligned} = \sum_{m=1}^M \left(g u_m \cdot \delta_m - c t b_m \cdot t b_m - \sum_{t=0}^{T-1} (d c a_{mt} \cdot \delta a_{mt} + d c d_{mt} \cdot \delta d_{mt}) \right. \\ \left. - \sum_{h=1}^H \sum_{t=0}^{T-T_h} \left((\lambda a_{ht} + \lambda t_{ht}) \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} \right. \right. \\ \left. \left. + (\lambda d_{ht} + \lambda t_{ht}) \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} \right) \right) \\ + \sum_{h=1}^H \sum_{t=0}^{T-T_h} (\lambda a_{ht} \cdot a c_{ht} + \lambda d_{ht} \cdot d c_{ht} + \lambda t_{ht} \cdot t c_{ht}) \end{aligned}$$

Para simplificar la expresión anterior, vamos a eliminar las consecuencias de borde de las restricciones rodantes cuando $T_h > 1$.

Extenderemos $\lambda a_{ht}, \lambda d_{ht}, \lambda t_{ht}$ más allá de la franja horaria $(T - T_h)$ para los multiplicadores que inicialmente no están que definidos en esas franjas horarias.

Para no modificar el problema, esos multiplicadores tendrán valor cero si $t > T - T_h$. Con esto, y definiendo $\theta_h = \min\{T - 1, t + T_h - 1\}$, el sumatorio de franjas horarias del segundo y tercer término de la función objetivo se puede poner entre 0 y $(T - 1)$ de forma que no depende de h . Así, la función objetivo del problema relajado se puede poner como:

$$\begin{aligned}
 L = \sum_{m=1}^M & \left(gu_m \cdot \delta_m - ctb_m \cdot tb_m - \sum_{t=0}^{T-1} (dca_{mt} \cdot \delta a_{mt} + dcd_{mt} \cdot \delta d_{mt}) \right. \\
 & - \sum_{t=0}^{T-1} \sum_{h=1}^H \left((\lambda a_{ht} + \lambda t_{ht}) \sum_{\tau=t}^{\theta_h} \delta a_{m\tau} \right. \\
 & \left. \left. + (\lambda d_{ht} + \lambda t_{ht}) \sum_{\tau=t}^{\theta_h} \delta d_{m\tau} \right) \right) \\
 & + \sum_{t=0}^{T-1} \sum_{h=0}^H (\lambda a_{ht} \cdot ac_{ht} + \lambda d_{ht} \cdot dc_{ht} + \lambda t_{ht} \cdot tc_{ht})
 \end{aligned} \tag{5.22}$$

Cumpléndose:

$$\lambda a_{ht} = \lambda d_{ht} = \lambda t_{ht} = 0 \quad \{h: T_h > 1\}, \quad t = T - T_h + 1, \dots, T - 1 \tag{5.23}$$

En la función objetivo anterior, el término $\delta a_{m\tau}$ aparece en la expresión multiplicado por $(\lambda a_{ht} + \lambda t_{ht})$ para cada h , para los tiempos que van desde $(t - T_h + 1)$ hasta t , siempre y cuando $(t - T_h + 1) \geq 0$.

Por lo tanto, si definimos:

$$pa_t = \sum_{h=1}^H \sum_{\tau=\max(0, t-T_h+1)}^t (\lambda a_{h\tau} + \lambda t_{h\tau}) \tag{5.24}$$

$$pd_t = \sum_{h=1}^H \sum_{\tau=\max(0, t-T_h+1)}^t (\lambda d_{h\tau} + \lambda t_{h\tau}) \tag{5.25}$$

La función objetivo del problema relajado se puede escribir de la forma:

$$\begin{aligned}
 L = \sum_{m=1}^M & \left(gu_m \cdot \delta_m - ctb_m \cdot \mathbf{tb}_m - \sum_{t=0}^{T-1} (dca_{mt} \cdot \delta \mathbf{a}_{mt} + dcd_{mt} \cdot \delta \mathbf{d}_{mt}) \right. \\
 & \left. - \sum_{t=0}^{T-1} (pa_t \cdot \delta \mathbf{a}_{mt} + pd_t \cdot \delta \mathbf{d}_{mt}) \right) \\
 & + \sum_{t=0}^{T-1} \sum_{h=0}^H (\lambda a_{ht} \cdot ac_{ht} + \lambda b_{ht} \cdot dc_{ht} + \lambda t_{ht} \cdot tc_{ht})
 \end{aligned} \tag{5.26}$$

Agrupando términos de la función anterior, el nuevo problema relajado queda como:

$$\begin{aligned}
 \max_{\delta_m, \delta a_m, \delta d_m} L = \sum_{m=1}^M & \left(gu_m \cdot \delta_m - ctb_m \cdot \mathbf{tb}_m \right. \\
 & \left. - \sum_{t=0}^{T-1} ((dca_{mt} + pa_t) \cdot \delta \mathbf{a}_{mt} + (dcd_{mt} + pd_t) \cdot \delta \mathbf{d}_{mt}) \right) \\
 & + \sum_{t=0}^{T-1} \sum_{h=0}^H (\lambda a_{ht} \cdot ac_{ht} + \lambda b_{ht} \cdot dc_{ht} + \lambda t_{ht} \cdot tc_{ht})
 \end{aligned} \tag{5.27}$$

Sujeto a las restricciones:

$$\delta_m = \sum_{t=0}^{T-1} \delta \mathbf{a}_{mt} \quad m = \{1, 2, \dots, M\} \tag{5.28}$$

$$\delta_m = \sum_{t=0}^{T-1} \delta \mathbf{d}_{mt} \quad m = \{1, 2, \dots, M\} \tag{5.29}$$

$$\sum_{t=0}^{T-1} t \cdot \delta \mathbf{d}_{mt} - \sum_{i=0}^{T-1} t \cdot \delta \mathbf{a}_{mt} - fk_m \cdot \mathbf{tb}_m = 0 \quad m = \{1, 2, \dots, M\} \tag{5.30}$$

$$\mathbf{tb}_m - mtb_m \cdot \delta_m \geq 0 \quad m = \{1, 2, \dots, M\} \tag{5.31}$$

$$\delta \mathbf{a}_{mt}, \delta \mathbf{d}_{mt}, \delta_m \in \{0, 1\} \quad \begin{array}{l} m = \{1, 2, \dots, M\} \\ t = \{1, 2, \dots, T-1\} \end{array} \tag{5.32}$$

Dado que las restricciones están separadas para cada m , y teniendo en cuenta que el último término de la función objetivo es una constante para unos $\lambda a_{ht}, \lambda b_{ht}, \lambda t_{ht}$ dados, el problema anterior se puede descomponer en M subproblemas independientes como el siguiente:

$$\begin{aligned}
 \max_{\delta_m, \delta a_m, \delta d_m} L_m = gu_m \cdot \delta_m - ctb_m \cdot \mathbf{tb}_m \\
 - \sum_{t=0}^{T-1} ((dca_{mt} + pa_t) \cdot \delta \mathbf{a}_{mt} + (dcd_{mt} + pd_t) \cdot \delta \mathbf{d}_{mt})
 \end{aligned} \tag{5.33}$$

Sujeto a las restricciones:

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} \quad (5.34)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta d_{mt} \quad (5.35)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - f k_m \cdot t b_m = 0 \quad (5.36)$$

$$t b_m - m t b_m \cdot \delta_m \geq 0 \quad (5.37)$$

$$\delta a_{mt}, \delta d_{mt}, \delta_m \in \{0, 1\} t = \{1, 2, \dots, T - 1\} \quad (5.38)$$

Si interpretamos que pa_t es el un precio asociado al uso de la franja horaria t para una llegada y que pd_t es el precio del uso de ese franja horaria para la salida, el problema anterior se resume en seleccionar los intervalos llegada y salida para el vuelo m , maximizando el excedente que resulta de restar a la utilidad global el coste de programación, el de permanencia en el aeropuerto, y el precio pagado por los slots.

Lo normal es que para unos precios seleccionados al azar, la resolución independiente de cada subproblema dé lugar a soluciones globales no factibles donde se violan muchas de las restricciones relajadas. Pero si tuviéramos unos precios adecuados, la resolución de todos estos subproblemas de forma independiente podría conducir a una solución cercana a la solución óptima del problema original. La cuestión está en cómo conocer los precios.

5.5.2. Método del subgradiente

Se va a utilizar el método del subgradiente explicado en el apartado 5.4, para calcular los multiplicadores de Lagrange, los precios y resolver el problema dual que aparece a continuación (5.39):

$$\begin{aligned}
 \min_{\lambda_m, \lambda a_m, \lambda d_m} \mathcal{L} = & \left(\max_{\delta_m, \delta a_m, \delta d_m} \sum_{i=1}^M \left(gu_m \cdot \delta_m - ctb_m \cdot \mathbf{t}b_m \right. \right. \\
 & - \sum_{t=0}^{T-1} (dca_{mt} \cdot \delta a_{mt} + dcd_{mt} \cdot \delta d_{mt}) \\
 & - \sum_{h=1}^H \sum_{t=0}^{T-T_h} \left(\lambda a_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau} - ac_{ht} \right) \right. \\
 & + \lambda d_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau} - dc_{ht} \right) \\
 & \left. \left. \left. + \lambda t_{ht} \left(\sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} (\delta a_{m\tau} + \delta d_{m\tau}) - tc_{ht} \right) \right) \right) \right) \quad (5.39)
 \end{aligned}$$

Sujeto a las restricciones:

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} \quad m = \{1, 2, \dots, M\} \quad (5.40)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta d_{mt} \quad m = \{1, 2, \dots, M\} \quad (5.41)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - fk_m \cdot \mathbf{t}b_m = 0 \quad m = \{1, 2, \dots, M\} \quad (5.42)$$

$$\mathbf{t}b_m - mtb_m \cdot \delta_m \geq 0 \quad m = \{1, 2, \dots, M\} \quad (5.43)$$

$$\delta a_{mt}, \delta d_{mt}, \delta_m \in \{0, 1\} \quad \begin{array}{l} m = \{1, 2, \dots, M\} \\ t = \{1, 2, \dots, T-1\} \end{array} \quad (5.44)$$

Vamos a utilizar para resolverlo, el método del subgradiente, que como ya vimos en el apartado 5.4, es un método iterativo en el que partiendo de unos multiplicadores iniciales $(\lambda a_{ht}^0, \lambda d_{ht}^0, \lambda t_{ht}^0)$, se resuelve el problema relajado para obtener unos valores de las variables $(\delta a_{mt}, \delta d_{mt}, \delta_m)$; valores que denominaremos $(\delta a_{mt}^0, \delta d_{mt}^0, \delta_m^0)$.

Pero para resolver el problema relajado, dado que tenemos unos multiplicadores fijos, en vez de resolver el problema global (5.15), se puede resolver de forma independiente los m subproblemas formulados en las ecuaciones (5.33), (5.34), (5.35), (5.36), (5.37) y (5.38). Para ello es necesario haber calculado previamente, a partir de los multiplicadores $(\lambda a_{ht}^0, \lambda d_{ht}^0, \lambda t_{ht}^0)$ y los precios de los *slots* de llegada y de salida (pa_{ht}^0, pd_{ht}^0) .

Una vez calculados, aplicamos el método y obtenemos los nuevos multiplicadores $(\lambda_{ht}^1, \lambda d_{ht}^1, \lambda t_{ht}^1)$, que utilizaremos para obtener unos nuevos precios (pa_{ht}^1, pd_{ht}^1) , y con ellos unos nuevas soluciones $(\delta a_{mt}^1, \delta a_{mt}^1, \delta_m^1)$. Se utilizaran estas últimas para actualizar de nuevo los multiplicadores, y repetir estos pasos en cada iteración.

Las soluciones de los subproblemas (resueltos de forma independiente) podrían ser casi factibles y la solución global que se podría obtener a partir de ellas podría ser bastante buena.

El procedimiento que se lleva a cabo para cada iteración i es:

1. Partimos de los precios calculados en la anterior iteración (pa_{ht}^i, pd_{ht}^i) . Si $i=0$, tomamos los multiplicadores iniciales $(\lambda a_{ht}^0, \lambda d_{ht}^0, \lambda t_{ht}^0)$.
2. Resolvemos los m subproblemas formulados con los precios de esta iteración que quedarán según (5.45), obteniendo los valores $(\delta a_{mt}^i, \delta a_{mt}^i, \delta_m^i)$ para las variables del problema:

$$\begin{aligned} \max_{\delta_m, \delta a_m, \delta d_m} L_m = & gu_m \cdot \delta_m - ctb_m \cdot tb_m \\ & - \sum_{t=0}^{T-1} \left((dca_{mt} + pa_t^i) \cdot \delta a_{mt} + (dcd_{mt} + pd_t^i) \cdot \delta d_{mt} \right) \end{aligned} \quad (5.45)$$

Sujeto a:

$$\delta_m = \sum_{t=0}^{T-1} \delta a_{mt} \quad (5.46)$$

$$\delta_m = \sum_{t=0}^{T-1} \delta d_{mt} \quad (5.47)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{mt} - \sum_{i=0}^{T-1} t \cdot \delta a_{mt} - fk_m \cdot tb_m = 0 \quad (5.48)$$

$$tb_m - mtb_m \cdot \delta_m \geq 0 \quad (5.49)$$

$$\delta a_{mt}, \delta d_{mt}, \delta_m \in \{0, 1\} t = \{1, 2, \dots, T - 1\} \quad (5.50)$$

3. A partir de las soluciones obtenidas en el paso previo, se calcula el gradiente (subgradiente) de la función objetivo (5.39) respecto a los multiplicadores.

$$ga_{ht}^{i)} = \frac{\partial \mathcal{L}}{\partial \lambda a_{ht}} = \sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta a_{m\tau}^{i)} - ac_{ht} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.51)$$

$$gd_{ht}^{i)} = \frac{\partial \mathcal{L}}{\partial \lambda d_{ht}} = \sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} \delta d_{m\tau}^{i)} - dc_{ht} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.52)$$

$$gt_{ht}^{i)} = \frac{\partial \mathcal{L}}{\partial \lambda t_{ht}} = \sum_{m=1}^M \sum_{\tau=t}^{t+T_h-1} (\delta a_{m\tau}^{i)} + \delta d_{m\tau}^{i)}) - tc_{ht} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.53)$$

4. Con el gradiente anterior, actualizamos los multiplicadores.

$$\lambda a_{ht}^{i+1)} = \max_{\lambda} \{0, \lambda a_{ht}^{i)} + \alpha^i \cdot ga_{ht}^{i)}\} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.54)$$

$$\lambda d_{ht}^{i+1)} = \max_{\lambda} \{0, \lambda d_{ht}^{i)} + \alpha^i \cdot gd_{ht}^{i)}\} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.55)$$

$$\lambda t_{ht}^{i+1)} = \max_{\lambda} \{0, \lambda t_{ht}^{i)} + \alpha^i \cdot gt_{ht}^{i)}\} \quad \begin{matrix} h = 1, 2, \dots, H \\ t = 0, 1, 2, \dots, T - T_h \end{matrix} \quad (5.56)$$

5. Se calculan los precios de los slots de llegada y salida a partir de los multiplicadores anteriores.

$$pa_t^{i+1)} = \sum_{h=1}^H \sum_{\tau=\max(0, t-T_h+1)}^t (\lambda a_{ht}^{i+1)} + \lambda t_{ht}^{i+1)}) \quad t = 0, 1, 2, \dots, T - T_h \quad (5.57)$$

$$pd_t^{i+1)} = \sum_{h=1}^H \sum_{\tau=\max(0, t-T_h+1)}^t (\lambda d_{ht}^{i+1)} + \lambda t_{ht}^{i+1)}) \quad t = 0, 1, 2, \dots, T - T_h \quad (5.58)$$

Factibilización de la solución

Las soluciones obtenidas mediante el método del subgradiente no siempre son factibles para el problema original. En ese caso hay que realizar una factibilización priorizando las soluciones de los subproblemas que mayores precios totales han aceptado por los slots de llegada y salida. En cada paso este precio vendrá dado por:

$$p_m^i) = \sum_{t=0}^{T-1} (pa_t^i) \cdot \delta a_{mt}^i) + pd_t^i) \cdot \delta d_{mt}^i) \quad (5.59)$$

Basándonos en este precio, se elabora un conjunto ordenado \aleph de los m vuelos con el criterio de $p_m^i)$ decreciente. $Seal(m)$ es una función, tipo permutación, entera de variable entera y definida para $m = \{1, 2, \dots, M\}$ que nos devuelve la

posición del elemento m en \aleph , y sea $m(j) = l^{-1}(j)$, su función inversa que nos devuelve el vuelo m que está en la posición j de \aleph . Obsérvese que se deberá cumplir que $p_{m(j)}^i \geq p_{m(j+1)}^i$

Una vez conocida $m(j)$, se puede obtener una solución factible resolviendo, por el orden indicado por j , los M subproblemas que resultan de añadir a cada subproblema $m(j)$, unas restricciones de capacidad similares a las del problema original pero con las siguientes modificaciones:

- Solo afectan a las variables del subproblema $m(j)$ que se está resolviendo ($\delta a_{m(j)t}$ y $\delta d_{m(j)t}$).
- la capacidad de cada restricción ($ac_{ht}, ac_{ht}, dc_{ht}$) se decrementa en función de los vuelos que ya han sido programados previamente cuando se han resuelto los subproblemas $m(1), m(2), \dots, m(j-1)$

A estos subproblemas los denominaremos subproblemas restringidos, y las soluciones obtenidas para cada uno de ellos se denotarán, para diferenciarla de las obtenidas de los anteriores, con un asterisco ($\delta a_{m(l)t}^{*i}, \delta d_{m(l)t}^{*i}, \delta_m^{*i}$).

$$\begin{aligned} \max_{\delta_m, \delta a_m, \delta d_m} L_m = & gu_{m(j)} \cdot \delta_{m(j)} - ctb_m \cdot tb_{m(j)} \\ & - \sum_{t=0}^{T-1} \left((dca_{mt} + pa_t^i) \cdot \delta a_{mt} + (dcd_{mt} + pd_t^i) \cdot \delta d_{mt} \right) \\ & - \sum_{t=0}^{T-1} \left((dca_{m(j)t} + pa_t^i) \cdot \delta a_{m(j)t} \right. \\ & \left. + (dcd_{m(j)t} + pd_t^i) \cdot \delta d_{m(j)t} \right) \end{aligned} \quad (5.60)$$

Sujeto a:

$$\sum_{\tau=t}^{t+T_h-1} \delta a_{m(j)\tau} \leq ac_{ht} - \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta a_{m(f)\tau}^{*i} \quad h = 1, 2, \dots, H \quad t = 0, 1, 2, \dots, T - T_h \quad (5.61)$$

$$\sum_{\tau=t}^{t+T_h-1} \delta d_{m(j)\tau} \leq dc_{ht} - \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta d_{m(f)\tau}^{*i} \quad h = 1, 2, \dots, H \quad t = 0, 1, 2, \dots, T - T_h \quad (5.62)$$

$$\begin{aligned} & \sum_{\tau=t}^{t+T_h-1} (\delta a_{m(j)\tau} + \delta d_{m(j)\tau}) \\ & \leq tc_{ht} - \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} (\delta a_{m(f)\tau}^{*i} - \delta d_{m(f)\tau}^{*i}) \quad h = 1, 2, \dots, H \quad t = 0, 1, 2, \dots, T - T_h \end{aligned} \quad (5.63)$$

$$\delta_{m(j)} = \sum_{t=0}^{T-1} \delta a_{m(j)t} \quad (5.64)$$

$$\delta_m(j) = \sum_{t=0}^{T-1} \delta d_{m(j)t} \quad (5.65)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{m(j)t} - \sum_{i=0}^{T-1} t \cdot \delta a_{m(j)t} - fk_{m(j)} \cdot tb_{m(j)} = 0 \quad (5.66)$$

$$tb_{m(j)} - mtb_{m(j)} \cdot \delta_m(j) \geq 0 \quad (5.67)$$

$$\delta a_{m(j)t}, \delta d_{m(j)t}, \delta_m(j) \in \{0, 1\} t = \{1, 2, \dots, T-1\} \quad (5.68)$$

Parecen unos subproblemas muy complicados, pero se pueden simplificar quedando de la siguiente manera:

$$\begin{aligned} & \max_{\delta_m, \delta a_m, \delta d_m} L_{m(j)} \\ & = gu_{m(j)} \cdot \delta_m(j) \\ & - \sum_{t=0}^{T-1} \left((dca_{m(j)t} + pa_t^i) \cdot \delta a_{m(j)t} \right. \\ & \left. + (dcd_{m(j)t} + pd_t^i) \cdot \delta d_{m(j)t} \right) \end{aligned} \quad (5.69)$$

Sujeto a:

$$\sum_{\tau=t}^{t+T_h-1} \delta a_{m(j)\tau} = 0 \quad \left\{ h, t : ac_{ht} = \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta a_{m(f)\tau}^* \right\} \quad (5.70)$$

$$\sum_{\tau=t}^{t+T_h-1} \delta d_{m(j)\tau} = 0 \quad \left\{ h, t : dc_{ht} = \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta d_{m(f)\tau}^* \right\} \quad (5.71)$$

$$\begin{aligned} & \sum_{\tau=t}^{t+T_h-1} (\delta a_{m(j)\tau} + \delta d_{m(j)\tau}) = 0 \\ & \left\{ h, t : tgm_{m(j)} \geq T_h \wedge dc_{ht} = \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta d_{m(f)\tau}^* \right\} \end{aligned} \quad (5.72)$$

$$\begin{aligned} & \sum_{\tau=t}^{t+T_h-1} (\delta a_{m(j)\tau} + \delta d_{m(j)\tau}) \leq 1 \\ & \left\{ h, t : tgm_{m(j)} < T_h \wedge dc_{ht} - \sum_{f=1}^{j-1} \sum_{\tau=t}^{t+T_h-1} \delta d_{m(f)\tau}^* \leq 1 \right\} \end{aligned} \quad (5.73)$$

$$\delta_{m(j)} = \sum_{t=0}^{T-1} \delta a_{m(j)t} \quad (5.74)$$

$$\delta_{m(j)} = \sum_{t=0}^{T-1} \delta d_{m(j)t} \quad (5.75)$$

$$\sum_{t=0}^{T-1} t \cdot \delta d_{m(j)t} - \sum_{i=0}^{T-1} t \cdot \delta a_{m(j)t} - f k_{m(j)} \cdot t b_{m(j)} = 0 \quad (5.76)$$

$$t b_{m(j)} - m t b_{m(j)} \cdot \delta_{m(j)} \geq 0 \quad (5.77)$$

$$\delta a_{m(j)t}, \delta d_{m(j)t}, \delta_{m(j)} \in \{0, 1\} t = \{1, 2, \dots, T-1\} \quad (5.78)$$

Si las soluciones de los subproblemas (5.45), cumplen estas nuevas restricciones del subproblema restringido, también serán soluciones de este problema, es decir:

$$\delta a_{m(l)t}^* = \delta a_{m(l)t}^i, \delta d_{m(l)t}^* = \delta d_{m(l)t}^i \text{ y } \delta_{m(l)}^* = \delta_{m(l)}^i$$

Paso de actualización

Para que este método converja hacia la solución óptima, se necesita un paso de actualización en cada iteración de valor α^i , que se calcula como:

$$\alpha^i = \gamma^i \frac{\mathcal{L}(\bar{\lambda}^i) - \mathcal{L}(\bar{\lambda}^{op})}{\|\bar{g}^i\|^2} \quad \text{con } 0 < \gamma^i \leq 2 \quad (5.79)$$

Siendo $\bar{\lambda}^i$ el vector de multiplicadores en la iteración i , $\bar{\lambda}^{op}$ el vector de multiplicadores para la solución óptima del problema dual, $\|\bar{g}^i\|$ el módulo del vector subgradiente en la iteración i , y γ^i un parámetro que regula la convergencia. En el numerador de la expresión anterior se calcula la diferencia entre el valor de la función dual $\mathcal{L}(\bar{\lambda}^i)$ en la iteración i y la cota Lagrangiana $\mathcal{L}(\bar{\lambda}^{op})$ o valor óptimo de la función dual.

En la práctica como $\mathcal{L}(\bar{\lambda}^{op})$ es desconocido, se suele sustituir por una cota inferior suya. Esta cota inferior suele ser el mejor valor de la función objetivo del problema original obtenido hasta el momento. Con ello el paso de actualización queda como:

$$\alpha^i = \gamma^i \frac{\mathcal{L}(\bar{\lambda}^i) - \max_j(U^j)}{\|\bar{g}^i\|^2} \quad \text{con } 0 < \gamma^i \leq 2, \quad j = 0, 1, \dots, i \quad (5.80)$$

Gap de dualidad

Es la diferencia (no negativa) entre los valores óptimos del problema original y del problema relajado. Como el problema que se está tratando es de maximización, \mathcal{L}^{op} será una cota superior de la solución del problema original y por lo tanto el gap de dualidad se puede poner como:

$$GAP = \mathcal{L}^{op} - U^{op} \quad (5.81)$$

Como los valores óptimos no se conocen, en cada iteración i se puede calcular un gap de dualidad con la siguiente expresión:

$$GAP^i = \mathcal{L}^i - \max_j(U^j) \quad j = 0, 1, \dots, i \quad (5.82)$$

Criterio de parada

Pueden utilizarse diferentes criterios de parada, como:

1. El valor calculado del gap de dualidad en la iteración es muy pequeño

$$GAP^i < \varepsilon \quad (5.83)$$

2. El vector de multiplicadores de Lagrange ha variado muy poco de una iteración a la siguiente.

$$\|\bar{\lambda}^i - \bar{\lambda}^{i-1}\| < \varepsilon \quad (5.84)$$

3. La mejor valor de la función objetivo original ($\max_j(U^j)$) no ha sido actualizada en un determinado número de iteraciones.

5.6. Conclusiones

La relajación lagrangiana es un método apropiado de resolución del problema de asignación de franjas horarias, que relaja sus restricciones más complejas y las incluye en la función objetivo en forma de penalización.

Una vez relajado el problema, el procedimiento de resolución descrito es análogo a una subasta combinatoria de tipo “*price driving*”. Los compradores serían los M vuelos (llegada + salida) que tendrían que seleccionar los intervalos (bienes subastados) que más les interesan. Esa selección depende del precio de los intervalos y de sus preferencias, que el aeropuerto iría actualizando en cada iteración según las expresiones descritas en el capítulo.

CAPÍTULO 6. Ajuste y comparación de métodos

6.1. Introducción

Una vez explicado el problema de reparto de franjas horarias, y algunos de los métodos más importantes para optimizar su resolución, vamos a llevar a cabo una serie de experimentos con el fin de evaluar el funcionamiento de dichos métodos, y ver cuáles son los más adecuados.

Para realizar este experimento se van a utilizar los programas Gurobi y Eclipse Java. Han sido elegidos por ser unas herramientas muy potentes para la optimización de problemas y por la posibilidad de documentación de las soluciones.

6.2. Generación de problemas

El primer paso para el análisis de los datos, es generarlos. Para ello, utilizamos la función `generateProblems()`, que crea una serie de problemas aleatorios de repartos de slots. Dichos problemas no superan los límites de capacidad (h) de un determinado aeropuerto, es decir, cumplen sus restricciones rodantes. Cada una de ellas afecta a T_h intervalos de coordinación consecutivos y cumplen con los valores indicados en Tabla 1:

Tabla 1. Constantes de capacidad del aeropuerto. Fuente: (de la Fuente, 2016)

h	T_h	nº max de llegadas	nº max de salidas	nº max de operaciones
1	1	2	2	3
2	3	5	4	8

Una vez establecidos estos límites, los problemas generados aleatoriamente van a contener los siguientes valores:

- Tipo de vuelo: 1 si sale y luego regresa al aeropuerto, 0 si viceversa.
- Tiempo mínimo transcurrido entre las operaciones. Para vuelos de tipo 0: uniformemente distribuido entre 1 y $\max\{3, \min\{T/100, 8\}\}$. Para vuelos de tipo 1: uniformemente distribuido entre $\max\{3, \min\{T/30, 20\}\}$ y $\max\{6, \min\{T/10, 40\}\}$.
- Coste asociado a que el avión esté fuera o dentro del aeropuerto: uniformemente distribuido entre 0.1 y 1 unidades monetarias
- Slot mínimo y máximo deseado para la llegada y salida de los vuelos. En este caso, si es una hora punta los datos se generan mediante una

Ajuste y comparación de métodos

distribución normal en torno a la hora punta, y si no lo es, mediante una distribución uniforme. Se generarán diferentes escenarios con más o menos vuelos en horas punta: a) no hay horas punta, b) el 20% de los vuelos se programan en horas punta, y c) el 40% de los vuelos se programan en horas puntas.

- Utilidad bruta en cada vuelo: uniformemente distribuida entre 20 y 100 unidades monetarias.
- Coste de programar las salidas y llegadas antes de las franjas horarias mínimas deseadas: uniformemente distribuido entre 40/T y 200/T, siendo T el número de intervalos temporales establecidos.
- Coste de programar las salidas y llegadas después de las franjas horarias máximas deseadas: uniformemente distribuido entre 40/T y 200/T, siendo T el número de intervalos totales establecidos.

Cada vez que se ejecuta la función con una determinada semilla, se generan 54 problemas con las características anteriores, que están recogidos en Tabla 2 con su descripción:

Tabla 2. Tipos de problemas aleatorios generados para cada semilla.

	Problema	Periodo analizado	Nº franjas horarias	Tamaño franja horaria (min)	Nº vuelos	% horas punta
1	d24x26a	1 día	24	60	26	0
2	d24x26b	1 día	24	60	26	20
3	d24x26c	1 día	24	60	26	40
4	d24x34a	1 día	24	60	34	0
5	d24x34b	1 día	24	60	34	20
6	d24x34c	1 día	24	60	34	40
7	d24x41a	1 día	24	60	41	0
8	d24x41b	1 día	24	60	41	20
9	d24x41c	1 día	24	60	41	40
10	d48x53a	1 día	48	30	53	0
11	d48x53b	1 día	48	30	53	20
12	d48x53c	1 día	48	30	53	40
13	d48x67a	1 día	48	30	67	0
14	d48x67b	1 día	48	30	67	20
15	d48x67c	1 día	48	30	67	40
16	d48x82a	1 día	48	30	82	0
17	d48x82b	1 día	48	30	82	20
18	d48x82c	1 día	48	30	82	40
19	d96x106a	1 día	96	15	106	0

20	d96x106b	1 día	96	15	106	20
21	d96x106c	1 día	96	15	106	40
22	d96x134a	1 día	96	15	134	0
23	d96x134b	1 día	96	15	134	20
24	d96x134c	1 día	96	15	134	40
25	d96x163a	1 día	96	15	163	0
26	d96x163b	1 día	96	15	163	20
27	d96x163c	1 día	96	15	163	40
28	s168x185a	1 semana	168	60	185	0
29	s168x185b	1 semana	168	60	185	20
30	s168x185c	1 semana	168	60	185	40
31	s168x235a	1 semana	168	60	235	0
32	s168x235b	1 semana	168	60	235	20
33	s168x235c	1 semana	168	60	235	40
34	s168x286a	1 semana	168	60	286	0
35	s168x286b	1 semana	168	60	286	20
36	s168x286c	1 semana	168	60	286	40
37	s336x370a	1 semana	336	30	370	0
38	s336x370b	1 semana	336	30	370	20
39	s336x370c	1 semana	336	30	370	40
40	s336x470a	1 semana	336	30	470	0
41	s336x470b	1 semana	336	30	470	20
42	s336x470c	1 semana	336	30	470	40
43	s336x571a	1 semana	336	30	571	0
44	s336x571b	1 semana	336	30	571	20
45	s336x571c	1 semana	336	30	571	40
46	s672x1142a	1 semana	672	15	1142	0
47	s672x1142b	1 semana	672	15	1142	20
48	s672x1142c	1 semana	672	15	1142	40
49	s672x739a	1 semana	672	15	739	0
50	s672x739b	1 semana	672	15	739	20
51	s672x739c	1 semana	672	15	739	40
52	s672x941a	1 semana	672	15	941	0
53	s672x941b	1 semana	672	15	941	20
54	s672x941c	1 semana	672	15	941	40

La función se ha ejecutado 3 veces, con diferentes semillas, para generar un total de $3 \times 54 = 162$ problemas.

6.3. Resolución mediante Relajación Lagrangiana

Como se ha visto en el CAPÍTULO 5, la relajación lagrangiana es uno de los métodos para resolver problemas de optimización. Vamos a ponerlo en práctica, a partir de la programación de éste en el programa eclipse.

Cargamos en la función llamada solveProblemsLR un vector con los problemas que queremos que sean resueltos. Los problemas de menor tamaño, que contienen datos para un día, se han resuelto para 3 semillas. Los más grandes, los que incluyen la información de una semana, han sido resueltos para 1 sola semilla, ya que el tiempo de ejecución es muy elevado y se pueden sacar conclusiones con un solo experimento.

El solver de la relajación lagrangiana contiene los siguientes parámetros:

- gamma: regula el paso de actualización de los multiplicadores.
- alfa: número de pasos sin mejora de la solución para que se actualice gamma.
- maxRound: número máximo de iteraciones.
- beta: factor de actualización de gamma, es decir, número por el que es multiplicado al cumplirse alfa.

6.3.1. Parámetros a modificar

Tanto gamma, como maxRound no van a ser modificados. Van a ser valores estándar para todos los experimentos.

Para ajustar la herramienta de resolución con el fin de encontrar los valores óptimos, vamos a hacer comprobaciones variando los valores de alfa y de beta. Los rangos de posibilidades que se han considerado son:

- Alfa: 5, 10, 60 o 100
- Beta: entre 0.6 y 0.95

Con estos rangos de valores, en la Tabla 3 se muestran las distintas combinaciones estudiadas:

Tabla 3: Combinaciones utilizadas para resolver cada uno de los problemas.

alfa	60	10	100	100	60	10	100	60	10	5
beta	0.85	0.85	0.85	0.6	0.6	0.6	0.95	0.95	0.95	0.85

Partiendo de todos los problemas generados, con las diferentes semillas, y utilizando estas posibles combinaciones, se han resuelto un total de 1059 problemas.

El objetivo de comprobar la mejor combinación de estos parámetros, es que el método de resolución tenga una mejor convergencia, es decir, que obtenga el mejor óptimo con el menor número de iteraciones posibles.

6.3.2. ¿Qué obtenemos?

Como se ha visto anteriormente, es un método iterativo. En la siguiente gráfica podemos ver en un ejemplo para un determinado problema, la evolución del valor objetivo para cada iteración.

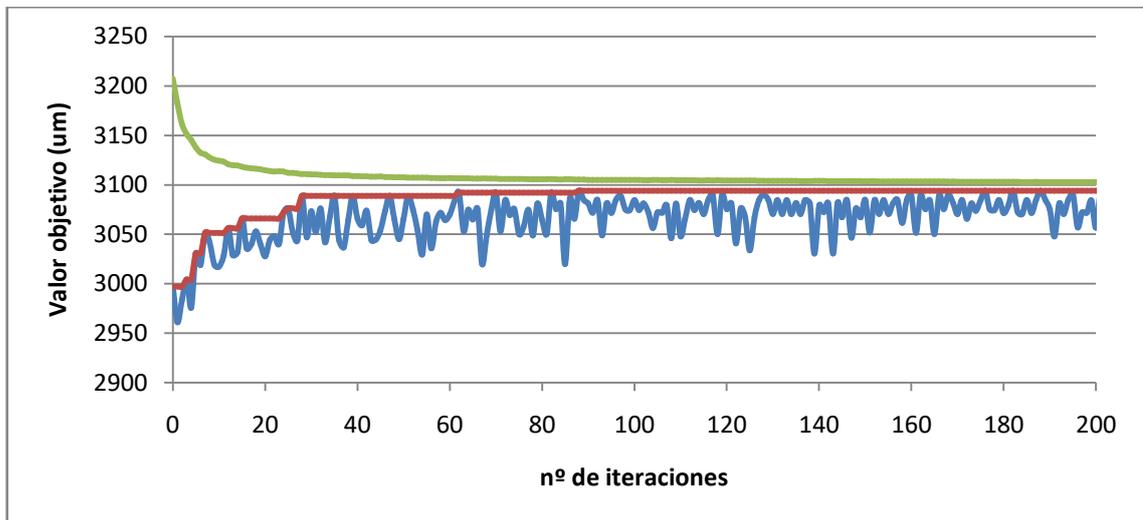


Ilustración 8. Convergencia de la resolución de un problema de optimización mediante RL.

En la Ilustración 8, se observa la convergencia del método. La línea azul nos muestra los valores de la función objetivo, la línea roja el mayor valor objetivo encontrado hasta ese instante (nunca puede ser descendente) y la línea verde los valores de la función objetivo del problema relajado.

Cada vez que ejecutamos la resolución de un problema, obtenemos los siguientes parámetros de cada uno de ellos:

- *Time intervals*: nº de franjas horarias.
- *Flights*: nº de vuelos.
- *Average search space*: nos indica con cuántas franjas horarias tiene que “tantear” un vuelo para obtener la mejor solución.

Ajuste y comparación de métodos

- *Load factor*: factor de carga. Es la relación entre el nº de solicitudes y la capacidad. Para que no esté saturado, el valor debe ser <1 .
- *Obj*: Es el óptimo de la función objetivo del problema original, factibilizando el obtenido en el problema relajado.
- *Gap*: es la distancia entre la solución obtenida y el óptimo.
- *Rounds*: nº de iteraciones realizadas hasta llegar a la solución.
- *Time*: tiempo de ejecución de la resolución del problema.
- *Allocated flights*: nº de vuelos asignados.
- *Allocated Gross Utility*: utilidad bruta * nº de vuelos
- *Scheduling Cost*: coste de no asignar el vuelo a la franja horaria deseada. Se calcula como la diferencia entre la utilidad y el objetivo.

Se ha programado una clase en JAVA que recoge los resultados obtenidos en cada uno de los problemas resueltos y los reúne todos en un nuevo fichero, así podemos exportar con más facilidad todos los resultados al archivo de Excel. (Véase programación en ANEXO)

6.3.3. Estudio 1: ajuste de parámetros

Tras resolver todos los problemas con el método de relajación lagrangiana, vamos a identificar la mejor combinación de parámetros para el método.

Tomamos como ejemplo uno de los problemas, exactamente el s168x185a. Hemos resuelto este problema 9 veces, 1 con cada combinación de parámetros. En la Ilustración 9 se representa para cada dicha combinación (representadas con números del eje x), su valor objetivo (eje y izquierdo) y su número de iteraciones (eje y derecho).

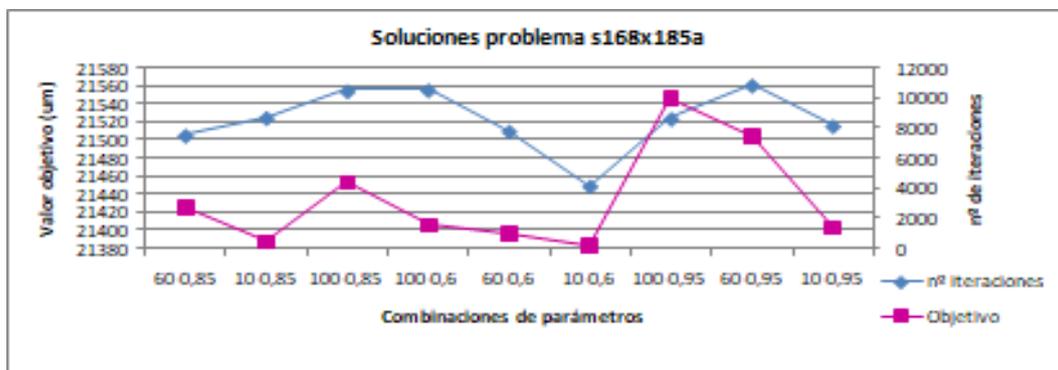


Ilustración 9. Representación del valor objetivo y del nº de iteraciones de 1 problema, para cada una de las combinaciones de parámetros.

La combinación que interesa es la de mayor valor objetivo (función azul) y menor número de iteraciones (función morada). En el gráfico podemos observar que ese caso no se cumple. El mínimo de la función morada, no corresponde con el máximo de la azul, lo que hace que en el estudio de la mejor combinación tengamos que separar ambos resultados, es decir, hay que buscar la mejor combinación para el menor número de iteraciones y por separado otra para el mayor valor objetivo.

Para este estudio los pasos seguidos han sido los siguientes:

1. Asignar a cada problema un tamaño: se ha creado una nueva variable, llamada tamaño del problema, multiplicando el número de franjas horarias por el número de vuelos en cada uno de los problemas.
2. Clasificamos según tamaño: desde 624 hasta 3916 serán pequeños, desde 10176 hasta 48048 serán medianos y desde 124320 hasta 767424 serán grandes.
3. Calcular promedios del valor objetivo y del número de iteraciones para cada tamaño de problema: Para cada combinación de parámetros en cada tamaño, hacemos la media de los valores obtenidos con todas las semillas.
4. Elegir el valor óptimo para cada tamaño de problema: para cada uno de los tamaños obtenidos, tenemos un promedio para cada combinación de parámetros. Para cada tamaño, serán 9 promedios de valor objetivo (elegimos el máximo) y otros nueve de número de iteraciones (elegimos el mínimo).
5. Elección de parámetros óptimos para cada tamaño: identificamos a qué combinación de parámetros corresponde ese máximo/mínimo. Se obtienen las siguientes tablas:

Tabla 4. Combinación óptima de parámetros para cada tamaño (maximiza valor objetivo).

Pequeños			Medianos			Grandes		
Tamaño	alfa	beta	Tamaño	alfa	beta	Tamaño	alfa	beta
624	60	0,95	10176	100	0,95	124320	60	0,85
816	100	0,95	12864	60	0,95	157920	60	0,95
984	60	0,85	15648	10	0,85	191856	60	0,95
2544	100	0,85	31080	60	0,95	496608	60	0,85
3216	100	0,85	39480	60	0,95	632352	60	0,85
3936	100	0,95	48048	100	0,95	767424	60	0,95

Ajuste y comparación de métodos

Tabla 5. Combinación óptima de parámetros para cada tamaño (minimiza número de iteraciones).

Pequeños			Medianos			Grandes		
Tamaño	alfa	beta	Tamaño	alfa	beta	Tamaño	alfa	beta
624	10	0,6	10176	10	0,6	124320	10	0,6
816	10	0,6	12864	10	0,6	157920	10	0,6
984	10	0,6	15648	10	0,6	191856	10	0,6
2544	10	0,6	31080	10	0,6	496608	10	0,6
3216	10	0,6	39480	10	0,6	632352	10	0,6
3936	10	0,6	48048	10	0,6	767424	10	0,6

6. Elección de parámetros óptimos finales en función del valor objetivo:

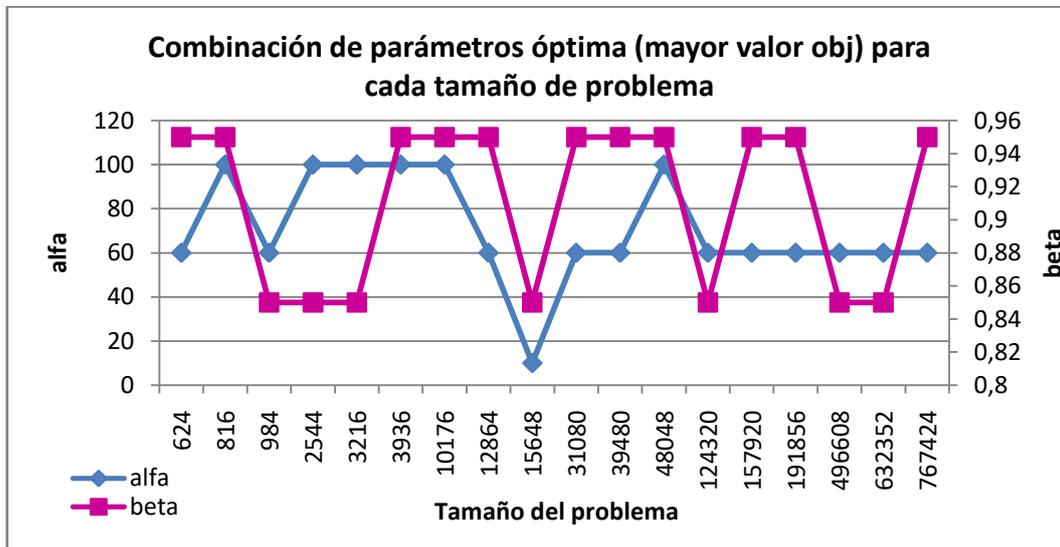


Ilustración 10. Combinación de parámetros óptima (mayor valor objetivo) para cada tamaño de problema.

En la Ilustración 10, podemos observar que la combinación de parámetros no depende del tamaño. No hay un patrón en función de si son grandes, medianos o pequeños, por lo que hay que ajustar dichos parámetros sin tener en cuenta el tamaño del problema.

El mayor valor de la función objetivo se obtiene para valores grandes de beta (entre 0.85 y 0.95), por lo que el parámetro ajustado se va a encontrar entre dichos valores. Para calcularlo hacemos la media de todos los valores que aparecen para beta en la Tabla 4. Combinación óptima de parámetros para cada tamaño (maximiza valor objetivo), y el resultado es 0,91.

Conclusión: el valor para beta que maximiza la función objetivo es 0,91.

En el caso de alfa, el valor que maximice la función objetivo va a estar entre 60 y 100. Realizamos el mismo procedimiento que en el caso anterior, con los datos de la columna alfa de la Tabla 4, y el promedio obtenido es 70,5. Este parámetro indica un número de iteraciones, por lo que vamos a utilizar un valor entero.

Conclusión: el valor de alfa para maximizar la función objetivo es 70.

8. Elección de los parámetros óptimos finales en función del número de iteraciones:

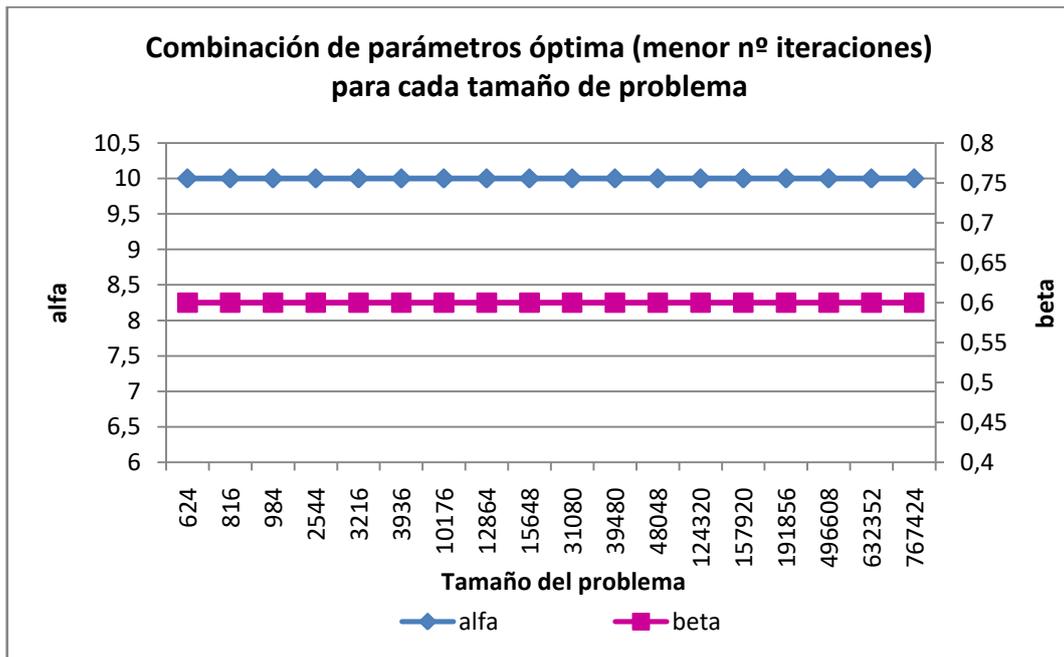


Ilustración 11. Combinación de parámetros óptima (menor número de iteraciones) para cada tamaño de problema.

En este caso, según la Tabla 5, y como se puede apreciar en la Ilustración 11, para todos los tamaños tenemos la misma combinación, así que consideramos 10 y 0.6 los parámetros óptimos.

Una vez realizados todos los pasos, los parámetros ajustados para el método de Relajación Lagrangiana son:

- Mayor valor objetivo: 70 y 0.91
- Menor número de iteraciones: 10 y 0.6

6.4. Comparación de los dos métodos

Una vez que se han ajustado los parámetros óptimos para el método de relajación lagrangiana, ya tenemos disponibles las dos opciones de resolución de problemas. Se van a realizar una serie de experimentos para compararlos entre sí.

El nuevo método, Gurobi (GRB), es una herramienta para resolver problemas de programación lineal basado en el algoritmo de branch and bound explicado en el apartado 3.3. Su criterio de parada es el tiempo de ejecución, y para estos experimentos se ha utilizado un `timeLimit = 10800s`, es decir 3 horas.

6.4.1. Estudio 2: ¿Cuál es el mejor método de resolución?

Partimos de los 54 problemas generados por una determinada semilla (A) y llevamos a cabo:

1. Resolución con el método de relajación lagrangiana (**RL1**), ajustado para obtener el mayor valor objetivo (parámetros 70 y 0,91).
2. Resolución también con el método de relajación lagrangiana (**RL2**), pero con los parámetros ajustados con el fin de obtener el menor número de iteraciones (10 y 0,6).
3. Se resuelven otra vez los problemas con la misma semilla de los 2 casos anteriores, pero esta vez con el método **GRB**.

Una vez obtenidos los resultados, las variables que son interesantes de comparar entre ellos son:

- Valor objetivo
- Número de iteraciones
- Tiempo de ejecución en la resolución

Para facilitar la comparación se van a recopilar estos parámetros en una sola tabla para todos los problemas y métodos de resolución. En la Tabla 6 están recogidos los problemas pequeños, en la Tabla 7 los medianos y en la Tabla 8 los grandes.

Tabla 6. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas pequeños con todos los métodos de resolución.

Problema	Nº franjas horarias	Nº de vuelos	RL1: Mayor valor obj (70-0,91)			RL2: Menor nº iteraciones (10-0,6)			Método GRB		
			Sol óptima (um)	nº de iteraciones	t ejecución (s)	Sol óptima (um)	nº de iteraciones	t ejecución (s)	Sol óptima (um)	nº de iteraciones	t ejecución (s)
d24x26a	24	26	2757,2641	2134	0,197	2670,8431	138	0,035	2771,2755	1.0	0,3011
d24x26b	24	26	3068,6607	852	0,051	3067,0668	91	0,012	3069,7932	1.0	0,0876
d24x26c	24	26	2901,931	302	0,018	2901,931	60	0,009	2902,306	1.0	0,1219
d24x34a	24	34	3712,2359	1013	0,094	3685,742	140	0,022	3712,2359	1.0	0,1504
d24x34b	24	34	3374,1932	1717	0,177	3333,2469	156	0,021	3389,5728	1.0	0,2986
d24x34c	24	34	3625,9408	3058	0,278	3586,845	138	0,018	3652,7779	1.0	0,4816
d24x41a	24	41	3836,1152	3568	0,342	3812,5689	158	0,019	3911,1054	1.0	0,4618
d24x41b	24	41	3761,6537	3386	0,326	3720,5149	138	0,015	3789,4412	297.0	0,7822
d24x41c	24	41	3907,2551	1045	0,107	3826,9337	133	0,014	3907,2551	1.0	0,0868
d48x53a	48	53	6293,9635	684	0,227	6287,1148	122	0,046	6297,3979	1.0	0,4962
d48x53b	48	53	6334,5895	2174	1,045	6276,3895	133	0,068	6356,4993	1.0	1,0332
d48x53c	48	53	6161,6405	972	0,351	6144,0518	120	0,046	6165,8111	1.0	1,1924
d48x67a	48	67	7368,2042	4112	2,138	7229,3774	187	0,106	7450,071	1.0	2,9547
d48x67b	48	67	7140,5665	3484	1,906	6965,4801	149	0,09	7180,8654	71.0	2,0193
d48x67c	48	67	7348,9412	1799	0,883	7065,6223	173	0,089	7349,0011	1.0	0,8
d48x82a	48	82	7963,3871	3703	2,059	7859,722	175	0,108	8090,9421	7.0	4,3738
d48x82b	48	82	7996,7557	4224	3,242	7809,0791	164	0,111	8125,3826	1.0	1,1358
d48x82c	48	82	7937,8959	4127	5,513	7906,1817	157	0,107	8105,0421	1.0	1,0788

Pequeños (semilla A)

Ajuste y comparación de métodos

Tabla 7. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas medianos con todos los métodos de resolución.

Problema	Nº franjas horarias	Nº de vuelos	RL1: Mayor valor obj (70-0,91)			RL2: Menor nº iteraciones (10-0,6)			Método GRB		
			Sol óptima (um)	nº de iteraciones	t ejecución (s)	Sol óptima (um)	nº de iteraciones	t ejecución	Sol óptima (um)	nº de iteraciones	t ejecución (s)
d96x106a	96	106	12781,2004	3737	15,54	12754,2559	160	0,427	12806,9964	101.0	7,5918
d96x106b	96	106	12440,2636	2378	5,926	12380,7726	152	0,432	12447,3804	1.0	3,096
d96x106c	96	106	12624,7812	3711	21,413	12586,743	143	0,445	12659,8977	1.0	18,5618
d96x134a	96	134	14627,3247	5724	50,528	14665,4162	199	0,755	15128,3223	862.0	65,3793
d96x134b	96	134	14197,001	5251	37,661	13650,8817	194	0,594	14677,8853	6285.0	178,5697
d96x134c	96	134	14680,0211	5483	22,948	14190,8647	204	0,723	15190,8654	1104.0	141,2723
d96x163a	96	163	15811,5849	4816	21,763	15964,3693	220	0,954	16625,4497	476.0	48,4592
d96x163b	96	163	15699,309	4676	16,989	15798,0487	189	0,781	16463,2453	42272.0	1096,264
d96x163c	96	163	15696,3313	4337	17,743	15475,497	189	0,909	16102,781	463.0	60,7001
s168x185a	168	185	21526,5724	5170	52,966	21448,7138	205	2,324	21623,4032	1801.0	99,957
s168x185b	168	185	21424,7733	4565	52,67	21367,6562	152	1,926	21469,2748	1421.0	123,3077
s168x185c	168	185	20739,6698	5052	73,327	20521,0555	188	6,295	21038,3351	6031.0	683,6592
s168x235a	168	235	25441,1691	5772	97,798	24471,0402	214	8,982	26735,6959	33091.0	4240,2112
s168x235b	168	235	26047,3885	6248	113,557	24748,5889	202	9,318	27494,785	19976.0	823,4979
s168x235c	168	235	23253,6179	4854	88,387	22936,4526	221	9,567	23794,9613	2588.0	552,1261
s168x286a	168	286	27654,8997	4950	92,337	27342,2453	192	6,582	28930,4561	153673.0	5015,2757
s168x286b	168	286	27712,7406	5014	92,846	27196,5341	193	5,186	29025,3643	1250.0	286,4791
s168x286c	168	286	25962,2214	5251	113,758	25486,0537	211	6,353	27086,4779	1254.0	334,4641

Tabla 8. Resultados obtenidos en solución óptima, nº de iteraciones y tiempo de ejecución para los problemas grandes con todos los métodos de resolución.

Problema	Nº franjas horarias	Nº de vuelos	RL1: Mayor valor obj (70-0,91)			RL2: Menor nº iteraciones (10-0,6)			Método GRB		
			Sol óptima (um)	nº de iteraciones	t ejecución (s)	Sol óptima (um)	nº de iteraciones	t ejecución (s)	Sol óptima (um)	nº de iteraciones	t ejecución (s)
s336x370a	336	370	43963,8025	5256	330,666	43899,6024	210	16,358	44096,4782	19637.0	1089,4753
s336x370b	336	370	43442,0425	5119	397,867	43177,6541	203	18,774	43730,1697	90980.0	10800,371
s336x370c	336	370	42225,259	5235	471,186	41479,6245	228	22,939	42782,5214	20621.0	10800,1673
s336x470a	336	470	51625,9357	6416	653,99	48304,726	238	27,807	54607,3975	8068.0	10800,2091
s336x470b	336	470	49226,4013	6095	654,508	47831,5768	212	27,624	50571,2596	5801.0	10800,7738
s336x470c	336	470	46250,4027	5547	616,627	45295,748	187	23,333	47451,316	6851.0	10800,1773
s336x571a	336	571	54244,7336	5099	531,233	53476,1405	218	24,644	57893,8355	11072.0	10805,6213
s336x571b	336	571	55236,1715	5251	646,072	53719,6342	208	27,456	58316,1916	8875.0	10800,3281
s336x571c	336	571	50290,3143	5378	699,037	48858,8944	214	29,029	52918,6639	947.0	10800,29
s672x1142a	672	1142	106804,279	5041	2784,524	103994,5599	223	130,806	116041,8722	1.0	10800,4836
s672x1142b	672	1142	107984,5945	5559	3971,361	103545,0402	201	151,132	115926,7268	1.0	10803,7342
s672x1142c	672	1142	101698,3592	5789	4017,57	96312,3303	225	152,053	107933,9914	1.0	10800,3
s672x739a	672	739	90133,8059	6171	2237,746	90101,8277	207	73,202	90379,2926	17692.0	10803,3444
s672x739b	672	739	84199,741	5446	2454,54	83140,0235	205	89,821	85670,9308	677.0	10803,6288
s672x739c	672	739	81742,6619	5613	2656,246	80331,3053	226	106,636	83269,7943	1.0	10800,21
s672x941a	672	941	99765,545	6382	2857,119	94254,3961	190	95,83	108603,3215	1.0	10800,7834
s672x941b	672	941	97879,4052	6447	3684,33	91140,6034	183	101,8	105296,7465	1.0	10802,9788
s672x941c	672	941	94862,9996	5590	3222,699	87649,017	181	101,039	96226,8997	1.0	10802,9799

Ajuste y comparación de métodos

A partir de todos los datos obtenidos en las tablas anteriores, vamos a representar gráficamente cada parámetro a comparar y así se puede ver cuál de los métodos es el más apropiado para obtener el mejor valor de cada uno de ellos.

- Valor objetivo

Para optimizar este parámetro, hay que maximizar el valor de la función objetivo. En las siguientes gráficas se puede ver la representación de ese valor para cada uno de los problemas. El valor azul corresponde al obtenido mediante el método RL1, el rojo mediante el RL2 y el verde mediante el GRB. Este código de colores se mantiene para todo el documento.



Ilustración 12. Representación gráfica del valor objetivo para cada problema pequeño.

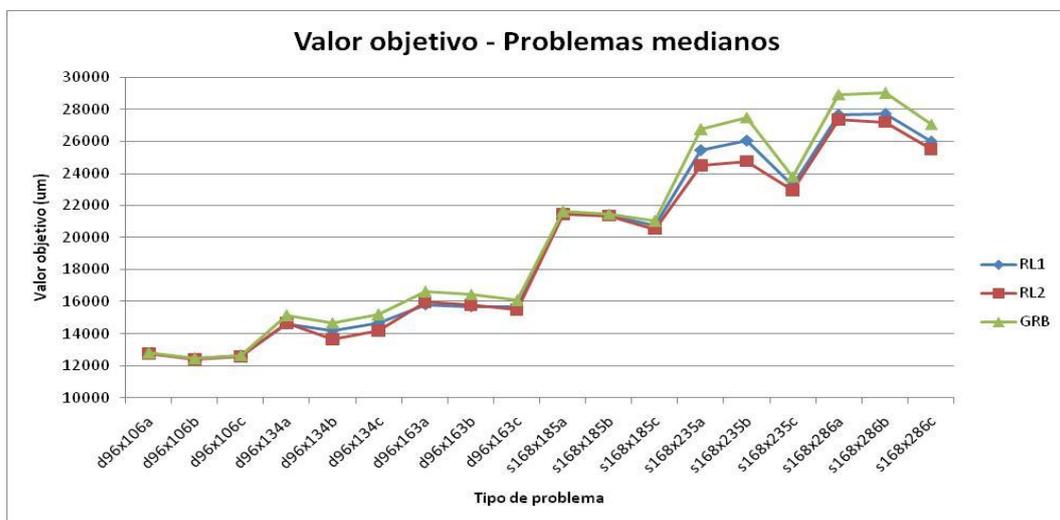


Ilustración 13. Representación gráfica del valor objetivo para cada problema mediano.

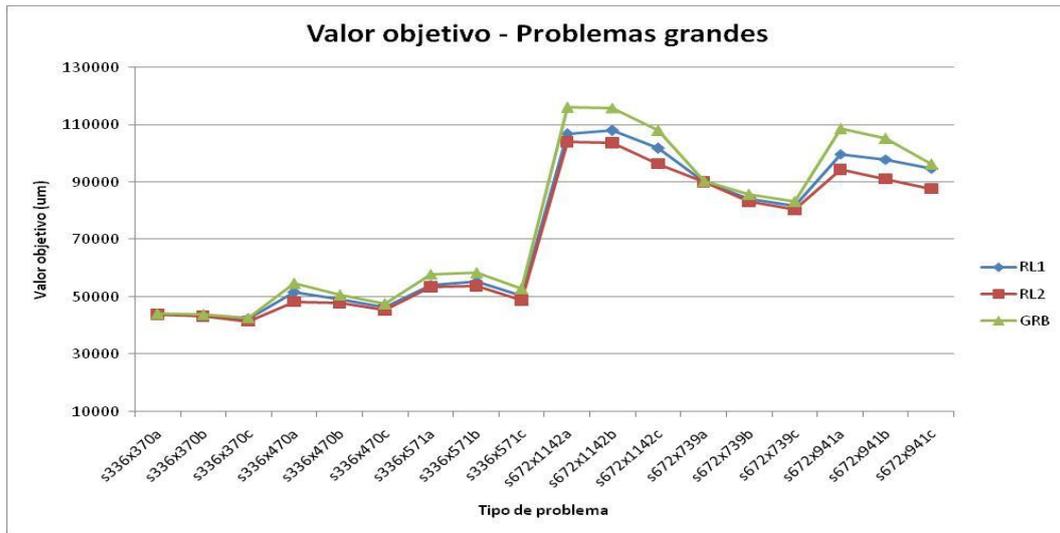


Ilustración 14. Representación gráfica del valor objetivo para cada problema grande.

Al observar las gráficas anteriores, se observa que para todos los tamaños de problema (pequeños en Ilustración 12, medianos en Ilustración 13 o grandes en Ilustración 14) la línea que se encuentra siempre a la altura o por encima del resto es la verde. Por lo que, el método que debemos usar para obtener un mayor valor objetivo es el GRB.

Conclusiones para el mayor valor objetivo:

- Mejor método: GRB
- Método intermedio: RL1
- Peor método: RL2

Ajuste y comparación de métodos

- Número de iteraciones:

Para obtener el valor óptimo en este caso, hay que encontrar el método que nos proporcione el menor número de iteraciones. En las gráficas que aparecen a continuación, se representan los valores obtenidos para cada método de resolución.

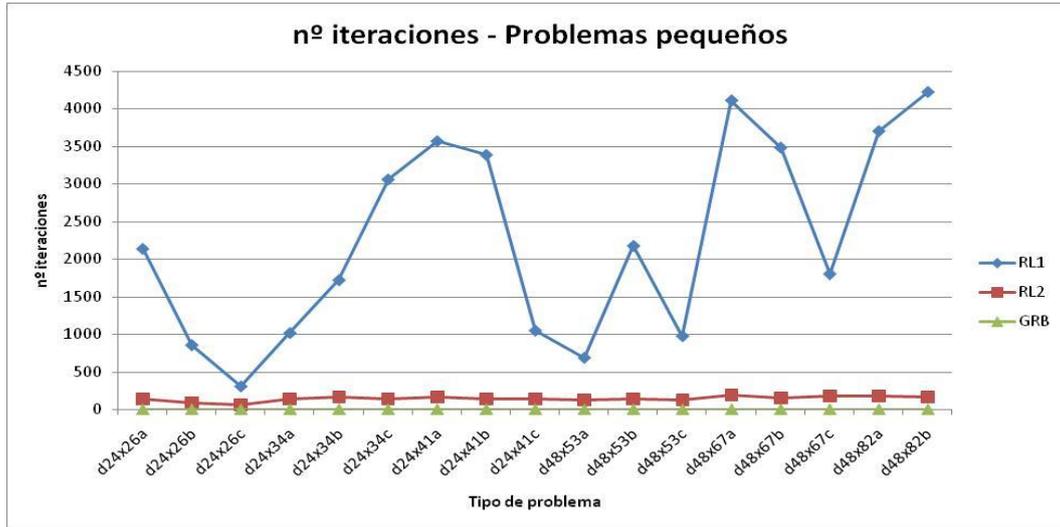


Ilustración 15. Representación gráfica del número de iteraciones para cada problema pequeño.

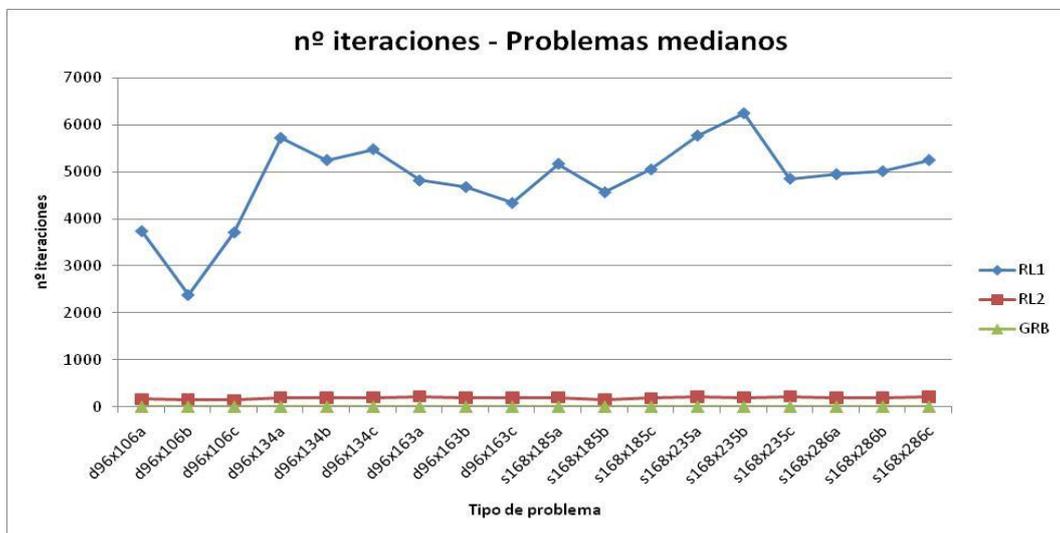


Ilustración 16 Representación gráfica del número de iteraciones para cada problema mediano.

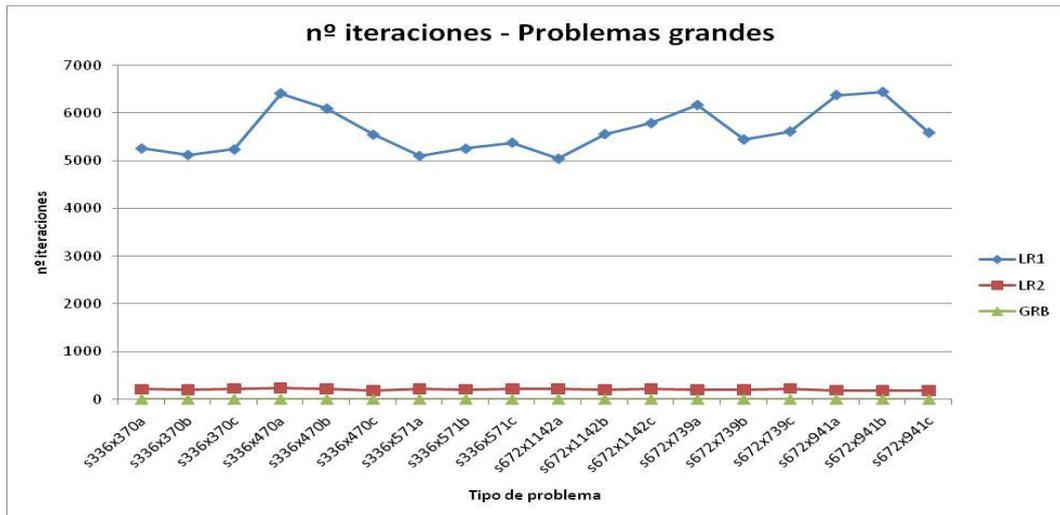


Ilustración 17. Representación gráfica del número de iteraciones para cada problema grande.

En las gráficas anteriores (Ilustración 15, Ilustración 16 e Ilustración 17) hay que encontrar el método que permite el menor número de iteraciones. Para este parámetro, no son comparables los resultados obtenidos para RL y GRB, ya que no tienen el mismo significado. Centrándonos en los dos tipos de RL, vemos que la línea azul (método RL1) está muy por encima del resto, por lo que es el peor. El método RL2 es el que nos proporciona un menor número de iteraciones.

Elección para el menor número de iteraciones:

- Mejor método: RL2
- Peor método: RL1

Ajuste y comparación de métodos

- Tiempo de ejecución:

El último parámetro que vamos a comparar es el tiempo de ejecución. Cuanto menor sea, obtendremos la solución del problema de forma más rápida. Los valores para cada método están representados en las siguientes gráficas (Ilustración 18, Ilustración 19 e Ilustración 20).

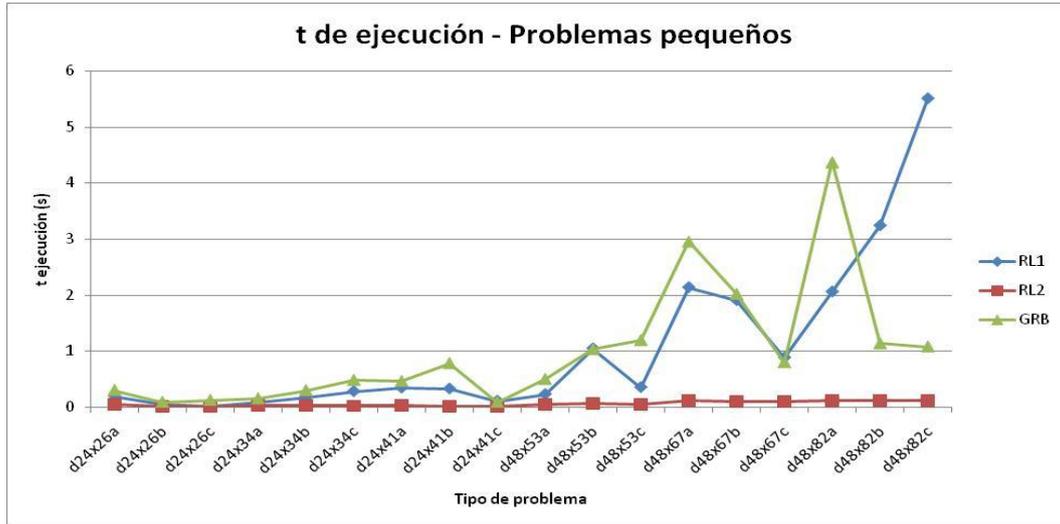


Ilustración 18 Representación gráfica del tiempo de ejecución para cada problema pequeño.

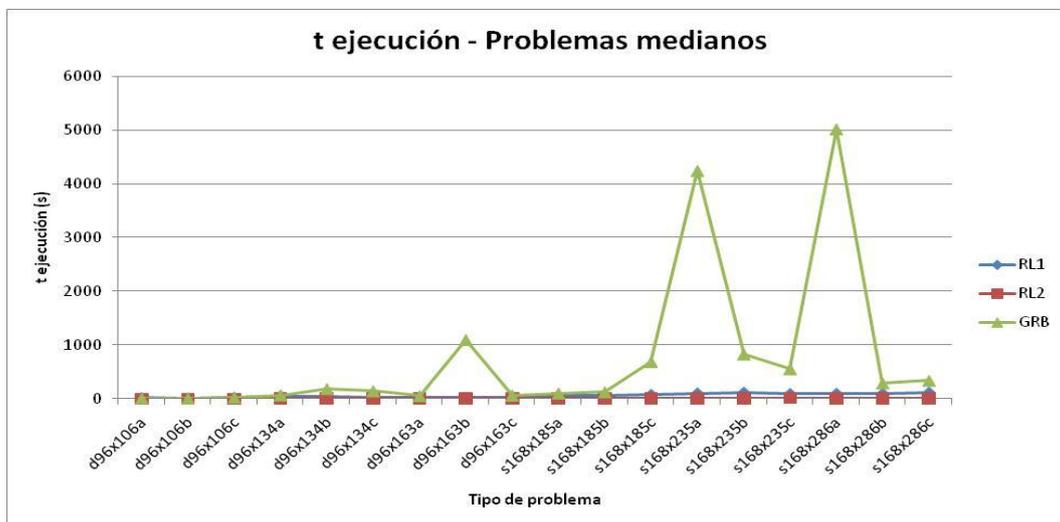


Ilustración 19. Representación gráfica del tiempo de ejecución para cada problema mediano.

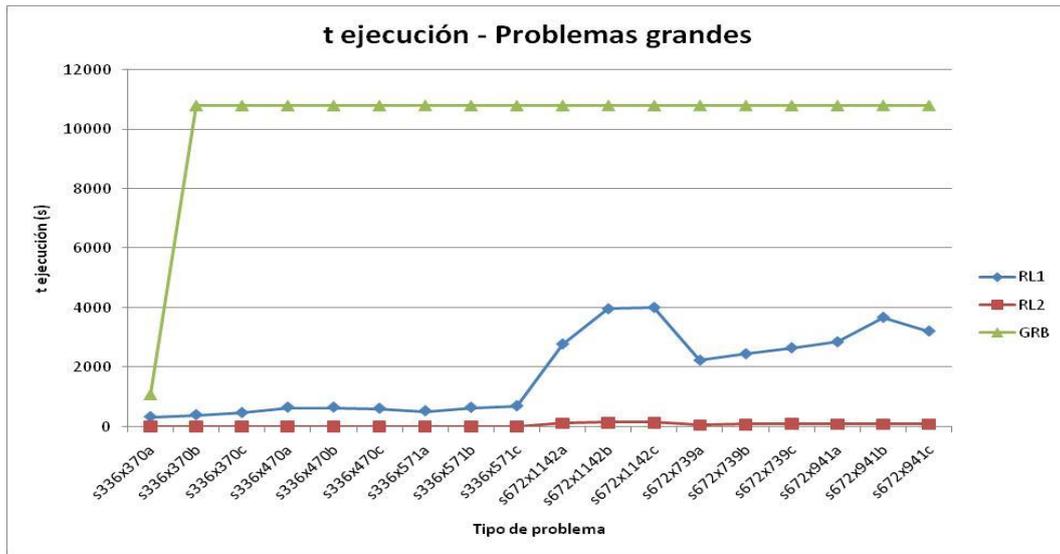


Ilustración 20. Representación gráfica del tiempo de ejecución para cada problema grande.

En este caso, el menor valor para el tiempo de ejecución está representado por la línea roja, es decir, viene dado por el método RL2.

Elección para el menor tiempo de ejecución:

- Mejor método: RL2
- Método intermedio: RL1
- Peor método: GRB

6.4.2. Estudio 2. Solución óptima RL vs GRB

Una vez estudiados todos los parámetros relevantes para cada problema, sin duda el más importante es el valor objetivo obtenido por cada método.

Según el estudio anterior, los dos mejores métodos para encontrar el mayor valor de la función objetivo son (en orden de prioridad) el método GRB y el método LR1.

Se va a estudiar la diferencia que hay entre los resultados obtenidos por ambos métodos, con el fin de encontrar algún tipo de relación entre el tipo de problema y dicha diferencia.

Para este estudio, se van a utilizar los datos obtenidos en el estudio anterior, recogidos en Tabla 6, Tabla 7 y Tabla 8, utilizando la columna de solución óptima.

Ajuste y comparación de métodos

Para obtener la relación adimensional entre el método GRB y LR1, para cada problema se va a realizar la operación:

$$\%diferencia = 100 - \left[\left(\frac{\text{sol óptima LR1}}{\text{sol óptima GRB}} \right) * 100 \right] \quad (6.1)$$

Los resultados obtenidos de la fórmula (6.1), se representan gráficamente en la Ilustración 21:

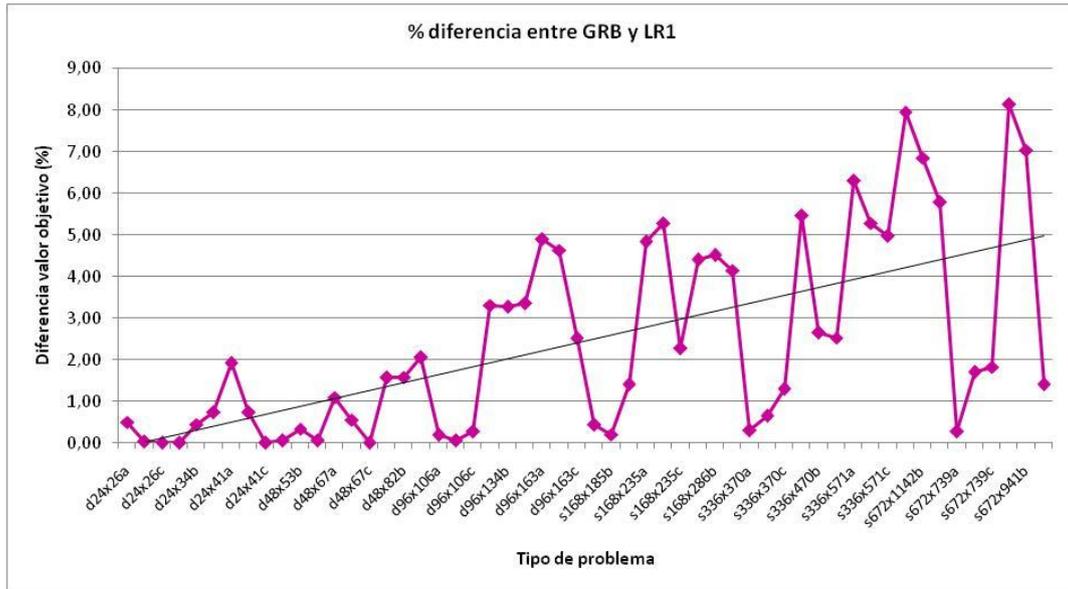


Ilustración 21. Porcentaje de diferencia del valor objetivo entre método LR1 y GRB.

Se ha añadido una línea de tendencia en la Ilustración 21. Como se puede observar, esta recta es ascendente, por lo que podemos afirmar que hay una dependencia entre el tamaño del problema y la diferencia entre el valor objetivo obtenido por ambos métodos.

Conclusión: cuanto mayor es el tamaño del problema, mejor es la solución del método GRB frente al método de relajación lagrangiana.

6.4.3. Estudio 3. Solución óptima en función de problema tipo a, b o c.

En el apartado anterior hemos observado que sí que existe una relación entre la diferencia de resultados de ambos métodos para el valor objetivo, con el tamaño del problema.

A continuación vamos a estudiar si existe una relación de dicha diferencia con el tipo de problema en función del porcentaje de horas punta que tenga, es decir, en función de si es tipo a (0% de horas punta), b (20%) o c (40%).

Para este estudio se van a utilizar los datos recogidos en la Tabla 6, Tabla 7 y Tabla 8, y además se van a resolver todos los problemas con una nueva semilla (B) mediante estos dos métodos, para obtener nuevos valores para la función objetivo y poder sacar conclusiones más firmes. A todos estos datos se les ha aplicado la fórmula (6.1) y se han clasificado según el tipo a, b o c en las siguientes tablas:

Tabla 9. Diferencia (% del valor objetivo) entre método GRB y LR1 con semillas A y B.

Sem A	% de diferencia GRB y LR1			Sem B	% de diferencia GRB y LR1		
	a	b	c		a	b	c
d24x26	0,51	0,04	0,01	d24x26	0,09	0,05	0,00
d24x34	0,00	0,45	0,73	d24x34	0,08	0,28	0,72
d24x41	1,92	0,73	0,00	d24x41	0,04	2,25	0,03
d48x53	0,05	0,34	0,07	d48x53	0,05	0,09	0,36
d48x67	1,10	0,56	0,00	d48x67	2,40	1,80	0,26
d48x82	1,58	1,58	2,06	d48x82	1,99	2,23	1,33
d96x106	0,20	0,06	0,28	d96x106	0,03	0,36	0,36
d96x134	3,31	3,28	3,36	d96x134	3,79	3,60	1,25
d96x163	4,90	4,64	2,52	d96x163	4,30	4,76	3,03
s168x185	0,45	0,21	1,42	s168x185	0,10	0,16	1,49
s168x235	4,84	5,26	2,28	s168x235	5,25	4,28	2,86
s168x286	4,41	4,52	4,15	s168x286	4,21	4,52	4,89
s336x370	0,30	0,66	1,30	s336x370	0,21	0,51	1,76
s336x470	5,46	2,66	2,53	s336x470	6,21	5,20	2,03
s336x571	6,30	5,28	4,97	s336x571	5,74	5,54	4,75
s672x1142	7,96	6,85	5,78	s672x1142	7,10	6,31	6,06
s672x739	0,27	1,72	1,83	s672x739	0,07	0,45	1,99
s672x941	8,14	7,04	1,42	s672x941	7,87	6,24	2,87

Los resultados recogidos en la Tabla 9, se muestran gráficamente a continuación:

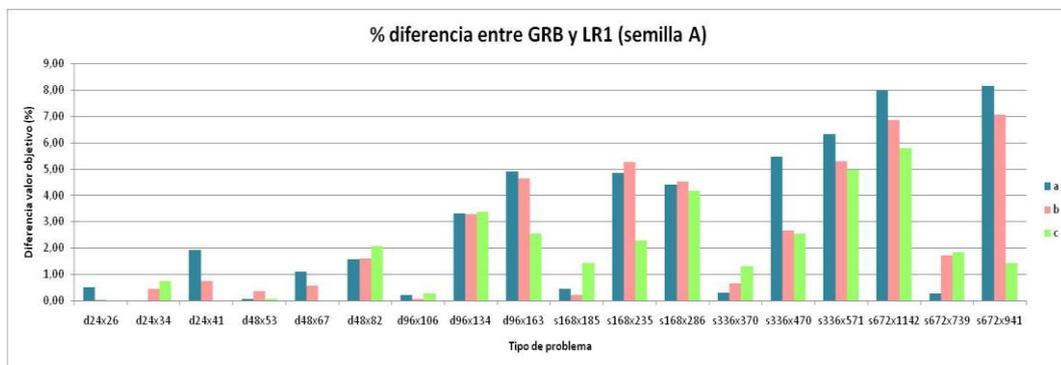


Ilustración 22. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla A.

Ajuste y comparación de métodos

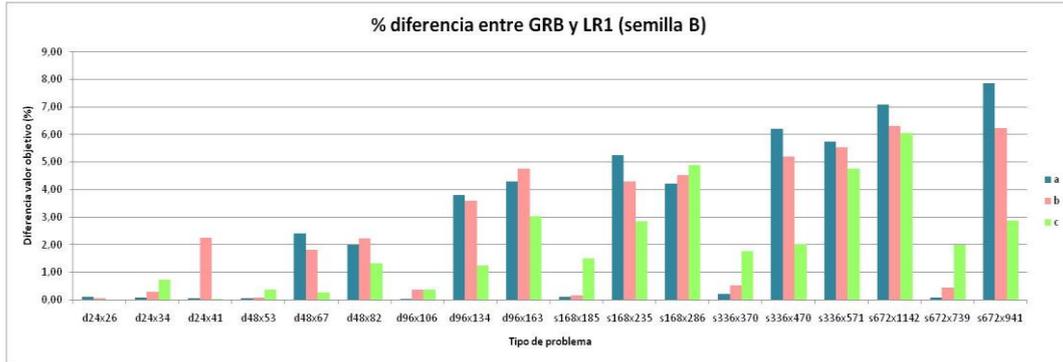


Ilustración 23. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla B.

Tanto en la Ilustración 22, como en la Ilustración 23, si se comparan para cada tipo problema las tres alturas de las barras, se observa que en la mayoría de los casos la más pequeña es la verde (sobre todos en los problemas grandes, que son los más representativos). Esto nos permite afirmar, que cuando el problema es de tipo c, la distancia entre el valor objetivo obtenido por ambos métodos es más pequeña.

El método GRB cuenta con una gran ventaja respecto al método LR1. Esa ventaja es el tiempo de ejecución, es mucho más elevado por lo que tiene más tiempo para lograr un mayor valor objetivo. Por ello, antes de confirmar las conclusiones, vamos a resolver todos los problemas para una de las semillas utilizadas (B) de nuevo, pero esta vez el tiempo de ejecución (criterio de parada) será el mismo que ha necesitado el método LR1 para obtener sus valores, es decir 4000 segundos. Los resultados obtenidos son:

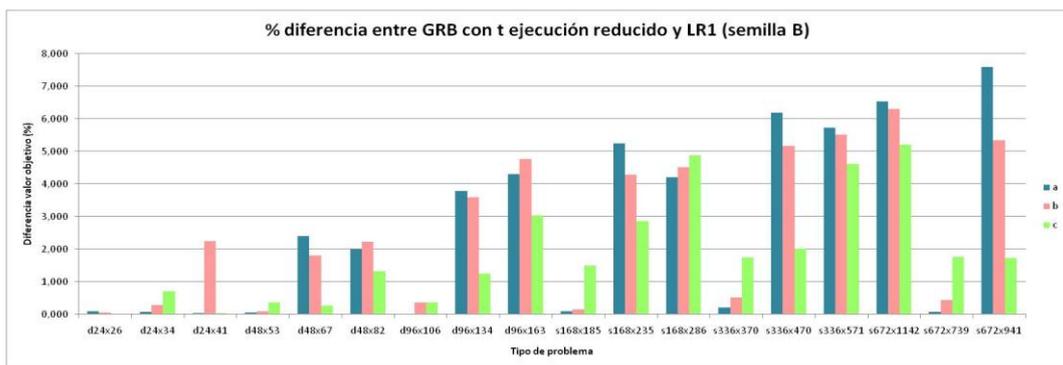


Ilustración 24. Diferencia (% del valor objetivo) entre método GRB con t de ejecución reducido y LR1 con semilla B.

Si comparamos las dos gráficas de la semilla B (Ilustración 23 e Ilustración 24), vemos que la diferencia es mínima, por lo que aunque igualemos el tiempo de

ejecución de ambos métodos las conclusiones obtenidas anteriormente no cambian.

Conclusión: Sí que hay relación entre el tipo de problema (a/b/c) y la diferencia entre los valores objetivos obtenidos mediante ambos métodos. En los problemas de tipo c, con una cantidad alta de horas punta (40%), la diferencia es menor.

6.4.4. Estudio 4: Resolución con horizonte de planificación periódico.

Todos los estudios realizados en este documento hasta el momento, como se indicó anteriormente, están llevados a cabo para un horizonte de planificación cerrado.

En este caso, con horizonte de planificación periódico nos referimos a la posibilidad de programas que se repiten periódicamente a lo largo de varios ciclos, lo que hace posible que la salida de un vuelo que llega al final del periodo de planificación, se produzca al comienzo del periodo.

Para este nuevo análisis se van a resolver los 54 problemas generados para una semilla (A), con el método GRB y LR1 pero con la condición de que el horizonte de planificación sea periódico. Posteriormente los compararemos para ver cuál de los dos obtiene un mayor valor objetivo.

El tiempo de ejecución de ambos métodos ha sido igualado para que los dos dispongan del mismo tiempo para alcanzar la solución óptima. Vamos a compararles en función del valor objetivo obtenido, cuyos resultados están recogidos en las siguientes tablas:

Ajuste y comparación de métodos

Tabla 10. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas pequeños.

	Problema	Nº franjas horarias	Nº de vuelos	Método RL1		Método GRB	
				Sol óptima (um)	nº de iteraciones	Sol óptima (um)	nº de iteraciones
Pequeños (semilla A)	d24x26a	24	26	2794,2115	1667	2801,9623	1.0
	d24x26b	24	26	3068,6607	1528	3069,7932	1.0
	d24x26c	24	26	2903,8135	439	2903,8135	1.0
	d24x34a	24	34	3685,6651	4690	3734,985	1.0
	d24x34b	24	34	3580,926	4882	3662,4376	84.0
	d24x34c	24	34	3619,5396	4325	3678,177	1.0
	d24x41a	24	41	3917,0785	3664	3965,4113	1.0
	d24x41b	24	41	3829,7455	2987	3903,3524	1.0
	d24x41c	24	41	3795,6184	4297	3893,1851	50.0
	d48x53a	48	53	6293,9635	756	6297,0885	1.0
	d48x53b	48	53	6432,0506	1284	6432,0506	1.0
	d48x53c	48	53	6161,6405	1011	6165,4654	1.0
	d48x67a	48	67	7365,3615	4631	7522,6435	2377.0
	d48x67b	48	67	7245,0117	3318	7285,7482	42.0
	d48x67c	48	67	7261,6792	5652	7472,5477	858.0
	d48x82a	48	82	7837,0134	4780	8144,9248	11.0
	d48x82b	48	82	8136,1912	1677	8141,1997	1.0
d48x82c	48	82	8101,1133	1696	8105,8935	1.0	

Tabla 11. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas medianos.

	Problema	Nº franjas horarias	Nº de vuelos	Método RL1		Método GRB	
				Sol óptima (um)	nº de iteraciones	Sol óptima (um)	nº de iteraciones
Medianos (semilla A)	d96x106a	96	106	12772,1756	4154	12807,063	38.0
	d96x106b	96	106	12440,2636	2686	12447,3804	1.0
	d96x106c	96	106	12647,255	2986	12660,4032	1.0
	d96x134a	96	134	14482,6636	5773	15147,666	3816.0
	d96x134b	96	134	13952,5057	6228	14736,0415	1816.0
	d96x134c	96	134	14569,1178	5673	15284,885	4838.0
	d96x163a	96	163	15965,8315	5088	16672,8426	4442.0
	d96x163b	96	163	15952,9177	4979	16576,7833	3840.0
	d96x163c	96	163	15461,6791	4928	16173,746	4857.0
	s168x185a	168	185	21573,6526	4850	21625,1751	1.0
	s168x185b	168	185	21476,7649	4644	21520,4647	2927.0
	s168x185c	168	185	21193,1494	4714	21386,6641	31000.0
	s168x235a	168	235	25247,8857	5974	26844,2821	20613.0
	s168x235b	168	235	25623,0495	6196	27684,2621	22445.0
	s168x235c	168	235	23834,6615	6163	25147,7867	10705.0
	s168x286a	168	286	27326,0771	5117	29054,2981	20765.0
	s168x286b	168	286	27403,6276	5352	29286,4419	26901.0
s168x286c	168	286	26726,0412	5674	28425,1895	8720.0	

Tabla 12. Resultados obtenidos para horizonte de planificación periódico en solución óptima y nº de iteraciones para problemas grandes.

			Método RL1		Método GRB	
Problema	Nº franjas horarias	Nº de vuelos	Sol óptima (um)	nº de iteraciones	Sol óptima (um)	nº de iteraciones
			Grandes (semilla A)			
s336x370a	336	370	43962,9299	5369	44096,49	2338.0
s336x370b	336	370	43792,1252	4746	43897,3666	7230.0
s336x370c	336	370	42892,6096	5691	43358,6425	1729.0
s336x470a	336	470	50903,6477	6572	54196,9316	243.0
s336x470b	336	470	47906,3522	6658	51512,2337	28.0
s336x470c	336	470	47738,0758	6831	48330,2232	1.0
s336x571a	336	571	54338,8266	5317	57783,1719	28.0
s336x571b	336	571	54438,6337	5065	57943,2316	27.0
s336x571c	336	571	52331,3986	5615	54710,5589	1.0
s672x1142a	672	1142	106447,788	5151	113625,59	1.0
s672x1142b	672	1142	109155,261	5355	107047,957	1.0
s672x1142c	672	1142	104560,566	5416	90171,8316	1.0
s672x739a	672	739	90122,5557	6217	90353,4604	801.0
s672x739b	672	739	85792,7225	6232	85289,7508	1.0
s672x739c	672	739	84796,9267	5896	83704,7635	1.0
s672x941a	672	941	100095,103	6418	105874,707	1.0
s672x941b	672	941	98232,0101	6821	102881,108	1.0
s672x941c	672	941	94031,905	6857	83424,7516	1.0

Para optimizar el valor objetivo, buscamos el método que nos proporcione un mayor valor para la solución. Usando los datos de Tabla 10, Tabla 11 y Tabla 12, obtenemos las siguientes representaciones:

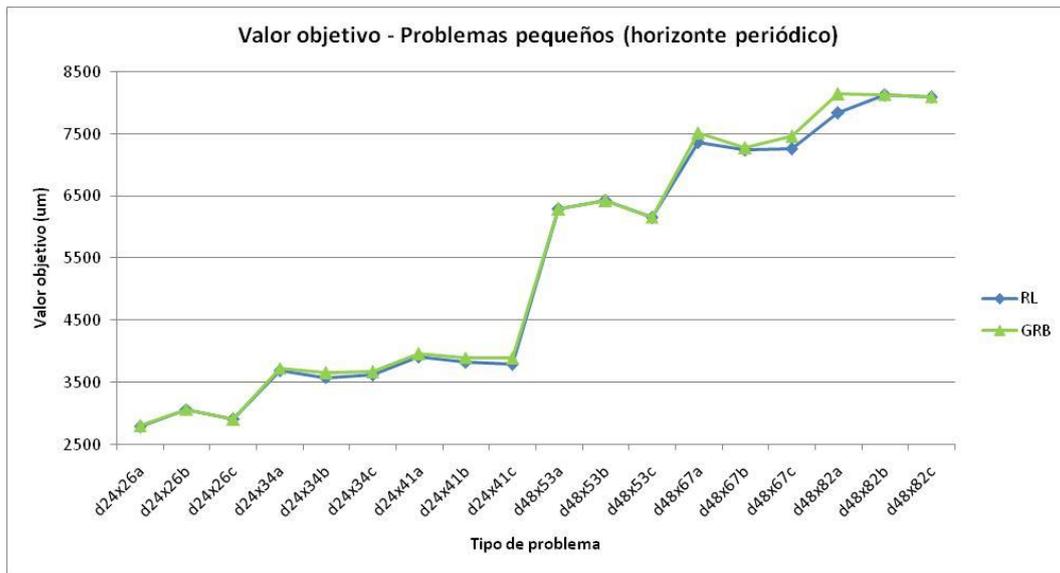


Ilustración 25. Valor objetivo para horizonte periódico métodos RL y GRB para problemas pequeños.

Ajuste y comparación de métodos

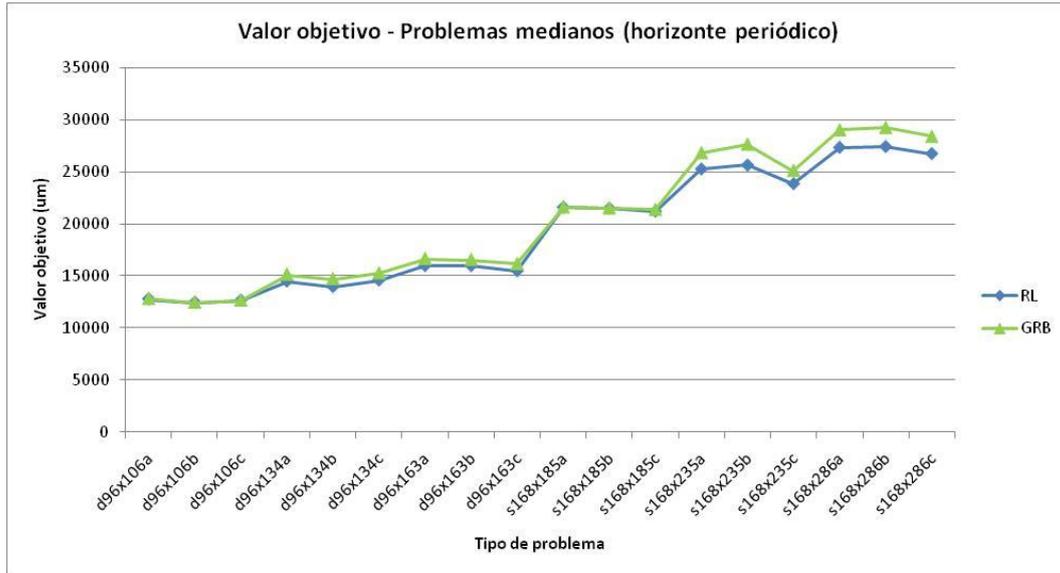


Ilustración 26. Valor objetivo para horizonte periódico métodos RL y GRB para problemas medianos.

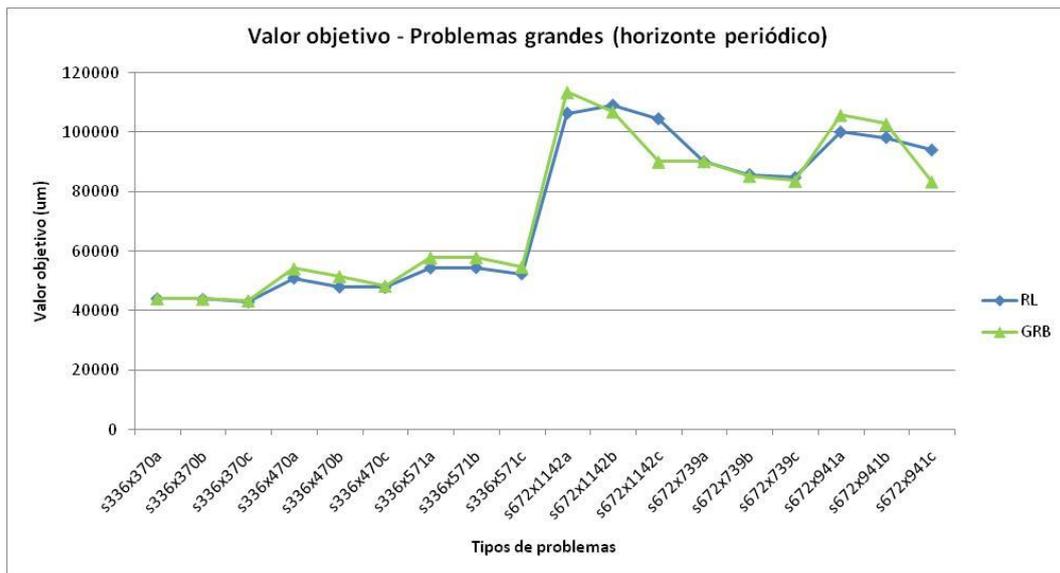


Ilustración 27. Valor objetivo para horizonte periódico métodos RL y GRB para problemas grandes.

Observando las gráficas anteriores, para los problemas pequeños y medianos (Ilustración 25 e Ilustración 26) el método GRB siempre nos da mayor valor para la función objetivo. En el caso de los problemas grandes (Ilustración 27), hay 3 casos en los que no se cumple el hecho anterior. Eso es debido, a que se ha reducido el tiempo de ejecución del método GRB y al ser problemas muy grandes no le ha dado tiempo a superar el valor del método RL. Por este motivo se considera también que es mejor el método GRB.

Conclusión: en los problemas con horizonte de planificación periódico, mediante el método GRB pase consiguen valores de la función objetivo más cercanas al óptimo que con el de Relajación Lagrangiana, al igual que pasaba con el horizonte de planificación cerrado.

Ahora vamos a estudiar la dependencia de la diferencia entre valores objetivo, con el tipo de problema (a, b o c). Para ello, aplicamos la fórmula utilizada en el apartado anterior (6.1) a los datos de Tabla 10, Tabla 11 y Tabla 12, y tras clasificarlos en función del porcentaje de horas punta programadas, obtenemos la siguiente representación:

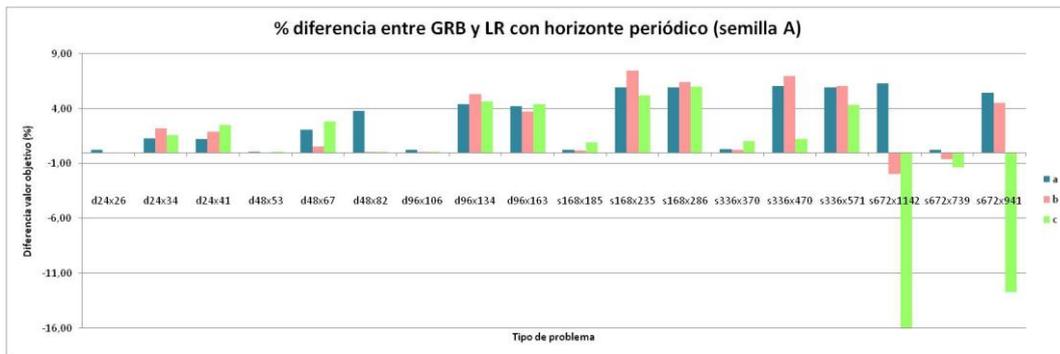


Ilustración 28. Diferencia (% del valor objetivo) entre método GRB y LR1 con semilla A para horizonte de planificación periódico.

En la Ilustración 28, observamos que para los problemas grandes, que son los que más similitud tienen con la realidad, los problemas tipo c obtienen valores más próximos al método LR que los otros tipos.

En los últimos tres casos, en los problemas más grandes, los valores obtenidos para GRB son incluso inferiores que los obtenidos mediante LR (véase en las barras del eje negativo). Y, en el caso de que el problema sea tipo c, esa diferencia es mucho mayor.

Conclusión: los problemas tipo c (40% de vuelos programados en horas punta), son los que menos % de mejoría tienen en cuanto al valor objetivo obtenido en RL, comparado con los otros dos tipos (a y b) de su mismo tamaño. Esto quiere decir, que cuanto mayor sea el % de vuelos programados en horas punta, más costoso es para el método GRB acercarse a la solución óptima.

6.5. Conclusiones

Una vez que se han realizado todos los estudios, se puede concluir:

Para optimizar el método de relajación lagrangiana se ha realizado el ajuste de dos parámetros: el número de pasos sin mejora de la solución para que se actualicen los multiplicadores (alfa) y el factor de actualización del parámetro que actualiza dichos multiplicadores (beta). Los resultados obtenidos han sido:

- Mayor valor objetivo: alfa=70, beta=0,91
- Mínimo número de iteraciones: alfa=10, beta=0,6

La optimización del número de iteraciones origina resultados de peor calidad. Por lo tanto, aunque de lugar a un procedimiento más rápido no es recomendable.

Una vez ajustados los parámetros y resueltos todos problemas mediante relajación lagrangiana (LR) (con parámetros alfa = 70, beta = 0,91) y el método de programación lineal con Gurobi (GRB), se llega a la conclusión de que el mejor método para obtener el mayor valor objetivo suele ser el GRB. Esto sucede siempre en problemas con horizonte de planificación limitado, pero en casos más complejos con problemas grandes y horizonte de planificación periódicos ambos métodos tienen similar desempeño.

También se ha comprobado, para un horizonte de planificación cerrado, que existe dependencia entre la diferencia de valores objetivos conseguidos por ambos métodos, y el tipo de problema. En primer lugar, cuanto mayor es el tamaño del problema, mejor es la solución del método GRB frente al método de relajación lagrangiana. Además, los problemas de tipo c (40% de horas punta), tienen una diferencia menor entre métodos que el resto.

Si el horizonte de planificación es periódico, hay más dificultad en encontrar la solución óptima. En estos casos, se observa que los resultados con LR son mejores que los obtenidos con Gurobi en los problemas con alto número de vuelos en horas punta (tipo c).

CAPÍTULO 7. Conclusiones

En este capítulo se recogen las conclusiones obtenidas tras la elaboración de este trabajo de fin de grado.

El aumento de la demanda de transporte aéreo en los últimos años ha provocado un gran desajuste de ésta con la capacidad de los aeropuertos. Para equilibrarlo hay que optimizar la asignación de las franjas horarias a las compañías aéreas. Este problema es posible modelarlo de forma efectiva mediante programación lineal entera mixta.

El problema puede ser resuelto mediante Relajación Lagrangiana, un método que nos permite simplificar el problema inicial y posteriormente resolverlo mediante el método del subgradiente basado en subastas combinatorias.

Para este método se ha realizado un ajuste de parámetros, con el fin de optimizar su resolución, es decir, que tenga una buena convergencia. Con este procedimiento vamos a conseguir una buena solución, pero no la óptima. Las ventajas que proporciona son: nos permite distribuir el problema (cada aerolínea puede resolver su parte por separado), no es necesario que los participantes conozcan la función objetivo, ya que se puede implementar como subasta. Por último, en el caso de que haya algún cambio en el modelo del problema, una vez que se conocen los precios (multiplicadores), pueden volver a utilizarse, no es necesaria una gran modificación.

El paquete de resolución mediante programación lineal de Gurobi (método GRB), nos permite encontrar un valor de la función objetivo más cercano al óptimo. Es un método capaz de resolver problemas grandes, pero con el aumento de tamaño debido a su elevada complejidad, la resolución baja de calidad. De hecho, en los problemas muy complejos (grandes y con horizonte de planificación periódico) el desempeño del método es similar al RL.

En cuanto a la diferencia (%) entre el valor objetivo obtenido por ambos métodos, se ha encontrado una relación con el tipo de problema según el porcentaje de vuelos programados en horas punta. En los problemas tipo c (con 40% de vuelos en horas puntas), tanto en horizonte de planificación cerrado como periódico, el método RL ha demostrado comportarse bastante eficazmente.

BIBLIOGRAFÍA

Alvarado Ortega, A. (2016). Trabajo de investigación. *Método de Branch and Bound*. Arequipa, Perú: Universidad Tecnológica del Perú. Facultad de Ingeniería Industrial y Mecánica.

Araúzo, J. A. (2018). *Relajación Lagrangiana para la asignación de slots aeroportuarios*. Informe Interno, Escuela de Ingenierías Industriales, Universidad de Valladolid, Dpto. de Organización de Empresas y CIM.

Ball, M. O., Ausubel, L. M., Berardino, F., Cramton, P., Donohue, G., Hansen, M., y otros. (2007). Ball, M. O., Ausubel, L. M., Berardino, F., Cramton, P., Market-based alternatives for managing congestion at new york's laguardia airport. *Ball, M. O., Ausubel, L. M., Berardino, F., Cramton, P., Donohue, G., Hansen, M., & Hoffman, K. (2007). AirNeth Annual Conference, The Hague*.

Ball, M., Berardino, F., & Hansen, M. (2018). The use of auctions for allocating airport access rights. *Transportation Research Part A: Policy and Practice*, 114, 186-202.

Bazaraa, M. (1998). *Programación lineal y flujo en redes*. México: Limusa.

Bertsekas, D. (1990). The auction algorithm for assignment and other network flow problems: a tutorial. *Interfaces* 20(4), 133–149.

Boirivant, J. (2009). La programación lineal, aplicación de las pequeñas y medianas empresas. *Reflexiones*, 88(1), 4.

Castillo, E., Conejo, A., Pedregal, P., Garcia, R., & Alguacil, N. (2002). Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia.

Czerny, A., & Tegner, H. (2002). Secondary Markets for Runway Capacity. Bruselas: Second Seminar of the IMPRINT- EUROPE.

de la Fuente, M. (2016). Algoritmo de Relajación Lagrangiana para la asignación de franjas horarias a vuelos. Universidad de Valladolid. Escuela de ingenierías industriales.

Dekrajangpetch, S. S. (1999). Auction implementation problems using Lagrangian relaxation. *IEEE Transactions on Power Systems*, doi:10.1109/59.744488.

Delgado Díez, R. (2014). Análisis de la red europea de aeropuertos mediante la teoría de redes. Universidad de Valladolid. Escuela de ingenierías Industriales.

Bibliografía

Dewan, P., & Joshi, S. (2002). Auction-based distributed scheduling in a dynamic job shop environment. *International Journal of Production Research*, 40(5) , 1173–1191.

Díaz Olariaga, O., & Benito Ruiz de Villa, A. (2010). Alternativas de mercado en la asignación de slots aeroportuarios. *Díaz Olariaga, O., & Benito Ruiz de Villa, A. (2010). AlternIngeniería Aeronáutica y Astronáutica*, (396) , 1-10.

Forde, T., & Doyle, L. (2008). A combinatorial clock auction for OFDMA-based cognitive wireless networks. *Forde TK, Doyle LE (2008). A combinatorial clock auction for OFDMA-based coProc of the 3rd International Symposium on Wireless Pervasive Computing* , 329–333.

Gale, I., & O'Brien, D. (2013). The Welfare Effects of Use-or-Lose Provisions in Markets with Dominant Firms. *Gale, Ian, and Daniel P. O'Brien. 2013. "The Welfare Effects of UseAmerican Economic Journal: Microeconomics*, 5 (1) , 175-93.

Gestión de Operaciones. (2011). Obtenido de <https://www.gestiondeoperaciones.net/programacion-entera/ejemplo-del-algoritmo-de-branch-and-bound-ramificacion-y-acotamiento/>

Guignard, M. (2003). Lagrangean relaxation. *TOP* , 11(2):151–200.

Herrera García, A. (2006). Alternativas de solución para problemas de capacidad aeroportuaria. *PUBLICACION TECNICA*, (284) .

Jacquillat, A. &. (2018). A roadmap toward airport demand and capacity management. *Transportation Research Part A: Policy and Practice*, 114 , 168-185.

Laviós Villahoz, J. (2013). Análisis de la Relajación Lagrangiana como método de programación de talleres flexibles en un entorno multiagente. Universidad de Burgos. Departamento de Ingeniería Civil.

López, M. L. (2003). Modelamiento a través de la programación lineal entera mixta del problema de programación de operaciones en el taller de trabajo. *UIS Ingenierías* , 2(1), 9-17.

Luciano Potenze, P. (2018). *Gaceta Aeronáutica*. Obtenido de <https://www.gacetaaeronautica.com/gaceta/wp-101/?p=27011>

Matthews, B., & Menaz, B. (Noviembre de 2003). Airport Capacity: The problem of Slot Allocation. *Working Paper* . Institute for Transport Studies. University of Leeds.

Medrano, J. (2015). *SlideShare*. Obtenido de <https://es.slideshare.net/jaimemedrano771/programacin-lineal-entera-y-binaria>

Noticias El Universo. (2018). Obtenido de <https://www.eluniverso.com/noticias/2018/01/17/nota/6571870/trafico-aereo-mundial-crecio-2017-gracias-companias-low-cost>

Odoni, A. R. (1987). The flow management problem in air traffic control. *Flow control of congested networks*. Springer, Berlin, Heidelberg. , 269-288.

Parkes, D., & Ungar, L. (2001). An auction-based method for decentralized train scheduling. *Parkes DC, Ungar LH (2001). An auction-based method Proceedings of the Fifth International Conference on Autonomous Agents* , 43-50.

Pascual Pulido, L. (2018). Modelos de Programación Lineal para la Asignación de Franjas Horarias a Vuelos. Universidad de Valladolid. Escuela de ingenierías industriales.

Ramón Vilarasán, D. (2014). *Hosteltur*. Obtenido de https://www.hosteltur.com/183900_infografia-trafico-aereo-mundial-sera-doble-20-anos-asia-pacifico-cabeza.html

Rassenti, S., Smith, V., & Bulfin, R. (1982). A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics* 13(2) , 402–417.

Roldán, P. N. (s.f.). *Economipedia*. Obtenido de <https://economipedia.com/definiciones/subasta.html>

T&E. (Noviembre de 2005). Airport capacity, efficiency and safety in Europe. *Position Paper* . Bruselas.

Zografos, K., Andreatta, G., & Odoni, A. (2013). *Modelling and managing airport performance*. John Wiley & Sons.

ANEXO

Programación de una clase en JAVA para que recoja los resultados obtenidos en cada uno de los problemas resueltos y los reúna todos en un nuevo fichero, con el fin de facilitar la exportación de datos.

```
package asa;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class FilesManipulator {
    static void jointLRSummaries(String folder, String fileName) {
        File dir=new File(folder);
        if(dir.isDirectory()) {
            try{
                FileWriter fw = new FileWriter(folder+File.separator+fileName+".txt");
                PrintWriter pw = new PrintWriter(fw);
```

```
pw.println("Problem\tTimeIntervals\tFlights\tASS\tLoadFactor\tObj\tGap\tRounds\tTime\tAllocFlights\tAllo
cGU\tSchCost");
for(File file: dir.listFiles()) {
    if(file.isFile() && file.getName().endsWith("_lr.txt")) {
        FileReader f=null;
        try {
            f = new FileReader(file);
            BufferedReader b = new BufferedReader(f);
            String line;
            boolean summaryFound=false;
            while((line = b.readLine()) != null) {
                if(line.equals("LR Summary results")) {
                    summaryFound=true;
                    break;
                }
            }
        }
        if(summaryFound) {
            String data=file.getName().split("_")[0];
            try {
                for(int i=0;i<11;i++) {
                    data=data+"\t"+b.readLine().split("\t")[1];
                }
                pw.println(data);
            }
        }
    }
}
```

```
        } catch(Exception e) {
            data=file.getName().split("_")[0]+"\\tformat error";
            pw.println(data);
        }
    }
    b.close();
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
if(f!=null) f.close();
}
}
fw.close();
pw.close();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
```

```
}
static void jointGRBSummaries(String folder, String fileName) {
    File dir=new File(folder);
    if(dir.isDirectory()) {
        try{
            FileWriter fw = new FileWriter(folder+File.separator+fileName+".txt");
            PrintWriter pw = new PrintWriter(fw);
            pw.println("Problem\tTimeIntervals\tFlights\tASS\tLoadFactor\tObj\tGap\tNodes\tTime\tAllocFlights\tAlloc
            GU\tSchCost");
            for(File file: dir.listFiles()) {
                if(file.isFile() && file.getName().endsWith("_grb.txt")) {
                    FileReader f=null;
                    try {
                        f = new FileReader(file);
                        BufferedReader b = new BufferedReader(f);
                        String line;
                        boolean summaryFound=false;
                        while((line = b.readLine()) != null) {
                            if(line.equals("Summary results")) {
                                summaryFound=true;
                                break;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
        if(summaryFound) {
            String data=file.getName().split("_")[0];
            try {
                for(int i=0;i<11;i++) {
                    data=data+"\t"+b.readLine().split("\t")[1];
                }
                pw.println(data);
            } catch(Exception e) {
                data=file.getName().split("_")[0)+"\tformat error";
                pw.println(data);
            }
        }
        b.close();
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    if(f!=null) f.close();
}
}
```

```
        fw.close();
        pw.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}
```