

# Trabajo Fin de Grado

## Ingeniería Aeroespacial

### Programa para resolver el problema de Aircraft Routing

Autor: Javier Verdejo Fernández

Tutor: José María del Castillo Granados

**Dpto. Ingeniería y Ciencia de los Materiales y  
del Transporte**  
**Escuela Técnica Superior de Ingeniería**

Sevilla, 2021





Trabajo Fin de Grado  
Ingeniería Aeroespacial

# **Programa para resolver el problema de Aircraft Routing**

Autor:

Javier Verdejo Fernández

Tutor:

José María del Castillo Granados

Catedrático de Universidad

Dpto. de Ingeniería y Ciencia de los Materiales y del Transporte

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Grado: Programa para resolver el problema de Aircraft Routing

Autor: Javier Verdejo Fernández

Tutor: José María del Castillo Granados

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal



# Agradecimientos

---

*A mi familia, por apoyarme durante todos estos años en la consecución de mis objetivos.*





El objetivo de este Trabajo Fin de Grado es desarrollar un programa informático para resolver el problema de rotación de aviones o 'Aircraft Routing'. Se utiliza el software *Matlab* para realizar esta tarea, cuyo resultado final consiste en la obtención de las rotaciones finales que deberán realizar las aeronaves de una compañía aérea. Se utilizarán diferentes archivos con el objetivo de poder realizar este proceso utilizando distintas estrategias de optimización. Además, se resolverán un ejemplo ficticio y real para comprobar el correcto funcionamiento de los algoritmos elaborados y, finalmente, se realizará un análisis respecto a posibles cambios horarios a introducir en la programación de vuelos con el fin de mejorar las soluciones finales obtenidas.



# Abstract

---

The aim of this Final Project is to develop a computer programme to solve the Aircraft Routing problem. The ultimate objective of this programme is to obtain the rotations to operate by the company and Matlab is used to achieve this purpose. Different files will be created so that various strategies of optimization could be applied to the results. Furthermore, both fictitious and real examples will be solved in order to check the operation of the algorithm. Finally, flight scheduling changes will be analyzed to improve the final solutions.



<b>Agradecimientos</b>	<b>vii</b>
<b>Resumen</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Índice</b>	<b>xiii</b>
<b>Índice de Tablas</b>	<b>xv</b>
<b>Índice de Figuras</b>	<b>xvii</b>
<b>Notación</b>	<b>xx</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Aircraft Routing</b>	<b>5</b>
2.1 <i>Requerimientos</i>	5
2.2 <i>Modelo matemático</i>	6
2.2.1 <i>Generación de rotaciones</i>	7
2.2.2 <i>Función objetivo</i>	8
2.2.3 <i>Restricciones</i>	10
2.3 <i>Solución</i>	12
<b>3 Programación Lineal</b>	<b>15</b>
3.1 <i>Complejidad del problema</i>	15
3.2 <i>Programación lineal entera</i>	16
3.3 <i>Programación lineal en Matlab</i>	17
3.3.1 <i>Intlinprog</i>	19
<b>4 Resolución del problema</b>	<b>25</b>
4.1 <i>Archivos</i>	25
4.2 <i>Ejemplo Ultimate Air</i>	27
4.2.1 <i>Boeing 737</i>	31
4.2.2 <i>Boeing 757</i>	38
4.2.3 <i>Cambios horarios</i>	43
<b>5 Aplicación práctica</b>	<b>57</b>
5.1 <i>Descripción general</i>	57
5.2 <i>Obtención de datos</i>	59
5.3 <i>Resultados</i>	61
5.3.1 <i>Maximizar las posibilidades de mantenimiento</i>	61
5.3.2 <i>Maximizar el número de conexiones de vuelos</i>	64
5.3.3 <i>Minimizar desviación respecto a horas de referencia</i>	66
5.3.4 <i>Minimizar costes</i>	68
<b>6 Conclusiones</b>	<b>71</b>
<b>Referencias</b>	<b>72</b>
<b>Índice de Conceptos</b>	<b>75</b>
<b>Glosario</b>	<b>77</b>

<b>ANEXOS</b>	<b>79</b>
<i>ANEXO A</i>	79
Routing1	79
Routing2	82
Routing3	84
Routing4	85
Rotaciones1	87
Rot1	89
<i>ANEXO B</i>	91

# ÍNDICE DE TABLAS

---

<b>Tabla 1:</b> Ejemplo restricciones	11
<b>Tabla 2:</b> Opciones intlinprog	20
<b>Tabla 3:</b> Valores exitflag	21
<b>Tabla 4:</b> Datos output intlinprog	22
<b>Tabla 5:</b> Programación de vuelos Ultimate Air	27
<b>Tabla 6:</b> Aeropuertos Ultimate Air	28
<b>Tabla 7:</b> Vuelos operados por cada aeronave	29
<b>Tabla 8:</b> B737. Solución routing1	31
<b>Tabla 9:</b> B737. Solución routing2	33
<b>Tabla 10:</b> B737. Solución routing3	35
<b>Tabla 11:</b> B737. Solución routing4	36
<b>Tabla 12:</b> B757. Solución routing1	38
<b>Tabla 13:</b> B757. Solución routing2	39
<b>Tabla 14:</b> B757. Solución routing3	40
<b>Tabla 15:</b> B757. Solución routing4	42
<b>Tabla 16:</b> Horarios vuelos B757	43
<b>Tabla 17:</b> Aeropuertos Kulula Airlines	60
<b>Tabla 18:</b> Kulula Airlines. Solución routing1	62
<b>Tabla 19:</b> Valores función objetivo de las rotaciones elegidas en routing1	63
<b>Tabla 20:</b> Kulula Airlines. Solución routing2	64
<b>Tabla 21:</b> Valores función objetivo de las rotaciones elegidas en routing2	65
<b>Tabla 22:</b> Kulula Airlines. Solución routing3	66
<b>Tabla 23:</b> Valores función objetivo de las rotaciones elegidas en routing3	67
<b>Tabla 24:</b> Kulula Airlines. Solución routing4	68
<b>Tabla 25:</b> Valores función objetivo de las rotaciones elegidas en routing4	70
<b>Tabla 26:</b> Datos vuelos Kulula	91





# ÍNDICE DE FIGURAS

---

<b>Figura 1:</b> Evolución tráfico aéreo pasajeros, [1]	1
<b>Figura 2:</b> Evolución porcentaje de beneficios de las aerolíneas, [2]	1
<b>Figura 3:</b> Mantenimiento en hangar, [4]	6
<b>Figura 4:</b> Rotación válida. Ciclos cerrados.	7
<b>Figura 5:</b> Rotación válida. Ciclos abiertos.	8
<b>Figura 6:</b> Rotación no válida.	8
<b>Figura 7:</b> Tabla de decisión de optimización	19
<b>Figura 8:</b> Rotaciones posibles cambios horarios intervalo [-45min, +45min]	45
<b>Figura 9:</b> Segmentación número de rotaciones posibles intervalo [-45min, +45 min]	46
<b>Figura 10:</b> Rotaciones posibles cambios horarios intervalo [-40min, +40min]	46
<b>Figura 11:</b> Segmentación número de rotaciones posibles intervalo [-40min, +40 min]	47
<b>Figura 12:</b> Rotaciones posibles cambios horarios intervalo [-35min, +35min]	47
<b>Figura 13:</b> Segmentación número de rotaciones posibles intervalo [-35min, +35 min]	48
<b>Figura 14:</b> Rotaciones posibles cambios horarios intervalo [-30min, +30min]	48
<b>Figura 15:</b> Segmentación número de rotaciones posibles intervalo [-30min, +30min]	49
<b>Figura 16:</b> Rotaciones posibles cambios horarios intervalo [-25min, +25min]	49
<b>Figura 17:</b> Segmentación número de rotaciones posibles intervalo [-25min, +25min]	50
<b>Figura 18:</b> Rotaciones posibles cambios horarios intervalo [-20min, +20min]	50
<b>Figura 19:</b> Segmentación número de rotaciones posibles intervalo [-20min, +20min]	51
<b>Figura 20:</b> Rotaciones posibles cambios horarios intervalo [-15min, +15min]	51
<b>Figura 21:</b> Segmentación número de rotaciones posibles intervalo [-15min, +15min]	52
<b>Figura 22:</b> Rotaciones posibles cambios horarios intervalo [-10min, +10min]	52
<b>Figura 23:</b> Segmentación número de rotaciones posibles intervalo [-10min, +10min]	53
<b>Figura 24:</b> Rotaciones posibles cambios horarios intervalo [-5min, +5min]	53
<b>Figura 25:</b> Segmentación número de rotaciones posibles intervalo [-5min, +5min]	54
<b>Figura 26:</b> Máximo número de rotaciones posibles tras cambios horarios	54
<b>Figura 27:</b> Porcentajes de rotaciones posibles mayores o iguales que las iniciales	55
<b>Figura 28:</b> Segmentación número de rotaciones posibles general	55
<b>Figura 29:</b> Rutas operadas por Kulula	58
<b>Figura 30:</b> Rutas operadas por Kulula a través de British Airways	58
<b>Figura 31:</b> Web oficial Kulula Airlines	59
<b>Figura 32:</b> Rutas operadas por Kulula actualmente	60

<b>Figura 33:</b> Rutas realizadas en la primera rotación de routing1	63
<b>Figura 34:</b> Rutas realizadas en la primera rotación de routing2	66
<b>Figura 35:</b> Rutas realizadas en la primera rotación de routing3	68
<b>Figura 36:</b> Rutas realizadas en la primera rotación de routing4	70



# Notación

---

$\lambda$	Número mínimo de días con estancia nocturna en base de mantenimiento
$\beta$	Tiempo de escala
P	Algoritmos polinómicos
NP	Algoritmos no polinómicos
$X_j$	Variable de decisión
$m_j$	Número de estancias nocturnas de la rotación $j$ en el aeropuerto base de mantenimiento
$v_j$	Número total de vuelos de la rotación $j$
H	Horas de referencia
$h_j$	Horas totales de vuelo de la rotación $j$
$c_j$	Coste de operación unitario de la rotación
f	Vector de coeficiente
A	Matriz de restricción de desigualdad lineal
b	Vector de restricción de desigualdad lineal
$A_{eq}$	Matriz de restricción de igualdad lineal
$b_{eq}$	Vector de restricción de igualdad lineal
intcon	Vector de restricciones de enteros
lb	Límites inferiores
ub	Límites superiores
$x_0$	Punto inicial
x	Vector solución
fval	Valor función objetivo
$A_m$	Aeropuerto base de mantenimiento
tmin	Tiempo mínimo de escala
$N_a$	Número de aviones disponibles
s	Matriz de horario de salida
l	Matriz de horario de llegada
rot	Matriz de posibles rotaciones de 1 día
R	Matriz de posibles rotaciones
m	Vector de estancias nocturnas de cada rotación en $A_m$
v	Vector de número total de vuelos de cada rotación
h	Vector de horas totales de vuelo de cada rotación
c	Vector de coste unitario de cada rotación
nr	Vector posición valores no nulos de x
time	Tiempo de ejecución del programa
nv	Número máximo de vuelos en un día
rotaciones	Matriz de rotaciones finales



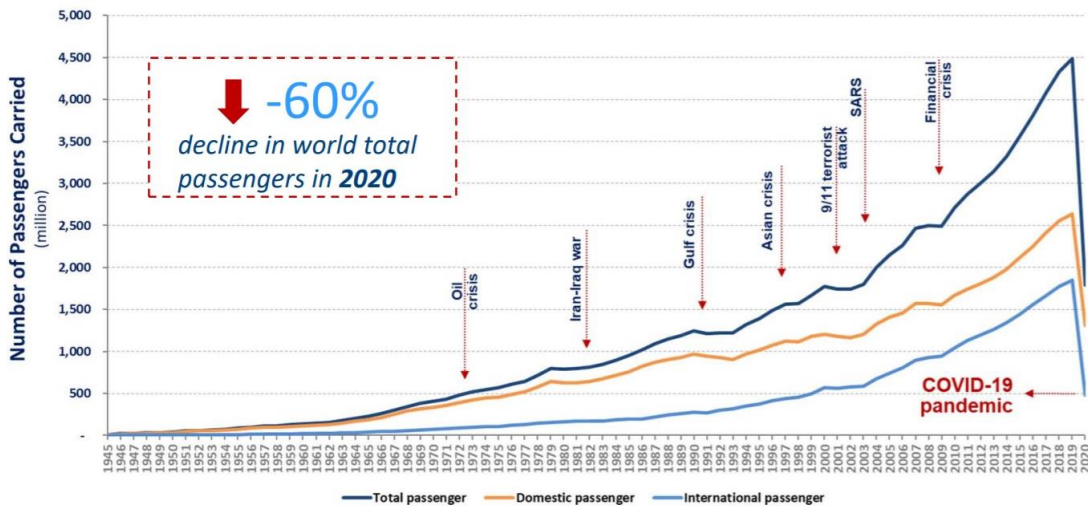


# 1 INTRODUCCIÓN

El transporte aéreo surge a principio de los años 20 a través de las empresas de correos, las cuales realizaban vuelos de transporte de correos y carga ligera utilizando aviones adaptados de la Primera Guerra Mundial. Tras la Segunda Guerra Mundial, aprovechando la consolidación de este tipo de transporte y la disponibilidad de aeronaves de un mayor tamaño, aparecen las primeras aerolíneas o compañías aéreas, las cuales comienzan a ofrecer vuelos regulares entre diferentes ciudades, enfocándose en el transporte de pasajeros.

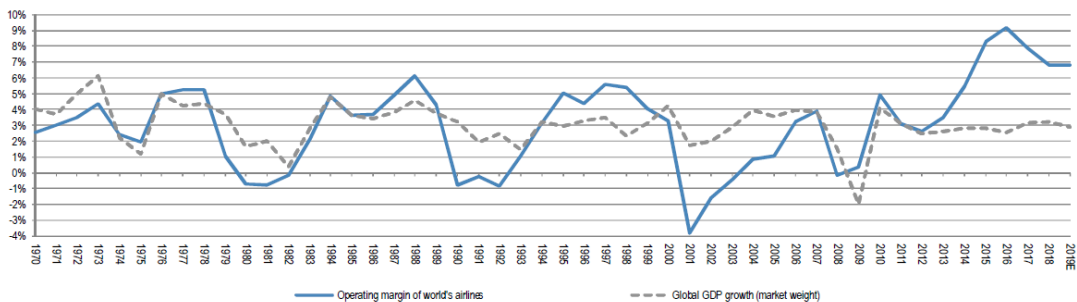
Desde los años 70, la evolución del tráfico aéreo de pasajeros presenta un claro crecimiento, a ritmo anual del 5%. A pesar de existir épocas donde este crecimiento se ha visto frenado o incluso reducido debido a diversos factores externos (guerras, pandemias, crisis económicas, etc.) el sector se caracteriza por disponer de capacidad de recuperación, hecho que se refleja en la **Figura 1**. Este crecimiento en el número de pasajeros se debe al aumento del tráfico aéreo propiciado por un aumento en el número de aerolíneas y una reducción de las tarifas.

**World passenger traffic evolution  
1945 – 2020**



**Figura 1:** Evolución tráfico aéreo pasajeros, [1]

A pesar de este crecimiento, las compañías aéreas se caracterizan principalmente por ser negocios con altos costes fijos y muy baja rentabilidad, siendo el promedio del 2% de beneficios. Además, la evolución de este margen presenta variaciones cíclicas, cambiando entre valores positivos y negativos cada 5 años aproximadamente.



**Figura 2:** Evolución porcentaje de beneficios de las aerolíneas, [2]

Por todos estos motivos, se identifica en este sector una alta competencia entre aerolíneas, determinada principalmente por las tarifas de los billetes ofertados con el objetivo de garantizar una cuota de mercado adecuada para maximizar los beneficios obtenidos.

Este hecho ha provocado la necesidad de utilizar diferentes técnicas en los diversos procesos que forman parte de la operación diaria de las compañías aéreas, buscando minimizar los costes y optimizar los recursos disponibles. Estos métodos han sido utilizados desde el inicio de las aerolíneas y han evolucionado con el tiempo acorde a la tecnología disponible, afectando a todas las áreas y departamentos que trabajan en este tipo de empresas.

La planificación dentro de una aerolínea puede dividirse en dos fases: estratégica y táctica. La fase estratégica se corresponde con la planificación a largo plazo, la cual puede extenderse entre varios meses hasta varios años, mientras que la fase táctica abarca el periodo comprendido entre días antes de la salida del vuelo hasta 3-4 meses antes.

En cada una de ellas se realizan diferentes tareas o procesos con el objetivo de obtener la planificación final de vuelos de la compañía y operar eficazmente las diversas rutas programadas. El objetivo principal de la fase estratégica consiste en analizar y definir el tipo de operativa de la aerolínea, las rutas a volar y el tipo de aeronaves que deberán utilizarse. Por otro parte, en la fase táctica se realizan las tareas de asignación y rotación de aviones, programación de tripulaciones y demás procesos que permitan realizar un control y seguimiento hasta la realización de los vuelos establecidos.

El objetivo de este trabajo es desarrollar un programa para resolver el problema de 'Aircraft Routing' o rotación de aviones. Este proceso se encuentra dentro de la fase táctica de la compañía y consiste en asignar una aeronave concreta a cada uno de los vuelos programados una vez que se conoce la asignación de aviones, es decir, el tipo de aeronaves que realiza cada trayecto, y la programación de vuelos existentes.

En primer lugar, se explica detenidamente el problema. En este capítulo se detallan los requerimientos que forman parte del problema y se expone el modelo matemático que lo define, presentando las diferentes funciones objetivos o metas de optimización y restricciones que se incorporan al programa informático.

A continuación, se expone de forma general las características que presentan este tipo de problema, y se detallan los modelos y técnicas utilizados para su resolución. De forma particular, se analizan los algoritmos que pueden utilizarse en el software utilizado para la implementación del programa a desarrollar.

Seguidamente, se presenta el programa elaborado para la resolución del problema. En concreto, se explica detalladamente el funcionamiento y objetivo de cada uno de los archivos y funciones creados, y se presenta un ejemplo ficticio para poner en práctica el programa.

Por último, se analiza el problema de rotación de aviones de una compañía existente, con el objetivo de comprobar el éxito del programa aplicado a un caso real.







## 2 AIRCRAFT ROUTING

---

El problema de rotación de aviones es el siguiente paso por resolver una vez que se conocen los resultados del problema de asignación de flota. Dada la programación de vuelos y los diferentes tipos de aviones que componen la flota de una compañía aérea, la asignación de flota tenía como objetivo asignar cada tipo de avión a un vuelo determinado minimizando los costes de explotación.

Una vez conocidos los horarios y frecuencias de los diferentes vuelos y el tipo de aeronave que realiza cada trayecto, el problema que surge a continuación es asignar un avión en concreto a cada uno de los vuelos existentes. En otras palabras, se conoce el tipo de avión que realizará cada trayecto y se quiere averiguar el avión específico que se le asignará entre todas las aeronaves disponibles de la flota. Este problema de asignación es lo que se conoce como rotación de aviones ('Aircraft Routing' en inglés).

El objetivo principal de la rotación de aviones es maximizar los ingresos o minimizar los costes de operación, debiéndose realizar para ello una asignación óptima y eficiente de cada uno de los aviones que forman la flota a cada uno de los vuelos de la compañía. Para ello se deben considerar ciertos requerimientos y restricciones que permitan abordar el problema desde una perspectiva lógica y real.

### 2.1 Requerimientos

Existen dos limitaciones importantes que se deben tener en cuenta antes de afrontar el problema: el mantenimiento de las aeronaves y el tiempo necesario entre la llegada y posterior salida de una misma aeronave.

Las actividades de mantenimiento son indispensables para mantener un nivel adecuado de seguridad operacional. Para ello, se elaboran programas de mantenimiento según las normas regidas por la autoridad competente, como pueden ser la EASA en Europa o la FAA en Estados Unidos, y el fabricante de las aeronaves.

Existen diferentes tipos de mantenimientos los cuales se diferencian por la frecuencia con que se realizan y el alcance de las inspecciones llevadas a cabo. A modo de ejemplo, es posible catalogar como mantenimiento desde la inspección visual realizada antes de un vuelo hasta la parada completa de la actividad del avión durante varias semanas para cometer reconocimientos más profundos.

Según la FAA, el mantenimiento de toda aeronave se puede dividir en 4 categorías principales, cada una de ellas identificadas por una letra. El mantenimiento de tipo A se realiza aproximadamente cada 60 horas de vuelo y tiene como objetivo la inspección visual de los sistemas principales de la aeronave. El de tipo B se realiza cada 300-600 horas de vuelo y, además de la inspección visual de estos sistemas principales, se lubrican las partes móviles de éstos. Finalmente, el mantenimiento de tipo C y D se realizan con una frecuencia que puede variar entre 1 y 4 años, donde se acometen trabajos más complejos y detallados que requieren la interrupción de la actividad del avión durante varios días o incluso semanas [3].



**Figura 3:** Mantenimiento en hangar, [4]

En Europa, el mantenimiento de tipo A se denomina mantenimiento en rampa. Este es un mantenimiento muy ligero realizado en pistas o aparcamientos de aeropuertos, sin necesidad de desviar a las aeronaves hacia hangares específicos. Las inspecciones consisten en revisiones visuales de los distintos equipos exteriores e interiores, así como chequeos simples de sistemas de vuelos y comunicaciones.

En la práctica, las aerolíneas realizan mantenimientos de tipo A o rampa cada 3 o 4 días. Esta forma de proceder permite asegurar el cumplimiento de la frecuencia de este tipo de inspecciones con bastante holgura y facilita el diseño de las planificaciones de horarios y rotaciones, así como el control de la operación diaria de la compañía. El tiempo necesario para elaborar estos reconocimientos varía entre 3 y 10 horas, por lo que se realiza normalmente en horario nocturno cuando las aeronaves no están operativas.

Debido a las restricciones que este tipo de mantenimiento impone en la operación diaria de las aerolíneas, se incorpora en el problema de rotación de aviones la necesidad de realizar este tipo de inspecciones. El resto de los tipos de mantenimientos nombrados no se considera en este problema debido a la menor frecuencia de realización.

Por otro lado, las aeronaves necesitan cierto tiempo entre la llegada a un aeropuerto concreto y su posterior salida, denominado tiempo de escala. Este tiempo varía normalmente entre 20 y 60 minutos, e incluye el tiempo de rodadura desde la pista de aterrizaje hasta el estacionamiento de la aeronave, la descarga del equipaje y los pasajeros, la limpieza del avión, las inspecciones realizadas, el repostaje y vaciado de aguas fecales, embarque de pasajeros, carga de equipaje y rodadura hasta la pista de aterrizaje para la salida del siguiente vuelo.

Los tiempos de escala suponen un importante factor dentro de la operación de una compañía aérea. Cuanto mayor es el tiempo durante el cual la aeronave se encuentra en tierra, mayores son las tasas que deben depositar y menor es el tiempo de vuelo de las aeronaves, por lo que se reducen los ingresos asociados al traslado de pasajeros y carga. De forma general, cuanto mayor es la distancia de la ruta a operar, mayores son los tiempos de escala de una aeronave, debido a la necesidad de realizar un mayor número de tareas para la preparación del vuelo.

## 2.2 Modelo matemático

El objetivo principal de la rotación de aviones es asignar un avión específico a una serie de vuelos de forma óptima, es decir, minimizando los costes o maximizando los ingresos, sujeto a una serie de requerimientos y restricciones.

Las necesidades de mantenimiento se incorporan al problema requiriendo que cada uno de los aviones de la flota realicen una estancia nocturna en algún aeropuerto en concreto como mínimo cada  $\lambda$  días. Normalmente el mantenimiento solo se realiza en algunos aeropuertos conocidos como bases de mantenimiento, donde las aerolíneas cuentan con las infraestructuras y servicios necesarios para la realización de estas inspecciones. De

este modo, antes de abordar el problema de rotación de aviones es necesario conocer en qué aeropuertos es posible llevar a cabo estas revisiones, las cuales se efectuarán en horario nocturno fuera de la actividad operativa de la aeronave. Además, se fija el número máximo de días consecutivos  $\lambda$  entre los que es necesario realizar el mantenimiento, que varía normalmente entre 3 y 4 días, tal y como se comentó anteriormente.

Se fija también  $\beta$ , el tiempo mínimo necesario entre la llegada y salida de una aeronave o tiempo de escala. Conocido el horario de los diferentes vuelos y una vez fijado este tiempo, es posible conocer si un mismo avión será capaz de realizar dos vuelos consecutivos o si, por el contrario, se deberá asignar otro avión de la flota debido a que el tiempo de escala sea menor que  $\beta$ .

### 2.2.1 Generación de rotaciones

Antes de definir matemáticamente el problema, se aclaran algunos términos y se exponen ciertas consideraciones o suposiciones que se van a utilizar más adelante en el estudio de este problema.

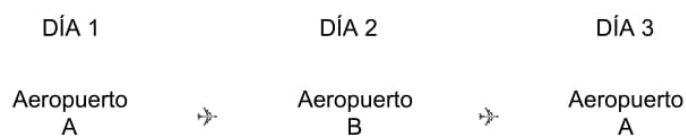
El primer paso para resolver el problema es generar todas las rotaciones válidas posibles. Una rotación es una secuencia de vuelos que realiza un mismo avión de modo que los aeropuertos iniciales y finales sean el mismo, es decir, que el último vuelo finaliza en el mismo aeropuerto desde el que se produjo la salida del primer vuelo de la rotación. Para que una rotación sea válida, debe cumplir los requerimientos de mantenimiento y tiempo entre la llegada y salida de un vuelo.

Las rotaciones se componen de una serie de días en concreto y de vuelos cada día. El caso más sencillo, utilizado como ejemplo por muchos autores para explicar el problema, consiste en considerar que la programación de vuelos se repite diariamente, es decir, se asume que los mismos vuelos se realizan todos los días. A partir de los horarios y frecuencias se generan en primer lugar todas las rotaciones posibles de un día, que permiten obtener las rotaciones del número de días deseado enlazando unas con otras y teniendo en cuentas las necesidades de mantenimiento y tiempo entre llegada y salida de un vuelo.

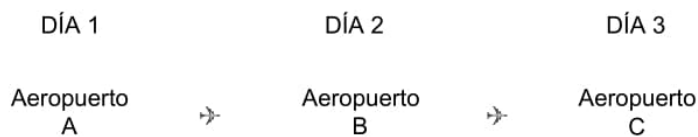
En el programa que se diseña más adelante para la resolución del problema, se va a asumir que la programación de vuelos se repite cada tres días. De este modo se permite incluir aquellos vuelos que no se repiten diariamente, siendo un caso que se aproxima más a la realidad del problema. Por tanto, para este programa se generarán rotaciones de 3 días, las cuales se repetirán en el tiempo cada semana.

Para reducir el número de las rotaciones posibles que pueden llegar a ser solución, se opta por introducir ciclos cerrados en las rotaciones que se generarán. Estas rotaciones se caracterizan por finalizar el último vuelo en el mismo aeropuerto donde se inició el primer vuelo. Este requerimiento de comenzar y finalizar en el mismo aeropuerto no es habitual en la operación de las aerolíneas, ya que limita las posibilidades de elegir un gran número de rotaciones. Las aerolíneas normalmente desarrollan planes mensuales de rotaciones de aviones, en los que una aeronave puede incluso tener rotaciones distintas cada día siempre que se cumplan las restricciones del problema, gracias a potentes programas específicos diseñados expresamente para este tipo de problemas. En este caso, dado que se trata de un problema académico y la falta de recursos para poder optar a software avanzado para este tipo de problemas, se introduce esta suposición de ciclos cerrados para poder obtener soluciones factibles [5].

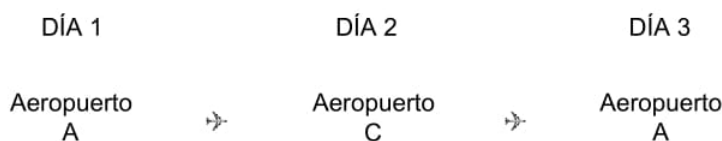
En las **Figuras 4, 5 y 6** se puede observar un ejemplo sencillo de dos tipos de rotaciones explicados. Se incluye también un ejemplo acerca de una rotación no válida. En este caso, es necesario realizar como mínimo una estancia nocturna en el aeropuerto de mantenimiento B.



**Figura 4:** Rotación válida. Ciclos cerrados



**Figura 5:** Rotación válida. Ciclos abiertos



**Figura 6:** Rotación no válida

## 2.2.2 Función objetivo

El segundo y definitivo paso para resolver el problema es seleccionar las rotaciones mejores o más eficientes, una vez obtenidas todas las rotaciones válidas posibles. Este proceso se efectúa de manera independiente para cada tipo de avión de la flota, teniendo en cuenta en cada uno de los casos solo los vuelos que realiza el tipo de avión que se está estudiando.

Para poder seleccionar las rotaciones mejores es necesario introducir una variable de decisión binaria  $X_j$  que toma los siguientes valores:

$$X_j = \begin{cases} 1 & \text{si la rotación } j \text{ es seleccionada} \\ 0 & \text{en caso contrario} \end{cases} \quad (2-1)$$

En total existirán tantas variables como rotaciones posibles hayan sido generadas.

Es posible diferenciar varias funciones objetivo según las variables que se deseen optimizar. A continuación, se explican cuatro funciones distintas que se incorporarán en el programa a elaborar.

### 2.2.2.1 Maximizar las posibilidades de mantenimiento

Esta función tiene como objetivo seleccionar aquellas rotaciones que permiten optar a mayores oportunidades de mantenimiento. Para ello, se introduce la variable  $m_j$ , que representa el número de estancias nocturnas que la rotación  $j$  realiza en uno de los aeropuertos designados como base de mantenimiento. Esta variable, para el programa que se diseñará, donde las rotaciones son de 3 días, puede tomar los valores 1,2 y 3, existiendo tantas variables como rotaciones posibles generadas.

La función objetivo es la siguiente:

$$\text{Maximizar } \sum_{j=1}^n m_j X_j \quad (2-2)$$

siendo  $n$  el número total de rotaciones posibles.

Es necesario aclarar los valores que puede tomar la variable  $m_j$ . El número de estancias nocturnas que realizará una determinada rotación en un aeropuerto base de mantenimiento dependerá de la frecuencia elegida para llevar a cabo estas inspecciones. Como se comentó al inicio de este capítulo, en la práctica las aerolíneas realizan mantenimiento cada 3 o 4 días. En el programa que se elabora en este proyecto este número de días será de uno como mínimo, de modo que todas las rotaciones que no cumplan esta condición no podrán formar parte de la solución. El aeropuerto de mantenimiento a elegir se introduce como una entrada del problema informático, de modo que el usuario pueda elegir aquél que le resulte más adecuado.

Se destaca también que el hecho de realizar una estancia nocturna en un aeropuerto base de mantenimiento no implica que sea obligatorio ejecutar el mantenimiento en ese preciso momento. Por ejemplo, si un avión de una rotación concreta pernocta 2 días en un aeropuerto base y es necesario realizar el mantenimiento como mínimo una vez cada 3 días, no es obligatorio que realice este reconocimiento los 2 días.

### 2.2.2.2 Maximizar el número de conexiones de vuelos

En este caso el objetivo principal es seleccionar aquellas rotaciones que cuenten con un mayor número de conexiones de vuelos, es decir, aquellas que permitan realizar más vuelos en el total de días que perduren las rotaciones.

Se introduce la variable  $v_j$ , que representa el número de vuelos totales que realiza la rotación  $j$ . Esta variable puede tomar los valores  $1, 2, \dots, z$ , siendo  $z$  el número total de vuelos existentes. En concreto, existen tantas variables  $v_j$  como rotaciones posibles generadas.

La función objetivo es la siguiente:

$$\text{Maximizar } \sum_{j=1}^n v_j X_j \quad (2-3)$$

De este modo se elegirán aquellas rotaciones con mayor número de vuelos, las cuales aportarán mayor beneficio económico a falta de datos económicos como costes e ingresos de cada uno de los vuelos. La rotación ideal sería, según este modelo, aquella que pudiera realizar todos los vuelos programados de la aerolínea, hecho muy complicado en cualquier aerolínea debido a la gran cantidad de destinos y frecuencias existentes.

### 2.2.2.3 Minimizar desviación respecto a horas de referencia

En esta función se introduce un número de horas de vuelo de referencia  $H$ . El objetivo es, conocido este número de horas de vuelo de referencia, que en el programa que se elabora más adelante será una variable de entrada, se elijan aquellas rotaciones cuyo número total de horas de vuelo se aproximen más a las horas de vuelo de referencia.

Se introduce la variable  $h_j$ , que representa el número horas de vuelo totales que se realiza en la rotación  $j$ . En concreto, existen tantas variables  $h_j$  como rotaciones posibles generadas.

La función objetivo consiste por tanto en minimizar la desviación respecto a este número de horas de vuelo de referencia  $H$ :

$$\text{Minimizar } \sum_{j=1}^n |h_j - H| X_j \quad (2-4)$$

### 2.2.2.4 Minimizar costes

Por último, otra de las posibles selecciones de las rotaciones más eficientes es teniendo en cuenta los costes. Un indicador muy utilizado en el transporte aéreo para comparar los costes de operación de las aerolíneas es el CASM ('cost per available seat mile').

El coste de operación de un vuelo se puede obtener mediante la siguiente expresión, y depende del tipo de aeronave que efectúe el trayecto.

$$\text{Coste de operación de un vuelo} = \text{CASM} \times \text{distancia(millas)} \times \text{número de plazas}$$

De este modo, conociendo la distancia total que se realiza en una rotación, el número de plazas de la aeronave utilizadas y el CASM se puede obtener el coste de operación total de los vuelos realizados en una rotación. Para evitar seleccionar aquellas rotaciones que tienen menores vuelos y por lo tanto tienen menores coste, se divide el coste total de operación de la rotación entre el número de vuelos realizados, con objeto de poder realizar una comparación unitaria entre las diferentes rotaciones [5].

El CASM se obtiene de dividir los costes de operación por ASM ('available seat miles'), y refleja el coste de volar una sola plaza por milla. Generalmente, cuanto más bajo es el CASM de una aerolínea, más eficiente y rentable es su actividad.

Para cuantificar el coste de cada rotación se introduce la variable  $c_j$ , que representa el coste de operación unitario de cada rotación descrito anteriormente. Esta variable será un dato que se calcula a partir del coste de operación de cada uno de los vuelos, dato que deberá introducir el usuario en el programa elaborado más adelante. Para no tener que recopilar datos reales de aerolíneas, en los ejemplos analizados posteriormente se opta por introducir estos costes operaciones de cada vuelo como variables aleatorias que pueden tomar cualquier valor en un determinado intervalo. De este modo, ya que el objetivo del trabajo es didáctico y no se centra en obtener resultados reales si no elaborar un programa para resolver el problema del 'Aircraft Routing' y comprobar el funcionamiento correcto del mismo, esta simplificación permite agilizar el proceso.

En concreto, existen tantas variables  $c_j$  como rotaciones posibles generadas. La función objetivo es la siguiente:

$$\text{Minimizar } \sum_{j=1}^n c_j X_j \quad (2-5)$$

En un caso real propio de las compañías aéreas, sería necesario incluir otra serie de costes como puede ser el combustible, coste por pérdidas de pasajeros o costes asociados a penalizaciones de aquellas rotaciones con menor número de conexiones. Sin embargo, dado que el programa que se elaborará tiene uso didáctico principalmente, no se entra en más detalle y se realiza esta selección de rotaciones teniendo en cuenta únicamente el coste  $c_j$ .

### 2.2.3 Restricciones

Hasta ahora se ha visto el proceso necesario para elaborar las rotaciones posibles y cuatro posibles casos o formas de elegir las más eficiente según el resultado que se pretende obtener. Sin embargo, el problema no está completamente definido, siendo necesario el uso de ciertas restricciones que permitan dotar de cierto realismo al planteamiento y a las posteriores soluciones que se obtengan.

En este apartado se exponen las dos principales limitaciones del problema de rotación de aviones: cobertura de vuelo y número de aviones disponibles.

#### 2.2.3.1 Cobertura de vuelo

Cada rotación cubre una serie de vuelos. Es necesario, una vez elegida las rotaciones que formen parte de la solución final, que todos los vuelos se realicen en algunas de estas rotaciones. Además, cada uno de los vuelos que forman parte de la programación de la aerolínea debe realizarse solo una vez al día. En otras palabras, cada vuelo debe realizarse todos los días en los que esté programado y solo una vez al día. Esto no quiere decir que todos los vuelos deban realizarse todos los días, ya que muchos de ellos se encuentran programados para solo algunos días de la semana.

Este tipo de restricciones tiene dos objetivos principales: por un lado, evitar repeticiones de vuelos, es decir, que un mismo vuelo se realice en más de una rotación, hecho que conllevaría a realizar un mismo vuelo con distintos aviones, lo cual no tiene ningún sentido; por otro lado, evitar la no realización de algunos vuelos.

En total existirán tantas restricciones de este tipo como vuelos y número de días que se realice cada vuelo haya.



Son por tanto restricciones que se imponen por vuelo y por día.

Para aclarar la forma de proceder al introducir este tipo de restricciones, se expone un breve ejemplo. Supongamos que la **Tabla 1** contiene 5 posibles secuencias de vuelos de una determinada aerolínea. La duración de las secuencias es de 3 días, y en total existen 4 vuelos.

**Tabla 1:** Ejemplo restricciones

Secuencias	Vuelos día 1	Vuelos día 2	Vuelos día 3
1	1 3	3 2	1 3
2	3	2 4	1 3
3	1 4	3	1
4	3 4	3 4	3
5	1	3 2	3

Como se ha comentado, las restricciones de balance de vuelo imponen que cada vuelo debe realizarse todos los días en los que esté programado y solo una vez al día.

En este caso, existen 4 vuelos: el vuelo número 1 se realiza solo los días 1 y 3; el vuelo número 2 se realiza solo el día 2; el vuelo número 3 se realiza todos los días; y el vuelo número 4 se realiza solo los días 1 y 2.

En total, se tienen tantas restricciones como vuelos y número de días que se realice cada vuelo haya. Por tanto, en este caso se tienen  $2 + 1 + 3 + 2 = 8$  restricciones. A continuación, se indican cada una de ellas, diferenciadas por vuelo y día.

Vuelo 1 día 1:

$$X_1 + X_3 + X_5 = 1 \quad (2-6)$$

Vuelo 1 día 3:

$$X_1 + X_2 + X_3 = 1 \quad (2-7)$$

Vuelo 2 día 2:

$$X_1 + X_2 + X_5 = 1 \quad (2-8)$$

Vuelo 3 día 1:

$$X_2 + X_4 = 1 \quad (2-9)$$

Vuelo 3 día 2:

$$X_1 + X_3 + X_4 + X_5 = 1 \quad (2-10)$$

Vuelo 3 día 3:

$$X_1 + X_2 + X_4 + X_5 = 1 \quad (2-11)$$

Vuelo 4 día 1:

$$X_3 + X_4 = 1 \quad (2-12)$$

Vuelo 4 día 2:

$$X_2 + X_4 = 1 \quad (2-13)$$

### 2.2.3.2 Número de aviones disponibles

A partir de este tipo de restricciones es posible tener en cuenta los resultados obtenidos del problema de asignación de aviones. Cada rotación candidata a ser solución es una secuencia de vuelos a los que se le puede asignar una aeronave específica, por lo que el número de rotaciones elegidas finalmente no debe superar al número de aeronaves disponibles de la flota.

Conociendo el número de aeronaves necesarias de un determinado tipo para efectuar una serie de vuelos, es posible introducir esta restricción. En el programa elaborado más adelante el número de aeronaves disponibles será una opción que pueda introducir el usuario.

Esta restricción se modela matemáticamente según la siguiente expresión:

$$\sum_{j=1}^n X_j \leq N^{\circ} \text{ aeronaves disponibles} \quad (2-14)$$

## 2.3 Solución

Finalmente, tras haber generado todas las rotaciones válidas posibles teniendo en cuenta las necesidades de mantenimiento y tiempo de escala, haber seleccionado la función objetivo adecuada según el propósito elegido y haber introducido las restricciones correspondientes se obtiene la solución final del problema. Esta solución está formada por las rotaciones elegidas teniendo en cuenta todos los criterios y restricciones mencionados, las cuales se corresponden con la asignación de una aeronave específica para cumplir con la programación inicial de vuelos establecida.

El número de rotaciones finales elegidas, que obviamente, se corresponde con el número de aeronaves necesarias, puede variar del número de aeronaves obtenidas a partir del problema de asignación de flota. Este suceso, aunque pueda parecer anormal, es muy posible que ocurra y se debe a las restricciones impuestas en el problema de rotación de aviones, las necesidades del mantenimiento requerido y el tiempo mínimo que debe existir entre la llegada y salida de una aeronave o tiempo de escala.

Este hecho puede provocar que el número de aeronaves necesarias tras realizar el problema de rotación de aviones sea superior al esperado a partir de problemas anteriores como la asignación de flota, de modo que al introducir la restricción del número de aviones disponibles no se pueda alcanzar la solución óptima del problema. Una posible solución rápida y eficaz es eliminar esta restricción para poder obtener el número de aeronaves que realmente se necesitan.

Por otra parte, existe la posibilidad de realizar mejoras en el problema. Es posible que ciertas rotaciones realicen muy pocos vuelos o conexiones ciertos días en comparación con el resto, o que el número de rotaciones y por tanto de aeronaves obtenidas sea muy elevado. Para ello, es normal realizar una revisión de horarios de aquellos vuelos problemáticos.

La revisión de horarios permite ajustar las horas de salidas y llegadas de ciertos vuelos con el objetivo de permitir la conexión entre ciertos trayectos que antes no eran posible, debido al insuficiente tiempo entre la llegada y salida de ellos. De este modo, ajustando estos horarios de aquellos vuelos que a priori puedan tener menos

conexiones, es posible mejorar las rotaciones aumentan el número de trayectos y conexiones realizados, e incluso disminuir el número de aeronaves necesarias para efectuarlos.

Este es un proceso muy común dentro de las compañías aéreas, en las que la programación inicial creada por el equipo de planificación es revisada continuamente por el equipo de operaciones. La relación continua entre departamentos permite ejecutar pequeños cambios para encontrar soluciones óptimas, maximizando los beneficios de la aerolínea.

Por último, cabe destacar que la asignación de una aeronave específica a cada rotación obtenida no es un proceso aleatorio. En este caso, no se profundiza en estos procesos por quedar fuera del objeto del trabajo, aunque es posible seguir ciertas reglas o criterios que tengan en cuenta las características de las rutas de cada rotación y de las aeronaves de la flota, como puede ser la edad, permitiendo una asignación más eficaz.



# 3 PROGRAMACIÓN LINEAL

---

En el capítulo anterior se presentó la rotación de aviones como un problema matemático compuesto por diferentes funciones objetivos, cada una de ellas adecuada para optimizar un caso particular según las variables utilizadas, y diferentes restricciones y requerimientos iniciales que limitan y definen las soluciones obtenidas.

Una vez analizado el modelo matemático que caracteriza este tipo de problemas es necesario el uso de algún software específico que permita encontrar una solución óptima en un periodo de tiempo aceptable. Una aerolínea de tamaño medio, compuesta por un número aceptable de aeronaves y de vuelos diarios, puede llegar a encontrarse con millones de variables a la hora de enfrentarse a este tipo de problemas. Por ello, la resolución del problema de rotación de aviones no puede ser resuelta a mano en estos casos, siendo necesario elaborar algoritmos de resolución que permitan implementar ciertas técnicas y métodos que habiliten al usuario para encontrar soluciones factibles y aceptables en un periodo de tiempo determinado.

Los modelos utilizados para la resolución de la rotación de aviones, así como los procedimientos empleados para la creación de un algoritmo que permita encontrar soluciones factibles a la rotación de aviones para un número limitado de vuelos y aviones, se exponen en el transcurso de este capítulo.

## 3.1 Complejidad del problema

Una pequeña idea acerca de la complejidad que presenta el problema debido al gran número de variables que se deben tener en cuenta puede observarse en uno de los ejemplos del libro ‘Airlines Operations and Scheduling’ [5].

En éste se presenta una compañía ficticia con una serie de vuelos diarios. En concreto, para los aviones de la flota del tipo B737-800 existen 30 vuelos diarios. Inicialmente se acuerda la necesidad de hacer noche al menos cada 3 días en el aeropuerto base de mantenimiento (JFK) y el tiempo entre la llegada y salida de una aeronave es de 45 minutos. El objetivo es obtener la rotación de aviones de 3 días optimizando las posibilidades de mantenimiento, sujeto a las restricciones de cobertura de vuelo. A pesar de ser un problema relativamente pequeño se llegan a obtener cerca de 6000 variables de decisión, lo que enfatiza la necesidad de resolver este tipo de problema con ciertos métodos o técnicas que permitan obtener soluciones aceptables en rangos de tiempo adecuados [5].

Como se acaba de demostrar, existen multitud de variables y por tanto de soluciones posibles a obtener ante este tipo de problemas. De este modo, han aparecido a lo largo de los años diferentes métodos o técnicas para resolverlos de una forma factible, algunas de las cuales se exponen más adelante. El hecho de que exista la posibilidad de utilizar diferentes opciones para su resolución pone de manifiesto la necesidad de encontrar aquellas más eficientes.

Para poder obtener la eficiencia de varios algoritmos se recurre a determinar el coste computacional, el cual permite conocer de forma aproximada la forma en que aumenta la complejidad del problema a medida que aumenta su tamaño. Una posible medida para calcular este coste sería hallar el tiempo empleado por el programa en ejecutarse y obtener una solución, aunque éste depende del lenguaje de programación utilizado y del procesador del ordenador, por lo que un mismo algoritmo puede presentar diferencias importantes según los dos aspectos comentados, impidiendo comparar adecuadamente diferentes programas.

De este modo, se introduce como medida del coste computacional de un programa o algoritmo el número de pasos u operaciones que necesita realizar para encontrar la solución final. En concreto, se busca el número de operaciones máximo, correspondiéndose con el peor de los casos. Así, el orden obtenido permite obtener el límite superior de pasos u operaciones necesarias, siendo este el coste máximo del programa.

Un algoritmo de orden  $n^2$ , representado como  $O(n^2)$ , indica que si el tamaño del problema aumentase por 2, el número de pasos necesario para resolverlos sería ahora de  $2^2 = 4$  veces más [5].

Durante los años 70 muchos investigadores realizaron estudios acerca de los diferentes tipos de algoritmos y su coste computacional. Como resultado surgió la clasificación de ellos en dos grandes grupos: polinómicos (P) y polinómicos no deterministas (NP) [5].

Los algoritmos polinómicos (P) se caracterizan por tener un orden polinómico. Por ejemplo, el algoritmo anterior de orden  $n^2$  es un ejemplo de este tipo, en el que al aumentar el tamaño del problema, es decir, al aumentar el número de variables, aumenta el coste del problema de forma polinómica. Este tipo de algoritmos es el más efectivo, debido a la posibilidad de resolver grandes problemas en tiempos relativamente aceptables.

Por otro lado, los algoritmos polinómicos no deterministas (NP) se caracterizan por tener un orden exponencial. Por ejemplo, un algoritmo de orden  $O(2^n)$  es un buen ejemplo. A medida que aumenta el tamaño del problema, aumenta el coste pero de forma exponencial, tratándose por ello del tipo de algoritmo más complicado de resolver.

El caso de la rotación de aviones se resuelve con algoritmos NP, por lo que pone de manifiesto la necesidad de implementar ciertos procedimientos a la hora de resolver el problema, con el objetivo de simplificar el proceso y poder obtener soluciones que, si bien no serán las óptimas, se acercarán bastante a ellas, en un periodo de tiempo razonable [5].

### 3.2 Programación lineal entera

La rotación de aviones es un tipo de problemas que puede ser resuelto mediante el uso de modelos de programación lineal entera. Esta programación se caracteriza por ser un tipo de programación lineal donde todas las variables son enteras y no negativas.

A continuación, se presenta la formulación matemática de este tipo de modelos con objeto de conocer profundamente las bases sobre las que se asienta el problema.

Un modelo de programación lineal entera (**IP**) es un problema de la forma

$$\begin{aligned} \min \quad & cx \\ \text{sujeto a} \quad & Ax \leq b \\ & x \geq 0 \text{ entera} \end{aligned} \tag{3-1}$$

donde los datos que se introducen son:

- El vector  $c = (c_1, c_2, c_3, \dots, c_n)$
- La matriz  $A = (a_{ij})$  de dimensiones  $m \times n$

- El vector columna  $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix}$

El vector columna  $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$  contiene las variables que serán optimizadas, que para el problema de rotación de

aviones, serán las rotaciones posibles generadas cumpliendo los requerimientos de mantenimiento y tiempos de escala. El vector  $x$ , de dimensiones  $n \times 1$ , se dice que es entero cuando todas sus variables son número enteros, es decir, cuando  $x \in \mathbb{Z}^n$  [6].

Se introduce también el concepto de programación lineal de enteros mixtos (**MILP**), el cual es un caso general de la programación lineal entera. Este es un tipo de problema de la forma

$$\begin{aligned}
 & \min cx + hy \\
 & \text{sujeto a } Ax + Gy \leq b \\
 & x \geq 0 \text{ entera} \\
 & y \geq 0
 \end{aligned}
 \tag{3-2}$$

donde los datos que se introducen son:

- El vector  $c = (c_1, c_2, c_3, \dots, c_n)$  y el vector  $h = (h_1, h_2, h_3, \dots, h_p)$
- La matriz  $A = (a_{ij})$  de dimensiones  $m \times n$  y la matriz  $G = (g_{ij})$  de dimensiones  $m \times p$

- El vector columna  $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix}$

Los vectores columna  $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$  e  $y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$  contienen las variables que serán optimizadas. Los valores de

las variables  $x_j$  se restringen a número enteros mayores de cero, mientras que las variables  $y_j$  pueden tomar cualquier valor real mayor que cero. Se asume además que siempre existe al menos una variable entera, es decir,  $n \geq 1$  [6].

Cabe destacar que modelo de programación lineal entera es un caso específico del modelo de programación lineal de enteros mixtos en el que  $p = 0$ .

La resolución de programaciones lineales de variables enteras, las cuales modelan el problema de rotación de aviones, es una tarea compleja de forma general. Tal y como se comentó en el **Capítulo 3.1**, los algoritmos utilizados para resolver este tipo de modelos crecen exponencialmente en el tiempo a medida que crece su tamaño, originando problemas importantes en cuanto a la eficiencia del código utilizado e impidiendo incluso en muchos casos la posibilidad de obtener una solución factible.

Dada la complejidad del problema y la necesidad de obtener soluciones óptimas en intervalos de tiempo aceptables, se utilizan diversos métodos para solucionar este inconveniente. Especialmente, se busca obtener soluciones cercanas al óptimo en un tiempo menor, que puedan considerarse lo suficientemente aceptables como para formar parte de la solución final. Este tipo de métodos se conocen como métodos heurísticos, y aunque no garantiza la obtención de la solución óptima, permite encontrar una solución cercana la cual se considera aceptable, en un tiempo mucho menor.

Este tipo de métodos se utilizarán en la programación de nuestro problema y se utilizará para tal fin el software *Matlab*, versión 2015. En concreto, se van a utilizar tres técnicas o métodos para encontrar una solución factible del problema: el método de la generación de corte, la heurística y el método de rama y encuadernado (conocido también como bifurcación y enlazado).

### 3.3 Programación lineal en Matlab

Los algoritmos utilizados para resolver problemas de programación lineal entera tienen como fin minimizar o maximizar, según los intereses del usuario y el propósito del problema en sí, una función objetivo sujeta a una serie de restricciones, que pueden ser de igualdad o desigualdad, e imponiendo que las variables utilizadas sean enteras.

Los algoritmos de programación lineal entera pueden implementarse en programas informáticos como *Matlab*. Para ello, se utilizará la función de *Matlab* **intlinprog**, que permite resolver problemas de programación lineal

entera mixta (**MILP**) y por tanto es aplicable a problemas de programación lineal entera (**IP**) como es el caso que se aborda en este trabajo. A continuación, se presenta de forma general el problema [7].

$$\min_x \{f^T x\} \quad (3-3)$$

Sujeto a:

$$\begin{aligned} Ax &\leq b \\ A_{eq}x &= b_{eq} \\ lb &\leq x \leq ub \\ x_i &\in \mathbb{Z} \end{aligned} \quad (3-4)$$

Como se acaba de adelantar, la función o solucionador de *Matlab* elegida para resolver el problema es **intlinprog**. Para llegar a esta conclusión se han tomado una serie de decisiones que se exponen a continuación.

En primer lugar, es necesario elegir un enfoque del problema al que nos enfrentamos. Existen dos formas de resolver problemas de optimización como el que nos encontramos: mediante un enfoque basado en el problema o un enfoque basado en el ‘Solver’. Las características principales de cada uno de ellos son las siguientes [8]:

1. Configuración de optimización basada en problemas
  - Mayor facilidad de creación y depuración.
  - No se aplica a la resolución de ecuaciones o mínimos cuadrados no lineales.
  - Representación del objetivo y las restricciones simbólicamente.
  - Tiempo de solución más largo debido a la necesidad de traducir el problema a forma matricial.
  - No permite la inclusión directa del gradiente o hessiana.
2. Configuración de optimización basada en ‘Solver’
  - Mayor dificultad de creación y depuración.
  - Representación del objetivo y las restricciones como funciones o matrices.
  - Tiempo de solución más corto debido a que no existe la necesidad de traducir el problema a forma matricial.
  - Permite la inclusión directa del gradiente o hessiana.

Teniendo en cuenta las características de cada uno de los dos enfoques y las necesidades del problema a resolver, se opta por utilizar la configuración basada en el ‘Solver’. Para este tipo de enfoque se siguen los siguientes pasos de forma general [9]:

1. Elección del solucionador de optimización.
2. Creación de la función objetivo.
3. Creación de las restricciones del problema.
4. Configuración de las opciones del solucionador, si no se desea utilizar las opciones predeterminadas.
5. Llamada al solucionador adecuado.

Para elegir el solucionador adecuado, se utiliza la siguiente tabla facilitada en la documentación del programa, que permite elegir el solucionador adecuado en función del tipo de restricciones y el tipo de objetivo [10].



Tipo de restricción	Tipo de objetivo				
	Lineal	Cuadrática	Mínimos cuadrados	Smooth no lineal	No liso
Ninguno	<code>n/a (= const o min =f -∞)</code>	<code>.quadprog</code>	<code>...lsqcurvefit</code>	<code>...fminsearch</code>	<code>...fminsearch</code>
Límite	<code>.linprog</code>	<code>.quadprog</code>	<code>...</code>	<code>...fminbnd</code>	<code>...fminbnd</code>
Lineal	<code>.linprog</code>	<code>.quadprog</code>	<code>...lsqcurvefit</code>	<code>...fmincon</code>	*
General Smooth	<code>...fmincon</code>	<code>...fmincon</code>	<code>...fmincon</code>	<code>...fmincon</code>	*
Discreto, con encuadernado o lineal	<code>...intlinprog</code>	*	*	*	*

Figura 7: Tabla de decisión de optimización

En nuestro caso la función objetivo es lineal y las restricciones toman valores discretos, por lo que se elige el solucionador **intlinprog**.

### 3.3.1 Intlinprog

#### 3.3.1.1 Descripción

Este solucionador o función de *Matlab* permite encontrar el mínimo de un problema de programación lineal de enteros mixtos como el descrito en las **ecuaciones 3-3 y 3-4**. La estructura completa de esta función es:

$$[x, fval, exitflag, output] = intlinprog(f, intcon, A, b, A_{eq}, b_{eq}, lb, ub, x_0, options) \quad (3-5)$$

A continuación, se explican los argumentos de entrada y salida disponibles [11].

- Argumentos de entrada
  - Vector de coeficiente **f**: se especifica como un vector real o una matriz real, y representa la función objetivo a introducir.
  - Vector de restricciones de enteros **intcon**: se especifica como un vector de enteros positivos, cuyos valores indican las componentes de la variable de decisión **x** que son valores enteros.
  - Matriz de restricción de desigualdad lineal **A**: se especifica como una matriz de dobles, cuyos valores indican las componentes de la variable de decisión **x** que forman parte de las restricciones de desigualdad. Si se trabaja con numerosas restricciones de recomienda utilizar una matriz dispersa para reducir el tiempo de resolución.
  - Vector de restricción de desigualdad lineal **b**: se especifica como vector de dobles y representa el valor constante de cada una de las restricciones de desigualdad.
  - Matriz de restricción de igualdad lineal **A<sub>eq</sub>**: se especifica como una matriz de dobles, cuyos valores indican las componentes de la variable de decisión **x** que forman parte de las restricciones de igualdad. Si se trabaja con numerosas restricciones de recomienda utilizar una matriz dispersa para reducir el tiempo de resolución.
  - Vector de restricción de desigualdad lineal **b<sub>eq</sub>**: se especifica como vector de dobles y representa el valor constante de cada una de las restricciones de igualdad.
  - Límites inferiores **lb**: se especifica como vector o array de dobles y representa el límite inferior de cada una de las componentes de la variable de decisión **x**. Es decir, permite introducir la restricción  $lb \leq x$ .
  - Límites superiores **ub**: se especifica como vector o array de dobles y representa el límite superior de cada una de las componentes de la variable de decisión **x**. Es decir, permite introducir la restricción  $x \leq ub$ .
  - Punto inicial **x<sub>0</sub>**: se especifica como una matriz real, e internamente el solucionador lo convierte en un vector **x<sub>0</sub>**. Utiliza los valores introducidos, siempre que éstos sean factibles, para comenzar la optimización del problema a partir de este punto inicial.
  - Opciones (**options**): se especifican como la salida de **optimoptions** y permite cambiar los valores predeterminados del solucionador. En la siguiente tabla se especifican las opciones

disponibles y se introduce una breve descripción de cada una de ellas.

**Tabla 2:** Opciones intlinprog

Opción	Descripción
AbsoluteGapTolerance	Número real no negativo que permite detener la función si la diferencia entre los límites superior e inferior calculados internamente en la función objetiva es menor.
BranchRule	Reglas para elegir el componente de la bifurcación del método de rama y encuadernado.
ConstraintTolerance	Máxima discrepancia o diferencia que las restricciones lineales pueden tener, sin llegar a detener el algoritmo.
CutGeneration	Nivel elegido del método de generación de corte.
CutMaxIterations	Número de pasos a través de los métodos de generación de corte antes de pasar a la fase de rama y encuadernado.
Display	Visualización de los valores finales o de todo el proceso iterativo.
Heuristics	Algoritmo utilizado para buscar puntos o soluciones factibles.
HeuristicsMaxNodes	Limita el número de nodos que puede explorar en el proceso de búsqueda de puntos factibles.
IntegerPreprocess	Tipo de preprocesamiento de enteros, permitiendo elegir entre pocos, moderados y todos los pasos.
IntegerTolerance	Tolerancia del preprocesamiento de enteros.
LPMaxIterations	Número máximo de iteraciones del algoritmo simplex por nodo durante los procesos de bifurcación y enlazado.
LPOptimalityTolerance	Real no negativo cuando los costos reducidos deben exceder para que una variable sea tomada en la base.
LPPreprocess	Tipo de preprocesado para la solución al programa lineal relajado.
MaxNodes	Número máximo de nodos explorados durante los procesos de bifurcación y enlazado.

MaxFeasiblePoints	Entero positivo, se detiene el algoritmo si encuentra puntos de enteros factibles.
MaxTime	Tiempo máximo en segundos para ejecutar el algoritmo.
NodeSelection	Elección del nodo que se desea explorar a continuación.
ObjectiveCutOff	Durante el proceso de bifurcación y enlazado descarta cualquier nodo en el que la solución de programación lineal tenga un valor objetivo superior.
ObjectiveImprovementThreshold	Cambia la solución factible actual solo cuando localiza otra con un valor de función objetivo que es al menos inferior.
OutputFcn	Permite especificar una o varias funciones a las que se les llama cuando suceden algunos eventos indicados.
PlotFcn	Traza varias medidas de progreso mientras se ejecuta el algoritmo.
RelativeGapTolerance	Número real no negativo que permite detener la función si la diferencia relativa entre los límites superior e inferior calculados internamente en la función objetiva es menor.
RootLPAlgorithm	Algoritmos utilizados para la resolución de programaciones lineales.
RootLPMaxIterations	Número máximo de iteraciones del algoritmo simplex para resolver el problema de programación lineal inicial.

- Argumentos de salida
  - Solución **x**: devuelve la solución en forma de vector, de manera que se minimice la función objetivo impuesta cumpliendo todas las restricciones introducidas. Cuando un problema es inviable o ilimitado, la solución es un vector vacío  $x = [ ]$ .
  - Valor objetivo **fval**: devuelve el valor de la función objetivo como una escalar
  - Condición de parada del algoritmo **exitflag**: devuelve como un entero la condición de parada del algoritmo. Los valores posibles y su significado se exponen en la siguiente tabla.

**Tabla 3:** Valores exitflag

3	Solución factible respecto a la tolerancia relativa pero no respecto a la tolerancia absoluta.
2	Detención prematura. Se encontró un punto

	entero factible.
1	Converge en la solución.
0	Detención prematura. No se encontró un punto entero factible.
-1	Detenido por una función de salida o función e trazado.
-2	No se encontró un punto entero factible.
-3	Las raíces del problema de programación lineal son ilimitadas.
-9	El solucionador o Solver perdió viabilidad.

- Resumen del proceso de solución (output): devuelve en forma de estructura información sobre el proceso de optimización. Los datos devueltos se exponen en la siguiente tabla.

**Tabla 4:** Datos output intlinprog

relativegap	Diferencia porcentual relativa entre los límites superior e inferior de la función objetiva que calcula en el algoritmo de bifurcación y enlazado
absolutegap	Diferencia entre los límites superior e inferior de la función objetiva que calcula en el algoritmo de bifurcación y enlazado
numfeaspoints	Número de puntos enteros factibles encontrados
numnodes	Número de nodos en el algoritmo de bifurcación y enlazado
constrviolation	Infracción de restricción positiva por las restricciones infringidas
message	Mensaje de salida

Esta es la estructura más completa que existe del solucionador utilizados. Sin embargo, en la mayoría de los casos no es necesario introducir tantos argumentos de entrada ni utilizar todos los argumentos de salida. En este caso, se opta por utilizar la siguiente sintaxis para la resolución del problema:

$$[x, fval, exitflag, output] = intlingprog(f, intcon, A, b, A_{eq}, b_{eq}, lb) \quad (3-6)$$

Es decir, no se va a introducir como valores de entrada el límite superior, el punto inicial y las opciones, y se utilizarán los valores predeterminados o por defecto que recomienda el software.

### 3.3.1.2 Funcionamiento

El funcionamiento o pasos que lleva a cabo esta función para resolver el problema se puede resumir en los siguientes puntos:

- Preprocesamiento lineal del programa
- Programación lineal
- Preprocesamiento de entero mixto del programa
- Métodos

En esta última parte de este capítulo, se detallan estos pasos para dar una idea del funcionamiento que realiza el algoritmo de *Matlab* con el objetivo de conocer en detalle la función utilizada para resolver nuestro problema [12].

En primer lugar, el solucionador busca reducir el tamaño del problema utilizando el preprocesamiento lineal del programa. Generalmente es posible reducir el número de variables y restricciones lineales, y aunque la realización de este proceso pueda suponer un aumento de tiempo para el solucionador, normalmente se consiguen reducciones totales de tiempo adecuadas para encontrar soluciones factibles de problemas mayores. En resumen, el objetivo principal de este primer paso es eliminar variables y restricciones redundantes para mejorar el tiempo de la solución final.

A continuación, se resuelve el problema de programación lineal relajado (**LP**) con la misma función objetivo y restricciones que el problema inicial, excepto las restricciones de enteros. Cabe destacar que dado que el problema **LP** resuelve la misma función objetivo pero con menos restricciones, se cumple que  $f^T x_{LP} \leq f^T x$ , siendo  $x_{LP}$  la solución del problema relajado y  $x$  la solución del problema original.

En tercer lugar, se realiza el pre-procesamiento de entero mixto, donde se analizan las restricciones de desigualdad lineal junto con las restricciones de enteros. El objetivo principal es simplificar los cálculos del método de bifurcación y enlazado subsiguientes, eliminando para ello algunas soluciones candidatas que conllevarían a subproblemas no productivos que de otra forma se analizarían, hecho que conllevaría a un mayor tiempo de ejecución del programa. Entre las principales conclusiones que se pueden determinar de este paso se encuentran: el problema es inviable, algunos límites se pueden apretar, algunas restricciones son redundantes, algunas restricciones pueden fortalecerse y algunas variables enteras pueden considerarse fijas.

Estos tres primeros procesos iniciales sirven de introducción a la resolución del problema y permite una reducción del tiempo de ejecución del algoritmo al eliminar cálculos innecesarios en el proceso de resolución del problema original.

A continuación, se realiza la optimización del problema, siendo el objetivo primordial minimizar la función objetivo cumpliendo con todas las restricciones impuestas, y teniendo en cuenta los resultados obtenidos de los tres pasos explicados anteriormente con el objetivo de mejorar la eficiencia del algoritmo y poder encontrar soluciones factibles en un menor tiempo.

Como se comentó al final del **Capítulo 3.2**, la resolución de problemas de programación lineal entera requiere el uso de una combinación de técnicas o métodos para estrechar el conjunto de soluciones posibles, descartar soluciones que no llegan a ser óptimas y encontrar soluciones factibles, dado el gran número de variables que intervienen en el problema. Entre estas técnicas comunes se incluyen el método de la generación de corte, la heurística y el método de rama y encuadrado (conocido también como bifurcación y enlazado), los cuales se utilizan internamente en la función **intlinprog** y se explican brevemente a continuación.

La generación de corte consiste en introducir una serie de desigualdades lineales adicionales al problema con el objetivo de restringir una región de soluciones factibles. Para ello, se ejecutan una serie de iteraciones y cortes que permiten definir un espacio adecuado alrededor de la solución del problema relajado **LP** que permita encontrar soluciones de enteros del problema original. Existen bastantes tipos de cortes, siendo utilizados en este caso aquellos que se encuentran dentro de las opciones predeterminadas del solucionador 'basic', por recomendación del software al ser nuestro problema de tipo entero íntegramente. La utilización de uno u otro tipo de corte se realiza por el mismo programa, escogiendo aquellos que permitan obtener las soluciones más eficientes. Se nombran los tipos de corte que pueden llegar a utilizarse sin entrar en más detalle:

- Cortes de redondeo de enteros mixtos
- Gomory corta
- Clique corta
- Cortes de cubierta
- Curbierta de flujo corta
- Strong Chvatal-Gomory corta
- Cero-medias cortes

Por otro lado, la heurística es un proceso que se utiliza para encontrar soluciones factibles, y permite obtener un límite superior de la función objetivo introducida. Algunas de las soluciones del problema relajado **LP** puede ser un número entero factible que pueda proporcionar un límite superior de la función objetivo en el problema original **IP**, permitiendo reducir el tiempo de procesamiento del algoritmo al reducir el tiempo de las iteraciones que se realizan durante el proceso de rama y encuadernado. Como su propio nombre indica, es una técnica heurística, lo cual significa que son algoritmos que pueden llegar a tener éxito, pero en su contra también tiene probabilidades de fallar.

Por último, el método de rama y encuadernado o bifurcación y enlazado permite construir una secuencia de subproblemas para intentar converger a la solución del problema original **IP**. Estos subproblemas acotan una región de soluciones, definidos por un límite superior e inferior. El primer límite superior es cualquier solución factible y el primer límite inferior es la solución del problema relajado, que como se comentó anteriormente tiene un valor de la función objetivo menor que la solución de la **IP**.

A partir de los límites iniciales, el método de bifurcación y enlazado construye nuevos subproblemas ramificando desde el nodo raíz. Este procedimiento permite por tanto elevar el límite inferior en cada iteración hasta llegar a la solución final. El paso de ramificación elegido se toma heurísticamente de acuerdo con varias reglas que dependen de las opciones introducidas al solucionador, que en nuestro caso será la opción predeterminada del programa.

# 4 RESOLUCIÓN DEL PROBLEMA

---

El objetivo principal de este trabajo es crear un programa en *Matlab* para resolver el problema de ‘Aircraft Routing’ o rotación de aviones. Una vez que se conocen las características de este tipo de problemas, como son los requerimientos y restricciones que se deben cumplir, las funciones objetivas que se van a optimizar, el modelado matemático y las técnicas y métodos disponibles en el software elegido para implementar los algoritmos necesarios para la resolución del problema, es posible comenzar con la programación para obtener las rotaciones que optimicen las funciones objetivas seleccionadas.

El programa elaborado consta de una serie de funciones y archivos los cuales se exponen en el **Anexo A**, dónde además se explica con detenimiento el proceso seguido para la creación de cada uno de ellos. En este capítulo se resumen y explican brevemente estos algoritmos implementados, y se presenta un ejemplo ficticio con objeto de poner en práctica el programa y analizar las soluciones obtenidas.

## 4.1 Archivos

Recopilando la información introducida en capítulos anteriores, para abordar el problema necesitamos conocer inicialmente la programación de vuelos, el tipo de avión que realiza cada uno de los vuelos, el aeropuerto donde se realiza el mantenimiento, el tiempo de escala y el número de días que componen la rotación.

Los diferentes archivos creados tienen como objetivo la resolución del problema para rotaciones de 3 días, habiéndose optado por este número al ser las más comunes. Para el resto de las variables, se deja la opción al usuario de introducir los datos que conozca o considere más oportuno, con el objetivo de que pueda cambiar algunas de ellas y observar las diferencias en los resultados obtenidos.

Con el propósito de facilitar las labores de introducción de los datos de la programación de vuelos, se elabora un archivo con la herramienta *Excel* llamado **Datos vuelos**, el cual se adjunta junto con los diferentes archivos de *Matlab* para poder ser utilizados por el usuario. En éste, el usuario introducirá los datos de los vuelos que realizan un mismo tipo de avión, ya que si existen diferentes tipos de aviones el problema se aborda por separado, es decir, un problema para cada tipo de avión. Los datos que deben introducirse en cada una de las columnas son:

- Columna A. Número de vuelo
- Columna B. Aeropuerto de salida
- Columna C. Aeropuerto de llegada
- Columna D. Día del vuelo
- Columna E. Horario salida (h)
- Columna F. Horario salida (min)
- Columna G. Horario llegada (h)
- Columna H. Horario llegada (min)
- Columna I. Duración
- Columna J. Coste

Se explica brevemente la forma en que deben de introducirse estos datos. En primer lugar, a cada aeropuerto de la red se le asigna un número de identificación, siendo éste el que debe introducirse en las columnas A y B. Para los horarios de salida y llegada, se crean las columnas E y G para introducir el número que representa la hora (0-24) y las columnas F y H para los minutos (0-59). En la columna I con la duración de cada uno de los trayectos se debe tener en cuenta que cada una de las horas de salida y llegada se corresponden con la hora local del

aeropuerto de salida y llegada. Por último, en la columna J se introduce el coste operativo de cada uno de los trayectos.

El programa completo consta de cuatro archivos, correspondientes a cada una de las funciones objetivos que se presentaron en el **Capítulo 2.1.2**, y dos funciones que se ejecutan dentro de cada uno de los archivos permitiendo la generación de todas las rotaciones posibles.

El archivo **routing1** permite obtener la solución del problema de modo que se maximice las posibilidades de mantenimiento, es decir, el número de estancias nocturnas en el aeropuerto de mantenimiento. Para ello, el usuario debe introducir al inicio los valores de las variables **Am**, **tmin** y **Na**, que hacen referencia al aeropuerto de mantenimiento, el tiempo mínimo entre la llegada y posterior salida de un mismo avión o tiempo de escala y el número máximo de aviones disponibles respectivamente.

El archivo **routing2** permite obtener la solución del problema de modo que se maximice el número de conexiones de vuelo, es decir, se consiguen las rotaciones con un mayor número de vuelos. Para ello, el usuario debe introducir al inicio los valores de las variables **Am**, **tmin** y **Na**, que hacen referencia al aeropuerto de mantenimiento, el tiempo de escala y el número máximo de aviones disponibles respectivamente.

El archivo **routing3** permite obtener la solución del problema de modo que se minimice la desviación respecto al número de horas de referencia. Para ello, el usuario debe introducir al inicio los valores de las variables **Am**, **tmin**, **H** y **Na**, que hacen referencia al aeropuerto de mantenimiento, el tiempo de escala, el número de horas de referencia y el número máximo de aviones disponibles respectivamente.

El archivo **routing4** permite obtener la solución del problema de modo que se minimicen los costes. Para ello, el usuario debe introducir al inicio los valores de las variables **Am**, **tmin** y **Na**, que hacen referencia al aeropuerto de mantenimiento, el tiempo de escala y el número máximo de aviones disponibles respectivamente. Conocer los costes de operación de cada uno de los vuelos no es una tarea fácil, y dado que el objetivo de este programa es meramente didáctico, se aconseja al usuario introducir estos costes como valores comprendidos en un rango designado según su criterio o si lo prefieren aleatoriamente, y observar los resultados obtenidos al variar estas cantidades.

El código de estos cuatro archivos es muy similar, diferenciándose únicamente en la función objetivo. Por ello, el funcionamiento de todos ellos se puede resumir en las siguientes líneas:

1. Introducción de datos por parte del usuario.
2. Lectura del archivo *Excel*.
3. Generación de todas las rotaciones posibles, teniendo en cuenta los requerimientos de mantenimiento y el tiempo mínimo entre la llegada y salida de un mismo avión. Para ello, se han creado 2 funciones: **rot1** y **rotaciones1**.
  - La función **rot1** permite obtener, a partir de la programación de vuelos y el tiempo mínimo entre llegada y salida de un mismo avión, todas las secuencias posibles de vuelos de 1 día, para cada uno de los 3 días en los que pueden realizarse vuelos.
  - La función **rotaciones1**, a partir de la programación de vuelos, el tiempo de escala permitido, el aeropuerto de mantenimiento y la salida de rot1 (secuencias posibles de vuelos de 1 día) permite obtener todas las rotaciones posibles de 3 días.
4. Creación de la función objetivo, una diferente para cada uno de los archivos explicados.
5. Creación de las restricciones de cobertura de vuelos y número máximo de aviones disponibles.
6. Ejecución de la función **intlinprog** de *Matlab* con los datos necesarios introducidos y calculados para optimizar el problema.
7. Obtención de la solución, es decir, de las rotaciones elegidas.

Los códigos de los archivos y funciones creados, así como una explicación más detallada de la implementación de todos ellos se pueden consultar en el **Anexo A**.



## 4.2 Ejemplo Ultimate Air

Para poner en práctica el programa desarrollado se recurre a un ejemplo ficticio presente en el libro ‘Airlines Operations and Scheduling’ [5]. En este se presenta una aerolínea ficticia, llamada Ultimate Air, que realiza 42 vuelos al día entre un total de 8 aeropuertos estadounidenses. Se asume que los mismos vuelos se repiten diariamente.

La programación de vuelos de un día de la compañía aérea es la siguiente:

**Tabla 5:** Programación de vuelos Ultimate Air

Nº vuelo	Aeropuerto salida	Aeropuerto llegada	Horario salida (h)	Horario salida (min)	Horario llegada (h)	Horario llegada (min)	Duración (h)
101	2	1	5	0	13	30	5.5
104	3	1	5	5	13	35	5.5
116	7	1	6	15	7	45	1.5
140	1	8	6	20	7	20	1
125	1	3	7	25	9	55	5.5
107	4	1	7	30	10	30	2
122	1	2	7	35	10	5	5.5
137	1	7	7	40	9	10	1.5
110	5	1	8	10	10	40	2.5
119	8	1	8	15	9	15	1
113	6	1	9	10	12	10	3
131	1	5	9	30	12	0	2.5
102	2	1	9	45	18	15	5.5
105	3	1	9	50	18	20	5.5
117	7	1	10	0	11	30	1.5
128	1	4	10	5	11	5	2
134	1	6	10	35	13	35	3
141	1	8	12	0	13	0	1
108	4	1	12	20	15	20	2
138	1	7	12	30	14	0	1.5

111	5	1	13	10	15	40	2.5
120	8	1	14	25	15	25	1
114	6	1	14	30	17	30	3
132	1	5	14	35	17	35	3
118	7	1	15	0	16	30	1.5
129	1	4	15	5	16	5	2
135	1	6	15	10	18	10	3
142	1	8	15	15	16	15	1
103	2	1	15	20	23	50	5.5
106	3	1	15	25	23	55	5.5
126	1	3	15	30	18	0	5.5
123	1	2	16	0	18	30	5.5
109	4	1	17	10	20	10	2
112	5	1	18	0	20	30	2.5
133	1	5	18	5	20	35	2.5
136	1	6	18	10	21	10	3
115	6	1	18	15	21	15	3
121	8	1	18	30	19	30	1
124	1	2	19	0	21	30	5.5
127	1	3	20	0	22	30	5.5
130	1	4	21	0	22	0	2
139	1	7	21	30	23	0	1.5

Cabe destacar que todas las horas de salidas y llegadas hacen referencia a la hora local de cada uno de los aeropuertos. Por otro lado, la numeración de aeropuertos escogida es:

**Tabla 6:** Aeropuertos Ultimate Air

Nº Aeropuerto	Código IATA	Nombre Aeropuerto
1	JFK	Aeropuerto Internacional John

		F.Kennedy
2	LAX	Aeropuerto Internacional de Los Ángeles
3	SFO	Aeropuerto Internacional de San Francisco
4	ORD	Aeropuerto Internacional de Chicago O'Hare
5	ATL	Aeropuerto Internacional de Atlanta
6	MIA	Aeropuerto Internacional de Miami
7	BOS	Aeropuerto Internacional Logan de Boston
8	IAD	Aeropuerto Internacional de Washington-Dulles

A continuación, es necesario conocer el tipo de aeronave que realiza cada vuelo. En este ejemplo, se da la posibilidad de realizar cada trayecto con dos tipos de aviones: el Boeing 737-800 y el Boeing 757-200. El proceso seguido para asignar cada tipo de aeronave a cada uno de los vuelos, conocido como asignación de flota, muestra que 12 vuelos se realizarán con el B757 y 30 vuelos con el B737. No se dan más detalles acerca del proceso seguido para obtener esta solución, pues corresponde a otro tipo de problema distinto del que nos ocupa en este trabajo.

**Tabla 7:** Vuelos operados por cada aeronave

B737	B757
101	125
104	110
116	113
140	131
107	105
122	138
137	11
119	114
102	118
117	135

---

128	133
134	136
141	
108	
120	
132	
129	
142	
103	
106	
126	
123	
109	
112	
115	
121	
124	
127	
130	
130	

Una vez conocido la programación de vuelos y el tipo de aeronave que realiza cada uno de ellos, es necesario separar el problema en tantos subproblemas como tipos de aeronaves distintas existan. En este caso, se realiza en primer lugar el proceso para obtener las rotaciones de 3 días para los vuelos que se operan con el B737 por un lado y seguidamente se ejecuta el mismo proceso para los vuelos operados por el B757 por otro lado. De esta forma, obtenemos todas las rotaciones para cada tipo de aeronave que cumplan con los requerimientos y restricciones iniciales.

El último paso antes de ejecutar el programa es designar un aeropuerto de mantenimiento y el tiempo mínimo entre la llegada y salida de un mismo avión. En este caso, se opta por introducir el aeropuerto JFK de Nueva York como aeropuerto de mantenimiento, siendo de obligado cumplimiento realizar una parada nocturna o pernoctación en éste como mínimo uno de los 3 días de la rotación. El tiempo de escala para esta compañía aérea se fija en 45 minutos.

### 4.2.1 Boeing 737

Se comienza resolviendo el problema para los vuelos realizados con la aeronave B737. Se introducen los datos de los vuelos, el aeropuerto 1 (JFK) como aeropuerto de mantenimiento y un tiempo de escala de 45 minutos. Para el número máximo de aeronaves disponibles se introducen 9, por ser el resultado obtenido en el problema de la asignación de flota. Sin embargo, para este número de aviones el problema es demasiado restrictivo y no es capaz de encontrar ninguna solución factible, por lo que se opta por relajar esta restricción e introducir 12 como el número máximo de aeronaves disponibles.

Se ejecutan los 4 archivos, de modo que se obtienen las diferentes soluciones minimizando o maximizando cada una de las funciones objetivos explicadas con anterioridad. Para este problema se obtienen un total de 86.967 rotaciones posibles, que es igual al número de variables del problema, y 91 restricciones. Cabe destacar el elevado número de variables y restricciones que conforman el problema con tan solo 30 vuelos al día, hecho que pone de manifiesto la complejidad de este tipo de problemas, tal y como se discutió en el **Capítulo 3.1**.

Para analizar el tiempo de ejecución del programa y obtener una aproximación de lo exigente que pueden ser este tipo de problemas al introducir un gran número de variables, se mide el tiempo que tarda en ejecutarse en programa con las funciones **tic** y **toc** de *Matlab*. La forma ideal de analizar la eficiencia del código sería medir el número de operaciones de cada uno de los algoritmos introducidos para poder obtener medidas de la eficiencia más acorde, ya que el tiempo de ejecución depende no solo de la eficiencia del algoritmo sino también de otros factores como el lenguaje de programación y el procesador del ordenador utilizado. Sin embargo, ya que todos los archivos creados se crean en el mismo lenguaje y se utiliza el mismo ordenador para ejecutar los diferentes problemas que se analizan, y cómo el objetivo principal sólo es obtener una aproximación de la eficiencia del código al comparar los diferentes tiempos de ejecución para los diferentes archivos y problemas que se solucionan, se ha decidido utilizar finalmente el tiempo como medida para obtener una idea de la eficiencia de los archivos.

De este modo, se ha obtenido las diferentes soluciones para cada función objetivo. A continuación, se presentan los resultados para cada uno de los archivos, pudiéndose observar las rotaciones que optimizan cada una de las funciones introducidas y que cumplen con todos los requerimientos y restricciones impuestos. Cada una de estas rotaciones puede estar formada en este problema por un máximo de hasta 5 vuelos diarios.

#### 4.2.1.1 Routing1

Para maximizar el número de días que se realiza una estancia nocturna en el aeropuerto de mantenimiento JFK, y maximizar por tanto las posibilidades de mantenimiento, se precisan operar 12 rotaciones compuestas por los siguientes vuelos:

**Tabla 8:** B737. Solución routing1

R	Vuelos						
	Día 1		Día 2			Día 3	
1	<b>106</b>	<b>141</b>	<b>120</b>	<b>124</b>	<b>102</b>	<b>127</b>	
	SFO-JFK	JFK-IAD	IAD-JFK	JFK-LAX	LAX-JFK	JFK-SFO	
	15:25-23:55	12:00-13:00	14:25-15:25	19:00-21:30	9:45-18:15	20:00-22:30	
2	<b>126</b>	<b>104</b>	<b>126</b>	<b>106</b>			
	JFK-SFO	SFO-JFK	JFK-SFO	SFO-JFK			
	15:30-18:00	5:05-13:35	15:30-18:00	15:25-23:55			
3	<b>104</b>	<b>132</b>	<b>112</b>		<b>137</b>	<b>117</b>	<b>126</b>
	SFO-JFK	JFK-ATL	ATL-JFK		JFK-BOS	BOS-JFK	JFK-SFO

	5:05-13:35	14:35-17:35	18:00-20:30					7:40-9:10	10:00-11:30	15:30-18:00
4	<b>122</b>	<b>103</b>	<b>140</b>	<b>119</b>	<b>129</b>	<b>109</b>	<b>130</b>	<b>107</b>	<b>141</b>	<b>120</b>
	JFK-LAX	LAX-JFK	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK	JFK-IAD	IAD-JFK
	7:35-10:05	15:20-23:50	6:20-7:20	8:15-9:15	15:05-16:05	17:10-20:10	21:00-22:00	7:30-10:30	12:00-13:00	14:25-15:25
5	<b>102</b>	<b>127</b>	<b>106</b>					<b>128</b>	<b>108</b>	<b>124</b>
	LAX-JFK	JFK-SFO	SFO-JFK					JFK-ORD	ORD-JFK	JFK-LAX
	9:45-18:15	20:00-22:30	15:25-23:55					10:05-11:05	12:20-15:20	19:00-21:30
6	<b>112</b>	<b>139</b>	<b>116</b>	<b>134</b>	<b>115</b>	<b>140</b>	<b>119</b>	<b>132</b>		
	ATL-JFK	JFK-BOS	BOS-JFK	JFK-MIA	MIA-JFK	JFK-IAD	IAD-JFK	JFK-ATL		
	18:00-20:30	21:30-23:00	6:15-7:45	10:35-13:35	18:15-21:15	6:20-7:20	8:15-9:15	14:35-17:35		
7	<b>116</b>	<b>128</b>	<b>108</b>	<b>128</b>	<b>108</b>	<b>127</b>	<b>104</b>	<b>129</b>	<b>109</b>	<b>139</b>
	BOS-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK	JFK-SFO	SFO-JFK	JFK-ORD	ORD-JFK	JFK-BOS
	6:15-7:45	10:05-11:05	12:20-15:20	10:05-11:05	12:20-15:20	20:00-22:30	5:05-13:35	15:05-16:05	17:10-20:10	21:30-23:00
8	<b>107</b>	<b>142</b>	<b>121</b>	<b>137</b>	<b>117</b>	<b>123</b>	<b>101</b>	<b>142</b>	<b>121</b>	<b>130</b>
	ORD-JFK	JFK-IAD	IAD-JFK	JFK-BOS	BOS-JFK	JFK-LAX	LAX-JFK	JFK-IAD	IAD-JFK	JFK-ORD
	7:30-10:30	15:15-16:15	18:30-19:30	7:40-9:10	10:00-11:30	16:00-18:30	5:00-13:30	15:15-16:15	18:30-19:30	21:00-22:00
9	<b>137</b>	<b>117</b>	<b>123</b>	<b>102</b>		<b>139</b>	<b>116</b>	<b>134</b>	<b>115</b>	
	JFK-BOS	BOS-JFK	JFK-LAX	LAX-JFK		JFK-BOS	BOS-JFK	JFK-MIA	MIA-JFK	
	7:40-9:10	10:00-11:30	16:00-18:30	9:45-18:15		21:30-23:00	6:15-7:45	10:35-13:35	18:15-21:15	
10	<b>141</b>	<b>120</b>	<b>124</b>	<b>101</b>		<b>132</b>	<b>112</b>			
	JFK-IAD	IAD-JFK	JFK-LAX	LAX-JFK		JFK-ATL	ATL-JFK			
	12:00-13:00	14:25-15:25	19:00-21:30	5:00-13:30		14:35-17:35	18:00-20:30			
11	<b>101</b>	<b>129</b>	<b>109</b>	<b>130</b>	<b>107</b>	<b>142</b>	<b>121</b>	<b>123</b>		
	LAX-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK	JFK-IAD	IAD-JFK	JFK-LAX		

	5:00-13:30	15:05-16:05	17:10-20:10	21:00-22:00	7:30-10:30	15:15-16:15	18:30-19:30	16:00-18:30		
12	<b>140</b>	<b>119</b>	<b>134</b>	<b>115</b>	<b>122</b>		<b>103</b>		<b>122</b>	<b>103</b>
	JFK-IAD	IAD-JFK	JFK-MIA	MIA-JFK	JFK-LAX		LAX-JFK		JFK-LAX	LAX-JFK
	6:20-7:20	8:15-9:15	10:35-13:35	18:15-21:15	7:35-10:05		15:20-23:50		7:35-10:05	15:20-23:50

El tiempo aproximado empleado en obtener la solución ha sido de 11,3 minutos.

#### 4.2.1.2 Routing2

Para maximizar el número de conexiones de vuelos, es decir, seleccionar las rotaciones que operen un mayor número de vuelos, se precisan operar 12 rotaciones compuestas por los siguientes vuelos:

**Tabla 9:** B737. Solución routing2

R	Vuelos									
	Día 1			Día 2			Día 3			
1	<b>106</b>			<b>123</b>			<b>101</b>	<b>126</b>		
	SFO-JFK			JFK-LAX			LAX-JFK	JFK-SFO		
	15:25-23:55			16:00-18:30			5:00-13:30	15:30-18:00		
2	<b>126</b>			<b>104</b>	<b>132</b>		<b>112</b>			
	JFK-SFO			SFO-JFK	JFK-ATL		ATL-JFK			
	15:30-18:00			5:05-13:35	14:35-17:35		18:00-20:30			
3	<b>107</b>	<b>132</b>		<b>112</b>			<b>141</b>	<b>120</b>	<b>130</b>	
	ORD-JFK	JFK-ATL		ATL-JFK			JFK-IAD	IAD-JFK	JFK-ORD	
	7:30-10:30	14:35-17:35		18:00-20:30			12:00-13:00	14:25-15:25	21:00-22:00	
4	<b>122</b>	<b>103</b>		<b>122</b>	<b>103</b>		<b>122</b>		<b>103</b>	
	JFK-LAX	LAX-JFK		JFK-LAX	LAX-JFK		JFK-LAX		LAX-JFK	
	7:35-10:05	15:20-23:50		7:35-10:05	15:20-23:50		7:35-10:05		15:20-23:50	
5	<b>102</b>		<b>127</b>		<b>106</b>			<b>123</b>		
	LAX-JFK		JFK-SFO		SFO-JFK			JFK-LAX		
	9:45-18:15		20:00-22:30		15:25-23:55			16:00-18:30		
6	<b>112</b>		<b>139</b>		<b>116</b>	<b>134</b>	<b>115</b>	<b>137</b>	<b>117</b>	<b>132</b>

	ATL-JFK		JFK-BOS		BOS-JFK		JFK-MIA		MIA-JFK		JFK-BOS		BOS-JFK		JFK-ATL	
	18:00-20:30		21:30-23:00		6:15-7:45		10:35-13:35		18:15-21:15		7:40-9:10		10:00-11:30		14:35-17:35	
7	<b>101</b>	<b>142</b>	<b>121</b>	<b>128</b>	<b>108</b>	<b>124</b>	<b>102</b>		<b>124</b>							
	LAX-JFK	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-LAX	LAX-JFK		JFK-LAX							
	5:00-13:30	15:15-16:15	18:30-19:30	10:05-11:05	12:20-15:20	19:00-21:30	9:45-18:15		19:00-21:30							
8	<b>104</b>	<b>129</b>	<b>109</b>	<b>137</b>	<b>117</b>	<b>142</b>	<b>121</b>	<b>139</b>	<b>116</b>	<b>128</b>	<b>108</b>	<b>127</b>				
	SFO-JFK	JFK-ORD	ORD-JFK	JFK-BOS	BOS-JFK	JFK-IAD	IAD-JFK	JFK-BOS	BOS-JFK	JFK-ORD	ORD-JFK	JFK-SFO				
	5:05-13:35	15:05-16:05	17:10-20:10	7:40-9:10	10:00-11:30	15:15-16:15	18:30-19:30	21:30-23:00	6:15-7:45	10:05-11:05	12:20-15:20	20:00-22:30				
9	<b>116</b>	<b>134</b>	<b>115</b>	<b>140</b>	<b>119</b>	<b>141</b>	<b>120</b>	<b>127</b>	<b>104</b>	<b>142</b>	<b>121</b>	<b>139</b>				
	BOS-JFK	JFK-MIA	MIA-JFK	JFK-IAD	IAD-JFK	JFK-IAD	IAD-JFK	JFK-SFO	SFO-JFK	JFK-IAD	IAD-JFK	JFK-BOS				
	6:15-7:45	10:35-13:35	18:15-21:15	6:20-7:20	8:15-9:15	12:00-13:00	14:25-15:25	20:00-22:30	5:05-13:35	15:15-16:15	18:30-19:30	21:30-23:00				
10	<b>137</b>	<b>117</b>	<b>123</b>	<b>102</b>					<b>140</b>	<b>119</b>	<b>134</b>	<b>115</b>				
	JFK-BOS	BOS-JFK	JFK-LAX	LAX-JFK					JFK-IAD	IAD-JFK	JFK-MIA	MIA-JFK				
	7:40-9:10	10:00-11:30	16:00-18:30	9:45-18:15					6:20-7:20	8:15-9:15	10:35-13:35	18:15-21:15				
11	<b>141</b>	<b>120</b>	<b>130</b>	<b>107</b>			<b>126</b>		<b>106</b>							
	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK			JFK-SFO		SFO-JFK							
	12:00-13:00	14:25-15:25	21:00-22:00	7:30-10:30			15:30-18:00		15:25-23:55							
12	<b>140</b>	<b>119</b>	<b>128</b>	<b>108</b>	<b>124</b>	<b>101</b>	<b>129</b>	<b>109</b>	<b>130</b>	<b>107</b>	<b>129</b>	<b>109</b>				
	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-LAX	LAX-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK				
	6:20-7:20	8:15-9:15	10:05-11:05	12:20-15:20	19:00-21:30	5:00-13:30	15:05-16:05	17:10-20:10	21:00-22:00	7:30-10:30	15:05-16:05	17:10-20:10				

El tiempo aproximado empleado en obtener la solución ha sido de 11,2 minutos.

#### 4.2.1.3 Routing3

Para minimizar la desviación del número de horas de vuelos de cada una de las rotaciones respecto al número de horas de referencia introducido, que se ha optado por que sea igual a 20, se precisan operar 12 rotaciones compuestas por los siguientes vuelos:



Tabla 10: B737. Solución routing3

R	Vuelos										
	Día 1		Día 2				Día 3				
1	<b>102</b>		<b>122</b>		<b>103</b>		<b>123</b>				
	LAX-JFK		JFK-LAX		LAX-JFK		JFK-LAX				
	9:45-18:15		7:35-10:05		15:20-23:50		16:00-18:30				
2	<b>106</b>		<b>137</b>	<b>117</b>	<b>142</b>	<b>121</b>	<b>130</b>	<b>107</b>		<b>126</b>	
	SFO-JFK		JFK-BOS	BOS-JFK	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK		JFK-SFO	
	15:25-23:55		7:40-9:10	10:00-11:30	15:15-16:15	18:30-19:30	21:00-22:00	7:30-10:30		15:30-18:00	
3	<b>112</b>		<b>128</b>		<b>108</b>		<b>127</b>		<b>104</b>	<b>132</b>	
	ATL-JFK		JFK-ORD		ORD-JFK		JFK-SFO		SFO-JFK	JFK-ATL	
	18:00-20:30		10:05-11:05		12:20-15:20		20:00-22:30		5:05-13:35	14:35-17:35	
4	<b>101</b>	<b>132</b>	<b>112</b>				<b>140</b>	<b>119</b>	<b>141</b>	<b>120</b>	<b>124</b>
	LAX-JFK	JFK-ATL	ATL-JFK				JFK-IAD	IAD-JFK	JFK-IAD	IAD-JFK	JFK-LAX
	5:00-13:30	14:35-17:35	18:00-20:30				6:20-7:20	8:15-9:15	12:00-13:00	14:25-15:25	19:00-21:30
5	<b>104</b>	<b>123</b>	<b>102</b>				<b>128</b>		<b>108</b>	<b>127</b>	
	SFO-JFK	JFK-LAX	LAX-JFK				JFK-ORD		ORD-JFK	JFK-SFO	
	5:05-13:35	16:00-18:30	9:45-18:15				10:05-11:05		12:20-15:20	20:00-22:30	
6	<b>107</b>	<b>126</b>	<b>106</b>				<b>137</b>	<b>117</b>	<b>142</b>	<b>121</b>	<b>130</b>
	ORD-JFK	JFK-SFO	SFO-JFK				JFK-BOS	BOS-JFK	JFK-IAD	IAD-JFK	JFK-ORD
	7:30-10:30	15:30-18:00	15:25-23:55				7:40-9:10	10:00-11:30	15:15-16:15	18:30-19:30	21:00-22:00
7	<b>122</b>	<b>103</b>	<b>140</b>	<b>119</b>	<b>134</b>	<b>115</b>	<b>122</b>		<b>103</b>		
	JFK-LAX	LAX-JFK	JFK-IAD	IAD-JFK	JFK-MIA	MIA-JFK	JFK-LAX		LAX-JFK		
	7:35-10:05	15:20-23:50	6:20-7:20	8:15-9:15	10:35-13:35	18:15-21:15	7:35-10:05		15:20-23:50		
8	<b>116</b>	<b>134</b>	<b>115</b>	<b>141</b>	<b>120</b>	<b>124</b>	<b>102</b>		<b>139</b>		
	BOS-JFK	JFK-MIA	MIA-JFK	JFK-IAD	IAD-JFK	JFK-LAX	LAX-JFK		JFK-BOS		

	6:15-7:45	10:35-13:35	18:15-21:15	12:00-13:00	14:25-15:25	19:00-21:30	9:45-18:15	21:30-23:00			
9	<b>128</b>	<b>108</b>	<b>124</b>	<b>101</b>		<b>123</b>		<b>101</b>	<b>129</b>	<b>109</b>	
	JFK-ORD	ORD-JFK	JFK-LAX	LAX-JFK		JFK-LAX		LAX-JFK	JFK-ORD	ORD-JFK	
	10:05-11:05	12:20-15:20	19:00-21:30	5:00-13:30		16:00-18:30		5:00-13:30	15:05-16:05	17:10-20:10	
10	<b>129</b>	<b>109</b>	<b>139</b>	<b>116</b>	<b>129</b>	<b>109</b>	<b>139</b>	<b>116</b>	<b>134</b>	<b>115</b>	
	JFK-ORD	ORD-JFK	JFK-BOS	BOS-JFK	JFK-ORD	ORD-JFK	JFK-BOS	BOS-JFK	JFK-MIA	MIA-JFK	
	15:05-16:05	17:10-20:10	21:30-23:00	6:15-7:45	15:05-16:05	17:10-20:10	21:30-23:00	6:15-7:45	10:35-13:35	18:15-21:15	
11	<b>140</b>	<b>119</b>	<b>141</b>	<b>120</b>	<b>127</b>	<b>104</b>		<b>132</b>		<b>112</b>	
	JFK-IAD	IAD-JFK	JFK-IAD	IAD-JFK	JFK-SFO	SFO-JFK		JFK-ATL		ATL-JFK	
	6:20-7:20	8:15-9:15	12:00-13:00	14:25-15:25	20:00-22:30	5:05-13:35		14:35-17:35		18:00-20:30	
12	<b>137</b>	<b>117</b>	<b>142</b>	<b>121</b>	<b>130</b>	<b>107</b>		<b>126</b>		<b>106</b>	
	JFK-BOS	BOS-JFK	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK		JFK-SFO		SFO-JFK	
	7:40-9:10	10:00-11:30	15:15-16:15	18:30-19:30	21:00-22:00	7:30-10:30		15:30-18:00		15:25-23:55	

El tiempo aproximado empleado en obtener la solución ha sido de 11,2 minutos.

#### 4.2.1.4 Routing4

Para minimizar los costes operativos se precisan operar 12 rotaciones compuestas por los siguientes vuelos:

**Tabla 11:** B737. Solución routing4

R	Vuelos			
	Día 1	Día 2	Día 3	
1	<b>132</b>	<b>112</b>	<b>122</b>	<b>103</b>
	JFK-ATL	ATL-JFK	JFK-LAX	LAX-JFK
	14:35-17:35	18:00-20:30	7:35-10:05	15:20-23:50
2	<b>106</b>	<b>123</b>	<b>102</b>	<b>127</b>
	SFO-JFK	JFK-LAX	LAX-JFK	JFK-SFO
	15:25-23:55	16:00-18:30	9:45-18:15	20:00-22:30

3	<b>123</b>				<b>102</b>		<b>127</b>			<b>106</b>							
	JFK-LAX				LAX-JFK		JFK-SFO			SFO-JFK							
	16:00-18:30				9:45-18:15		20:00-22:30			15:25-23:55							
4	<b>112</b>				<b>122</b>		<b>103</b>			<b>132</b>							
	ATL-JFK				JFK-LAX		LAX-JFK			JFK-ATL							
	18:00-20:30				7:35-10:05		15:20-23:50			14:35-17:35							
5	<b>122</b>		<b>103</b>		<b>132</b>						<b>112</b>						
	JFK-LAX		LAX-JFK		JFK-ATL						ATL-JFK						
	7:35-10:05		15:20-23:50		14:35-17:35						18:00-20:30						
6	<b>102</b>		<b>127</b>		<b>106</b>						<b>123</b>						
	LAX-JFK		JFK-SFO		SFO-JFK						JFK-LAX						
	9:45-18:15		20:00-22:30		15:25-23:55						16:00-18:30						
7	<b>116</b>	<b>134</b>	<b>115</b>	<b>140</b>	<b>119</b>	<b>128</b>	<b>108</b>	<b>124</b>	<b>101</b>	<b>142</b>	<b>121</b>	<b>139</b>					
	BOS-JFK	JFK-MIA	MIA-JFK	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-LAX	LAX-JFK	JFK-IAD	IAD-JFK	JFK-BOS					
	6:15-7:45	10:35-13:35	18:15-21:15	6:20-7:20	8:15-9:15	10:05-11:05	12:20-15:20	19:00-21:30	5:00-13:30	15:15-16:15	18:30-19:30	21:30-23:00					
8	<b>107</b>	<b>141</b>	<b>120</b>	<b>137</b>		<b>117</b>		<b>126</b>		<b>104</b>	<b>129</b>	<b>109</b>	<b>130</b>				
	ORD-JFK	JFK-IAD	IAD-JFK	JFK-BOS		BOS-JFK		JFK-SFO		SFO-JFK	JFK-ORD	ORD-JFK	JFK-ORD				
	7:30-10:30	12:00-13:00	14:25-15:25	7:40-9:10		10:00-11:30		15:30-18:00		5:05-13:35	15:05-16:05	17:10-20:10	21:00-22:00				
9	<b>137</b>		<b>117</b>		<b>126</b>		<b>104</b>	<b>129</b>	<b>109</b>	<b>130</b>		<b>107</b>		<b>141</b>		<b>120</b>	
	JFK-BOS		BOS-JFK		JFK-SFO		SFO-JFK	JFK-ORD	ORD-JFK	JFK-ORD		ORD-JFK		JFK-IAD		IAD-JFK	
	7:40-9:10		10:00-11:30		15:30-18:00		5:05-13:35	15:05-16:05	17:10-20:10	21:00-22:00		7:30-10:30		12:00-13:00		14:25-15:25	
10	<b>101</b>	<b>142</b>	<b>121</b>	<b>139</b>	<b>116</b>		<b>134</b>		<b>115</b>		<b>140</b>	<b>119</b>	<b>128</b>	<b>108</b>	<b>124</b>		
	LAX-JFK	JFK-IAD	IAD-JFK	JFK-BOS	BOS-JFK		JFK-MIA		MIA-JFK		JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-LAX		
	5:00-13:30	15:15-16:15	18:30-19:30	21:30-23:00	6:15-7:45		10:35-13:35		18:15-21:15		6:20-7:20	8:15-9:15	10:05-11:05	12:20-15:20	19:00-21:30		
11	<b>104</b>	<b>129</b>	<b>109</b>	<b>130</b>	<b>107</b>		<b>141</b>		<b>120</b>		<b>137</b>		<b>117</b>		<b>126</b>		
	SFO-JFK	JFK-ORD	ORD-JFK	JFK-ORD	ORD-JFK		JFK-IAD		IAD-JFK		JFK-BOS		BOS-JFK		JFK-SFO		

	5:05-13:35	15:05-16:05	17:10-20:10	21:00-22:00	7:30-10:30	12:00-13:00	14:25-15:25	7:40-9:10	10:00-11:30	15:30-18:00		
12	<b>140</b>	<b>119</b>	<b>128</b>	<b>108</b>	<b>124</b>	<b>101</b>	<b>142</b>	<b>121</b>	<b>139</b>	<b>116</b>	<b>134</b>	<b>115</b>
	JFK-IAD	IAD-JFK	JFK-ORD	ORD-JFK	JFK-LAX	LAX-JFK	JFK-IAD	IAD-JFK	JFK-BOS	BOS-JFK	JFK-MIA	MIA-JFK
	6:20-7:20	8:15-9:15	10:05-11:05	12:20-15:20	19:00-21:30	5:00-13:30	15:15-16:15	18:30-19:30	21:30-23:00	6:15-7:45	10:35-13:35	18:15-21:15

El tiempo aproximado empleado en obtener la solución ha sido de 11,1 minutos.

### 4.2.2 Boeing 757

De manera similar al problema resultado para la flota de aviones anteriores, se introducen los datos de los vuelos, el aeropuerto 1 (JFK) como aeropuerto de mantenimiento y un tiempo mínimo de 45 minutos entre llegada y salida de un mismo avión. Para el número máximo de aeronaves disponibles se introducen en este caso 6, por ser el resultado obtenido en el problema de la asignación de flota. Sin embargo, para este número de aviones el problema es demasiado restrictivo y no es capaz de encontrar ninguna solución factible, por lo que se opta por relajar esta restricción e introducir 8 como el número máximo de aeronaves disponibles.

Se ejecutan los 4 archivos, de modo que se obtienen las diferentes soluciones minimizando o maximizando cada una de las funciones objetivos explicadas con anterioridad. Para este problema se obtienen un total de 455 rotaciones posibles, que es igual al número de variables del problema, y 37 restricciones. Recordamos que esta aeronave operaba un total de 12 vuelos diarios.

A continuación, se presentan los resultados para cada uno de los archivos, igual que en el caso anterior. Cada una de estas rotaciones puede estar formada en este problema por secuencias con un máximo de hasta 4 vuelos diarios.

#### 4.2.2.1 Routing1

Para maximizar el número de días que se realiza una estancia nocturna en el aeropuerto de mantenimiento JFK, y maximizar por tanto las posibilidades de mantenimiento, se precisan operar 8 rotaciones compuestas por los siguientes vuelos:

**Tabla 12:** B757. Solución routing1

R	Vuelos					
	Día 1		Día 2		Día 3	
1	<b>125</b>		<b>105</b>		<b>138</b>	<b>118</b>
	JFK-SFO		SFO-JFK		JFK-BOS	BOS-JFK
	7:25-9:55		9:50-18:20		12:30-14:00	15:00-16:30
2	<b>113</b>		<b>131</b>	<b>111</b>	<b>136</b>	<b>113</b>
	MIA-JFK		JFK-ATL	ATL-JFK	JFK-MIA	MIA-JFK
	9:10-12:10		9:30-12:00	13:10-15:40	18:10-21:10	9:10-12:10

3	<b>105</b>		<b>138</b>	<b>118</b>	<b>125</b>	
	SFO-JFK		JFK-BOS	BOS-JFK	JFK-SFO	
	9:50-18:20		12:30-14:00	15:00-16:30	7:25-9:55	
4	<b>114</b>		<b>133</b>		<b>110</b>	<b>136</b>
	MIA-JFK		JFK-ATL		ATL-JFK	JFK-MIA
	14:30-17:30		18:05-20:35		8:10-10:40	18:10-21:10
5	<b>136</b>		<b>114</b>		<b>131</b>	<b>111</b>
	JFK-MIA		MIA-JFK		JFK-ATL	ATL-JFK
	18:10-21:10		14:30-17:30		9:30-12:00	13:10-15:40
6	<b>110</b>	<b>135</b>	<b>113</b>		<b>133</b>	
	ATL-JFK	JFK-MIA	MIA-JFK		JFK-ATL	
	8:10-10:40	15:10-18:10	9:10-12:10		18:05-20:35	
7	<b>138</b>	<b>118</b>	<b>125</b>		<b>105</b>	
	JFK-BOS	BOS-JFK	JFK-SFO		SFO-JFK	
	12:30-14:00	15:00-16:30	7:25-9:55		9:50-18:20	
8	<b>131</b>	<b>111</b>	<b>133</b>	<b>110</b>	<b>135</b>	<b>114</b>
	JFK-ATL	ATL-JFK	JFK-ATL	ATL-JFK	JFK-MIA	MIA-JFK
	9:30-12:00	13:10-15:40	18:05-20:35	8:10-10:40	15:10-18:10	14:30-17:30

El tiempo aproximado empleado en obtener la solución ha sido de 1,7 segundos.

#### 4.2.2.2 Routing2

Para maximizar el número de conexiones de vuelos, es decir, seleccionar las rotaciones que operen un mayor número de vuelos, se precisan operar 8 rotaciones compuestas por los siguientes vuelos:

**Tabla 13:** B757. Solución routing2

R	Vuelos			
	Día 1	Día 2	Día 3	
1	<b>125</b>	<b>105</b>	<b>138</b>	<b>118</b>
	JFK-SFO	SFO-JFK	JFK-BOS	BOS-JFK
	7:25-9:55	9:50-18:20	12:30-14:00	15:00-16:30

2	<b>113</b>		<b>131</b>	<b>111</b>	<b>133</b>	<b>110</b>	<b>136</b>	
	MIA-JFK		JFK-ATL	ATL-JFK	JFK-ATL	ATL-JFK	JFK-MIA	
	9:10-12:10		9:30-12:00	13:10-15:40	18:05-20:35	8:10-10:40	18:10-21:10	
3	<b>105</b>		<b>138</b>		<b>118</b>		<b>125</b>	
	SFO-JFK		JFK-BOS		BOS-JFK		JFK-SFO	
	9:50-18:20		12:30-14:00		15:00-16:30		7:25-9:55	
4	<b>114</b>		<b>135</b>			<b>113</b>	<b>135</b>	
	MIA-JFK		JFK-MIA			MIA-JFK	JFK-MIA	
	14:30-17:30		15:10-18:10			9:10-12:10	15:10-18:10	
5	<b>135</b>		<b>114</b>			<b>131</b>	<b>111</b>	
	JFK-MIA		MIA-JFK			JFK-ATL	ATL-JFK	
	15:10-18:10		14:30-17:30			9:30-12:00	13:10-15:40	
6	<b>110</b>	<b>133</b>	<b>110</b>			<b>133</b>		
	ATL-JFK	JFK-ATL	ATL-JFK			JFK-ATL		
	8:10-10:40	18:05-20:35	8:10-10:40			18:05-20:35		
7	<b>138</b>	<b>118</b>	<b>125</b>			<b>105</b>		
	JFK-BOS	BOS-JFK	JFK-SFO			SFO-JFK		
	12:30-14:00	15:00-16:30	7:25-9:55			9:50-18:20		
8	<b>131</b>	<b>111</b>	<b>136</b>	<b>113</b>	<b>136</b>	<b>114</b>		
	JFK-ATL	ATL-JFK	JFK-MIA	MIA-JFK	JFK-MIA	MIA-JFK		
	9:30-12:00	13:10-15:40	18:10-21:10	9:10-12:10	18:10-21:10	14:30-17:30		

El tiempo aproximado empleado en obtener la solución ha sido de 1,8 segundos.

#### 4.2.2.3 Routing3

Para minimizar la desviación del número de horas de vuelos de cada una de las rotaciones respecto al número de horas de referencia introducido, que se ha optado por que sea igual a 20, se precisan operar 8 rotaciones compuestas por los siguientes vuelos:

**Tabla 14:** B757. Solución routing3

R	Vuelos		
	Día 1	Día 2	Día 3

1	<b>125</b>		<b>105</b>		<b>138</b>	<b>118</b>
	JFK-SFO		SFO-JFK		JFK-BOS	BOS-JFK
	7:25-9:55		9:50-18:20		12:30-14:00	15:00-16:30
2	<b>113</b>		<b>136</b>		<b>113</b>	<b>135</b>
	MIA-JFK		JFK-MIA		MIA-JFK	JFK-MIA
	9:10-12:10		18:10-21:10		9:10-12:10	15:10-18:10
3	<b>105</b>		<b>131</b>	<b>111</b>	<b>125</b>	
	SFO-JFK		JFK-ATL	ATL-JFK	JFK-SFO	
	9:50-18:20		9:30-12:00	13:10-15:40	7:25-9:55	
4	<b>114</b>		<b>138</b>	<b>118</b>	<b>136</b>	
	MIA-JFK		JFK-BOS	BOS-JFK	JFK-MIA	
	14:30-17:30		12:30-14:00	15:00-16:30	18:10-21:10	
5	<b>135</b>		<b>113</b>	<b>133</b>	<b>110</b>	
	JFK-MIA		MIA-JFK	JFK-ATL	ATL-JFK	
	15:10-18:10		9:10-12:10	18:05-20:35	8:10-10:40	
6	<b>110</b>	<b>136</b>	<b>114</b>		<b>131</b>	<b>111</b>
	ATL-JFK	JFK-MIA	MIA-JFK		JFK-ATL	ATL-JFK
	8:10-10:40	18:10-21:10	14:30-17:30		9:30-12:00	13:10-15:40
7	<b>138</b>		<b>118</b>	<b>125</b>		<b>105</b>
	JFK-BOS		BOS-JFK	JFK-SFO		SFO-JFK
	12:30-14:00		15:00-16:30	7:25-9:55		9:50-18:20
8	<b>131</b>	<b>111</b>	<b>133</b>	<b>110</b>	<b>135</b>	<b>114</b>
	JFK-ATL	ATL-JFK	JFK-ATL	ATL-JFK	JFK-MIA	MIA-JFK
	9:30-12:00	13:10-15:40	18:05-20:35	8:10-10:40	15:10-18:10	14:30-17:30

El tiempo aproximado empleado en obtener la solución ha sido de 1,9 segundos.

#### 4.2.2.4 Routing4

Para minimizar los costes operativos se precisan operar 8 rotaciones compuestas por los siguientes vuelos:

Tabla 15: B757. Solución routing4

R	Vuelos						
	Día 1		Día 2			Día 3	
1	<b>125</b>		<b>105</b>			<b>138</b>	<b>118</b>
	JFK-SFO		SFO-JFK			JFK-BOS	BOS-JFK
	7:25-9:55		9:50-18:20			12:30-14:00	15:00-16:30
2	<b>105</b>		<b>138</b>	<b>118</b>	<b>125</b>		
	SFO-JFK		JFK-BOS	BOS-JFK	JFK-SFO		
	9:50-18:20		12:30-14:00	15:00-16:30	7:25-9:55		
3	<b>114</b>		<b>133</b>		<b>110</b>	<b>135</b>	
	MIA-JFK		JFK-ATL		ATL-JFK	JFK-MIA	
	14:30-17:30		18:05-20:35		8:10-10:40	15:10-18:10	
4	<b>133</b>		<b>110</b>	<b>135</b>	<b>114</b>		
	JFK-ATL		ATL-JFK	JFK-MIA	MIA-JFK		
	18:05-20:35		8:10-10:40	15:10-18:10	14:30-17:30		
5	<b>110</b>	<b>135</b>	<b>114</b>		<b>133</b>		
	ATL-JFK	JFK-MIA	MIA-JFK		JFK-ATL		
	8:10-10:40	15:10-18:10	14:30-17:30		18:05-20:35		
6	<b>113</b>	<b>136</b>	<b>113</b>		<b>131</b>	<b>111</b>	<b>136</b>
	MIA-JFK	JFK-MIA	MIA-JFK		JFK-ATL	ATL-JFK	JFK-MIA
	9:10-12:10	18:10-21:10	9:10-12:10		9:30-12:00	13:10-15:40	18:10-21:10
7	<b>131</b>	<b>111</b>	<b>131</b>	<b>111</b>	<b>136</b>	<b>113</b>	
	JFK-ATL	ATL-JFK	JFK-ATL	ATL-JFK	JFK-MIA	MIA-JFK	
	9:30-12:00	13:10-15:40	9:30-12:00	13:10-15:40	18:10-21:10	9:10-12:10	
8	<b>138</b>	<b>118</b>	<b>125</b>		<b>105</b>		
	JFK-BOS	BOS-JFK	JFK-SFO		SFO-JFK		
	12:30-14:00	15:00-16:30	7:25-9:55		9:50-18:20		

El tiempo aproximado empleado en obtener la solución ha sido de 1,7 segundos.

Por último, se comprueba que todas las rotaciones sean correctas. Para ello, nos fijamos en que el aeropuerto de salida del primer vuelo de la rotación coincida con el aeropuerto de llegada del último vuelo de la misma



rotación, y en que se respeten los tiempos de escala entre cada uno de los vuelos. Se verifica además que se realizan todos los vuelos y no se repite ninguno de ellos.

Como principal diferencia entre ambos problemas se puede destacar el tamaño de ellos. A pesar de que el número de vuelos no es excesivamente desigual, siendo la diferencia de 22 vuelos diarios, el número de variables del problema del avión B737 difiere bastante del número de variables del problema del avión B757.

Este fenómeno demuestra dos hechos: por un lado, el número de rotaciones obtenidos no depende estrictamente del número de vuelos, sino del número de posibilidades de secuenciación que existen entre ellos, aunque de forma general a mayor número de vuelos mayor será la posibilidad de formar distintas secuencias de vuelos. Por otra parte, se reafirma la idea expuesta acerca del crecimiento no lineal que sufre este tipo de problema a medida que se aumenta su tamaño, hecho que provoca grandes inversiones de tiempo en la obtención de soluciones factibles a medida que aumenta el tamaño de este, como se acaba de manifestar con estos dos problemas.

### 4.2.3 Cambios horarios

Una vez seleccionadas las rotaciones a operar por una compañía aérea para cubrir todas las rutas planeadas en función de los diferentes tipos de aeronaves incluidas en la flota, es muy útil realizar una revisión de los horarios de salidas y llegadas de los vuelos planificados, con el objetivo de mejorar los resultados obtenidos.

La revisión de horarios permite ajustar las horas de salidas y llegadas de ciertos vuelos con el objetivo de permitir la conexión entre ciertos trayectos que antes no eran posible, debido al insuficiente tiempo entre la llegada y salida de ellos. De este modo, ajustando estos horarios de aquellos vuelos que a priori puedan tener menos conexiones, es posible mejorar las rotaciones aumentando el número de trayectos y conexiones realizados, e incluso disminuir el número de aeronaves necesarias para efectuarlos.

Aunque parezca un proceso sencillo, se encuentra sujeto a diversos factores los cuales pueden reducir libertades a la hora de la nueva selección de horarios. La planificación horaria elaborada en un primer análisis se ejecuta otorgando los horarios más adecuados a cada tipo de vuelos, de modo que se maximice el factor de ocupación de las aeronaves utilizados y se optimice el precio de los billetes vendidos, por lo que un cambio significativo en el horario de salida y llegada puede suponer una pérdida considerable de beneficios. Por otro lado, existen bastantes aeropuertos con un volumen de tráfico elevado, donde es necesario la obtención de slots aeroportuarios o intervalos de tiempo específicos en los que poder operar un vuelo específico, siendo difícil la obtención de nuevos slots en otros horarios deseados debido a la falta de capacidad por parte del aeropuerto.

Por estos motivos, aunque efectuando varios cambios horarios se puedan obtener un mayor número de secuencias de vuelos en cada una de las rotaciones, hecho que podría parecer muy ventajoso para la operación de la aerolínea, es necesario discutir estos cambios con el resto de los departamentos de la compañía con el objetivo de obtener un equilibrio entre todos los factores que afectan a la operatividad de cada vuelo para obtener las soluciones finales óptimas.

Para demostrar cómo pueden variar las soluciones obtenidas al realizar un proceso de revisión de horarios, se va a analizar el ejemplo del avión B-757 de la aerolínea ficticia Ultimate Air, debido a que este caso permite obtener soluciones del problema en pocos segundos. Este proceso se podría realizar del mismo modo tanto para la otra aeronave de la flota de esta compañía como para cualquier otro caso real que quiera estudiarse.

Una forma de proceder a la hora de analizar los horarios de una serie de vuelos en este tipo de problemas sería observar aquellos vuelos potencialmente favorables a secuenciarse, es decir, vuelos en los que el aeropuerto de llegada de uno de ellos coincide con el aeropuerto de salida de otro de los vuelos, en los que el tiempo entre la llegada del primero de ellos y la salida del siguiente sea inferior al tiempo de escala mínimo necesario para la compañía aérea.

Un ejemplo de este hecho pueden ser los vuelos 125 y 105. En este caso, dado que el primer vuelo (125) llega 5 minutos después de la salida del segundo vuelo (105) es imposible poder realizar ambos con la misma aeronave.

**Tabla 16:** Horarios vuelos B757

Nº vuelo	Aeropuerto salida	Aeropuerto llegada	Horario salida	Horario salida (min)	Horario llegada (h)	Horario llegada	Duración (h)
----------	-------------------	--------------------	----------------	----------------------	---------------------	-----------------	--------------

			(h)			(min)	
125	1	3	7	25	9	55	5.5
110	5	1	8	10	10	40	2.5
113	6	1	9	10	12	10	3
131	1	5	9	30	12	0	2.5
105	3	1	9	50	18	20	5.5
138	1	7	12	30	14	0	1.5
111	5	1	13	10	15	40	2.5
114	6	1	14	30	17	30	3
118	7	1	15	0	16	30	1.5
135	1	6	15	10	18	10	3
133	1	5	18	5	20	35	2.5
136	1	6	18	10	21	10	3

Una posible solución sería adelantar el vuelo 125 o retrasar el vuelo 105, con el objetivo de obtener un margen mínimo de 45 minutos, que es igual al tiempo de escala de la compañía en estudio, para poder encadenar ambas rutas y realizar estos dos vuelos con el mismo avión.

A pesar de ser un proceso aparentemente sencillo, el gran número de vuelos que componen la operación de una aerolínea típica, así como el factor de que adelantar o retrasar un solo vuelo puede mejorar el enlace con otro, pero a la vez puede perjudicar otras combinaciones que anteriormente a este cambio eran posibles, reflejan la dificultad del caso. Además, a estos hechos se les debe unir los motivos externos como los explicados con anterioridad los cuales provocan el hecho de que no por tener un mayor número de vuelos en una rotación los resultados económicos finales sean mejores.

Dado la complejidad de analizar los resultados reales obtenidos al realizar diversos cambios horarios en la planificación inicial de los vuelos, así como debido a la falta de un mayor número de datos que permitan encontrar ciertas vías para obtener un análisis más concreto de este tipo de problemas, se va a realizar un análisis estadístico en función del número de rotaciones posibles o potenciales a formas parte de la solución final obtenidas con los archivos y funciones creados. Es decir, el siguiente análisis se basa en la idea de que a un mayor número de posibles rotaciones entre las cuales se encuentran las soluciones finales, mayores probabilidades existen de encontrar mejores resultados. Si bien esta suposición puede no ser real a ciencia cierta ya que depende de muchos factores y motivos externos a los cuales no se tiene acceso, nos permite realizar una primera aproximación al problema y obtener unos resultados objetivos en función de los datos disponibles, cumpliendo con la finalidad del trabajo.

El análisis elaborado consiste, de forma resumida, en elaborar cambios aleatorios en los horarios de los vuelos del avión B-757 del ejemplo de Ultimate Air, dentro unos márgenes designados, repitiendo el proceso durante un número de iteraciones adecuado de modo que permita obtener unos resultados aceptables. De este modo, para cada iteración, se ejecuta el archivo **rotaciones1** y se obtiene la matriz **R**, la cual nos informa acerca del número de rotaciones posibles o potenciales a formar parte de la solución final, así como del número máximo de vuelos que es posible realizar con los datos introducidos.

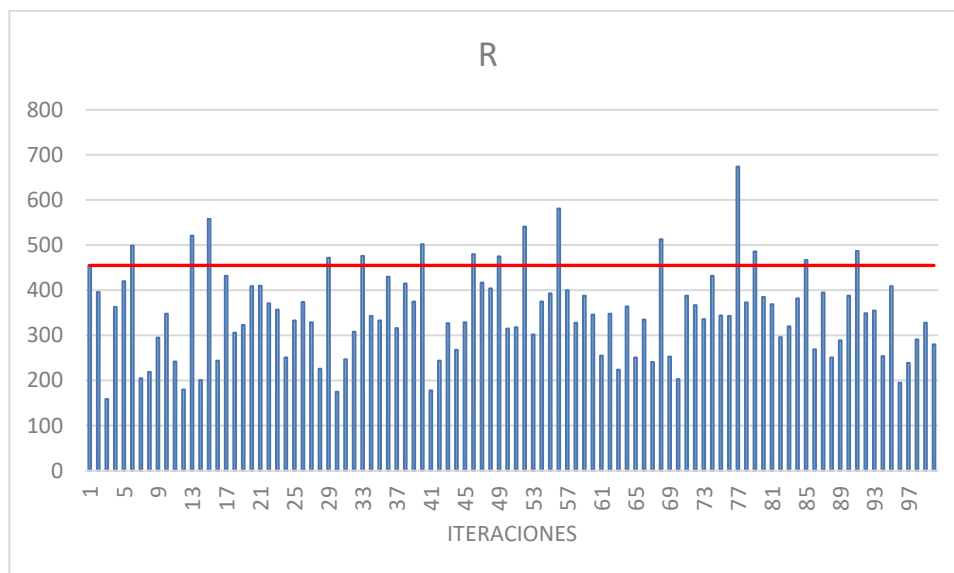
En concreto, se establecen nueve intervalos entre los cuales pueden variar los horarios de los vuelos operados

por esta aeronave: 45 min, 40 min, 35 min, 30 min, 25 min, 20 min, 15 min y 5 min. Por ejemplo, dentro del margen de 45 minutos, los horarios de salida y llegada pueden variar en un intervalo de  $\pm 45$  min, es decir, un vuelo cualquiera dentro de este margen puede adelantarse o retrasarse un máximo de 45 minutos. Además, la modificación del horario es independiente para cada uno de los vuelos, es decir, el primer vuelo puede adelantarse 15 min, mientras que el segundo vuelo no tiene por qué seguir este patrón, sino que puede atrasarse, por ejemplo, 20 min. Lo mismo sucede para cada uno de los márgenes o intervalos de tiempo a estudiar.

Por otro lado, en cada intervalo se toma una muestra de 100 datos. Es decir, se modifica aleatoriamente el horario de los diferentes vuelos y se ejecuta el archivo **rotaciones1** cien veces para cada intervalo, obteniendo en cada iteración el número de rotaciones posibles y el número máximo de vuelos posibles a realizar.

En los siguientes párrafos se detallan las diferentes comparaciones realizadas y los resultados obtenidos. Finalmente, se analizan estos datos obtenidos.

En primer lugar, para el intervalo de 45 minutos, se muestra el número de rotaciones posibles obtenidas para cada iteración en la **Figura 8**. Cada barra del gráfico representa este número de posibles rotaciones para cada una de las iteraciones ejecutadas, mientras que la línea roja horizontal representa el número de posibles rotaciones obtenido con los datos horarios de partida, es decir, sin ningún cambio horario, que tomaba el valor de 455.



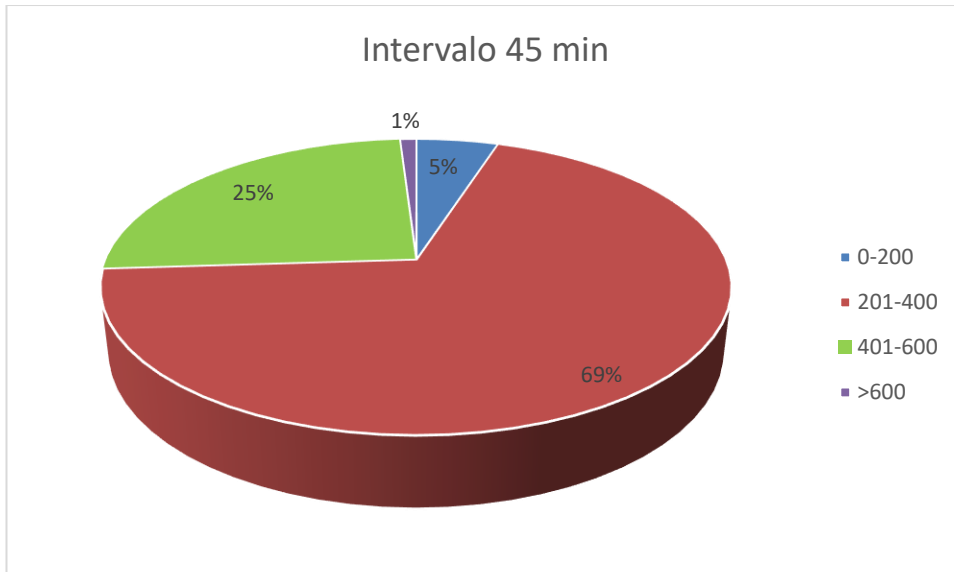
**Figura 8:** Rotaciones posibles cambios horarios intervalo [-45min, +45min]

De este modo, todas las iteraciones cuyos horarios mejoran aparentemente el problema, permitiendo obtener un mayor número de posibles rotaciones, aparecen por encima de esta línea roja horizontal. Mientras tanto, aquellas en las que se obtiene un menor número de rotaciones potenciales aparecen representadas por encima de esta línea.

El porcentaje de ocasiones en el que se mejoran los resultados, es decir, aquellas en las que los cambios horarios permiten obtener un igual o mayor número de posibles rotaciones, es del 16% en este caso. Por su parte, el mayor número de posibles rotaciones obtenidas es de 674.

Por otro lado, en cuanto al número máximo de vuelos obtenidos para cada una de las rotaciones, el 91% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 9% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación.

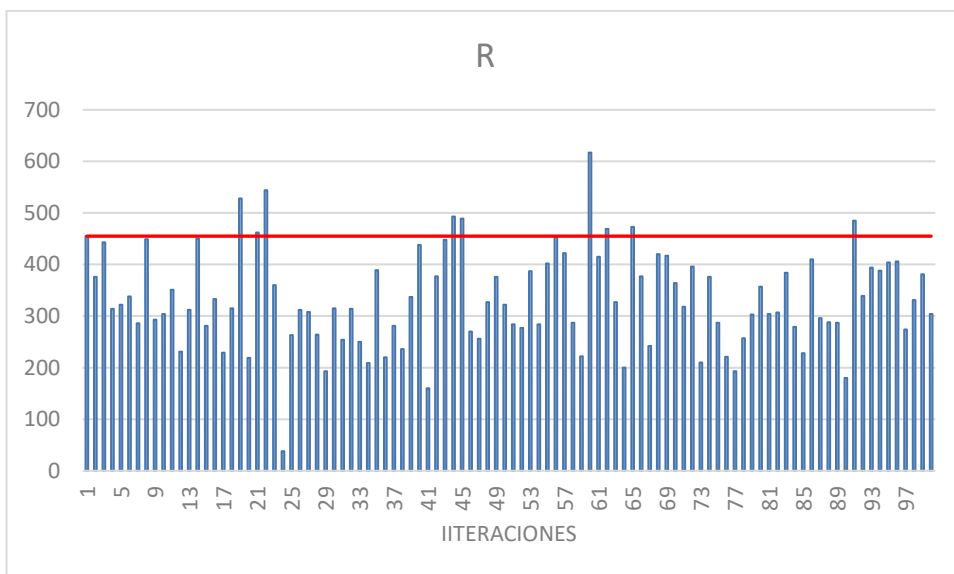
Con el objetivo de visualizar los datos por grupos, se ha realizado además una comparativa segmentando el número de posibles rotaciones obtenidas. De este modo, las rotaciones se han dividido en cuatro grupos diferentes, abarcando los siguientes valores: 0-200, 201-400, 401-600 y >600. Se muestra en la **Figura 9** el reparto alcanzado para los diferentes grupos en el intervalo de 45 minutos.



**Figura 9:** Segmentación número de rotaciones posibles intervalo [-45min, +45 min]

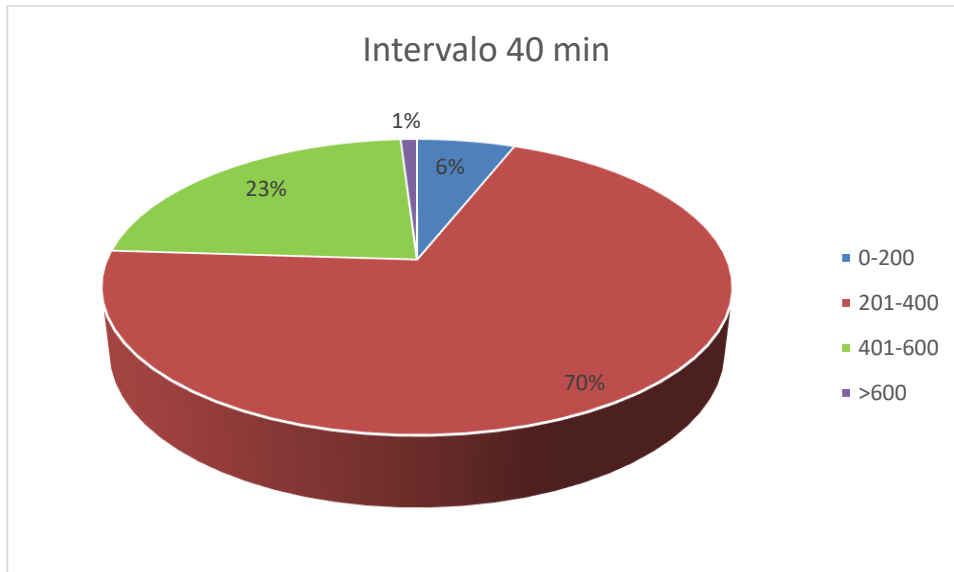
Se observa como el mayor volumen de posibles rotaciones se encuentra en el segmento entre las 201 y 400 rotaciones, seguido del segmento perteneciente a las 401-600 rotaciones. Tan solo el 1% de ocasiones se superan las 600, mientras que el 5% de veces obtenemos un número de rotaciones por debajo de 200.

Para el intervalo de 40 minutos, en la **Figura 10** se muestra el número de posibles rotaciones alcanzado. El porcentaje de ocasiones en el que se mejoran los resultados es del 11%, disminuyendo respecto al caso anterior. Por su parte, el mayor número de posibles rotaciones obtenidas es de 617, también inferior al intervalo anterior.



**Figura 10:** Rotaciones posibles cambios horarios intervalo [-40min, +40min]

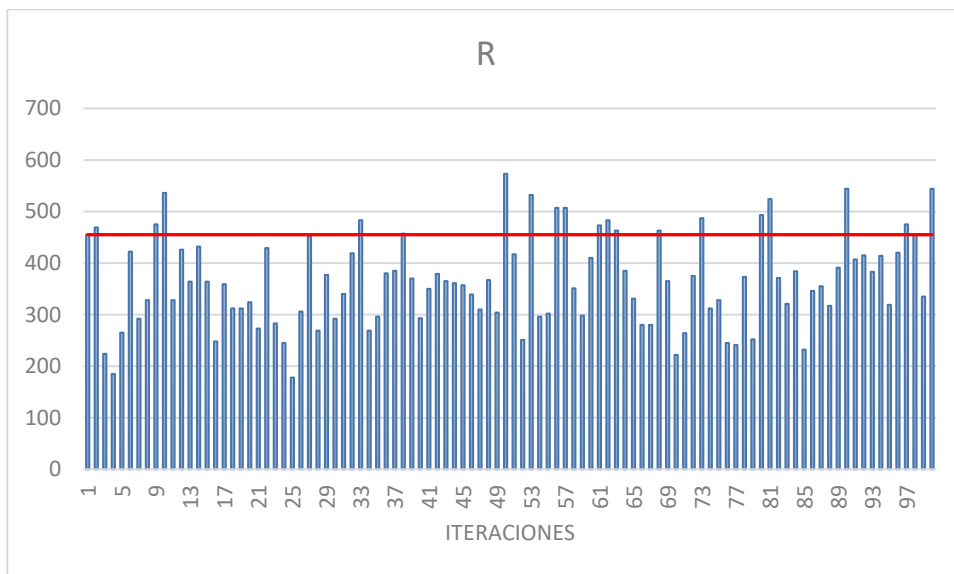
En cuanto al número máximo de vuelos obtenidos para cada una de las rotaciones, el 96% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 4% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación. Se observa una mejora en los porcentajes logrados respecto al intervalo anterior.



**Figura 11:** Segmentación número de rotaciones posibles intervalo [-40min, +40 min]

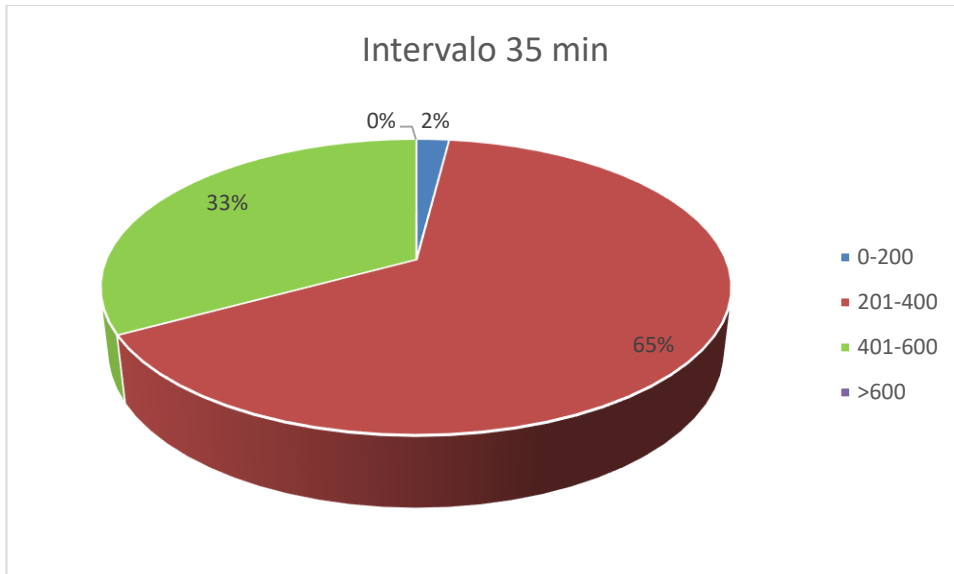
En este intervalo, los resultados obtenidos respecto al número de posibles rotaciones agrupadas por segmentos son casi idéntico al anterior.

Respecto al intervalo de 35 minutos, en la **Figura 12** se muestra el número de posibles rotaciones alcanzado. El porcentaje de ocasiones en el que se mejoran los resultados es del 21%, aumentando respecto a los casos anteriores. Por su parte, el mayor número de posibles rotaciones obtenidas es de 573, inferior al intervalo anterior.



**Figura 12:** Rotaciones posibles cambios horarios intervalo [-35min, +35min]

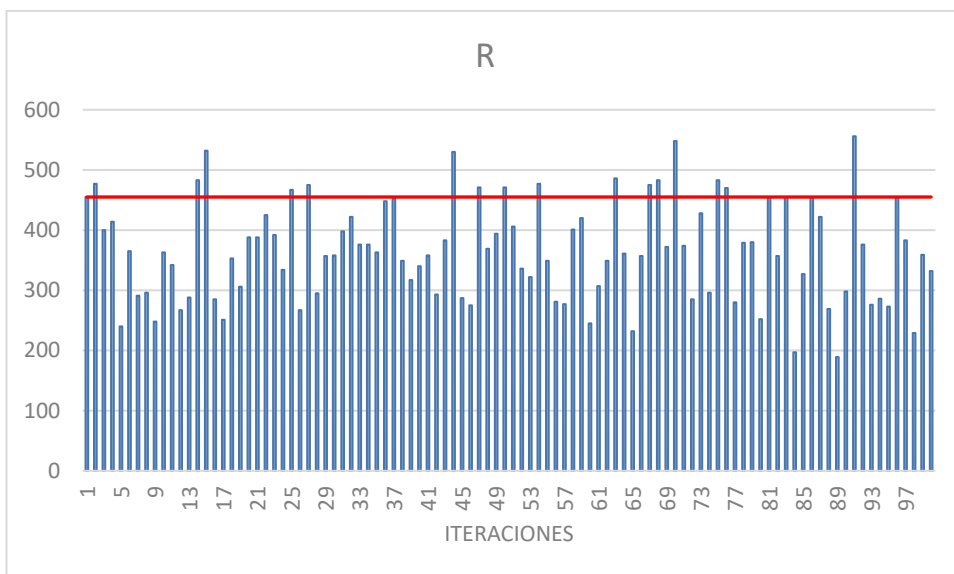
En relación con el número máximo de vuelos obtenidos para cada una de las rotaciones, el 98% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 2% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación. Se observa una mejora en los porcentajes logrados respecto al intervalo anterior.



**Figura 13:** Segmentación número de rotaciones posibles intervalo [-35min, +35 min]

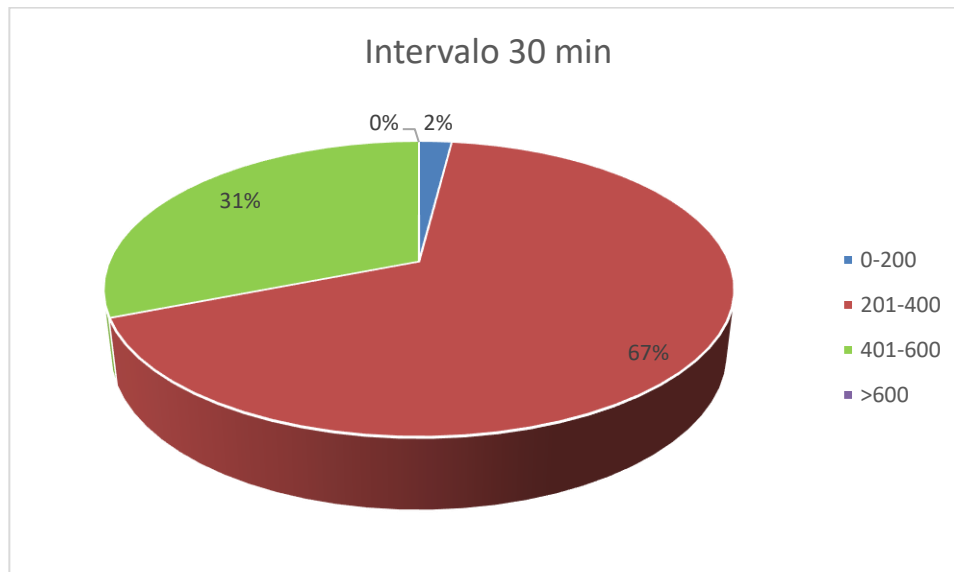
Por último, se muestran los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos. Se observa como aumentan los valores del grupo entre 401 y 600 rotaciones, mientras que el resto de ellos disminuye.

Para el intervalo de 30 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 21%, igual que en el caso anterior. Por su parte, el mayor número de posibles rotaciones obtenidas es de 556, inferior al intervalo anterior.



**Figura 14:** Rotaciones posibles cambios horarios intervalo [-30min, +30min]

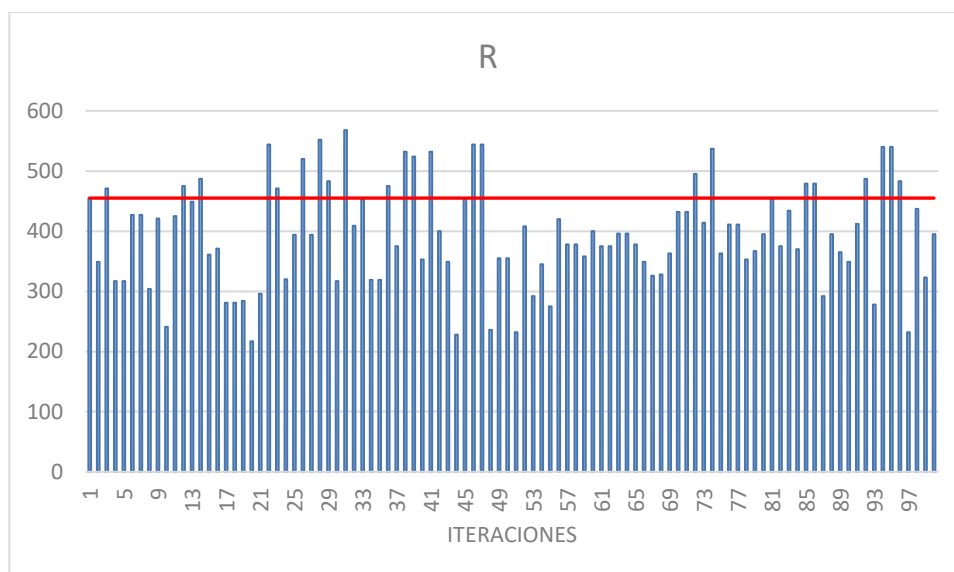
En cuanto al número máximo de vuelos obtenidos para cada una de las rotaciones, el 97% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 3% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación. Se observa una mínima caída en los porcentajes logrados respecto al intervalo anterior.



**Figura 15:** Segmentación número de rotaciones posibles intervalo [-30min, +30min]

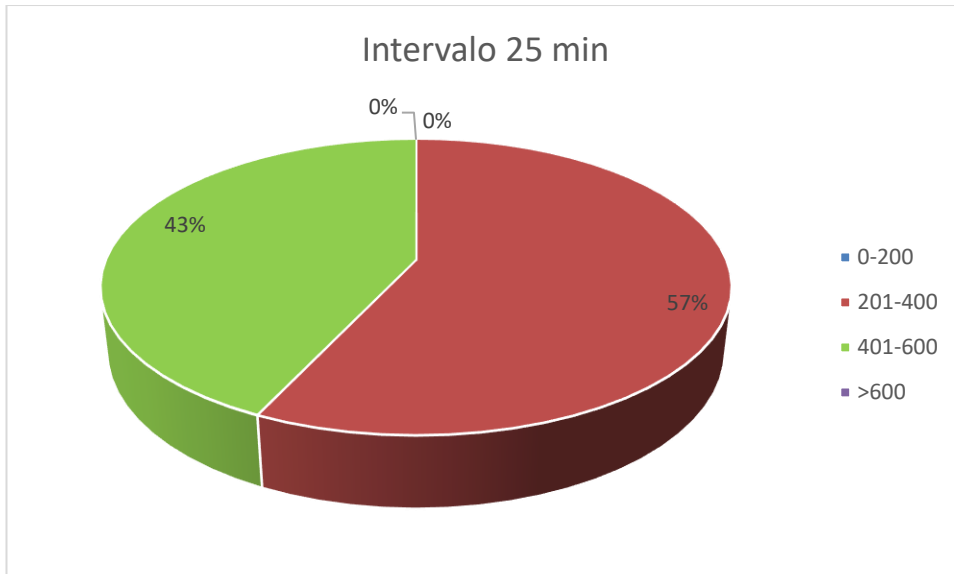
Respecto a los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como aumentan los valores del grupo entre 201 y 400 rotaciones, mientras que el resto de ellos disminuye.

En el siguiente intervalo, el de 25 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 26%, mejorando los datos de los anteriores márgenes o intervalos. Por su parte, el mayor número de posibles rotaciones obtenidas es de 568, variable que aumenta también respecto al intervalo anterior.



**Figura 16:** Rotaciones posibles cambios horarios intervalo [-25min, +25min]

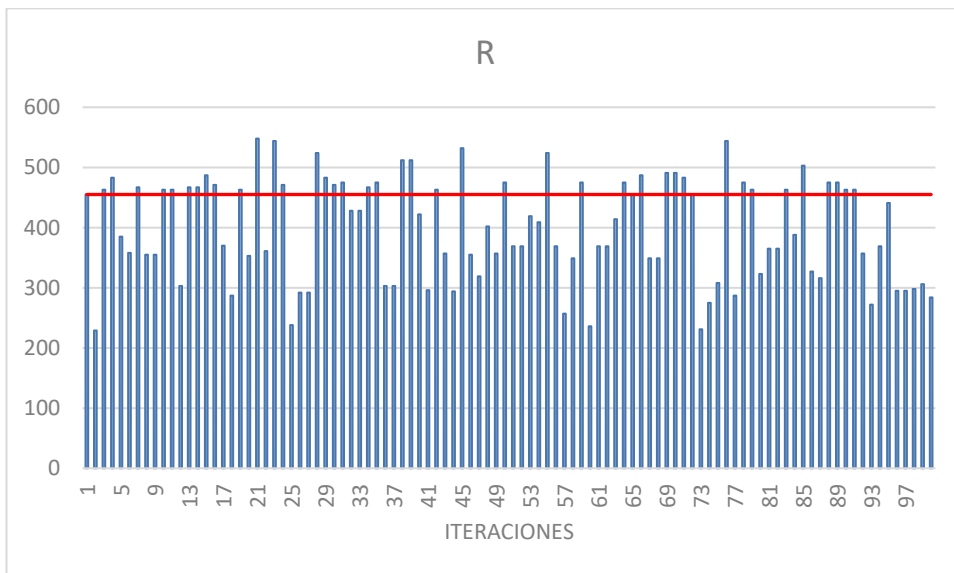
El 98% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 2% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación. Se observa una mínima mejora en los porcentajes logrados respecto al intervalo anterior.



**Figura 17:** Segmentación número de rotaciones posibles intervalo [-25min, +25min]

Analizando los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como aumentan los valores del grupo entre 401 y 600 rotaciones, mientras que el resto de ellos disminuye. En concreto, se aprecia una homogenización de los valores obtenidos, concentrándose todos ellos en el grueso entre las 200 y 600 posibles rotaciones. Por debajo y por encima de estos valores no se obtienen resultados.

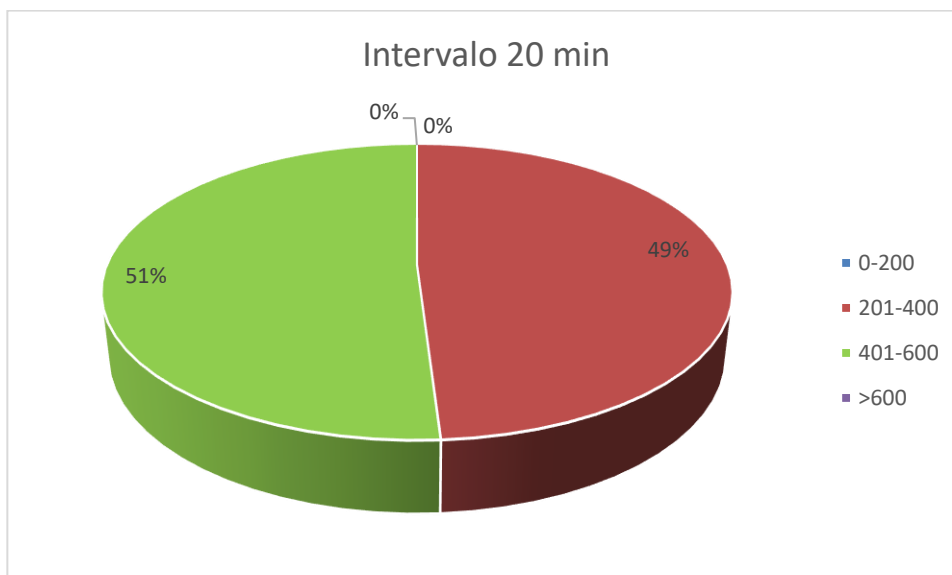
En el sexto intervalo, el de 20 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 43%, mejorando considerablemente los datos de los anteriores márgenes o intervalos, y alcanzando casi el doble del apartado anterior. Por su parte, el mayor número de posibles rotaciones obtenidas es de 548, disminuyendo en relación con los resultados de grupos anteriores.



**Figura 18:** Rotaciones posibles cambios horarios intervalo [-20min, +20min]

El 99% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente, mientras que el 1% de ocasiones ha disminuido a 9 vuelos máximo durante los tres días de la rotación. Se observa una mínima mejora en los porcentajes logrados respecto al intervalo anterior.

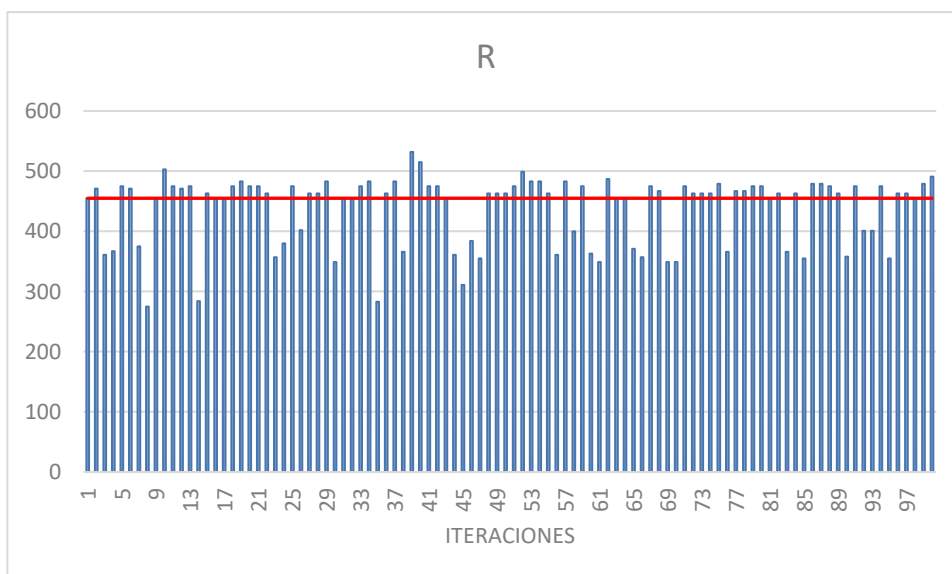




**Figura 19:** Segmentación número de rotaciones posibles intervalo [-20min, +20min]

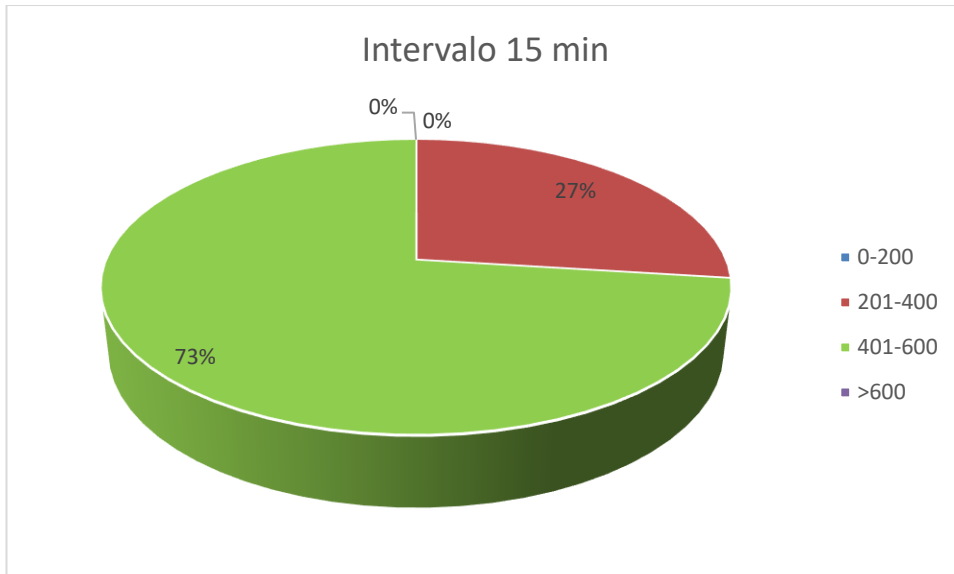
Respecto a los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como aumentan los valores del grupo entre 401 y 600 rotaciones, mientras que el resto de ellos disminuye. En concreto, al igual que en el caso anterior se aprecia una homogenización de los valores obtenidos, concentrándose todos ellos en el grueso entre las 200 y 600 posibles rotaciones. Por debajo y por encima de estos valores no se obtienen resultados. En este intervalo, se observa específicamente como el grupo de valores comprendidos entre las 401 y 600 rotaciones posibles, el cual iba aumentando a medida que disminuían los intervalos de tiempo, supera ya al grupo de 201-400 rotaciones.

Por otro lado, en el intervalo de 15 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 70%, mejorando considerablemente los datos de los anteriores márgenes o intervalos. Por su parte, el mayor número de posibles rotaciones obtenidas es de 532, disminuyendo en relación con los resultados de grupos anteriores.



**Figura 20:** Rotaciones posibles cambios horarios intervalo [-15min, +15min]

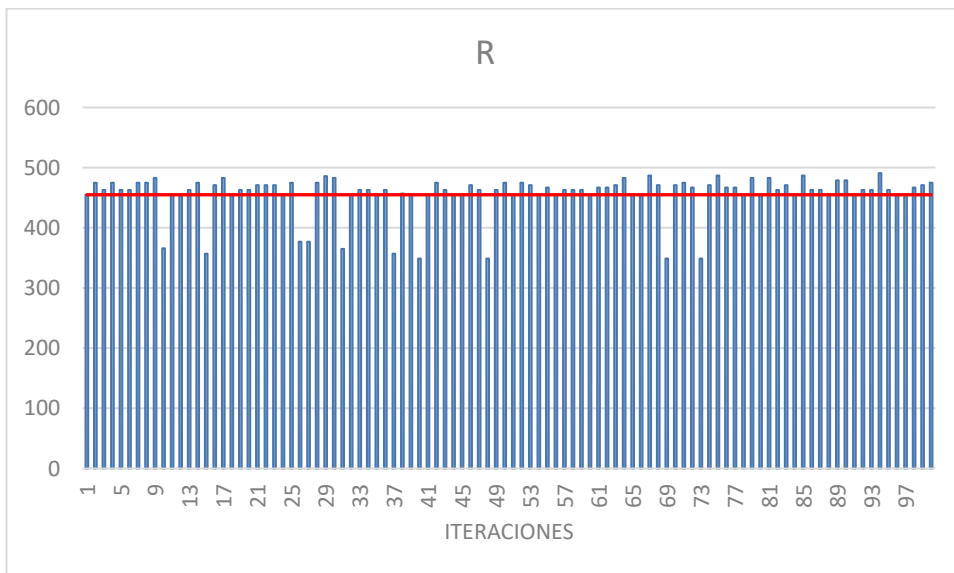
En este caso, el 100% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente. Se observa una mínima mejora en los porcentajes logrados respecto al intervalo anterior.



**Figura 21:** Segmentación número de rotaciones posibles intervalo [-15min, +15min]

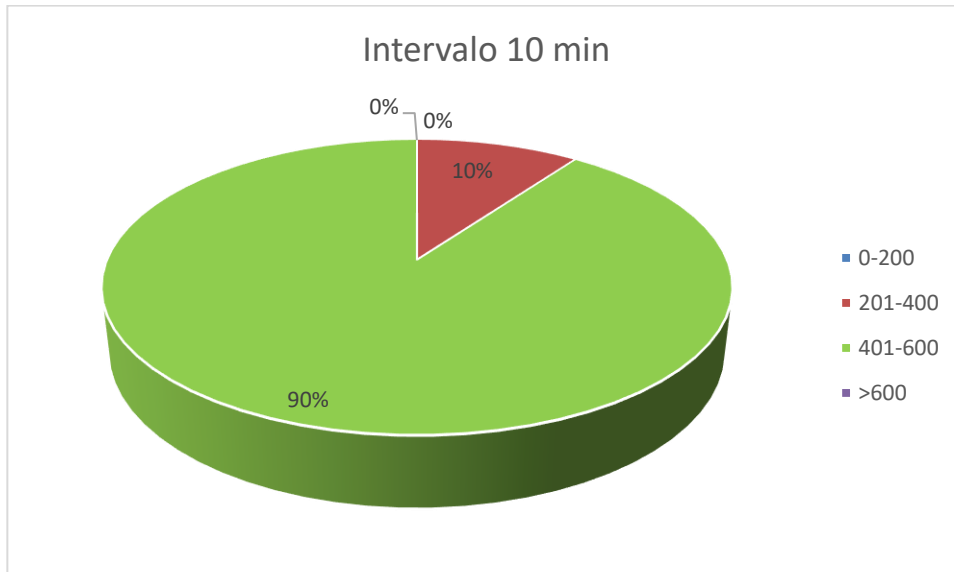
Respecto a los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como aumentan los valores del grupo entre 401 y 600 rotaciones. En concreto, se observa cómo se mantiene la tendencia al alza del grupo de 401-600 posibles rotaciones, el cual supone ya tres cuartas partes del total de resultados obtenidos.

Para el intervalo de 10 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 90%, mejorando considerablemente los datos de los anteriores márgenes o intervalos. Por su parte, el mayor número de posibles rotaciones obtenidas es de 491, disminuyendo en relación con los resultados de grupos anteriores.



**Figura 22:** Rotaciones posibles cambios horarios intervalo [-10min, +10min]

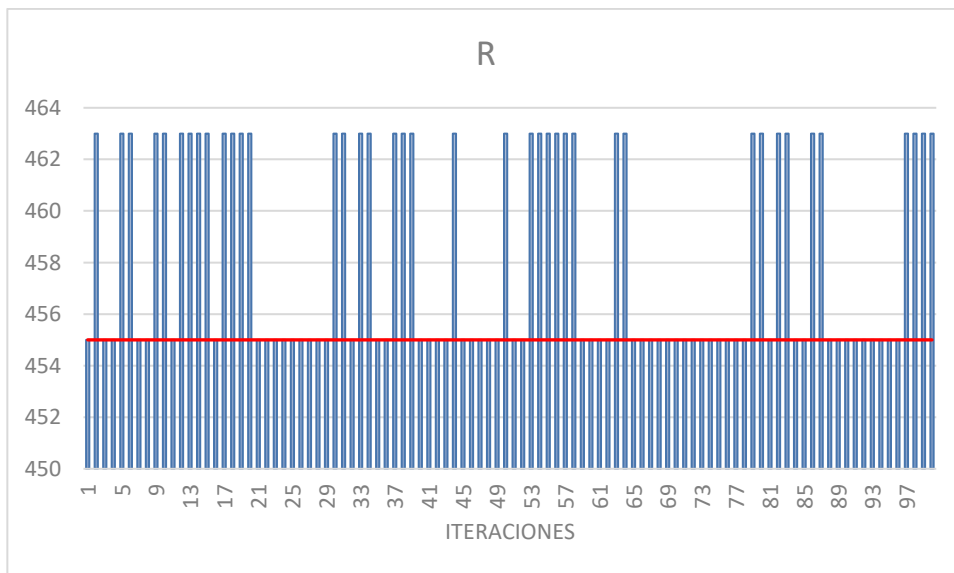
Al igual que en el caso anterior, el 100% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente.



**Figura 23:** Segmentación número de rotaciones posibles intervalo [-10min, +10min]

Por otro lado, en relación con los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como continúan aumentando los valores del grupo entre 401 y 600 rotaciones. En este caso, se observa cómo se tiende a una concentración entre los valores de este segmento o grupo, existiendo un pequeño margen del 10% en el que se concentran valores entre las 201 y 400 posibles rotaciones.

Por último, para el intervalo de 5 minutos, el porcentaje de ocasiones en el que se mejoran los resultados es del 100%. En concreto, esta variación tan pequeña de los cambios horarios provoca poder optar a un menor número de diferentes opciones, obteniéndose exclusivamente dos valores respecto al número de posibles rotaciones: 455 y 463.



**Figura 24:** Rotaciones posibles cambios horarios intervalo [-5min, +5min]

Al igual que en el caso anterior, el 100% de las ocasiones se mantienen los 12 vuelos máximos por rotación que existían inicialmente.

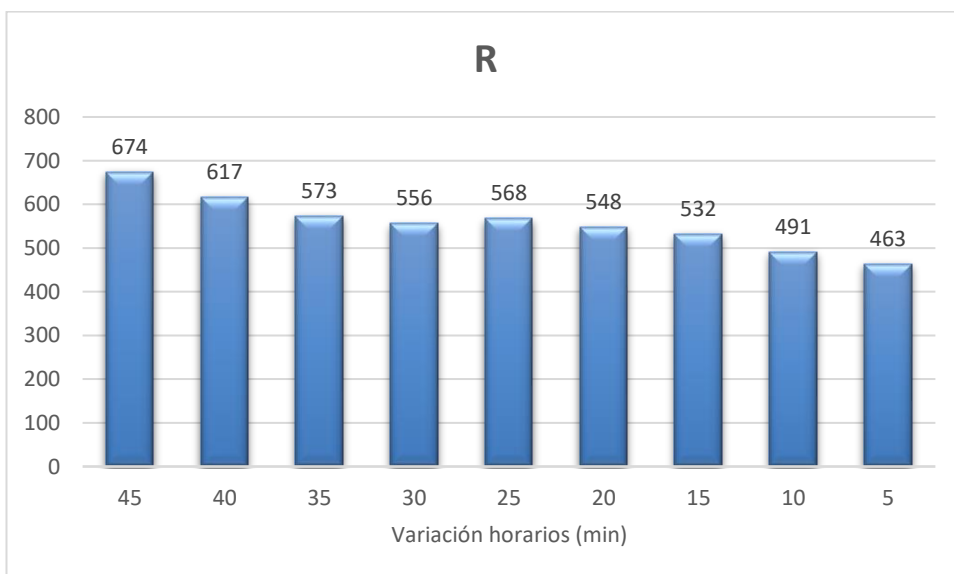


**Figura 25:** Segmentación número de rotaciones posibles intervalo [-5min, +5min]

Finalmente, en relación con los porcentajes obtenidos respecto al número de rotaciones agrupadas por segmentos, se observa como continúan aumentando los valores del grupo entre 401 y 600 rotaciones. En este caso, se observa cómo se tiende a una concentración entre los valores de este grupo, mientras que en el resto de los segmentos no se obtienen resultados. Por tanto, se pone de manifiesto la tendencia positiva de intervalos anteriores en el segmento 401-600.

De forma general, se pueden extraer tres conclusiones principales a partir de los datos analizados y resultados extraídos.

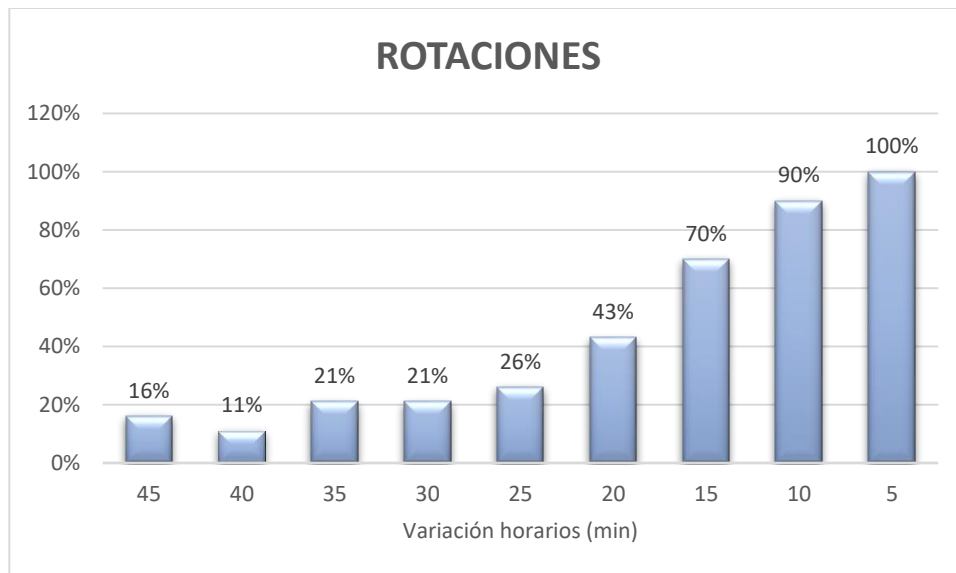
En primer lugar, se observa una tendencia descendente en el número máximo de posibles rotaciones obtenidas a medida que disminuyen los intervalos de tiempo. Este hecho se explica fácilmente, debido a que a medida que disminuyen los intervalos de tiempo, menores son las posibilidades de cambios horarios introducidos y, por tanto, menores son las probabilidades de obtener un mayor número de posibles rotaciones.



**Figura 26:** Máximo número de rotaciones posibles tras cambios horarios

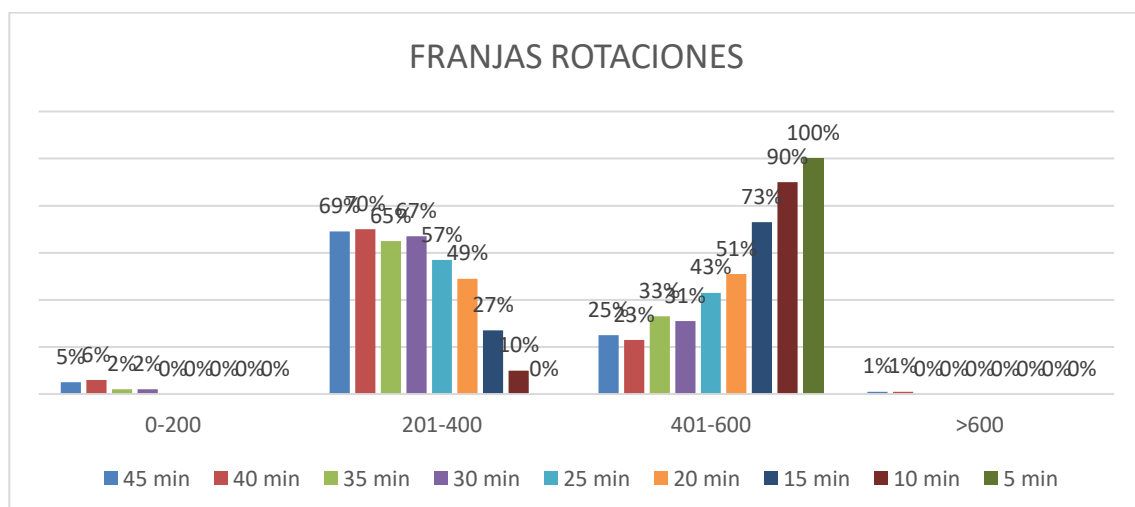
En segundo lugar, se observa una tendencia positiva y ascendente en el porcentaje de ocasiones en que se superan

o igualan las 455 posibles rotaciones obtenidas sin aplicar ningún cambio horario a medida que disminuyen los tiempos. La explicación reside en la idea anterior: cuanto menor es la posibilidad de introducir cambios horarios, es decir, cuanto más pequeño son estos intervalos, menor es la probabilidad de encontrar nuevas posibles rotaciones que puedan formar parte de la solución y, por tanto, menor será el grado de afección a los resultados iniciales.



**Figura 27:** Porcentajes de rotaciones posibles mayores o iguales que las iniciales

En último lugar, se aprecia una clara concentración de resultados en los valores obtenidos respecto al número de posibles rotaciones de cada iteración a medida que disminuyen los tiempos. La misma idea se aplica en este caso, es decir, a menos posibilidades de cambio, menores probabilidades de encontrar nuevas posibles rotaciones. Además, menores cambios de tiempo implican que los nuevos resultados obtenidos sean más igualados, hecho que se demuestra en la **Figura 28**. Se aprecia claramente una mayor dispersión de los resultados en los intervalos de tiempo mayores, mientras que a medida que disminuyen los resultados tienden a agruparse primero entre dos segmentos y, finalmente, en uno solo.



**Figura 28:** Segmentación número de rotaciones posibles general

Como se comentó con anterioridad, la revisión de horarios es un proceso que depende de multitud de factores,

por lo cual las conclusiones que se extraigan para este ejemplo no pueden ser extrapolables para otros casos. Es por ello por lo que, debido a la diferencia de horarios y al resto de factores y variables que afectan a los beneficios en la operación aérea de una aerolínea, en este tipo de procesos no pueden seguirse pautas generales, siendo más factible analizar cada caso individualmente.

## 5 APLICACIÓN PRÁCTICA

En el capítulo anterior se presentó el programa elaborado para la resolución del problema del ‘Aircraft Routing’, el cual permite obtener cuatro soluciones diversas según el criterio de optimización elegido. Para comprobar el funcionamiento del programa desarrollado, éste se puso en práctica resolviendo un ejemplo ficticio del libro ‘Airlines Operations and Scheduling’ [5]. Una vez comprobado el buen funcionamiento del programa, en este apartado se pretende ponerlo en práctica analizando la operativa de una compañía aérea real.

Cabe destacar que la mayoría de las aerolíneas cuentan con programas informáticos específicos y avanzados para la optimización de este tipo de cuestiones, permitiéndoles obtener soluciones factibles a problemas con un número más elevado de variables de los que se resuelven en este trabajo. Por este motivo, dado que el objetivo principal de la creación de este programa es meramente académico, se ha decidido buscar diferentes aerolíneas con un volumen de operación pequeño, con el objetivo de poder resolver el problema del ‘Aircraft Routing’ en intervalos de tiempos admisibles.

### 5.1 Descripción general

Entre las diferentes aerolíneas buscadas, se ha optado por analizar el caso de **Kulula Airlines**, debido al volumen de vuelos que realiza diariamente y al uso de un solo tipo de avión en su flota, hecho que nos permite resolver el problema únicamente para una aeronave.

Kulula es una compañía aérea sudafricana de bajo coste, fundada el 6 de julio de 2001, la cual opera rutas domésticas en las principales ciudades del país. El aeropuerto base de la compañía es el Aeropuerto Internacional O. R. Tambo, localizado en la ciudad de Johannesburgo [13].

Kulula ofrece una red de vuelos que conecta las principales ciudades del país: Johannesburgo, Ciudad del Cabo, Durban y George. Además, dado que Kulula es la aerolínea de bajo coste de la compañía ‘Comair Limited’, otra aerolínea basada en Sudáfrica que ofrece vuelos domésticos como franquicia de la aerolínea británica ‘British Airways’, y tiene acuerdos de código compartido con ‘Kenya Airways’, es posible ampliar esta red de ciudades y viajar a otros países africanos. En las **Figuras 29 y 30** se aprecia la oferta de destinos ofrecidos [14].

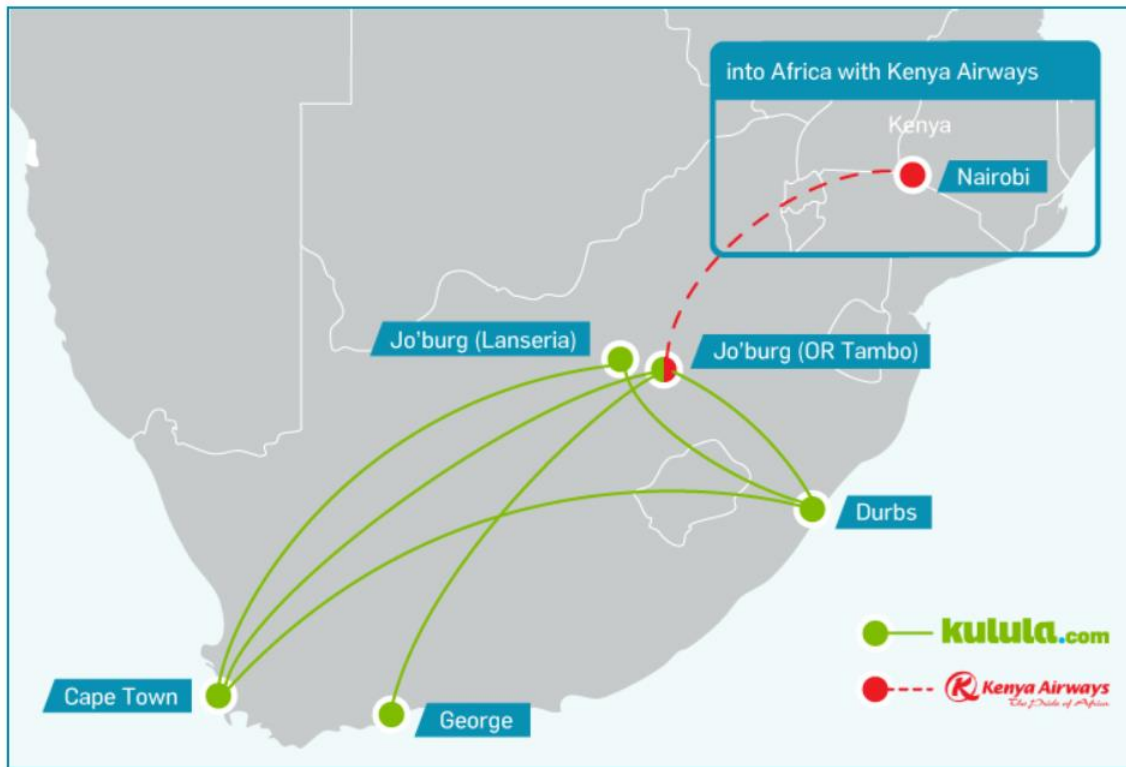


Figura 29: Rutas operadas por Kulula

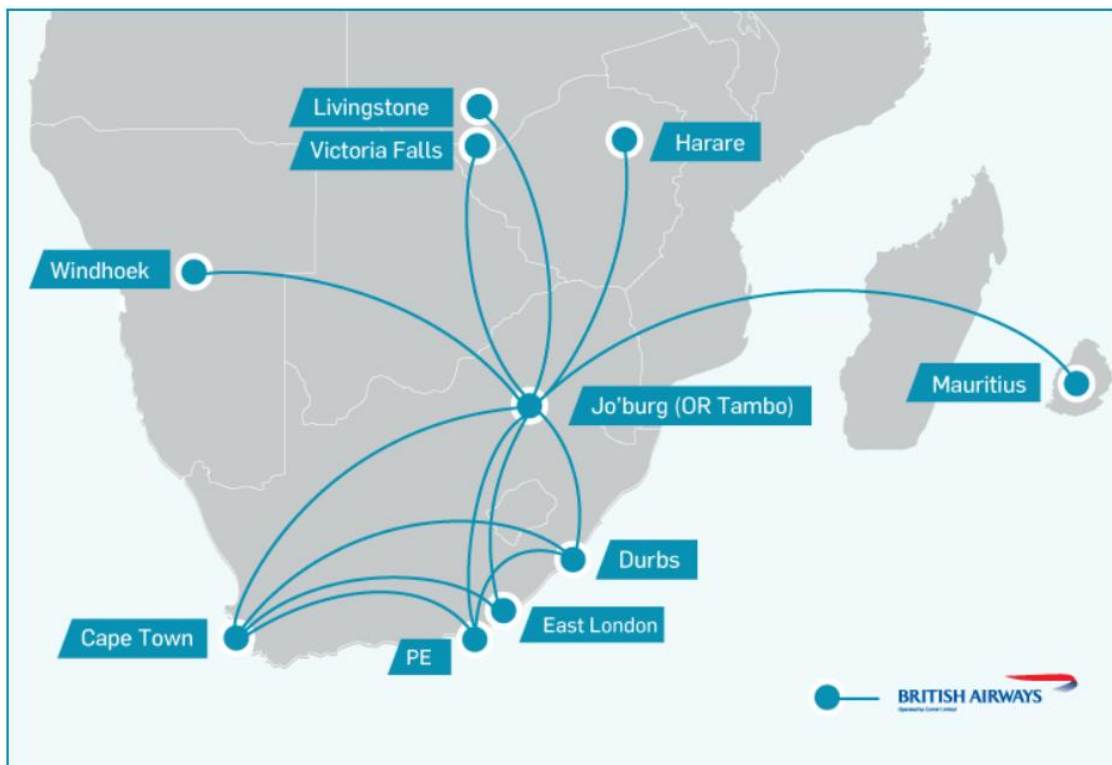


Figura 30: Rutas operadas por Kulula a través de British Airways

En este caso, para la resolución del problema del 'Aircraft Routing' se van a tomar solo los vuelos operados en exclusividad por Kulula. Estos vuelos operan entre los siguientes aeropuertos:



- Jo'burg - O R Tambo International (Johannesburgo)
- Lanseria - Lanseria Airport (norte de Johannesburgo)
- Durbs - King Shaka International (Durban)
- George - George Airport (George)
- Cape Town - Cape Town International (Ciudad del Cabo)

Para la realización de estos trayectos, la aerolínea sudafricana cuenta con una flota formada por 9 aviones Boeing 737-800 (B737-800).

## 5.2 Obtención de datos

El primer paso antes de poder abordar el problema es conocer la programación de vuelos de la compañía. En este caso, vamos a analizar el problema del 'Aircraft Routing' para la aerolínea Kulula, siendo las rotaciones de aviones válidas para 3 días, tal y como se diseñó en el programa. En concreto, los tres días elegidos para resolver el problema son el 6, 7 y 8 del mes de agosto de 2021, coincidiendo con el primer fin de semana de ese mes.

De este modo, es necesario obtener la programación de vuelos de la aerolínea para estos tres días especificados. Para ello, utilizamos la página web de la compañía [15], a través de la cual es posible conocer todos los vuelos realizados en los días seleccionados, así como los horarios de salida y llegada de cada uno de ellos. Las horas de salida y llegada de todos los vuelos obtenidos corresponden al mismo huso horario (GTM+2).

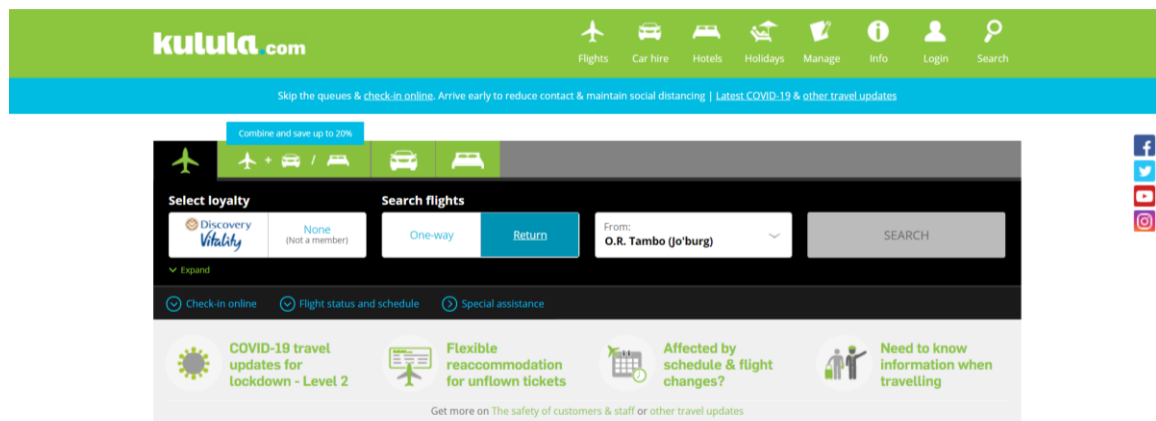


Figura 31: Web oficial Kulula Airlines

Debido a los efectos de la pandemia por Covid-19, la operación de la aerolínea se canceló en julio de 2020. A principios de diciembre del mismo año se inició la recuperación de la operatividad, aunque todavía no se ha alcanzado los niveles de años anteriores. Es por ello por lo que algunas rutas no se encuentran todavía operativas, siendo operadas actualmente las que aparecen en la **Figura 32**.



**Figura 32:** Rutas operadas por Kulula actualmente

Para la introducción de los datos en el archivo Excel utilizado en el programa elaborado, es necesario asignar un número a cada uno de los aeropuertos que forman la red de destinos de la compañía. En este caso el orden introducido es el siguiente:

**Tabla 17:** Aeropuertos Kulula Airlines

Nº Aeropuerto	Código IATA	Nombre Aeropuerto
1	JNB	O R Tambo International
2	DUR	King Shaka International
3	GRJ	George Airport
4	CPT	Cape Town International

El archivo *Excel* con la programación de vuelos y los datos requeridos para analizar los cuatro diferentes casos introducidos en el programa puede consultarse en el **Anexo B**. Se realizan un total de 62 vuelos para los tres días seleccionados: 24 vuelos el 06/08/2021, 19 vuelos el 07/08/2021 y 19 vuelos el 08/08/2021.

Por otro lado, es necesario conocer el aeropuerto base donde se realiza el mantenimiento y el tiempo mínimo de escala. Dado que el aeropuerto base de la compañía es O R Tambo International, Johannesburgo, es aquí donde se realiza el mantenimiento tipo A o mantenimiento en rampa. En cuanto el tiempo de escala, dado que la aerolínea realiza vuelos domésticos de corto radio cuya duración varía entre 1 y 3 horas, se opta por fijarlo en 40 minutos.

En cuanto al número de aviones disponibles, la aerolínea dispone de 9 aviones Boeing 737-800.

### 5.3 Resultados

Una vez se conocen todos los datos necesarios para poder abordar el problema, se ejecutan los cuatro archivos elaborados con *Matlab* para la resolución del problema. Como datos comunes se han introducido, a parte de la programación de vuelos de cada uno de los días seleccionados, el aeropuerto de Johannesburgo como aeropuerto de mantenimiento, un tiempo de escala mínimo de 40 minutos y un máximo de 9 aviones para operar todas las rotaciones.

En este caso, al ejecutar cada uno de los cuatro archivos elaborados no se encuentra ninguna solución posible. Este hecho no había aparecido en ninguno de los ejemplos anterior, si bien puede llegar a ocurrir con facilidad cuando las restricciones impuestas son demasiado estrictas.

De forma general, cuando no se encuentran soluciones factibles para la resolución de este tipo de problema, se recomienda relajar algunos de los requerimientos o restricciones. Posibles actuaciones por realizar son ampliar el tiempo mínimo de escala, ampliar el número de aviones disponibles o introducir otro aeropuerto base de mantenimiento. Todas estas modificaciones de los algoritmos son muy sencillas, otorgándole al usuario libertad a la hora de escoger diferentes alternativas para resolver el problema.

En cuanto al problema que se plantea en este capítulo, se decide en primer lugar relajar la restricción del número de aviones máximos. Para ello, se elimina esta restricción del código, con el objetivo poder encontrar alguna solución factible que no se encuentre condicionada por un número concreto de aeronaves a utilizar. Sin embargo, a pesar de eliminar esta restricción, el resultado obtenido sigue siendo negativo, ya que no es posible obtener ninguna rotación posible. De esta forma, queda claro que el problema no reside en esta restricción, por lo que se vuelve a introducir en el programa.

A continuación, se decide relajar el problema eliminando una particularidad de las rotaciones: los ciclos cerrados. Tanto en el código inicial elaborado como en los ejemplos resueltos en el **Capítulo 4.2**, todas las rotaciones tenían la particularidad de formar ciclos cerrados, es decir, el aeropuerto desde el que partía el primer vuelo de una rotación era igual al aeropuerto donde finalizaba el último vuelo de esta misma rotación. Este condicionante permite obtener rotaciones que pueden repetirse indefinidamente a lo largo del tiempo, encajando perfectamente. Sin embargo, introduce mayores restricciones al no permitir aeropuertos distintos en la salida y llegada del primer y último vuelo, hecho que conlleva a la obtención de un menor número de rotaciones posibles o rotaciones potenciales a formar parte de la solución final. Por ello, con el objetivo de aumentar este conjunto de posibles rotaciones finales y poder hallar una solución factible del problema, se elimina esta condición de la función **routing1**.

Al ejecutar el programa, se obtienen un total de 310.950 rotaciones posibles o variables, entre las cuales se eligen las más adecuadas para cada uno de los cuatro casos planteados según el objetivo de optimización, y cumpliendo con todas las restricciones impuestas. En concreto, se obtienen 63 restricciones.

Por otro lado, el máximo de vuelos que pueden llegar a realizarse en un mismo día y una misma rotación, de modo que se cumplan los requerimientos de mantenimiento y tiempo de escala, es de 6 vuelos.

A continuación, se presentan los resultados para cada uno de los diferentes casos o archivos elaborados. En cada uno de ellos se muestran las rotaciones elegidas, en las cuales aparecen los vuelos que se realizan en cada uno de los días que forman la rotación, así como el origen-destino y horarios de salida y llegada. Además, en cada uno de los apartados se muestra el valor que toma la función objetivo para cada una de las rotaciones finalmente seleccionadas.

#### 5.3.1 Maximizar las posibilidades de mantenimiento

Se utiliza el archivo **routing1** para obtener la solución del problema optimizando las posibilidades de mantenimiento, es decir, maximizando el número de días que se realiza una estancia nocturna en el aeropuerto JNB. Se obtienen un total de 9 rotaciones, por lo que deberán ser utilizados todos los aviones de la compañía.

Tabla 18: Kulula Airlines. Solución routing1

R	Vuelos									
	06/08/2021			07/08/2021			08/08/2021			
1	<b>6</b>			<b>33</b>		<b>29</b>		<b>51</b>		
	JNB-CPT			CPT-JNB		JNB-CPT		CPT-JNB		
	8:50-11:00			10:50-12:50		13:40-15:50		17:45-19:45		
2	<b>5</b>	<b>20</b>	<b>27</b>	<b>34</b>	<b>25</b>	<b>59</b>	<b>42</b>	<b>48</b>	<b>40</b>	<b>62</b>
	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB
	7:55-10:05	17:45-19:45	7:55-10:05	11:45-13:45	14:35-15:40	19:20-20:25	7:50-10:05	10:50-12:50	14:35-15:40	19:20-20:25
3	<b>21</b>		<b>8</b>	<b>35</b>			<b>43</b>	<b>49</b>	<b>41</b>	
	GRJ-JNB		JNB-CPT	CPT-JNB			JNB-CPT	CPT-JNB	JNB-DUR	
	9:15-10:55		14:50-17:00	14:40-16:40			8:50-11:00	14:40-16:40	17:30-18:35	
4	<b>1</b>	<b>14</b>	<b>11</b>	<b>38</b>			<b>46</b>			
	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB			JNB-GRJ			
	5:55-7:00	13:30-14:35	15:25-17:20	12:20-14:00			8:45-11:40			
5	<b>9</b>	<b>22</b>	<b>4</b>	<b>32</b>	<b>30</b>	<b>37</b>	<b>39</b>			
	JNB-GRJ	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR			
	6:45-8:40	12:20-14:00	17:30-18:35	7:50-8:55	14:50-17:00	17:45-19:45	11:45-12:50			
6	<b>12</b>	<b>10</b>	<b>23</b>	<b>24</b>	<b>58</b>	<b>55</b>	<b>61</b>	<b>56</b>		
	DUR-JNB	JNB-GRJ	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-DUR		
	6:00-7:05	8:45-11:40	18:00-19:40	11:45-12:50	16:20-17:25	19:25-20:30	16:20-17:25	19:25-20:30		
7	<b>17</b>	<b>7</b>	<b>19</b>	<b>28</b>		<b>36</b>	<b>44</b>	<b>50</b>		
	CPT-JNB	JNB-CPT	CPT-JNB	JNB-CPT		CPT-JNB	JNB-CPT	CPT-JNB		
	10:50-12:50	13:40-15:50	16:35-18:35	11:50-14:00		16:35-18:35	11:40-14:00	16:35-18:35		
8	<b>18</b>	<b>3</b>	<b>16</b>	<b>31</b>			<b>52</b>	<b>45</b>		
	CPT-JNB	JNB-DUR	DUR-JNB	JNB-GRJ			GRJ-JNB	JNB-CPT		

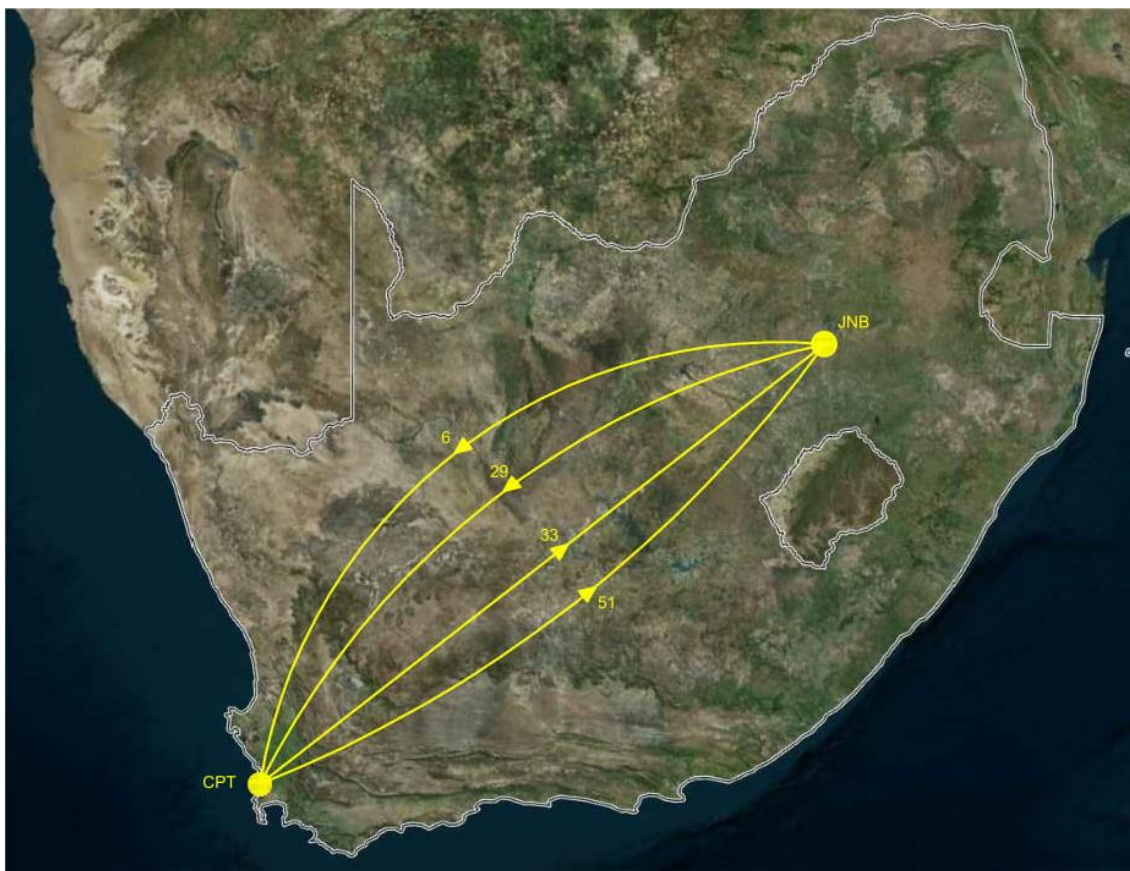
	11:45-13:45	14:35-15:40	19:20-20:25	8:45-11:40		12:20-14:00	14:50-17:00		
9	<b>13</b>	<b>2</b>	<b>15</b>	<b>54</b>	<b>57</b>	<b>26</b>	<b>60</b>	<b>47</b>	<b>53</b>
	DUR-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB
	7:50-8:55	11:45-12:50	16:20-17:25	19:25-20:30	13:30-14:35	17:30-18:35	13:30-14:35	15:25-17:20	18:00-19:40

Por otra parte, se muestra también el número de estancias nocturnas en el aeropuerto base de mantenimiento (JNB) de cada una de las rotaciones elegidas, pudiendo tomar los valores 1, 2 ó 3.

**Tabla 19:** Valores función objetivo de las rotaciones elegidas en routing1

Rotaciones	1	2	3	4	5	6	7	8	9
Nº estancias nocturnas en JNB	1	2	1	1	1	1	3	1	1

Se muestra también a modo de ejemplo una imagen donde se refleja la secuencia de vuelos que se realiza en la primera de las rotaciones seleccionadas.



**Figura 33:** Rutas realizadas en la primera rotación de routing1

### 5.3.2 Maximizar el número de conexiones de vuelos

Se utiliza el archivo **routing2** para obtener la solución del problema optimizando el número de conexiones de vuelos, es decir, maximizando el número de vuelos que se realiza en cada rotación. Se obtienen un total de 9 rotaciones, por lo que deberán ser utilizados todos los aviones de la compañía.

**Tabla 20:** Kulula Airlines. Solución routing2

R	Vuelos										
	06/08/2021			07/08/2021				08/08/2021			
1	<b>9</b>			<b>38</b>		<b>30</b>		<b>50</b>			
	JNB-GRJ			GRJ-JNB		JNB-CPT		CPT-JNB			
	6:45-8:40			12:20-14:00		14:50-17:00		16:35-18:35			
2	<b>19</b>			<b>25</b>				<b>62</b>			
	CPT-JNB			JNB-DUR				DUR-JNB			
	16:35-18:35			14:35-15:40				19:20-20:25			
3	<b>20</b>			<b>31</b>				<b>53</b>			
	CPT-JNB			JNB-GRJ				GRJ-JNB			
	17:45-19:45			8:45-11:40				18:00-19:40			
4	<b>13</b>		<b>7</b>	<b>34</b>	<b>26</b>	<b>59</b>	<b>46</b>	<b>52</b>	<b>45</b>	<b>51</b>	
	DUR-JNB		JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB	JNB-CPT	CPT-JNB	
	7:50-8:55		13:40-15:50	11:45-13:45	17:30-18:35	19:20-20:25	8:45-11:40	12:20-14:00	14:50-17:00	17:45-19:45	
5	<b>6</b>	<b>18</b>	<b>4</b>	<b>32</b>	<b>28</b>	<b>37</b>	<b>43</b>	<b>49</b>	<b>41</b>		
	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR		
	8:50-11:00	11:45-13:45	17:30-18:35	7:50-8:55	11:50-14:00	17:45-19:45	8:50-11:00	14:40-16:40	17:30-18:35		
6	<b>10</b>	<b>22</b>	<b>8</b>	<b>35</b>				<b>44</b>			
	JNB-GRJ	GRJ-JNB	JNB-CPT	CPT-JNB				JNB-CPT			
	8:45-11:40	12:20-14:00	14:50-17:00	14:40-16:40				11:40-14:00			
7	<b>1</b>	<b>14</b>	<b>11</b>	<b>23</b>	<b>27</b>	<b>33</b>	<b>29</b>	<b>36</b>	<b>42</b>	<b>48</b>	<b>40</b>
	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR
	5:55-7:00	13:30-14:35	15:25-17:20	18:00-19:40	7:55-10:05	10:50-12:50	13:40-15:50	16:35-18:35	7:50-10:05	10:50-12:50	14:35-15:40

8	<b>21</b>	<b>2</b>	<b>15</b>	<b>54</b>	<b>57</b>			<b>39</b>	<b>61</b>	<b>56</b>
	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB			JNB-DUR	DUR-JNB	JNB-DUR
	9:15-10:55	11:45-12:50	16:20-17:25	19:25-20:30	13:30-14:35			11:45-12:50	16:20-17:25	19:25-20:30
9	<b>12</b>	<b>5</b>	<b>17</b>	<b>3</b>	<b>16</b>	<b>24</b>	<b>58</b>	<b>55</b>	<b>60</b>	<b>47</b>
	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-GRJ
	6:00-7:05	7:55-10:05	10:50-12:50	14:35-15:40	19:20-20:25	11:45-12:50	16:20-17:25	19:25-20:30	13:30-14:35	15:25-17:20

Por otra parte, se muestra también el número de vuelos que se realiza en cada una de las rotaciones elegidas, pudiendo variar entre los valores 3 y 18, ambos inclusive. En otras palabras, se muestra el valor de la función objetivo para cada una de las rotaciones finalmente seleccionadas.

**Tabla 21:** Valores función objetivo de las rotaciones elegidas en routing2

Rotaciones	1	2	3	4	5	6	7	8	9
Nº vuelos	4	3	3	9	9	5	11	8	10

Se muestra también a modo de ejemplo una imagen donde se refleja la secuencia de vuelos que se realiza en la primera de las rotaciones seleccionadas.

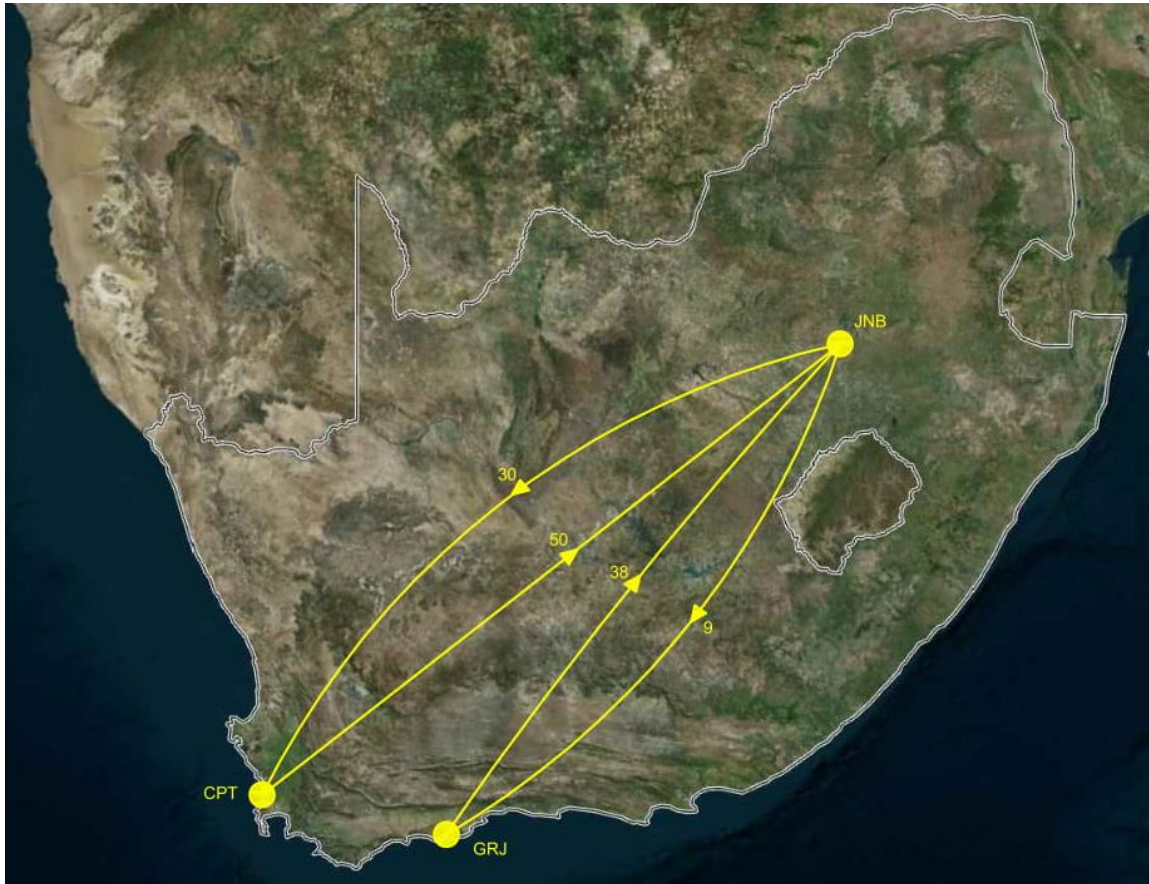


Figura 34: Rutas realizadas en la primera rotación de routing2

### 5.3.3 Minimizar desviación respecto a horas de referencia

Se utiliza el archivo **routing3** para obtener la solución del problema optimizando la desviación respecto al número de horas de referencia introducido (10 horas), es decir, minimizando la diferencia entre ésta y el número de horas totales de los vuelos de cada rotación. Se obtienen un total de 9 rotaciones, por lo que deberán ser utilizados todos los aviones de la compañía.

Tabla 22: Kulula Airlines. Solución routing3

R	Vuelos							
	06/08/2021		07/08/2021			08/08/2021		
1	<b>6</b>		<b>35</b>			<b>46</b>	<b>52</b>	<b>45</b>
	JNB-CPT		CPT-JNB			JNB-GRJ	GRJ-JNB	JNB-CPT
	8:50-11:00		14:40-16:40			8:45-11:40	12:20-14:00	14:50-17:00
2	<b>10</b>	<b>23</b>	<b>24</b>	<b>58</b>	<b>55</b>	<b>61</b>	<b>56</b>	
	JNB-GRJ	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-DUR	
	8:45-11:40	18:00-19:40	11:45-12:50	16:20-17:25	19:25-20:30	16:20-17:25	19:25-20:30	



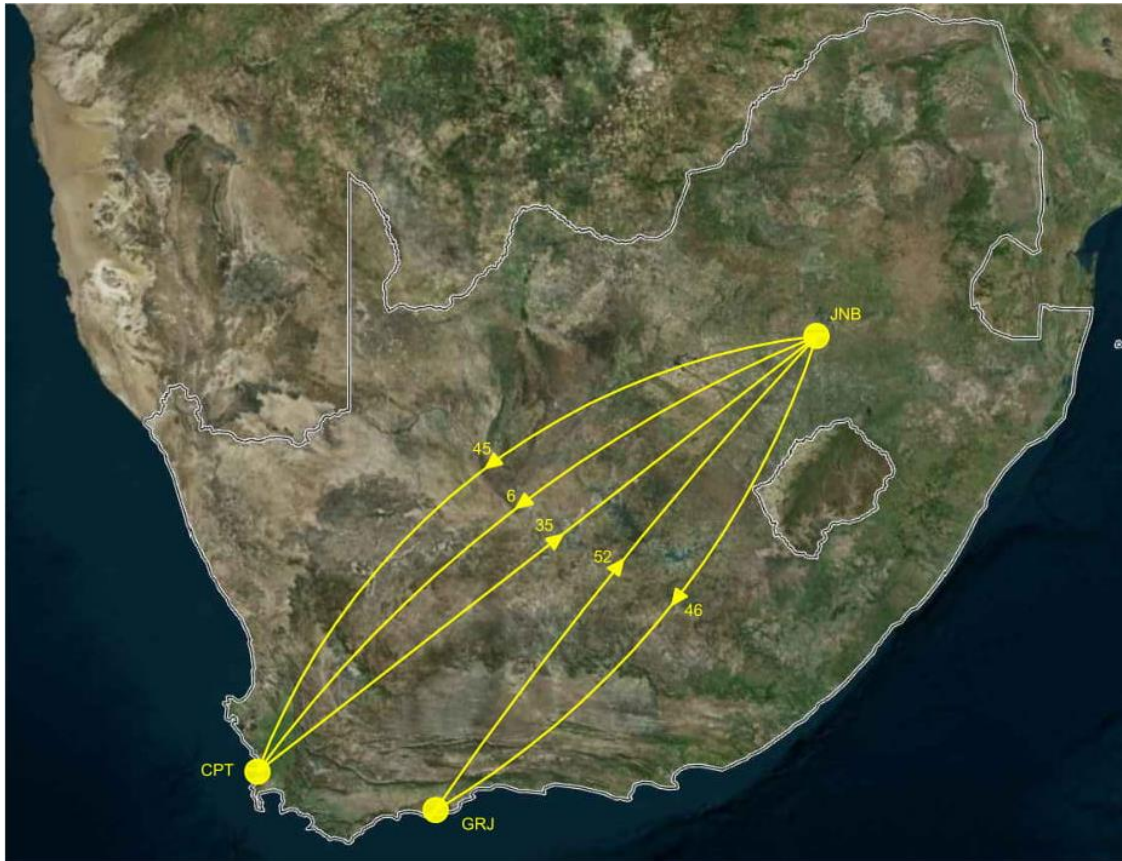
3	<b>12</b>	<b>5</b>	<b>34</b>	<b>30</b>	<b>37</b>	<b>39</b>					
	DUR-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR					
	6:00-7:05	7:55-10:05	11:45-13:45	14:50-17:00	17:45-19:45	11:45-12:50					
4	<b>18</b>	<b>3</b>	<b>57</b>	<b>26</b>	<b>59</b>	<b>43</b>	<b>49</b>	<b>41</b>			
	CPT-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR			
	11:45-13:45	14:35-15:40	13:30-14:35	17:30-18:35	19:20-20:25	8:50-11:00	14:40-16:40	17:30-18:35			
5	<b>1</b>	<b>14</b>	<b>11</b>	<b>38</b>			<b>44</b>		<b>51</b>		
	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB			JNB-CPT		CPT-JNB		
	5:55-7:00	13:30-14:35	15:25-17:20	12:20-14:00			11:40-14:00		17:45-19:45		
6	<b>13</b>	<b>8</b>	<b>20</b>	<b>27</b>	<b>33</b>	<b>29</b>	<b>36</b>	<b>42</b>	<b>48</b>	<b>40</b>	<b>62</b>
	DUR-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB
	7:50-8:55	14:50-17:00	17:45-19:45	7:55-10:05	10:50-12:50	13:40-15:50	16:35-18:35	7:50-10:05	10:50-12:50	14:35-15:40	19:20-20:25
7	<b>17</b>	<b>7</b>	<b>19</b>	<b>25</b>			<b>60</b>		<b>47</b>		
	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR			DUR-JNB		JNB-GRJ		
	10:50-12:50	13:40-15:50	16:35-18:35	14:35-15:40			13:30-14:35		15:25-17:20		
8	<b>9</b>	<b>22</b>	<b>4</b>	<b>16</b>	<b>31</b>			<b>53</b>			
	JNB-GRJ	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-GRJ			GRJ-JNB			
	6:45-8:40	12:20-14:00	17:30-18:35	19:20-20:25	8:45-11:40			18:00-19:40			
9	<b>21</b>	<b>2</b>	<b>15</b>	<b>54</b>	<b>32</b>		<b>28</b>		<b>50</b>		
	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB		JNB-CPT		CPT-JNB		
	9:15-10:55	11:45-12:50	16:20-17:25	19:25-20:30	7:50-8:55		11:50-14:00		16:35-18:35		

Por otra parte, se muestra también la diferencia, en valor absoluto, entre el número de horas totales de cada una de las rotaciones elegidas y el número de horas de referencia introducido (10 horas).

**Tabla 23:** Valores función objetivo de las rotaciones elegidas en routing3

Rotaciones	1	2	3	4	5	6	7	8	9
Diferencia (horas)	0,25	0,00	0,50	1,58	0,08	10,00	0,25	0,33	0,17

Se muestra también a modo de ejemplo una imagen donde se refleja la secuencia de vuelos que se realiza en la primera de las rotaciones seleccionadas.



**Figura 35:** Rutas realizadas en la primera rotación de routing3

### 5.3.4 Minimizar costes

Se utiliza el archivo **routing4** para obtener la solución del problema optimizando el aspecto económico, es decir, minimizando los costes. Se obtienen un total de 9 rotaciones, por lo que deberán ser utilizados todos los aviones de la compañía.

**Tabla 24:** Kulula Airlines. Solución routing4

R	Vuelos						
	06/08/2021		07/08/2021			08/08/2021	
1	<b>5</b>		<b>33</b>			<b>42</b>	
	JNB-CPT		CPT-JNB			JNB-CPT	
	7:55-10:05		10:50-12:50			7:50-10:05	
2	<b>9</b>	<b>38</b>	<b>30</b>	<b>37</b>	<b>44</b>	<b>50</b>	<b>56</b>

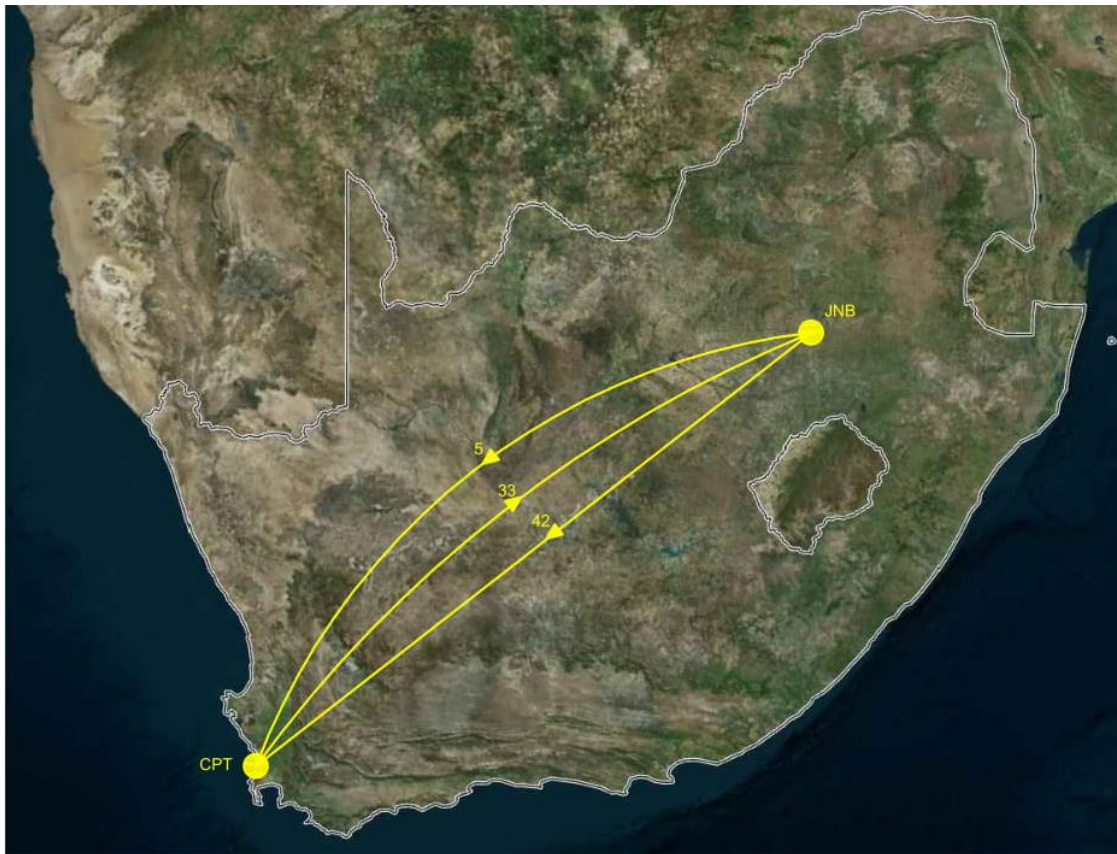
	JNB-GRJ				GRJ-JNB	JNB-CPT	CPT-JNB	JNB-CPT	CPT-JNB	JNB-DUR		
	6:45-8:40				12:20-14:00	14:50-17:00	17:45-19:45	11:40-14:00	16:35-18:35	19:25-20:30		
3	<b>20</b>				<b>24</b>		<b>58</b>		<b>46</b>			
	CPT-JNB				JNB-DUR		DUR-JNB		JNB-GRJ			
	17:45-19:45				11:45-12:50		16:20-17:25		8:45-11:40			
4	<b>13</b>	<b>3</b>			<b>32</b>	<b>28</b>	<b>36</b>	<b>39</b>				
	DUR-JNB	JNB-DUR			DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR				
	7:50-8:55	14:35-15:40			7:50-8:55	11:50-14:00	16:35-18:35	11:45-12:50				
5	<b>10</b>	<b>22</b>		<b>8</b>	<b>34</b>		<b>25</b>		<b>61</b>			
	JNB-GRJ	GRJ-JNB	JNB-CPT		CPT-JNB		JNB-DUR		DUR-JNB			
	8:45-11:40	12:20-14:00	14:50-17:00		11:45-13:45		14:35-15:40		16:20-17:25			
6	<b>17</b>	<b>7</b>	<b>19</b>		<b>31</b>			<b>52</b>	<b>45</b>	<b>51</b>		
	CPT-JNB	JNB-CPT	CPT-JNB		JNB-GRJ			GRJ-JNB	JNB-CPT	CPT-JNB		
	10:50-12:50	13:40-15:50	16:35-18:35		8:45-11:40			12:20-14:00	14:50-17:00	17:45-19:45		
7	<b>1</b>	<b>14</b>	<b>11</b>	<b>23</b>	<b>29</b>			<b>48</b>	<b>40</b>	<b>62</b>		
	JNB-DUR	DUR-JNB	JNB-GRJ	GRJ-JNB	JNB-CPT			CPT-JNB	JNB-DUR	DUR-JNB		
	5:55-7:00	13:30-14:35	15:25-17:20	18:00-19:40	13:40-15:50			10:50-12:50	14:35-15:40	19:20-20:25		
8	<b>21</b>	<b>2</b>	<b>15</b>	<b>54</b>	<b>57</b>		<b>55</b>		<b>60</b>	<b>47</b>	<b>53</b>	
	GRJ-JNB	JNB-DUR	DUR-JNB	JNB-DUR	DUR-JNB		JNB-DUR		DUR-JNB	JNB-GRJ	GRJ-JNB	
	9:15-10:55	11:45-12:50	16:20-17:25	19:25-20:30	13:30-14:35		19:25-20:30		13:30-14:35	15:25-17:20	18:00-19:40	
9	<b>12</b>	<b>6</b>	<b>18</b>	<b>4</b>	<b>16</b>	<b>27</b>	<b>35</b>	<b>26</b>	<b>59</b>	<b>43</b>	<b>49</b>	<b>41</b>
	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR	DUR-JNB	JNB-CPT	CPT-JNB	JNB-DUR
	6:00-7:05	8:50-11:00	11:45-13:45	17:30-18:35	19:20-20:25	7:55-10:05	14:40-16:40	17:30-18:35	19:20-20:25	8:50-11:00	14:40-16:40	17:30-18:35

Por otra parte, se muestra también el coste unitario de cada una de las rotaciones elegidas, es decir, el coste de cada uno de los vuelos de la rotación entre el número de vuelos que realiza cada una de ellas. Al haber introducido costes unitarios de cada uno de los vuelos que varían entre 50 y 100, estos serán los valores que puede tomar.

**Tabla 25:** Valores función objetivo de las rotaciones elegidas en routing4

Rotaciones	1	2	3	4	5	6	7	8	9
Coste unitario	68,00	76,29	73,50	68,83	64,50	86,00	73,25	85,89	78,92

Se muestra también a modo de ejemplo una imagen donde se refleja la secuencia de vuelos que se realiza en la primera de las rotaciones seleccionadas.

**Figura 36:** Rutas realizadas en la primera rotación de routing4

## 6 CONCLUSIONES

En primer lugar, se destaca la principal dificultad que se presenta a la hora de resolver el problema del ‘Aircraft Routing’, que es el tiempo de ejecución de los algoritmos que se utilizan para obtener la solución final. Se ha visto como este tipo de problema se caracteriza por utilizar un gran número de variables, y el hecho de que los algoritmos que se utilizan para su resolución son de tipo polinómicos no deterministas, es decir, el tamaño del problema aumenta exponencialmente a medida que aumenta el número de variables, provoca que el tiempo de ejecución del programa sea el principal factor limitante respecto a la obtención de soluciones factibles.

Sin embargo, se ha comprobado como este tiempo de ejecución no es proporcional al número de vuelos que forma parte de la programación de una aerolínea, sino que depende del número de combinaciones posibles que pueden obtenerse. Es decir, cuanto mayores son las posibilidades de formar distintas combinaciones de rotaciones posibles, mayor será el problema y por tanto el tiempo de ejecución será más amplio. Este hecho no depende por tanto del número de vuelos de la compañía, pues puede resultar que aerolínea con un menor número de vuelos presenten un mayor número de combinaciones posibles debido a una mejor distribución de horarios, menores tiempos de escala o una distribución de aeropuertos y bases de mantenimiento más eficiente.

Los algoritmos elaborados han sido desarrollados teniendo en cuenta este factor limitante, siendo especialmente importante el uso correcto y eficiente de las diferentes utilidades que ofrece el software *Matlab* para optimizar el código creado y obtener las soluciones en el menor tiempo posible. De este modo, se han obtenido archivos que permiten resolver el problema en cuestión de segundos y admite el análisis de casos de compañías aéreas reales con un mayor número de variables en cuestión de minutos. A pesar de ello, puede resultar que para aerolíneas con un mayor volumen de operaciones o una mayor posibilidad de combinaciones de vuelos el tiempo de ejecución de este programa se extienda hasta varias horas. Por ello, se propone para futuros trabajos el análisis de los archivos elaborados para mejorar la eficiencia del código utilizado y permitir mejorar en el tiempo de obtención de soluciones.

Los diferentes archivos creados permiten su aplicación para cualquier programación de vuelos disponible, siendo necesario la introducción de los diversos datos iniciales explicados a lo largo de este trabajo. Además, se han elaborado cuatro opciones distintas de resolución con el objetivo de obtener soluciones en función de diferentes criterios de optimización, permitiendo comparar los resultados obtenidos. En futuros trabajos relacionado, se propone la elaboración de otros algoritmos que permitan tener en cuenta nuevos objetivos de optimización.

Además, la secuenciación del programa en diferentes pasos permite comprobar en cada uno de ellos el proceso seguido y sirve para entender fácilmente el desarrollado seguido en la resolución del problema. De este modo, la mejor comprensión del proceso por parte del usuario permite un mejor análisis de las soluciones obtenidas y ayuda a que pueda introducir diferentes variantes en el código en el caso en que las restricciones impuestas no permitan obtener un resultado concreto, hecho que se utiliza en este trabajo para resolver el caso práctico introducido.

Por último, se ha demostrado como la revisión de horarios de la programación inicial de vuelos permite obtener mayores combinaciones de rotaciones posibles y, por tanto, ayuda a poder obtener mejores resultados finales. Aunque para poder aplicar cambios efectivos finales en la programación final de la compañía es necesario analizar otros factores que no dependen de este proceso de rotación de aviones, el hecho de elaborar un simple análisis en los horarios programados puede servir como inicio del proceso de mejora a desarrollar para obtener la solución final más eficiente. Para futuros trabajos en relación con el tema, sería interesante analizar estos factores y desarrollar algún tipo de programa que pueda tenerlos en cuenta para realizar una revisión de horarios más efectiva.

## REFERENCIAS

- [1] Aviación online. [consulta: 26 junio 2021]. Disponible en: <https://www.aviacionline.com/2021/01/documento-el-transporte-aereo-perdio-casi-2-8-mil-millones-de-pasajeros-en-2020/>
- [2] Organización de Aviación Civil Internacional. [consulta: 26 junio 2021]. Disponible en: <https://www.icao.int/Pages/default.aspx>
- [3] LACASSE-GUAY, E., DESAULNIERS, G., & SOUMIS, F. Aircraft routing under different business processes. En: *Journal of Air Transport Management*. 2010, 16(5), 258–263.
- [4] Mantenimiento en hangar. [consulta: 22 junio 2021]. Disponible en: <https://fly-news.es/wp-content/uploads/IBEMANTENIMIENTO.jpg>
- [5] BAZARGAN, M. *Airline Operations and Scheduling*. Aldershot England : Ashgate, 2004. ISBN 075463616X
- [6] CONFORTI, M., CORNUÉJOLS, G., & ZAMBELLI, G. *Integer Programming*. Springer, 2014. ISBN 978-3-319-11007-3
- [7] Integer Programming - MATLAB & Simulink. [consulta: 12 febrero 2021]. Disponible en: <https://es.mathworks.com/discovery/integer-programming.html>
- [8] Elija primero el enfoque basado en problemas o basado en Solver - MATLAB & Simulink - MathWorks España. [consulta: 15 febrero 2021]. Disponible en: <https://es.mathworks.com/help/optim/ug/first-choose-problem-based-or-solver-based-approach.html>
- [9] Configuración de problemas de optimización basada en Solver - MATLAB & Simulink - MathWorks España. [consulta: 15 febrero 2021]. Disponible en: <https://es.mathworks.com/help/optim/optimization-problem-setup-solver-based.html>
- [10] Tabla de decisión de optimización - MATLAB & Simulink - MathWorks España. [consulta: 26 febrero 2021]. Disponible en: <https://es.mathworks.com/help/optim/ug/optimization-decision-table.html>
- [11] Programación lineal de enteros mixtos (MILP) - MATLAB intlinprog - MathWorks España. [consulta: 26 febrero 2021]. Disponible en: <https://es.mathworks.com/help/optim/ug/intlinprog.html#bts3f9f-2>
- [12] Algoritmos de programación lineal de enteros mixtos - MATLAB & Simulink - MathWorks España. [consulta: 2 marzo 2021]. Disponible en: <https://es.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html#btzwtmv>
- [13] kulula - Wikipedia. [consulta: 20 abril 2021]. Disponible en: <https://en.wikipedia.org/wiki/Kulula.com>
- [14] Routes - kulula.com. [consulta: 28 abril 2021]. Disponible en: <https://www.kulula.com/flights/routes-and->

schedules/routes

[15] Kulula Airlines - kulula.com. [consulta: 30 abril 2021]. Disponible en: <https://www.kulula.com/>

[16] JOSÉ MARÍA DEL CASTILLO GRANADOS. Aplicación de la programación lineal al transporte aéreo: Problemas de explotación y gestión, Explotación del Transporte Aéreo, 4º de GIA.





# ÍNDICE DE CONCEPTOS

**Compañía aérea:** Toda empresa de transporte aéreo que posee una licencia de explotación válida, expedida por la Dirección General de Aviación Civil.

**Tiempo de escala:** Tiempo entre la llegada y posterior salida de una misma aeronave en un aeropuerto. Incluye el tiempo de rodadura desde la pista de aterrizaje hasta el estacionamiento de la aeronave, la descarga del equipaje y los pasajeros, la limpieza del avión, las inspecciones realizadas, el repostaje y vaciado de aguas fecales, embarque de pasajeros, carga de equipaje y rodadura hasta la pista de aterrizaje para la salida del siguiente vuelo.

**Secuencia de vuelos:** conjunto sucesivo de vuelos.

**Rotación:** secuencia de vuelos que realiza una misma aeronave.

**Frecuencia de vuelos:** número de veces que se realiza un determinado vuelo.

**Función objetivo:** expresión matemática que contiene las variables a optimizar.

**Optimización:** maximizar o minimizar una función real eligiendo sistemáticamente valores de entrada y computando el valor de la función.

**Coste computacional:** cantidad de memoria requerida (suma total del espacio que ocupan las variables del algoritmo) antes, durante y después de su ejecución.

**CASM:** coste unitario de producción, se obtiene dividiendo el coste total entre el ASM.

**ASM:** número total de asientos ofrecidos por milla volados.

**B737-800:** avión de pasajeros bimotor, de reacción, fuselaje estrecho, de corto a medio alcance, fabricado por la compañía estadounidense 'Boeing Commercial Airplanes'.

**B757-200:** avión de pasajeros bimotor, de reacción, fuselaje estrecho, de medio a largo alcance, fabricado por la compañía estadounidense 'Boeing Commercial Airplanes'.



## GLOSARIO

---

ICAO: International Civil Aviation Organization	1
EASA: European Union Aviation Safety Agency	5
FAA: Federal Aviation Administration	5
CASM: Cost per Available Seat Mile	10
ASM: Available Seat Mile	10
JFK: John F. Kennedy International Airport, IATA code	15
IP: Programación lineal entera	16
MILP: Programación lineal de enteros mixtos	16
IATA: International Air Transport Association	28
LAX: Los Angeles International Airport, IATA code	29
SFO: San Francisco International Airport, IATA code	29
ORD: O'Hare International Airport, IATA code	29
ATL: Hartsfield–Jackson Atlanta International Airport, IATA code	29
MIA: Miami International Airport, IATA code	29
BOS: Logan International Airport, IATA code	29
IAD: Washington Dulles International Airport, IATA code	29
JNB: O R Tambo International Airport, IATA code	60
DUR: King Shaka International Airport, IATA code	60
GRJ: George Airport, IATA code	60
CPT: Cape Town International Airport, IATA code	60



# ANEXOS

## ANEXO A

En este anexo se adjunta el código creado en los diferentes archivos y funciones explicados y utilizados anteriormente para la resolución del problema del 'Aircraft Routing'. Además, al final de cada uno de los códigos se explica detenidamente el objetivo de cada uno de ellos, las principales variables introducidas y su funcionamiento, con el propósito de aclarar cualquier duda que pueda existir a la hora de ejecutar los diferentes algoritmos que resuelven el problema.

### Routing1

#### Código

```
%Aircraft routing maximizando posibilidades mantenimiento
tic;
Am=1; %Aeropuerto de mantenimiento
tmin=45; %Tiempo entre la llegada y salida de un mismo avión (min)
Vuelos=xlsread('Datos vuelos'); %Programación de vuelos
R=rotaciones1(Am,tmin,Vuelos);
Na=5; %Número max de aviones disponibles

%Función objetivo
[nf,nc]=size(R);
nv=nc/3;
m=zeros(nf,1);
for j=1:nf
    for k=1:3
        v=find(R(j,k*nv-(nv-1):k*nv),1,'last');
        if Vuelos(find(Vuelos(:,1)==R(j,v+nv*(k-1)),1),3)==Am
            m(j)=m(j)+1;
        end
    end
end
f=-m;
intcon=(1:length(f));

%Restricciones
%Cobertura de vuelos Aeq*x=beq
u=1;
Aeq=sparse(3*size(Vuelos,1),nf);
for j=1:3
    vdj=Vuelos(Vuelos(:,4)==j,1);
    for k=1:length(vdj)
        for i=1:nf
            if any(R(i,j*nv-(nv-1):j*nv)==vdj(k))
                Aeq(u,i)=1;
            end
        end
        u=u+1;
    end
end
Aeq(sum(abs(Aeq),2)==0,:)=[];
beq=ones(size(Aeq,1),1);
```

```

%Número max de aviones disponibles  $A \cdot x \leq b$ 
A=ones(1,nf);
b=Na;

%Límite inferior (como mínimo  $x=0$ )
lb=zeros(nf,1);

%Optimización
[x,fval,exitflag,output]=intlinprog(f,intcon,A,b,Aeq,beq,lb);

%Solución
nr=find(x);
rotaciones=R(nr,:);
time=toc;
disp('Tiempo en ejecutar el programa');disp(time);
disp('Número máximo de vuelos en un mismo día');disp(nv);
disp('Las rotaciones elegidas están formadas por los
vuelos');disp(rotaciones);

```

## Descripción

El archivo **routing1** permite obtener entre el total de todas las rotaciones posibles, según las necesidades de mantenimiento y el tiempo de escala, aquellas que maximizan las posibilidades de mantenimiento, es decir, aquellas que maximizan el número de noches en el aeropuerto de mantenimiento elegido, cumpliendo con las restricciones de cobertura de vuelo y número máximo de aviones disponibles.

En las primeras líneas de este archivo se introducen los datos por partes del usuario, necesitando completar la siguiente información: aeropuerto de mantenimiento, tiempo de escala, programación de vuelos (en *Excel*) y número máximo de aeronaves disponibles para operar las rotaciones.

Todos estos datos sirven como entrada para la función **rotaciones1**, la cual permite hallar todas las rotaciones posibles a operar cumpliendo con los requerimientos de mantenimiento (al menos una noche en el aeropuerto base de mantenimiento de los 3 días que consta la rotación) y el tiempo de escala. Todas estas rotaciones posibles se almacenan en la matriz **R**, cuyas filas representan cada una de las rotaciones y sus columnas cada uno de los vuelos realizados en cada una de las rotaciones.

El siguiente paso consiste en implementar la función objetivo, la cual se quiere optimizar. Siendo el objetivo maximizar las posibilidades de mantenimiento, es decir, el número de estancias nocturnas en el aeropuerto base de mantenimiento, es necesario calcular las pernoctaciones en este aeropuerto de cada una de las rotaciones. Para ello, se crea el vector **m** cuya longitud coincide con el número de filas de la matriz **R**, es decir, con el número de rotaciones posibles. A este vector **m**, compuesto de ceros, se le sumará una unidad en cada rotación si el último vuelo del día finaliza en el aeropuerto de mantenimiento, realizándose este proceso para los 3 días que componen la rotación y para todas las rotaciones. De este modo, el vector **m** será un vector compuesto por los valores 0, 1, 2 ó 3. Por último, como el objetivo no es minimizar la función objetivo, sino maximizarla, la función objetivo **f** será el inverso de **m**.

Por otro lado, es necesario identificar las variables restringidas a tomar valores enteros. Este proceso se realiza mediante el vector de restricciones de enteros **intcon**, que está formado por un vector de 0 y 1, siendo los valores positivos las componentes de la variable de decisión (**x**) que son valores enteros. En este caso, todas las componentes de la variable de decisión son enteras, por lo que **intcon** es un vector con las mismas dimensiones que la función objetivo de con todas sus componentes iguales a 1.

A continuación, es necesario definir las restricciones del problema. Este paso es idéntico en los cuatro archivos donde aparecen, por lo que sólo se explica en este momento. Existen dos tipos de restricciones: de igualdad y de desigualdad.

Las restricciones de igualdad  $A_{eq}x = b_{eq}$  están formadas en nuestro problema por las restricciones de cobertura de vuelo. Esta restricción tenía por objetivo que cada vuelo se realizase todos los días en los que esté programado y solo una vez al día, permitiendo que no se realice un mismo vuelo con diferentes aviones y que se operen todos

los vuelos.

Para tener en cuenta este tipo de restricciones se crean la matriz de restricción de igualdad lineal  $A_{eq}$  y el vector de restricción de desigualdad lineal  $b_{eq}$ . La matriz tiene tantas filas como restricciones existan y el número de columnas es igual al número de rotaciones posibles, es decir, al número de variables de decisión del problema. Para cada una de las restricciones o filas se tomará el valor 1 si el vuelo que se está analizando aparece en la rotación, y 0 en caso contrario. Por otro lado, el vector tiene también tantas filas como restricciones existan y todas sus componentes son iguales a la unidad.

En concreto, existen tantas restricciones de este tipo como vuelos y número de días que se realice cada vuelo haya, es decir, una restricción por cada vuelo y por cada día que se realice el vuelo. Se indica como se formaría esta restricción en el programa con el ejemplo introducido en el **Capítulo 2.1.3**.

Restricciones de igualdad en *Matlab*:

$$A_{eq}x = b_{eq}$$

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (0-1)$$

Para ahorrar espacio y tiempo de ejecución se crea  $A_{eq}$  como una matriz dispersa.

$$b_{eq} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (0-2)$$

Las restricciones de desigualdad  $Ax \leq b$  la forman en nuestro problema las restricciones de número máximo de aviones disponibles. Esta restricción tenía por objetivo limitar el número de aeronaves que operen las rotaciones a elegir y, en concreto, solo existirá una restricción de este tipo.

Para tener en cuenta este tipo de restricciones se crean la matriz de restricción de desigualdad lineal  $A$  y el vector de restricción de desigualdad lineal  $b$ . La matriz tiene tantas filas como restricciones de este tipo existan, una en este caso, y el número de columnas es igual al número de variables de decisión existentes. El vector tiene tantas filas como restricciones, por lo que en este caso está formado por una sola componente.

Para el ejemplo anterior, suponiendo que el número máximo de aeronaves disponibles es de 3, las restricciones de desigualdad en *Matlab* serían:

$$Ax \leq b$$

$$A = [1 \ 1 \ 1 \ 1 \ 1] \quad (0-3)$$

$$b = [3]$$

Por último, se especifica el límite inferior de las variables de decisión. Para ello se introduce el vector  $lb$ , en el que se almacenan los valores de los límites inferiores que toman cada una de las variables de decisión de modo que  $lb \leq x$ . Se sabe que estas variables tomarán los valores 0 si la rotación no es elegida o 1 si la rotación es

elegida, por lo que se el vector será de longitud igual al número de variables existentes y todas sus componentes son iguales a cero. Este paso también es idéntico en los cuatro archivos elaborados.

Una vez definidos la función objetivo, las restricciones de cobertura de vuelo y número máximo de aviones disponibles, y el límite inferior de la variable de decisión, se utiliza la función **intlinprog** de *Matlab* para optimizar el problema y obtener la solución buscada. La información detallada del funcionamiento y las variables necesarias para utilizar esta función han sido expuestas en el **Capítulo 3.3.1**.

El último paso consiste en obtener la información deseada a partir de la solución obtenida. El resultado de ejecutar la función **intlinprog** se almacena en la variable de decisión  $x$ , siendo un vector de longitud igual al número de rotaciones posibles cuyas componentes toman los valores: 0, si la rotación no ha sido elegida, y 1, si la rotación es elegida.

Para ello, se buscan los valores no nulos de  $x$  y su posición se almacenan en el vector **nr**. Conociendo la posición de las rotaciones elegidas, se buscan en la matriz **R** donde se almacenan todas las rotaciones posibles, y el resultado se guarda en la matriz rotaciones, que será la que se muestre por pantalla.

Finalmente, se muestran por pantalla los siguientes resultados:

- Tiempo en ejecutar el programa (**time**): para ello se han utilizado las funciones tic y toc de Matlab.
- Número máximo de vuelos en un día (**nv**): indica el mayor número de vuelos seguidos que pueden realizarse en un mismo día, aunque no tiene que coincidir con el mayor número de vuelos que se realice en un día de las rotaciones finales elegidas.
- Rotaciones elegidas (**rotaciones**): en esta matriz se muestran las rotaciones elegidas y, por tanto, la solución del problema cumpliendo con todos los requerimientos y restricciones impuestos. La matriz tiene un número de filas igual al número de rotaciones finales elegidas y un número de columnas igual al número máximo de vuelos en un día por el número de días de la rotación (3). Los valores de esta matriz indican los números de los vuelos que se realizan en cada una de las rotaciones finales.

## Routing2

### Código

```
%Aircraft routing maximizando conexiones de vuelos
tic;
Am=1; %Aeropuerto de mantenimiento
tmin=45; %Tiempo entre la llegada y salida de un mismo avión (min)
Vuelos=xlsread('Datos vuelos'); %Programación de vuelos
R=rotaciones1(Am,tmin,Vuelos);
Na=5; %Número max de aviones disponibles

%Función objetivo
[nf,nc]=size(R);
nv=nc/3;
v=zeros(nf,1);
for j=1:nf
    v(j)=length(find(R(j,:)));
end
f=-v;
intcon=(1:length(f));

%Restricciones
%Cobertura de vuelos Aeq*x=beq
u=1;
Aeq=sparse(3*size(Vuelos,1),nf);
for j=1:3
    vdj=Vuelos(Vuelos(:,4)==j,1);
    for k=1:length(vdj)
        for i=1:nf
```



```

        if any(R(i, j*nv-(nv-1):j*nv)==vdj(k))
            Aeq(u, i)=1;
        end
    end
    u=u+1;
end
end
Aeq(sum(abs(Aeq), 2)==0, :)=[]; %
beq=ones(size(Aeq, 1), 1);
%Número de aviones disponibles A*x<=b
A=ones(1, nf);
b=Na;

%Límite inferior (como mínimo x=0)
lb=zeros(nf, 1);

%Optimización
[x, fval, exitflag, output]=intlinprog(f, intcon, A, b, Aeq, beq, lb);

%Solución
nr=find(x);
rotaciones=R(nr, :);
time=toc;
disp('Tiempo en ejecutar el programa'); disp(time);
disp('Número máximo de vuelos en un mismo día'); disp(nv);
disp('Las rotaciones elegidas están formadas por los
vuelos'); disp(rotaciones);

```

## Descripción

El archivo **routing2** permite obtener entre el total de todas las rotaciones posibles, según las necesidades de mantenimiento y el tiempo de escala, aquellas que maximizan las conexiones de vuelo, es decir, aquellas que maximizan el número de vuelos que se realizan en cada una de las rotaciones.

En las primeras líneas de este archivo se introducen los datos por partes del usuario, necesitando completar la misma información que para el archivo anterior: aeropuerto de mantenimiento, tiempo de escala, programación de vuelos (en *Excel*) y número máximo de aeronaves disponibles para operar las rotaciones.

Todos estos datos sirven como entrada para la función **rotaciones1**, la cual permite hallar todas las rotaciones posibles a operar cumpliendo con los requerimientos de mantenimiento (al menos una noche en el aeropuerto base de mantenimiento de los 3 días que consta la rotación) y el tiempo de escala. Todas estas rotaciones posibles se almacenan en la matriz **R**, cuyas filas representan cada una de las rotaciones y sus columnas cada uno de los vuelos realizados en cada una de las rotaciones.

El siguiente paso consiste en implementar la función objetivo, la cual se quiere optimizar. Siendo el objetivo maximizar las conexiones de vuelo, es decir, el número de vuelos que se realiza en cada rotación, es necesario calcular este dato para cada una de ellas. Para ello, se crea el vector **v** cuya longitud coincide con el número de filas de la matriz **R**, es decir, con el número de rotaciones posibles. A este vector **v**, compuesto de ceros en un principio, se le añade el número de vuelos que existen en la rotación, contando para ello el número de componentes no nulos de cada una de las rotaciones, es decir, de cada una de las filas de la matriz **R**. Por último, como el objetivo no es minimizar la función objetivo, sino maximizarla, la función objetivo **f** será el inverso de **v**.

Por otro lado, es necesario identificar las variables restringidas a tomar valores enteros. Este proceso se realiza mediante el vector de restricciones de enteros **intcon**. De igual modo que en el archivo anterior, todas las componentes son iguales a 1 debido a que todas las variables deben de tomar valores enteros.

A continuación, se incluyen las restricciones de cobertura de vuelo y número máximo de aviones disponibles exactamente igual que en el caso anterior. Lo mismo ocurre para la creación de los límites inferiores.

Por último, se ejecuta la función **intlinprog** de *Matlab* y se obtienen las rotaciones elegidas. Los resultados

mostrados por pantalla son los mismo que para el archivo anterior, si bien se obtendrán valores diferentes debido a que la optimización se realiza con un objetivo diferente.

### Routing3

#### Código

```
%Aircraft routing minimizando desviación respecto a horas de referencia
tic;
Am=1; %Aeropuerto de mantenimiento
tmin=45; %Tiempo entre la llegada y salida de un mismo avión (min)
Vuelos=xlsread('Datos vuelos'); %Programación de vuelos
R=rotaciones1(Am,tmin,Vuelos);
H=20; %Hora de referencia (h)
Na=5; %Número max de aviones disponibles

%Función objetivo
[nf,nc]=size(R);
h=zeros(nf,1);
for j=1:nf
    v=R(j,(find(R(j,:)))));
    for k=1:length(v)
        pos=find(v(k)==Vuelos(:,1),1);
        h(j)=h(j)+Vuelos(pos,9);
    end
end
f=abs(h-H);
intcon=(1:length(f));

%Restricciones
%Cobertura de vuelos Aeq*x=beq
nv=nc/3;
u=1;
Aeq=sparse(3*size(Vuelos,1),nf);
for j=1:3
    vdj=Vuelos(Vuelos(:,4)==j,1);
    for k=1:length(vdj)
        for i=1:nf
            if any(R(i,j*nv-(nv-1):j*nv)==vdj(k))
                Aeq(u,i)=1;
            end
        end
        u=u+1;
    end
end
Aeq(sum(abs(Aeq),2)==0,:)=[];
beq=ones(size(Aeq,1),1);
%Número de aviones disponibles A*x<=b
A=ones(1,nf);
b=Na;

%Límite inferior (como mínimo x=0)
lb=zeros(nf,1);

%Optimización
[x,fval,exitflag,output]=intlinprog(f,intcon,A,b,Aeq,beq,lb);

%Solución
nr=find(x);
rotaciones=R(nr,:);
```

```
time=toc;
disp('Tiempo en ejecutar el programa');disp(time);
disp('Número máximo de vuelos en un mismo día');disp(nv);
disp('Las rotaciones elegidas están formadas por los
vuelos');disp(rotaciones);
```

### Descripción

El archivo **routing3** permite obtener entre el total de todas las rotaciones posibles, según las necesidades de mantenimiento y el tiempo de escala, aquellas que minimizan la desviación del número total de horas de vuelo respecto a un número de horas de referencia.

En las primeras líneas de este archivo se introducen los datos por partes del usuario, necesitando completar la misma información que para el archivo anterior: aeropuerto de mantenimiento, tiempo de escala, programación de vuelos (en *Excel*) y número máximo de aeronaves disponibles para operar las rotaciones. En este caso, es necesario introducir también el número de horas de referencia, a elección del usuario.

Todos estos datos sirven como entrada para la función **rotaciones1**, la cual permite hallar todas las rotaciones posibles a operar cumpliendo con los requerimientos de mantenimiento (al menos una noche en el aeropuerto base de mantenimiento de los 3 días que consta la rotación) y el tiempo mínimo entre la llegada y posterior salida de un mismo avión. Todas estas rotaciones posibles se almacenan en la matriz **R**, cuyas filas representan cada una de las rotaciones y sus columnas cada uno de los vuelos realizados en cada una de las rotaciones.

El siguiente paso consiste en implementar la función objetivo, la cual se quiere optimizar. Siendo el objetivo minimizar la desviación entre el número de horas de vuelo y el número de horas de referencia, es necesario calcular este dato para cada una de ellas. Para ello, se crea el vector **h** cuya longitud coincide con el número de filas de la matriz **R**, es decir, con el número de rotaciones posibles. A este vector **h**, compuesto de ceros en un principio, se va sumando la duración de cada uno de los vuelos que forman cada rotación, a partir de los datos de la programación de vuelos introducido. Para ello, en primer lugar se buscan los valores no nulos de las filas de **R**, que son los vuelos que forman cada rotación, y se almacenan en el vector **v**. Seguidamente, para cada una de las componentes de **v**, es decir, para cada vuelo, se busca su posición dentro de la matriz de la programación de vuelos **y**, una vez conocida, se añade la duración de cada vuelo a la fila correspondiente de **h**. Por último, como el objetivo es minimizar la desviación, la función objetivo **f** es igual al valor absoluto de la diferencia de cada una de las filas de **h** con respecto al número de horas de referencia introducido **H**.

Por otro lado, es necesario identificar las variables restringidas a tomar valores enteros. Este proceso se realiza mediante el vector de restricciones de enteros **intcon**. De igual modo que en el archivo anterior, todas las componentes son iguales a 1 debido a que todas las variables deben de tomar valores enteros.

A continuación, se incluyen las restricciones de cobertura de vuelo y número máximo de aviones disponibles exactamente igual que en el caso anterior. Lo mismo ocurre para la creación de los límites inferiores.

Por último, se ejecuta la función **intlinprog** de *Matlab* y se obtienen las rotaciones elegidas. Los resultados mostrados por pantalla son los mismo que para el archivo anterior, si bien se obtendrán valores diferentes debido a que la optimización se realiza con un objetivo diferente.

## Routing4

### Código

```
%Aircraft routing minimizando costes
tic;
Am=1; %Aeropuerto de mantenimiento
tmin=45; %Tiempo entre la llegada y salida de un mismo avión (min)
Vuelos=xlsread('Datos vuelos'); %Programación de vuelos
R=rotaciones1(Am,tmin,Vuelos);
Na=5; %Número max de aviones disponibles

%Función objetivo
[nf,nc]=size(R);
c=zeros(nf,1);
```

```

for j=1:nf
    v=R(j, (find(R(j, :)))));
    for k=1:length(v)
        pos=find(v(k)==Vuelos(:,1),1);
        c(j)=c(j)+Vuelos(pos,10);
    end
    c(j)=c(j)/length(v);
end
f=c;
intcon=(1:length(f));

%Restricciones
%Cobertura de vuelos Aeq*x=beq
nv=nc/3;
u=1;
Aeq=sparse(3*size(Vuelos,1),nf);
for j=1:3
    vdj=Vuelos(Vuelos(:,4)==j,1);
    for k=1:length(vdj)
        for i=1:nf
            if any(R(i,j*nv-(nv-1):j*nv)==vdj(k))
                Aeq(u,i)=1;
            end
        end
        u=u+1;
    end
end
Aeq(sum(abs(Aeq),2)==0,:)=[];
beq=ones(size(Aeq,1),1);
%Número de aviones disponibles A*x<=b
A=ones(1,nf);
b=Na;

%Límite inferior (como mínimo x=0)
lb=zeros(nf,1);

%Optimización
[x,fval,exitflag,output]=intlinprog(f,intcon,A,b,Aeq,beq,lb);

%Solución
nr=find(x);
rotaciones=R(nr,:);
time=toc;
disp('Tiempo en ejecutar el programa');disp(time);
disp('Número máximo de vuelos en un mismo día');disp(nv);
disp('Las rotaciones elegidas están formadas por los
vuelos');disp(rotaciones);

```

## Descripción

El archivo **routing4** permite obtener entre el total de todas las rotaciones posibles, según las necesidades de mantenimiento y el tiempo de escala, aquellas que minimizan los costes introducidos en la programación de vuelos.

En las primeras líneas de este archivo se introducen los datos por partes del usuario, necesitando completar la misma información que para el archivo anterior: aeropuerto de mantenimiento, tiempo de escala, programación de vuelos (en *Excel*) y número máximo de aeronaves disponibles para operar las rotaciones.

Todos estos datos sirven como entrada para la función **rotaciones1**, la cual permite hallar todas las rotaciones posibles a operar cumpliendo con los requerimientos de mantenimiento (al menos una noche en el aeropuerto base de mantenimiento de los 3 días que consta la rotación) y el tiempo mínimo entre la llegada y posterior salida

de un mismo avión. Todas estas rotaciones posibles se almacenan en la matriz **R**, cuyas filas representan cada una de las rotaciones y sus columnas cada uno de los vuelos realizados en cada una de las rotaciones.

El siguiente paso consiste en implementar la función objetivo, la cual se quiere optimizar. Siendo el objetivo minimizar los costes, es necesario calcular el coste unitario de cada rotación. Para ello, se crea el vector **c** cuya longitud coincide con el número de filas de la matriz **R**, es decir, con el número de rotaciones posibles. A este vector **c**, compuesto de ceros en un principio, se va sumando los costes de cada uno de los vuelos que forman cada rotación, a partir de los datos de la programación de vuelos introducido. Para ello, en primer lugar se buscan los valores no nulos de las filas de **R**, que son los vuelos que forman cada rotación, y se almacenan en el vector **v**. Seguidamente, para cada una de las componentes de **v**, es decir, para cada vuelo, se busca su posición dentro de la matriz de la programación de vuelos y, una vez conocida, se añade el coste de cada vuelo a la fila correspondiente de **c**. Por último, como el objetivo es minimizar el coste unitario, se divide el coste total de cada rotación entre el número de vuelos de cada una. Esta es por tanto la función objetivo **f**.

Por otro lado, es necesario identificar las variables restringidas a tomar valores enteros. Este proceso se realiza mediante el vector de restricciones de enteros **intcon**. De igual modo que en el archivo anterior, todas las componentes son iguales a 1 debido a que todas las variables deben de tomar valores enteros.

A continuación, se incluyen las restricciones de cobertura de vuelo y número máximo de aviones disponibles exactamente igual que en el caso anterior. Lo mismo ocurre para la creación de los límites inferiores.

Por último, se ejecuta la función **intlinprog** de *Matlab* y se obtienen las rotaciones elegidas. Los resultados mostrados por pantalla son los mismo que para el archivo anterior, si bien se obtendrán valores diferentes debido a que la optimización se realiza con un objetivo diferente.

## Rotaciones1

### Código

```
function R=rotaciones1(Am,tmin,Vuelos)
v=Vuelos(:,1:4);
s=Vuelos(:, [5 6]);
l=Vuelos(:, [7 8]);
rot=rot1(tmin,v,s,l);
d1=find(rot(:,1)==1); l1=length(d1);
d2=find(rot(:,1)==2); l2=length(d2);
d3=find(rot(:,1)==3); l3=length(d3);
l=[l1 l2 l3]; lmax=max(l);
d=zeros(lmax,3);
d(1:l1,1)=d1; d(1:l2,2)=d2; d(1:l3,3)=d3;
rot2=zeros(1);
    for j=1:2
        u=1;
        for i=1:find(d(:,j),1,'last')
            for k=1:find(d(:,j+1),1,'last')
                vpi=find(rot(d(i,j,:),:),1,'last');
                vni=find(rot(d(i,j),vpi)==v(:,1),1);
                vnk=find(rot(d(k,j+1),2)==v(:,1),1);
                if v(vni,3)==v(vnk,2)
                    rot2(u,j+j-1)=d(i,j);
                    rot2(u,j+j)=d(k,j+1);
                    u=u+1;
                end
            end
        end
    end
end
u=1;
z=size(rot,2);
R=zeros(1);
for i=1:size(rot2(:,2))
    for k=1:size(rot2(:,3))
```

```

        if rot2(i,2)==rot2(k,3)
            rot3([1 2])=rot2(i,[1 2]);
            rot3(3)=rot2(k,4);
            rp=[find(rot(rot3(1),:),1,'last')
find(rot(rot3(2),:),1,'last') find(rot(rot3(3),:),1,'last')];
            rv=diag(rot(rot3,rp));
vf=[find(rv(1)==v(:,1),1);find(rv(2)==v(:,1),1);find(rv(3)==v(:,1),1)];
        if any(v(vf,3)==Am) &&
v(vf(3),3)==v((find(v(:,1)==rot(rot3(1),2),1)),2)
            for j=1:3
                R(u,j*(z-1)-(z-2):j*(z-1))=rot(rot3(j),2:end);
            end
            u=u+1;
        end
    end
end
end
end
end

```

## Descripción

La función **rotaciones1** tiene por objetivo encontrar todas las posibles rotaciones de 3 días que pueden realizarse teniendo en cuenta el aeropuerto de mantenimiento y el tiempo de escala. Las variables de entrada a esta función son el aeropuerto de mantenimiento (**Am**), el tiempo de escala (**tmin**) y los datos sobre la programación de vuelos introducidos en *Excel*, que se almacenan en la matriz **Vuelos**. La variable de salida es la matriz **R**, donde se albergan todos los vuelos que se realizan en cada una de las rotaciones posibles.

El primer paso de esta función consiste en extraer de la matriz de la programación de vuelos las columnas con la información relativa a los números de vuelos, día del vuelo y aeropuertos de salida y llegada, que se almacenan en la matriz **v**, y por otro lado las columnas con la información del horario de salida y llegada, que se almacenan en las matrices **s** y **l** respectivamente.

Seguidamente, se llama a la función **rot1** creada para obtener todas las posibles rotaciones de un solo día teniendo en cuenta **tmin**. Esta función nos da como resultado la matriz **rot**, donde aparecen en la primera columna el día en que se realiza cada una de las rotaciones, y en el resto de ellas los vuelos que componen cada una. Cada fila de esta matriz es una rotación de un día.

A continuación, a partir de la matriz de rotaciones de un día **rot**, se obtiene la matriz **d**. En ésta se albergan los números de las rotaciones de un día, de modo que cada columna se identifica con un día en concreto, existiendo por tanto 3 columnas.

El siguiente paso consiste en ir enlazando combinaciones de rotaciones de un día hasta crear las rotaciones de 3 días. Para ello, en primer lugar se obtiene la matriz **rot2**, cuyas dos primeras columnas indican las combinaciones posibles de rotaciones del día 1 con el día 2, y dos columnas siguientes indican las combinaciones posibles de rotaciones del día 2 con el día 3. Una vez realizado este proceso, se combinan las columnas segunda y tercera de esta matriz **rot2** para obtener las rotaciones de 3 días en **rot3**. Este vector está formado por 3 columnas, cada una de las cuales refleja la rotación de un día del día correspondiente a su columna.

Para mejorar la eficiencia del código, no se obtiene la matriz de rotaciones de 3 días completa y después se analizan si estas rotaciones son posibles en cuanto a los requisitos de mantenimiento impuesto. En vez de esto, cada vez que se obtiene una rotación posible de 3 días, se almacena en **rot3** y se analiza si el último vuelo de cada uno de los días finaliza en el aeropuerto de mantenimiento indicado, y si la rotación es del tipo ciclo cerrado, es decir, si el aeropuerto del que realiza la salida el primer vuelo del primer día es el mismo que el aeropuerto de llegada del último vuelo del último día. De ser así, esta rotación de 3 días ya sería adecuada para nuestros requerimientos y se almacenaría en la matriz final **R**, donde se almacenan todas las posibles rotaciones de 3 días que sí cumplen con los requerimientos indicados. En caso negativo, esta rotación no se almacenaría en esta matriz y se analiza la siguiente posible rotación de 3 días, borrándose de la memoria al no guardarse en ninguna variable.

Como se ha explicado inicialmente, una vez finalizado todo el proceso se obtienen todas las rotaciones posibles de 3 día en la matriz **R**. Esta matriz tendrá por tanto tantas filas como rotaciones posibles existan, y tantas columnas como número máximo de vuelos existan en un día multiplicado por los 3 días de la rotación. La matriz contendrá en cada una de las celdas los números de vuelos que forman cada una de las rotaciones para cada uno de los días establecidos.

Esta matriz se utiliza en los 4 archivos introducidos anteriormente para una vez determinadas todas las rotaciones posibles o potenciales, seleccionar dentro de este amplio abanico aquellas que permitan optimizar la función objetivo seleccionada.

## Rot1

### Código

```
function rot=rot1(tmin,v,s,l)
tmin=tmin/60;
s=(s(:,1)+s(:,2)/60);
l=(l(:,1)+l(:,2)/60);
%Rotaciones de 1 vuelo
m=size(v,1);
rot=v(:, [4 1]);
u=1;
%Rotaciones de 2 vuelos
for j=1:3
    a=find(v(:,4)==j);
    n=length(a);
    for i=1:n
        for k=1:n
            if v(a(i),3)==v(a(k),2) && s(a(k),1)-l(a(i))>=tmin
                rot(m+u,1:3)=[j v(a(i),1) v(a(k),1)];
                u=u+1;
            end
        end
    end
end
%Rotaciones de n vuelos
n=3;
b=find(rot(:,3));
z=length(b);
c=b;
s=z;
while c~=0
    for i=c(1):c(s)
        for k=b(1):b(z)
            if rot(i,n)==rot(k,2) && rot(i,1)==rot(k,1)
                rot(m+u,1:n+1)=[rot(k,1) rot(i,2:n) rot(k,3)];
                u=u+1;
            end
        end
    end
    n=n+1;
    y=size(rot,2);
    if n>y
        c=0;
    else
        c=find(rot(:,n));
        s=length(c);
    end
end
end
```

## Descripción

La función **rot1** tiene por objetivo hallar todas las rotaciones de 1 día posibles, según el requerimiento de tiempo mínimo de escala necesario. Las variables de entrada de esta función son el tiempo de escala (**tmin**), y las matrices con la información de los vuelos (**v**), horario de salida (**s**) y horario de llegada (**l**). La variable de salida es la matriz **rot**, donde se albergan todos los vuelos que se realizan en cada una de las rotaciones posibles de un solo día.

La primera tarea que realiza el algoritmo es pasar todos los horarios y tiempo a una misma unidad. En este caso se escoge pasar la variable **tmin** y las matrices **s** y **l** a horas, en formato decimal.

A continuación, se comienza a crear la matriz **rot**. En primer lugar, se consideran las rotaciones de un día que realicen solo un vuelo, siendo en este caso necesario pasar cada uno de los vuelos de la matriz **v** a la matriz **rot**.

Seguidamente se inicia el proceso de encontrar las rotaciones de un día que contengan solo 2 vuelos por día. Para ello, se busca para cada uno de los días aquellos vuelos cuyos aeropuertos de llegada y salida coinciden; si además de ello, el tiempo entre la llegada al aeropuerto del primer vuelo y la salida del segundo vuelo es mayor o igual que **tmin**, la secuencia es posible y se almacenan en las siguientes filas de la matriz **rot**. Este proceso se realiza para todos los vuelos de cada uno de los días que aparecen en la programación de vuelos introducida.

Por último, se continua el proceso de una forma similar, pero con el objetivo de encontrar rotaciones de más de 2 vuelos. Se comienza por tanto con 3 vuelos y se va aumentando sucesivamente hasta alcanzar el número máximo de vuelos que puede obtenerse en una rotación de un día. El proceso que se sigue en este paso consiste en combinar las secuencias de 2 vuelos con las secuencias de 1 solo vuelo que sean posibles, para obtener las rotaciones de 3 vuelos. Para obtener rotaciones de  $n$  vuelos, se combinan las secuencias de  $n-1$  vuelos con las secuencias de 1 vuelo. De esta forma, se consiguen obtener todas las rotaciones posibles de un día.

Los resultados obtenidos se almacenan en la matriz **rot**. Esta matriz contiene tantas filas como número posible de rotaciones de 1 día existan. En a las columnas, la primera de ellas contiene la información acerca del día en que se realiza la rotación (1, 2 ó 3) y en el resto de las columnas se almacenan los números de los vuelos que componen la rotación. Como existen rotaciones con mayor número de vuelos que otras, algunas columnas de una rotación contendrán ceros para rellenar las dimensiones de la matriz.

Esta función se utiliza dentro de la función **rotaciones1**. De este modo, una vez conocida todas las rotaciones de un día posible, se combinan teniendo en cuenta las necesidades de mantenimiento para obtener todas las rotaciones posibles de 3 días.



## ANEXO B

Se adjunta la tabla de *Excel* utilizada para introducir en los archivos elaborados con *Matlab* los datos sobre la programación horaria de los vuelos seleccionados para los días 6, 7 y 8 de agosto de 2021 de la aerolínea Kulula.

Los datos que aparecen en cada una de las columnas y que son necesarios de introducir en el programa son: número de vuelo, aeropuerto de salida, aeropuerto de llegada, hora de salida, hora de llegada, duración y coste de operación de cada vuelo.

Todos los datos recopilados se han obtenido a partir del buscador de vuelos de la página web de la compañía aérea, siendo por tanto un caso real de estudio, a excepción de los costes de operación, los cuales se introducen aleatoriamente y se utilizan simplemente para comprobar el correcto funcionamiento del archivo que permite minimizar los costes de operación, dado que los resultados obtenidos no serán reales por falta de conocimiento de estos mismos costes en la práctica.

**Tabla 26:** Datos vuelos Kulula

Número de vuelo	Aeropuerto salida	Aeropuerto llegada	Día del vuelo	Hora salida (h)	Hora salida (min)	Hora llegada (h)	Hora llegada (min)	Duración	Coste
1	1	2	1	5	55	7	0	1.08	90
2	1	2	1	11	45	12	50	1.08	71
3	1	2	1	14	35	15	40	1.08	86
4	1	2	1	17	30	18	35	1.08	65
5	1	4	1	7	55	10	5	2.17	75
6	1	4	1	8	50	11	0	2.17	84
7	1	4	1	13	40	15	50	2.17	77
8	1	4	1	14	50	17	0	2.17	62
9	1	3	1	6	45	8	40	1.92	83
10	1	3	1	8	45	11	40	2.92	87
11	1	3	1	15	25	17	20	1.92	80
12	2	1	1	6	0	7	5	1.08	66
13	2	1	1	7	50	8	55	1.08	95
14	2	1	1	13	30	14	35	1.08	86
15	2	1	1	16	20	17	25	1.08	96
16	2	1	1	19	20	20	25	1.08	53
17	4	1	1	10	50	12	50	2.00	63
18	4	1	1	11	45	13	45	2.00	86
19	4	1	1	16	35	18	35	2.00	63
20	4	1	1	17	45	19	45	2.00	58
21	3	1	1	9	15	10	55	1.67	67
22	3	1	1	12	20	14	0	1.67	84
23	3	1	1	18	0	19	40	1.67	86
24	1	2	2	11	45	12	50	1.08	93
25	1	2	2	14	35	15	40	1.08	83
26	1	2	2	17	30	18	35	1.08	80
27	1	4	2	7	55	10	5	2.17	65
28	1	4	2	11	50	14	0	2.17	54
29	1	4	2	13	40	15	50	2.17	69
30	1	4	2	14	50	17	0	2.17	60

31	1	3	2	8	45	11	40	2.92	73
32	2	1	2	7	50	8	55	1.08	87
33	4	1	2	10	50	12	50	2.00	84
34	4	1	2	11	45	13	45	2.00	52
35	4	1	2	14	40	16	40	2.00	90
36	4	1	2	16	35	18	35	2.00	70
37	4	1	2	17	45	19	45	2.00	97
38	3	1	2	12	20	14	0	1.67	88
39	1	2	3	11	45	12	50	1.08	86
40	1	2	3	14	35	15	40	1.08	70
41	1	2	3	17	30	18	35	1.08	52
42	1	4	3	7	50	10	5	2.25	89
43	1	4	3	8	50	11	0	2.17	81
44	1	4	3	11	40	14	0	2.33	68
45	1	4	3	14	50	17	0	2.17	60
46	1	3	3	8	45	11	40	2.92	65
47	1	3	3	15	25	17	20	1.92	72
48	4	1	3	10	50	12	50	2.00	79
49	4	1	3	14	40	16	40	2.00	84
50	4	1	3	16	35	18	35	2.00	99
51	4	1	3	17	45	19	45	2.00	75
52	3	1	3	12	20	14	0	1.67	77
53	3	1	3	18	0	19	40	1.67	85
54	1	2	1	19	25	20	30	1.08	60
55	1	2	2	19	25	20	30	1.08	86
56	1	2	3	19	25	20	30	1.08	52
57	2	1	2	13	30	14	35	1.08	98
58	2	1	2	16	20	17	25	1.08	57
59	2	1	2	19	20	20	25	1.08	84
60	2	1	3	13	30	14	35	1.08	78
61	2	1	3	16	20	17	25	1.08	81
62	2	1	3	19	20	20	25	1.08	75