

Survey de Problemas de Optimización Combinatoria en la Gestión Aeronáutica.

Memòria del Treball Fi de Grau  
Gestió Aeronàutica  
realitzat per  
David Roldan Martínez  
i dirigit per  
Daniel Riera Terrén  
Sabadell, 6 de Juliol de 2016.

El sotasignat, Daniel Riera Terrén  
Professor/a de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en/na David Roldan Martínez.

I per tal que consti firma la present.

Signat:



Sabadell, 6 de Juliol de 2016.

## DE RESUM – TREBALL FI DE GRAU DE L'ESCOLA D'ENGINYERIA

<b>Títol del Treball Fi de Grau (obligatori en tres idiomes: Català, Castellà, Anglès)</b>	
<b>Autor[a]:</b> David Roldan Martínez	<b>Data:</b> 07-2016
<b>Tutor[a]/s[es]:</b> Daniel Riera Terrén	
<b>Titulació:</b> Gestió Aeronàutica.	
<b>Paraules clau</b> (mínim 3) <ul style="list-style-type: none"><li>• Català: Optimització Combinatoria, Algoritmes, Aeronàutica, Problemes.</li><li>• Castellà: Optimización Combinatoria, Algoritmos, Aeronáutica, Problemas.</li><li>• Anglès: Combinatorial Optimization, Algorithms, Aeronautics, Problems.</li></ul>	
<b>Resum del Treball Fi de Grau</b> (extensió màxima 100 paraules) <ul style="list-style-type: none"><li>• Català:<p>Aquest treball tracta de reunir problemas d'optimització combinatoria de l'àmbit de la gestió aeronàutica i de trobar diferents solucions que s'han proposat per resoldre'ls. Es recullen sis problemas distints i s'ofereix una breu explicació de les solucions trobades, proporcionant referències al treball original.</p></li><li>• Castellà:<p>Este trabajo trata de reunir problemas de optimización combinatoria en el ámbito de la gestión aeronáutica, así como describir distintas soluciones que se han propuesto para resolverlos. Se recogen seis problemas distintos y se ofrece una pequeña explicación de las soluciones encontradas, proporcionando referencias al trabajo original.</p></li><li>• Anglès:<p>This paper tries to gather combinatorial optimization problems in the field of aviation management and describes the different solutions that have been proposed to solve them. Six different issues are collected and a brief explanation of the solutions is provided. Also the references to the original work are given.</p></li></ul>	

# Índice

1. Organización de la memoria. ....	6
2. Objetivos .....	7
3. Introducción. ....	8
3.1 Un ejemplo de problema combinatorio: El problema del viajero. ....	9
4. Estudio de viabilidad del proyecto. ....	11
5. Plantilla. ....	12
5.1 Modelo de la plantilla.....	12
6. Exposición Problemas. ....	13
Problema 1. Asignación de tripulaciones. ....	13
Solución 1.1 .....	14
Solución 1.2 .....	14
Solución 1.3 .....	16
Solución 1.4 .....	16
Solución 1.5 .....	17
Solución 1.6 .....	18
Problema 2. Asignación de flotas. ....	19
Solución 2.1 .....	20
Solución 2.2 .....	20
Solución 2.3 .....	21
Solución 2.4 .....	21
Solución 2.5 .....	22
Problema 3. Gestión de Ingresos. ....	24
Solución 3.1 .....	25
Solución 3.2 .....	26
Solución 3.3 .....	26
Solución 3.4 .....	27
Solución 3.5 .....	27
Solución 3.6 .....	28
Solución 3.7 .....	29
Problema 4. Capacidad del espacio aéreo. ....	30
Solución 4.1 .....	31
Solución 4.2 .....	31
Solución 4.3 .....	32

Solución 4.4 .....	32
Solución 4.5 .....	33
Solución 4.6 .....	33
Problema 5: Ground Holding Policy. ....	35
Solución 5.1 .....	36
Solución 5.2 .....	36
Solución 5.3 .....	37
Solución 5.4 .....	37
Solución 5.5 .....	38
Solución 5.6 .....	38
Problema 6: Asignación de rutas.....	39
Solución 6.1 .....	40
Solución 6.2 .....	40
Solución 6.3 .....	41
Solución 6.4 .....	42
Solución 6.5 .....	43
Solución 6.6 .....	44
7. Explicación algoritmos.....	45
7.1 Algoritmos geneticos.....	45
7.2 Branch and Bound.....	46
7.3 Algoritmo de optimización de la colonia de hormigas.....	46
7.4 Simulación de recocido. ....	47
7.5 Algoritmo de Generación de columnas.....	48
7.6 Algoritmo de evolución diferencial. ....	48
7.7 Algoritmo adaptativo. ....	49
7.8 Algoritmo Heurístico. ....	49
7.9 Descomposición de Benders. ....	50
8. Tabla de algoritmos utilizados. ....	51
9. Conclusiones.....	54
10. Bibliografía.....	56
10.1 Artículos .....	56
10.2 Webs.....	59

## **1. Organización de la memoria.**

---

La presente memoria está dividida en diez apartados. Desde el apartado 2 al 5, se presentan los objetivos del presente trabajo, así como una introducción a la temática tratada, un estudio de viabilidad del proyecto y la exposición de la plantilla que se utilizará para presentar las soluciones de los problemas escogidos.

En el apartado 6, se exponen los seis problemas encontrados, así como sus soluciones, debidamente descritas y referenciadas.

Por último, del apartado 7 al 10, encontramos una pequeña explicación a algunos de los algoritmos utilizados en la resolución de los problemas, así como una tabla donde se presentan las metodologías de solución escogidas, ordenadas cronológicamente. Por último, encontramos el apartado de las conclusiones, así como la bibliografía del trabajo.

## 2. Objetivos

---

- Reunir toda la información posible sobre problemas combinatorios relacionados con el mundo aeronáutico.
- Analizar y clasificar de forma adecuada la información recogida. Esto incluye una explicación del problema concreto, así como las soluciones que se han dado al mismo y los algoritmos y metodologías utilizados.
- Proporcionar referencias, así como citar y proporcionar links a los documentos originales de los autores para que el lector pueda acceder al documento original.
- Que este documento sirva de ayuda para aquellas personas que busquen información relacionada con problemas de optimización combinatoria en el mundo de la aeronáutica.
- Agrupar todas las metodologías de resolución de los problemas en una tabla al final del documento, donde se pueda ver a qué problema pertenecen, el año en el que el documento fue elaborado y la metodología de solución utilizada.

### 3. Introducción.

---

Se considera optimización aquel proceso que persigue el encontrar la mejor solución existente para un problema dado.

En un problema de optimización, en general, existe un número finito de variables que pueden tomar determinados valores, unas restricciones (que limitan los posibles valores que tomarán las variables), y una función objetivo. La función de las herramientas de optimización (como por ejemplo, la programación lineal y el método del simplex) consiste en encontrar valores para las variables que consigan que la función objetivo alcance su máximo o mínimo valor (dependiendo de si el objetivo es maximizar o minimizar dicho valor).

La optimización combinatoria está dentro de las matemáticas aplicadas y de la ciencia computacional teórica. Su función consiste en tratar de encontrar una solución o un objeto óptimos, dado un problema con un número finito de soluciones u objetos.

Existen muchos problemas de optimización. Algunos de ellos se pueden resolver con facilidad dependiendo de su complejidad y/o el tamaño de la instancia. Un ejemplo son los problemas lineales. En este tipo de problemas, se da el caso de que tanto la función objetivo como las restricciones proporcionadas son expresiones del tipo lineal y pueden (normalmente) resolverse de manera bastante eficiente con métodos como el simplex.

Sin embargo, la mayoría de problemas prácticos que nos encontramos suelen ser complejos, y por lo tanto difíciles de resolver.

Este hecho queda definido por el término científico de NP-Hard, un término empleado para definir la complejidad de un problema. Definido de forma simple, podríamos decir que en este tipo de problemas de optimización “difíciles”, no se puede asegurar que podamos encontrar la solución óptima en un término de tiempo razonable.

La optimización combinatoria es un componente muy importante de la optimización matemática, relacionada con algoritmos complejos. Los algoritmos basados en la optimización combinatoria, tratan de dar solución a problemas complejos como los Np-Hard, que cuentan con un espacio de soluciones inmenso. En este tipo de problemas, debido a lo amplio del espacio de soluciones, se hace imposible encontrar la solución más óptima para el problema, ya que aun utilizando miles de ordenadores a la vez, el tiempo para encontrar la solución del problema es demasiado grande.

Así pues, según lo que sabemos hasta ahora, podemos dividir los algoritmos que se encargan de encontrar soluciones a los problemas de optimización en dos grandes grupos:



**Algoritmos exactos/ de búsqueda completa:** Exploran todas las soluciones posibles del problema y dan como resultado la óptima (es decir, la mejor solución posible). Sin embargo, este tipo de algoritmos no suele resultar eficaz en problemas complejos o con tamaños de entrada muy grandes, ya que el tiempo que se necesita para explorar todo el espacio de soluciones es excesivo.

**Algoritmos inexactos/de búsqueda incompleta:** Tratan de obtener una solución suficientemente buena, aunque no se pueda probar que sea la óptima. Este tipo de algoritmos reduce el espacio de soluciones y trata de buscar soluciones eficientes, de forma la relación tiempo/calidad de la solución final sean aceptablemente buenos.

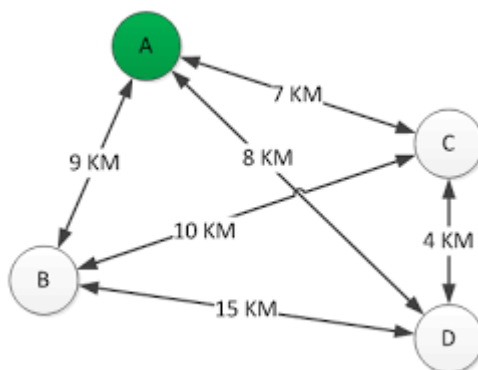
Un excelente ejemplo para discernir la aplicación de los algoritmos de búsqueda incompleta frente a los algoritmos de búsqueda completa, lo podemos encontrar en el problema del viajero, un problema NP-Hard muy estudiado y complejo, ya que su espacio de soluciones crece de forma factorial.

### 3.1 Un ejemplo de problema combinatorio: El problema del viajero.

Este problema se formuló por primera vez en 1930 y consiste en resolver la siguiente pregunta: Si tenemos una lista de ciudades, y la conocemos la distancia entre cada par de estas, ¿Cuál es el camino más corto para visitar cada una de estas ciudades una sola vez y regresar a la ciudad de origen?

A simple vista puede parecer un problema fácil de resolver y de entender. Desde luego entenderlo y representarlo es algo sencillo, y resolverlo (si hay pocas ciudades en la lista) tampoco resulta complicado.

El reto real lo encontramos cuando pensamos en la escalabilidad del problema a medida que aumentamos el número de ciudades a visitar. Vamos a suponer tres casos diferentes para poder comprobar como el número de rutas posibles (o espacio de soluciones) crece de forma exponencial a medida que aumentamos las ciudades que el viajero tiene que visitar.



*Figura 1. Representación gráfica del problema. [1]*

Para empezar, debemos saber que la cantidad de rutas posibles en una red está determinada por la siguiente ecuación:  $x = (n-1)!/2$ , donde  $x$  es el número de rutas posibles y  $n$  la cantidad de ciudades en nuestra lista.

Dicho esto, podemos realizar unos simples cálculos para ver cómo escala el problema.

*-Para 5 ciudades posibles, habrían 12 rutas diferentes. Podríamos solucionar el problema a mano sin mucha dificultad.*

*-Para 9 ciudades posibles, existirían 20.160 rutas diferentes. Esto ya no podría ser resuelto a mano, pero sería viable resolver el problema con un ordenador utilizando un **algoritmo exacto** (es decir, obteniendo la mejor solución posible). En términos de tiempo, esto sería algo factible, ya que actualmente incluso un procesador de gama baja es capaz de realizar millones de operaciones por segundo, resolviendo el problema en unos instantes.*

*-Para 20 ciudades diferentes, existen más de  **$6 \cdot 10^{16}$**  rutas posibles. Esto significa que, explorando todo el espacio de soluciones con un algoritmo exacto, un ordenador capaz de calcular un millón de rutas por segundo tardaría 1928 años en conseguir la mejor ruta.*

Como podemos observar, las soluciones del problema escalan de forma factorial, haciendo que incluso computacionalmente sea imposible resolver el problema de forma exacta (en teoría el problema se resuelve, pero en millones de años). Así pues en un problema de este tipo, se ha optado por utilizar algoritmos de optimización incompletos, que si bien no nos ofrecen una solución óptima, sí que nos ofrecen una solución eficiente en términos de tiempo y resultados.

Como vemos, existen a grandes rasgos, dos tipos de problemas, P y NP, además de diferentes algoritmos completos y algoritmos incompletos. Dado que la temática de este trabajo se centra en problemas combinatorios (normalmente NP), se tratará de recolectar diferentes tipos de problemas combinatorios relacionados con la aeronáutica y que métodos se utilizan para darles solución.

#### **4. Estudio de viabilidad del proyecto.**

---

En este apartado, veremos cuáles son los principales riesgos que tiene este trabajo de que los objetivos propuestos no se cumplan:

- **Temática concreta, pero a la vez relacionada con muchos campos distintos:**
  - La optimización combinatoria está relacionada con muchos campos a la vez, como la teoría computacional, las matemáticas y problemas de todo tipo. Es por este hecho que encontrar información al respecto puede ser bastante difícil, ya que se encontrará muy fraccionada y estará especializada en aquel campo en el que se esté consultando.
- **Información disponible en inglés:**
  - Este es un hecho que puede poner en riesgo los diferentes tiempos del trabajo, ya que toma mucho más tiempo el leer un texto en una lengua diferente a la materna. Es por eso que el riesgo de que los tiempos previstos para la lectura y documentación sobre el tema aumenten, pese a haber intentado realizarlos desde la perspectiva de que estamos leyendo textos en inglés.
- **Se buscan problemas concretos:**
  - Al tratarse principalmente de un trabajo de investigación donde se pretenden exponer todos los ejemplos de problemas combinatorios posibles relacionados con la aeronáutica, es posible que la búsqueda de problemas concretos para la temática analizada pueda poner en riesgo al proyecto. Al ser problemas de un tema muy concreto, esto puede resultar en una gran dificultad para encontrarlos, de forma que la duración y el tiempo de realización del proyecto puedan verse seriamente afectados.
- **Algoritmos y problemas muy complejos:**
  - Al hablar de problemas combinatorios, estamos tratando con una temática sumamente compleja. Si a esto además, le añadimos el tener que dar una explicación razonablemente simplificada de cómo actúan los algoritmos diseñados para resolver este tipo de problema, podemos toparnos con dificultades como un lenguaje o procedimientos matemáticos que sobrepasen nuestros conocimientos. Este hecho supondrá un retraso en el tiempo total de realización del trabajo, ya que probablemente por cada algoritmo o problema nuevo que tratemos de describir, deberemos emplear cierto tiempo en documentarnos a cerca de los procedimientos o lenguaje matemático usado por cada algoritmo o problema.

## 5. Plantilla.

---

Debido a que el objetivo del presente trabajo consiste en recopilar, presentar y ofrecer fuentes sobre distintos problemas combinatorios, se ha decidido optar por realizar una plantilla. Esto permitirá presentar los distintos problemas encontrados de una forma clara, concisa y uniforme, cosa que facilitará la lectura de este artículo al lector.

### 5.1 Modelo de la plantilla.

<b>Título del problema.</b>	
<b>Descripción del problema:</b> <i>En este punto se realizará una pequeña introducción para describir el problema tratado.</i>	
<b>Soluciones Propuestas</b>	
<b>Solución 1</b>	
<b>Autores:</b>	<i>Nombre del autor/es que propusieron la solución.</i>
<b>Año:</b>	<i>Año de propuesta de la solución.</i>
<b>Algoritmo Utilizado:</b>	<i>Nombre del algoritmo/s utilizados.</i>
<b>Explicación:</b>	<i>Breve explicación de la solución.</i>
<b>Más información:</b>	<i>Enlace al documento original del autor, donde se explica la solución en detalle.</i>
<b>Referencia:</b>	<i>Referencia del artículo o documento del autor donde se encuentra la solución propuesta. El formato de cita utilizado es el ISO 690.</i>

## 6. Exposición Problemas.

En este apartado se presentarán los seis problemas escogidos, así como las diferentes soluciones encontradas para cada uno de ellos. Los problemas se presentarán dentro del formato de la plantilla presentada en el apartado 4.

### **Problema 1. Asignación de tripulaciones.**

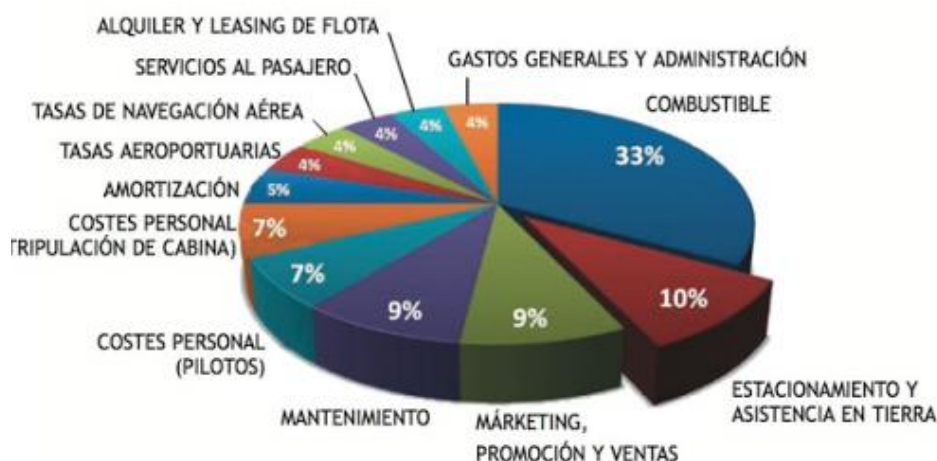
#### **Descripción del problema:**

Uno de los problemas fundamentales de la optimización combinatoria es el problema de asignación. Este trata de asignar, para cada caso concreto, la mejor persona o recurso para realizar un trabajo determinado con el menor coste posible.

Este problema se traslada a la industria aérea con el problema de asignación de tripulaciones.

La asignación de tripulaciones supone una tarea muy importante en las líneas aéreas, ya que si se gestiona de una forma eficiente, se pueden reducir muchos costes asociados al personal y así conseguir aumentar los beneficios de la aerolínea.

**ESTRUCTURA CARACTERÍSTICA DE COSTES EN UNA COMPAÑÍA AÉREA DE TIPO TRADICIONAL**



*Gráfico 1. Estructura de costes en una compañía aérea tradicional [2].*

Como se puede apreciar en el gráfico, el coste de personal supone el 14% de los costes totales de una aerolínea (sumando tripulación de cabina y pilotos), quedando solo por detrás del gasto en combustible. Por ello, el realizar una buena gestión de las tripulaciones puede resultar en un ahorro importante, tanto en tiempo como en dinero que no debe ser desdeñado.

Este problema viene siendo estudiado por las compañías aéreas desde hace décadas y se han empleado múltiples técnicas para resolverlo. Un ejemplo temprano de un estudio sobre cómo enfocar y resolver este problema de forma efectiva, fue hecho por Arabeyre en 1969, con el objetivo de optimizar la adecuación de los diferentes tripulantes a cada vuelo.

Sin embargo, la escala del problema fue creciendo junto con las aerolíneas, convirtiéndose en un problema NP-Hard, con matrices de información inmensas ( $10^4$  filas y entre 500 y 1000 columnas), haciendo que el espacio de soluciones fuese intratable con métodos convencionales.

## Soluciones Propuestas

### Solución 1.1

<b>Autores:</b>	Walid Moudani, Lebanese University.
<b>Año:</b>	2000.
<b>Algoritmo Utilizado:</b>	División de problema complejo en sub-problemas de menor dificultad, algoritmo genético.
<b>Explicación:</b>	Se propone dividir el problema inicial de asignación en varios problemas más pequeños y asequibles. Es este caso, el algoritmo utilizado no es completo (no se consigue la solución óptima).  La aplicación de esta solución se hace utilizando datos de 75 miembros de tripulación de vuelo, realizándose 275 emparejamientos y más de 4000 horas de vuelo.
<b>Más información:</b>	<a href="https://www.researchgate.net/publication/228774764_A_heuristic_genetic_algorithm_approach_for_the_airline_crew_scheduling_problem">https://www.researchgate.net/publication/228774764_A_heuristic_genetic_algorithm_approach_for_the_airline_crew_scheduling_problem</a>
<b>Referencia:</b>	KERATI, Sangcham, et al. A heuristic genetic algorithm approach for the airline crew scheduling problem. En Workshop on Multiple Objective Metaheuristics, Paris. 2002.

### Solución 1.2

<b>Autores:</b>	Onur Ozemir y Chilukuri Mohan, Syracuse University.
<b>Fecha:</b>	2001.
<b>Algoritmo Utilizado:</b>	Algoritmo Genético (GraGA).
<b>Explicación:</b>	En su tratamiento del problema, tratan de respetar las distintas necesidades de los tripulantes, minimizando el número de tripulantes que se necesitan y intentado equilibrar la carga de trabajo de los mismos.  Se utiliza un algoritmo genético que es aplicado a un grafo dirigido. En él, se representan varias variables del problema

así como restricciones específicas.

El algoritmo utilizado resulta ser un muy buen método, ya que comparativamente con otros algoritmos, este consigue una mejor distribución de la carga de trabajo entre tripulaciones, así como un menor número de tripulaciones requeridas para realizar los vuelos. Se concluye que la representación en grafo propia de esta solución es muy útil para problemas de este tipo (que cuentan con muchas restricciones), ya que tiende a reducir el espacio de soluciones.

Por último, los autores presentan esta tabla de resultados, donde se muestra la efectividad del citado algoritmo en la resolución de 28 problemas de asignación de tripulaciones reales. En esta tabla pueden verse los resultados de 5 algoritmos de resolución de este problema, incluido el GraGA.

The cost of best solution obtained using various algorithms; #F denotes the number of flights, #C denotes the number of columns, and #D denotes the number of days spanned by the schedule

Problem Name	#F	#C	#D	Chvatal SCP [5]	Bäck SCP [6]	SCP based CBGA [2]	Greedy SCP(new)	GraGA
GBXORD_M_26	26	16	1	<b>3600</b>	<b>3600</b>	<b>3600</b>	<b>3600</b>	<b>3600</b>
Delta_B727_43	43	40	1	9698	9730	9356	9730	<b>8912</b>
MandarinAir	28	52	7	22609.7	17609	18546.3	17609	<b>15992.1</b>
GBXSEA_M_55	55	76	1	8340	7540	7120	7553	<b>7112</b>
OrlandoAir	54	81	1	9636.6	8914.1	<b>8790.3</b>	8914.1	8975.33
GBXDFW_M_63	63	107	1	8400	<b>7054</b>	7080	7107	7092
GBXSFO_M_46	46	122	2	6900	<b>4500</b>	<b>4500</b>	4527	<b>4500</b>
Delta_B737_85	85	123	2	16070	15903	15598	15903	<b>15576</b>
LynxAir	32	133	7	10441.5	9088.5	9756.5	9118.5	<b>8349.5</b>
GBXDFW_M_86	86	182	1	11400	9444	<b>9120</b>	9432	9452
GBXSEA_M_75	75	184	1	12940	10310	<b>9680</b>	10310	10100
GBXMIA_M_93	93	205	1	14752.8	12072.8	12100.8	12072.8	<b>11707.8</b>
GBXDEN_M_72	72	295	1	10050	8750	8290	8750	<b>8180</b>
Delta_MD88_173	173	399	2	39279.7	38020.5	<b>35000.7</b>	37696.5	39435.5
GBXSEA_M_110	110	453	1	19270	<b>13680</b>	14070	14010	14500
Delta_102	102	857	1	16840	13272	12592	13310	<b>12210</b>
GBXATL_M_118	118	904	1	14114	9755	10679	9740	<b>9580</b>
GBXORD_MT_73	73	1008	2	14984	11414	10784	11530	<b>10678</b>
GBXBOS_M_73	73	1105	1	11400	6360	6380	6360	<b>5800</b>
Delta_B727_86	86	1466	2	20248.9	18744.5	<b>16946</b>	18440.5	20732.3
Delta_133	133	1698	1	23806.9	20335.9	20043.1	20365.8	<b>18961.3</b>
Delta_MD88_333	333	3428	2	78343.8	71984.7	78421	<b>71714.7</b>	77440.8
Delta_MD88_380	380	4415	2	89482.1	82463.9	88907.5	<b>82121.7</b>	90517
OrcaAir	83	5318	7	22042.8	18207.3	19356.5	17593.5	<b>16914.2</b>
Delta_217	217	5618	2	40398.6	<b>29178.6</b>	33340.6	29224.6	30006.6
GBXSFO_MT_97	97	15250	2	16593.5	<b>11298.5</b>	11473.5	11319.5	<b>11298.5</b>
GBXDFW_MT_158	158	14254	2	27444.2	<b>23184</b>	23854.8	24327	30026.2
THY_74	74	21308	7	22636.8	15431.9	17085	19420.9	<b>12321.5</b>
Number of best results				1	7	7	3	17

*Tabla 1. Comparación de resultados obtenidos por el algoritmo GraGA.[4]*

Como vemos, el algoritmo consigue encontrar una mejor solución frente a los otros cuatro en 17 de los 28 problemas propuestos, una cifra significativa que demuestra la efectividad del algoritmo.

**Más información:**

<http://www.pcporoje.com/filedata/644547.pdf>

<b>Referencia:</b>	<i>OZDEMIR, H. Timucin; MOHAN, Chilukuri K. Flight graph based genetic algorithm for crew scheduling in airlines. Information Sciences, 2001, vol. 133, no 3, p. 165-173.</i>
--------------------	---

### **Solución 1.3**

<b>Autores:</b>	Anne Mercier.
<b>Fecha:</b>	2005.
<b>Algoritmo Utilizado:</b>	Descomposición de Benders, generación de columnas y algoritmo de Branch-and-Bound.
<b>Explicación:</b>	<p>En este enfoque de resolución del problema, resuelve de forma integrada tanto el problema de asignación de rutas, como el problema de Asignación de tripulaciones, utilizando la Descomposición de Benders.</p> <p>Se divide el problema en dos, el problema maestro (que será el de asignación de rutas) y el sub-problema que será el problema de asignación de tripulaciones. Debido a las características de los problemas, estos se resuelven por generación de columnas.</p> <p>Una vez solucionados por separado, se utiliza el algoritmo de Branch-and-Bound para encontrar una solución conjunta que minimiza tanto el coste de la ruta como el coste de la tripulación asignada para realizar los vuelos.</p> <p>Los datos utilizados son reales y, en comparación con los métodos usuales de planificación, esta solución del problema consigue reducir los costes derivados por la asignación de tripulaciones y la asignación de rutas en un 9'4%.</p>
<b>Más información:</b>	<a href="https://www.researchgate.net/publication/222403543_A_computational_study_of_Benders_decomposition_for_the_integrated_aircraft_routing_and_crew_scheduling_problem">https://www.researchgate.net/publication/222403543_A_computational_study_of_Benders_decomposition_for_the_integrated_aircraft_routing_and_crew_scheduling_problem</a>

<b>Referencia:</b>	<i>MERCIER, Anne; CORDEAU, Jean-François; SOUMIS, François. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers &amp; Operations Research, 2005, vol. 32, no 6, p. 1451-1476.</i>
--------------------	---

### **Solución 1.4**

<b>Autores:</b>	David Levine.
<b>Fecha:</b>	1996.
<b>Algoritmo Utilizado:</b>	Algoritmo genético híbrido (con heurística de búsqueda local).



<b>Explicación:</b>	<p>El autor se propone resolver el problema de asignación de tripulaciones con un algoritmo genético híbrido, ya que explica que el hecho de combinar un algoritmo genético junto con una heurística local se ha demostrado beneficioso en otras ocasiones.</p> <p>El autor realiza comparaciones con otros algoritmos utilizando cuarenta problemas reales de asignación de tripulaciones. El algoritmo genético híbrido, consigue encontrar la mejor solución en la mitad de los casos, y buenas soluciones para otros nueve. Comparativamente con los algoritmos de branch-and-bound y branch-and-cut, el algoritmo genético híbrido demuestra ser ligeramente inferior al algoritmo branch-and-bound, pero obtiene peores resultados que el algoritmo de branch-and-cut.</p>
<b>Más información:</b>	<a href="http://140.221.6.23/pub/tech_reports/reports/P557.pdf">http://140.221.6.23/pub/tech_reports/reports/P557.pdf</a>
<b>Referencia:</b>	LEVINE, David. Application of a hybrid genetic algorithm to airline crew scheduling. <i>Computers &amp; Operations Research</i> , 1996, vol. 23, no 6, p. 547-558.
<b><u>Solución 1.5</u></b>	
<b>Autores:</b>	Budi Santosa, Andiek Sunarto, Arief Rahman.
<b>Fecha:</b>	2010.
<b>Algoritmo Utilizado:</b>	Algoritmo de evolución diferencial (personalizado).
<b>Explicación:</b>	<p>En este artículo se propone utilizar el algoritmo de evolución diferencial para resolver el problema de asignación de tripulaciones, aunque el algoritmo se ha modificado por los autores. Esta diferenciación se basa en utilizar intercambios al azar (<i>random swap</i>) como operador de la mutación.</p> <p>En la creación de las restricciones se contemplan algunas tales como las horas máximas de vuelo para piloto y copiloto (110 horas por mes), los horas máximas de vuelos para el personal de cabina (120 horas al mes) y la condición de que el número máximo de despegues que puede realizar un piloto al mes es de 90, además de otras restricciones de las tripulaciones.</p> <p>Se explica que este algoritmo se ha comparado con dos métodos utilizados para resolver el problema, la generación de columnas y MOSI (el método utilizado por la aerolínea colaboradora para realizar su asignación de tripulaciones).</p> <p>Se expone un gran número de tablas y de resultados que pueden ser de interés para el lector (se recomienda ver el consultar el documento referenciado).</p>

	Resumiendo, los autores concluyen que la clave para la efectividad de este algoritmo, es la selección del factor de probabilidad de mutación y de “crossover” (cuyo valor óptimo fijan entre 0,1 y 0,5). También explican que el algoritmo de evolución diferencial es superior las dos soluciones con las que se compara cuando se simula con pequeños conjuntos de datos, ofreciendo mejores soluciones tanto en el cumplimiento de restricciones, como en la calidad de la asignación de tareas a la tripulación.
<b>Más información:</b>	<a href="http://www.scirp.org/journal/PaperInformation.aspx?PaperID=2910">http://www.scirp.org/journal/PaperInformation.aspx?PaperID=2910</a>
<b>Referencia:</b>	SANTOSA, Budi, et al. Using differential evolution method to solve crew rostering problem. <i>Applied Mathematics</i> , 2010, vol. 1, no 04, p. 316.
<b>Solución 1.6</b>	
<b>Autores:</b>	Guang-Feng Deng y Woo-Tsong Lin.
<b>Fecha:</b>	2011.
<b>Algoritmo Utilizado:</b>	Algoritmo de optimización la colonia de hormigas.
<b>Explicación:</b>	<p>El problema se plantea como el problema del viajero y una vez planteado, se resuelve mediante la aplicación del algoritmo de optimización por colonia de hormigas. Dicho algoritmo se muestra efectivo en este problema concreto para conseguir minimizar los costes derivados del uso de tripulaciones. Esto se logra tratando de programar vuelos que resulten efectivos para evitar que los tripulantes deban pasar tiempo sin una ocupación (ya sea en hoteles, horas muertas o realizando escalas).</p> <p>Los autores realizan comparaciones de las soluciones obtenidas con este algoritmo y las comparan con las soluciones ofrecidas por algoritmos genéticos, utilizando datos reales proporcionados por aerolíneas y concluyen que este algoritmo resulta más efectivo que los algoritmos genéticos para resolver el problema de asignación de tripulaciones. Por ultimo acaban añadiendo que se realizara un estudio posterior para que el algoritmo considere algunos factores adicionales que se centren en mejorar la satisfacción de los miembros de la tripulación.</p>
<b>Más información:</b>	<a href="http://www.sciencedirect.com/science/article/pii/S0957417410011978">http://www.sciencedirect.com/science/article/pii/S0957417410011978</a>
<b>Referencia:</b>	DENG, Guang-Feng; LIN, Woo-Tsong. Ant colony optimization-based algorithm for airline crew scheduling problem. <i>Expert Systems with Applications</i> , 2011, vol. 38, no 5, p. 5787-5793.

## **Problema 2. Asignación de flotas.**

### **Descripción del problema:**

Una asignación de flota eficiente es una de las premisas obligatorias que debe seguir una compañía aérea para lograr beneficios. Si tomamos como referencia el gráfico de costes de las aerolíneas que se presentó con anterioridad, podemos concluir que, aproximadamente un 65% de los costes de las aerolíneas pueden verse directamente afectados (positivamente o negativamente) por la gestión de la flota (33% combustible + 10% estacionamiento y servicios en tierra +14% costes de tripulación + 8% tasas).

El problema de asignación de flotas es precisamente complejo por este hecho, porque debe tener en cuenta muchas variables y también porque debe ser lo suficientemente dinámico y rápido para adaptarse a los hechos presentes. Se trata básicamente de, una vez creados los horarios de los vuelos, ver que aviones asignamos a cada uno de los vuelos, teniendo en cuenta las limitaciones de cada tipo de avión (capacidad de pasajeros, alcance, límites operacionales, límites de mantenimiento etc) y tratando de encontrar la mejor solución posible a nivel de coste global.

Vamos a suponer, por ejemplo que contamos con un algoritmo que consigue resolver el problema de asignación de flotas de forma óptima en un tiempo de ocho horas. Esta solución sería perfectamente aplicable en términos de planificación a largo y medio plazo, pero en términos operacionales sería totalmente inviable. Si durante la operativa uno de nuestros aviones quedase AOG (Aircraft On Ground), la aerolínea sería incapaz de reorganizar de forma eficiente su flota para cubrir la baja de esta aeronave.

De esta forma vemos que los modelos destinados a resolver este problema, deberán resolverse de forma relativamente rápida, o su efectividad para la aplicación operacional sería nula. También podría ser interesante contar con un algoritmo óptimo para el largo/medio plazo y otro para el corto plazo, cuando necesitemos una solución al momento que sea suficientemente óptima.

## Soluciones Propuestas

### Solución 2.1

<b>Autor/es:</b>	Bélanger.
<b>Fecha:</b>	2006.
<b>Algoritmo Utilizado:</b>	Modelo matemático estructura de red multi-artículo no lineal y concepto de homogeneidad, aproximación heurística.
<b>Explicación:</b>	<p>En esta solución del problema, se toma como condición que se asignará el tipo de avión del que se consiga el mayor beneficio por vuelo.</p> <p>Se consideran los siguientes factores:</p> <ul style="list-style-type: none"><li>-Ventanas de tiempo de salida.</li><li>-Trayectos cuya duración dependa del tipo de avión asignado.</li><li>-Principio de homogeneidad aplicado para asignar siempre el mismo tipo de avión para cada trayecto concreto, independientemente de su fecha de realización.</li></ul>
<b>Más información:</b>	<a href="https://www.researchgate.net/profile/Jacques_Desrosiers/publication/223805452_Weekly_airline_fleet_assignment_with_homogeneity/links/0fcfd5086aa67b7a49000000.pdf">https://www.researchgate.net/profile/Jacques_Desrosiers/publication/223805452_Weekly_airline_fleet_assignment_with_homogeneity/links/0fcfd5086aa67b7a49000000.pdf</a>
<b>Referencia:</b>	BÉLANGER, Nicolas, et al. Weekly airline fleet assignment with homogeneity. <i>Transportation Research Part B: Methodological</i> , 2006, vol. 40, no 4, p. 306-318.

### Solución 2.2

<b>Autor/es:</b>	Suhl.
<b>Fecha:</b>	1999.
<b>Algoritmo Utilizado:</b>	Programación entera mixta y reducción de coeficientes.
<b>Explicación:</b>	<p>Se toma como principal objetivo la optimización de la ventana de tiempos de salida. Los problemas se presentan de forma independiente para cada flota de aviones.</p> <p>Nuevamente el objetivo es el de conseguir la maximización de beneficios, sin embargo Shul establece que es muy difícil aproximar el beneficio de los diferentes vuelos en el largo plazo. De esta forma, trata de conseguir la minimización de los aviones utilizados en la operativa.</p>
<b>Más información:</b>	<a href="http://link.springer.com/chapter/10.1057%2F9780230372924_7">http://link.springer.com/chapter/10.1057%2F9780230372924_7</a>

<b>Referencias:</b>	SUHL, Uwe H.; SUHL, Leena M. Solving Airline-fleet Scheduling Problems with Mixed-integer Programming. En <i>Operational Research in Industry</i> . Palgrave Macmillan UK, 1999. p. 135-156.
<b><u>Solución 2.3</u></b>	
<b>Autor/es:</b>	Götz.
<b>Fecha:</b>	1999.
<b>Algoritmo Utilizado:</b>	Simulación de Recocido (SA).
<b>Explicación:</b>	<p>En este enfoque de problema, Götz demuestra que los resultados obtenidos con la aplicación de Simulación de Recocido, son similares a los obtenidos por programación lineal entera. El algoritmo de Simulación de Recocido tiene como objetivo encontrar un resultado aproximado al óptimo.</p> <p>Respecto a la programación lineal, el SA muestra ventajas muy útiles:</p> <ul style="list-style-type: none"> <li>-Encontrar la solución requiere un menor tiempo de computación.</li> <li>-Resulta más fácil cambiar y añadir nuevas restricciones al problema.</li> </ul> <p>Estas dos ventajas la convierten en una solución muy versátil y que puede ser perfectamente aplicable en la planificación a corto plazo.</p>
<b>Referencia:</b>	GÖTZ, S., et al. Solving the weekly fleet assignment problem for large airlines. En <i>Proceedings of the III Metaheuristic International Conference</i> . 1999.
<b><u>Solución 2.4</u></b>	
<b>Autores:</b>	Francisco Javier Diego Martín, Jose Ángel González Manteca, Javier Carrasco Arias. Universidad Politécnica de Madrid.
<b>Fecha:</b>	2007.
<b>Algoritmo Utilizado:</b>	Optimización por colonia de hormigas.

<b>Explicación:</b>	<p>Se toma como objetivos principales el conseguir modelar el problema de asignación de flotas de forma que se puedan introducir todos aquellos eventos y requerimientos que puedan surgir en una situación real. Adicionalmente, se pretende resolver el problema obteniendo soluciones óptimas o cercanas a la mejor solución en un tiempo razonablemente pequeño.</p> <p>Así pues, el problema se modela aplicando los siguientes conceptos:</p> <ul style="list-style-type: none"> <li>-Se generalizan las ventanas de salida.</li> <li>-Se permite la modificación de los tiempos de duración de los trayectos para un mismo tipo de avión.</li> <li>-Permite la introducción de restricciones que simulan las preferencias de salida.</li> <li>-Se aplican relaciones de precedencia entre los diferentes trayectos.</li> </ul> <p>Para demostrar la eficacia de su solución se crean 3 sets de datos: 20 trayectos, 106 trayectos y 412 trayectos, y se resuelven utilizando su algoritmo y programación lineal.</p> <p>Según las soluciones obtenidas, en el primer caso (solamente 20 trayectos), la programación lineal es el método más eficiente (encuentra la solución más óptima en poco tiempo). Sin embargo en los otros dos casos, que se aproximan más a una situación real, el algoritmo de colonia de hormigas encuentra soluciones muy próximas a la óptima manteniendo un tiempo de resolución corto, mientras el tiempo de solución de la programación lineal aumenta hasta resultar inútil para su aplicación en casos reales.</p>
<b>Más información:</b>	<a href="http://www.revistadyo.com/index.php/dyo/article/view/57/57">http://www.revistadyo.com/index.php/dyo/article/view/57/57</a>
<b>Referencia:</b>	MARTÍN, Francisco Javier Diego; MANTECA, José Ángel González; ARIAS, Javier Carrasco. Análisis del problema de asignación de flotas flexible con ventanas de tiempo discretas, duración variable, preferencias de salida y relaciones de precedencia. Optimización por colonias de hormigas. <i>Dirección y Organización</i> , 2008, no 35, p. 61-68.
<b>Solución 2.5</b>	
<b>Autor/es:</b>	Yaohua Li y Na Tan. Facultad Aeronáutica de ingeniería, Universidad de Aviación Civil, China.
<b>Fecha:</b>	2013.

<b>Algoritmo Utilizado:</b>	Algoritmo Genético.
<b>Explicación:</b>	<p>En este documento, se representa el problema de Asignación de Flota modelado con datos reales de aerolíneas. Para solucionarlo, se utiliza un algoritmo genético mejorado por los autores, especialmente diseñado para solucionar el problema. Adicionalmente, se asigna al algoritmo un método heurístico en cada ciclo para constatar que la solución obtenida es factible. Si el algoritmo detecta que la solución obtenida no es factible, la ajusta, basándose en unas premisas previamente establecidas.</p> <p>Las conclusiones del trabajo, determinan que se obtienen soluciones rápidamente y que, de aplicarse este modelo en el proceso de asignación de flotas en las aerolíneas, esto mejoraría el proceso de automatización y conseguiría reducir sus costes.</p>
<b>Más información:</b>	<a href="http://www.hindawi.com/journals/mpe/2013/581586/">http://www.hindawi.com/journals/mpe/2013/581586/</a>
<b>Referencia:</b>	LI, Yaohua; TAN, Na. Study on Fleet Assignment Problem Model and Algorithm. <i>Mathematical Problems in Engineering</i> , 2013, vol. 2013.

### **Problema 3. Gestión de Ingresos.**

#### **Descripción del problema:**

La gestión de ingresos es un concepto muy importante para todas las compañías con ánimo de lucro, se trate de aerolíneas o no. Este campo tiene como objetivo establecer los precios de cada producto ofertado y analizar el número de clientes potenciales. De esta forma, se tratará de maximizar los beneficios obtenidos utilizando para ello todos los datos que tengamos a nuestra disposición.

Por ejemplo, uno de los problemas principales en este sentido suele ser el precio de venta de los billetes. Obviamente, a las aerolíneas no les interesa perder clientes fijando precios demasiado altos, pero a la vez no pueden permitirse perder ingresos potenciales con una política de bajos precios. Este hecho supone que la aerolínea deba tener un gran conocimiento acerca de qué tipo de clientes tiene, ya que un cliente de negocios no reaccionará de igual forma a una subida de precios que un cliente de ocio.

Según la definición ofrecida por Wikipedia:

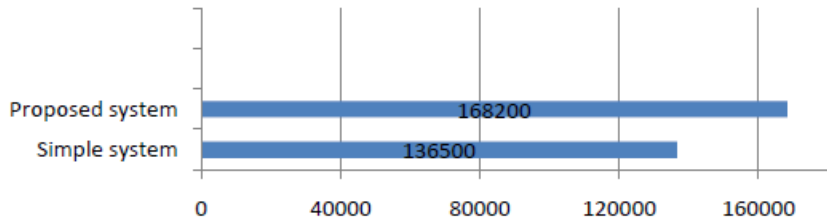
*“La disciplina de gestión de ingresos combina la minería de datos y la investigación de operaciones con la estrategia, la comprensión del comportamiento del cliente, y la asociación con la fuerza de ventas.”*

Así pues, como se puede intuir la experiencia en cuanto a demanda y comportamiento de los clientes puede ser un factor diferenciador entre aerolíneas para una correcta gestión del problema, otorgando ventaja a compañías con más años de experiencia (y por tanto con más conocimiento del mercado). Usualmente, cuanto mejor sea la calidad de los datos históricos recolectados por la aerolínea en cuestión, mejor será su gestión de ingresos.

Si tenemos en cuenta la cantidad de reservas y de operaciones que se llevan a cabo durante un año en una aerolínea comercial, la diferencia entre aplicar una buena política de gestión de ingresos o no, puede suponer la diferencia entre obtener beneficios o pérdidas. Los sistemas de gestión de ingresos han supuesto un aumento del 4 al 10% en los beneficios de las compañías.

Un ejemplo que muestra claramente la importancia de contar con un buen sistema de gestión de ingresos es el de la compañía American Airlines que en 1985 implementó un sistema de gestión de ingresos, que ofrecía descuentos cuando los vuelos se quedaban con algún asiento vacío debido a alguna cancelación. El resultado de la aplicación de este sistema fue todo un éxito; en el siguiente año, American Airlines aumentó sus ingresos en un 14'5% y sus ganancias en un 47'8%.



Soluciones Propuestas							
Solución 3.1							
<b>Autores:</b>	Rosey Plopi. Department of Computer Science and Applications, Kurukshetra University.						
<b>Fecha:</b>	2014.						
<b>Algoritmo Utilizado:</b>	Algoritmo genético. Matlab.						
<b>Explicación:</b>	<p>Se propone maximizar los beneficios de las aerolíneas utilizando los algoritmos genéticos. Este algoritmo se usa de forma que cada vez que varios clientes tratan de reservar un mismo asiento, el sistema analiza las diferentes propuestas y escoge la que maximice los beneficios totales. El problema distingue entre dos tipos de viajeros, los viajeros por motivos de negocio y los viajeros de ocio.</p> <p>El sistema se implementa utilizando el programa matlab, y el algoritmo genético se utiliza como herramienta de optimización.</p> <p>Los resultados son desglosados en la parte final del documento, y se exponen en un gráfico que muestra la diferencia a nivel de ingresos que se ha conseguido utilizando la solución propuesta frente a un sistema de gestión de ingresos simple.</p>  <table border="1"> <thead> <tr> <th>Sistema</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Proposed system</td> <td>168200</td> </tr> <tr> <td>Simple system</td> <td>136500</td> </tr> </tbody> </table> <p><i>Gráfico 2. Resultados de sistema propuesto frente a sistema simple [15].</i></p> <p>La diferencia entre los dos sistemas es bastante significativa (el resultado del sistema propuesto es un 19'85% mejor).</p>	Sistema	Valor	Proposed system	168200	Simple system	136500
Sistema	Valor						
Proposed system	168200						
Simple system	136500						
<b>Más información:</b>	<a href="http://www.ijarcsse.com/docs/papers/Volume_4/1_January2014/V4I1-0366.pdf">http://www.ijarcsse.com/docs/papers/Volume_4/1_January2014/V4I1-0366.pdf</a>						
<b>Referencia:</b>	Publicado en "International Journey of advanced Research in Computer Science and Software Engineering". Volumen 4, problema 1. ISSN: 2277 128X.						

### **Solución 3.2**

<b>Autores:</b>	Dimitris Bertsimas, Ioana Popescu.
<b>Fecha:</b>	2003.
<b>Algoritmo Utilizado:</b>	Algoritmo basado en la programación dinámica aproximada.
<b>Explicación:</b>	<p>Se propone un nuevo algoritmo basado en la programación dinámica aproximada. Además, el algoritmo se extiende para ser capaz de manejar cancelaciones de vuelos, gracias a la formulación de la programación lineal.</p> <p>En los datos que obtienen, los autores afirman que mejoran los beneficios y la robustez que utilizando otros métodos para el control del precio y la oferta.</p>
<b>Más información:</b>	<a href="http://www.mit.edu/~dbertsim/papers/Revenue%20Management/Revenue%20Management%20in%20a%20Dynamic%20Network%20Environment.pdf">http://www.mit.edu/~dbertsim/papers/Revenue%20Management/Revenue%20Management%20in%20a%20Dynamic%20Network%20Environment.pdf</a>
<b>Referencia:</b>	BERTSIMAS, Dimitris; POPESCU, Ioana. Revenue management in a dynamic network environment. <i>Transportation science</i> , 2003, vol. 37, no 3, p. 257-277.

### **Solución 3.3**

<b>Autores:</b>	Yingjie Lan, Huina Gao, Michael Ball, Itir Karaesmen.
<b>Fecha:</b>	2006.
<b>Algoritmo Utilizado:</b>	Algoritmos online.
<b>Explicación:</b>	<p>Se aplica el uso de algoritmos online al problema de Revenue Management, teniendo en cuenta que la información sobre la demanda y registros anteriores es muy limitada. Este es un enfoque algo más restrictivo que los anteriores, pero nos permite tener la seguridad de contar con buenas adaptaciones para cualquier escenario de demanda que pueda presentarse.</p> <p>Los autores explican que, la mayoría de algoritmos utilizados para resolver este problema son demasiado dependientes de información anterior, es decir de contar con una buena base de datos respecto al comportamiento de la demanda en el pasado. Se realizan muchas comparaciones de este nuevo método con anteriores, y los resultados parecen favorecer a este tipo de nuevos métodos, que no dependen en exceso de otros datos.</p>
<b>Más información:</b>	<a href="https://www.gsb.stanford.edu/sites/default/files/documents/CompetitiveAnalysisOfOnlineAlgorithmsInRevenueManagementAModelingPerspective.pdf">https://www.gsb.stanford.edu/sites/default/files/documents/CompetitiveAnalysisOfOnlineAlgorithmsInRevenueManagementAModelingPerspective.pdf</a>

<b>Referencia:</b>	LAN, Yingjie, et al. Revenue management with limited demand information. <i>Management Science</i> , 2008, vol. 54, no 9, p. 1594-1609.
--------------------	---

### **Solución 3.4**

<b>Autores:</b>	Sebastian Vock, Departamento de Información de sistemas, Feie Universität, Berlín. Steffen Enz, Catherine Cleophas.
<b>Fecha:</b>	2014.
<b>Algoritmo Utilizado:</b>	Algoritmos Genéticos.
<b>Explicación:</b>	<p>Se proponen dos nuevas aproximaciones al problema de la gestión de ingresos utilizando para ello los algoritmos genéticos y la simulación. En el primer caso, el algoritmo genético se limita a llevar a cabo la simulación y aprender a cerca de los resultados del problema.</p> <p>En el segundo caso, es el algoritmo quien ajusta los parámetros iniciales de la demanda, para tratar de encontrar los principales indicadores de beneficio y poder dar una respuesta en términos de efectiva frente a diferentes fluctuaciones de demanda.</p>
<b>Más información:</b>	<a href="http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=7019894&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D7019894">http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=7019894&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D7019894</a>
<b>Referencia:</b>	VOCK, Sebastian; ENZ, Steffen; CLEOPHAS, Catherine. Genetic algorithms for calibrating airline revenue management simulations. En <i>Proceedings of the Winter Simulation Conference 2014</i> . IEEE, 2014. p. 264-275.

### **Solución 3.5**

<b>Autores:</b>	F. Etebari, A. Aaghaie , F. Khoshalhan.
<b>Fecha:</b>	2012.
<b>Algoritmo Utilizado:</b>	Algoritmo Genético.
<b>Explicación:</b>	<p>Para el modelaje del problema, los autores utilizan un popular y eficiente metodo llamado “choice based deterministic linear programming” (CDLP). La idea principal de este modelo, es que asume que cada cliente pertenece a un determinado segmento, que obviamente estará definido de forma diferente al resto, y por tanto su comportamiento también será diferente. El problema resultante es del tipo NP-Hard y, en términos de esfuerzo computacional, muy difícil de resolver mediante métodos como, por ejemplo la generación de columnas.</p> <p>La resolución del problema se realiza mediante el uso del algoritmo genético y de la generación de columnas. Al</p>

	<p>compararse las soluciones obtenidas por estos dos métodos, se observa que al aumentar la escala del problema, el algoritmo genético consigue mejores soluciones.</p> <p>Según la escala del problema, se utilizan dos tipos de operadores de mutación diferentes en el algoritmo genético.</p>
<b>Más información:</b>	<a href="http://www.iors.ir/journal/article-1-137-fa.pdf">http://www.iors.ir/journal/article-1-137-fa.pdf</a>
<b>Referencia:</b>	ETEBARI, F.; AAGHAIE, A.; KHOSHALHAN, F. A Genetic Algorithm for Choice-Based Network Revenue Management. <i>Iranian Journal of Operations Research</i> , 2011, vol. 3, no 1, p. 89-103.
<b><u>Solución 3.6</u></b>	
<b>Autores:</b>	Garrett van Ryzin. Jeff McGill Graduate School of Business, Columbia University, New York
<b>Fecha:</b>	2000.
<b>Algoritmo Utilizado:</b>	Algoritmo adaptativo.
<b>Explicación:</b>	<p>Se propone como solución al problema un algoritmo adaptativo, cuya efectividad será comprobada comparándolo con otro método más tradicional (EMRS-b con método de previsión censurada).</p> <p>Se demuestra que el algoritmo adaptativo es más efectivo que los métodos tradicionales en algunos casos. Cuando la demanda está fuertemente censurada, este tipo de algoritmo saca ventaja a los tradicionales, mientras que en caso contrario sus resultados son peores.</p> <p>Así pues, los algoritmos adaptativos demuestran ser útiles para aerolíneas con sistemas tradicionales de gestión de ingresos como herramientas útiles cuando, por ejemplo se den cambios drásticos en el mercado. Sin embargo, este tipo de algoritmos puede ser de gran utilidad a las aerolíneas pequeñas o que no cuenten con grandes recursos para crear un sistema de gestión de ingresos al completo.</p>
<b>Más información:</b>	<a href="https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/3921/airline_seat_protection.pdf">https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/3921/airline_seat_protection.pdf</a>
<b>Referencia:</b>	VAN RYZIN, Garrett; MCGILL, Jeff. Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. <i>Management Science</i> , 2000, vol. 46, no 6, p. 760-775.

### **Solución 3.7**

<b>Autores:</b>	Sumit Kunnumkal Indian School of Business, Huseyin Topaloglu School of Operations Research and Information Engineering.
<b>Fecha:</b>	2010.
<b>Algoritmo Utilizado:</b>	Algoritmo de aproximación estocástico.
<b>Explicación:</b>	<p>Una de las características principales de este enfoque es que la probabilidad de que un pasajero escoja viajar en una ruta o otra depende del precio establecido en esa ruta. De esta forma, los autores pretenden encontrar, en una red de destinos, cuales son aquellos rangos de precios que ofrecen un mayor beneficio total.</p> <p>Una vez realizada la simulación, los autores afirman que los resultados obtenidos por su algoritmo de aproximación estocástico son bastante mejores que las estrategias utilizadas normalmente, como la programación lineal determinista.</p>
<b>Más información:</b>	<a href="https://people.orie.cornell.edu/huseyin/publications/pricing_network.pdf">https://people.orie.cornell.edu/huseyin/publications/pricing_network.pdf</a>
<b>Referencia:</b>	KUNNUMKAL, Sumit; TOPALOGU, Huseyin. A stochastic approximation algorithm for making pricing decisions in network revenue management problems. <i>Journal of Revenue and Pricing Management</i> , 2010, vol. 9, no 5, p. 419-442.

#### **Problema 4. Capacidad del espacio aéreo.**

##### **Descripción del problema:**

En las últimas décadas, el sector aéreo ha crecido a un ritmo constante y esto, pese a ser un factor positivo para el sector, también ha traído nuevos problemas consigo. La saturación del espacio aéreo, que en los inicios de la aviación civil parecía imposible. El hecho es que el crecimiento del tráfico aéreo ha hecho que algunos espacios aéreos tengan más capacidad de la que poseen, causando retrasos y demoras por culpa de este hecho.

Como definición, podríamos establecer que la capacidad del espacio aéreo estaría determinada por el número máximo de aviones que pueden estar a la vez en un sector concreto de forma segura.

El principal problema de esto es que, si bien la demanda del sector aéreo continúa creciendo a buen ritmo y las aerolíneas aumentan estimuladas por la competitividad propia de la competencia, el tráfico aéreo sigue gestionado por los estados. Esto representa un gran inconveniente para la correcta optimización del espacio aéreo, ya que los estados suelen resolver los problemas de forma muy lenta, debido principalmente a sus largos procesos administrativos. Esta forma de gestión representa, en la actualidad un gran inconveniente para la gestión del tráfico aéreo europeo, generando retrasos y aumentando los gastos operacionales de las aerolíneas. El componente del tráfico aéreo es de los pocos que aún no ha sido liberalizado por Europa.

De hecho, el aspecto de la capacidad y las restricciones existentes en el sistema de ATM actual, son los que llevaron a la Unión Europea a crear SESAR (Single European Sky ATM Research). Este proyecto tiene como objetivo conseguir implementar, para el año 2020 una red de gestión del tráfico aéreo europea, que tenga una visión y enfoque comunes sobre el ATM. Actualmente este proyecto se encuentra en fase de desarrollo. Las soluciones presentadas en la siguiente parte representan modelos antiguos y nuevos enfoques para tratar de optimizar la gestión del tráfico aéreo, para optimizar su capacidad y reducir los costes de la operación.



*Figura 2 .Crecimiento en 2015 del tráfico de pasajeros en términos de RPK (Revenue passengers per Kilometer) [III].*

## Soluciones Propuestas

### Solución 4.1

<b>Autores:</b>	Martin Lipicnik, Brane Lucovnik.
<b>Año:</b>	2000
<b>Algoritmo Utilizado:</b>	Modelo teórico algoritmo propio.
<b>Explicación:</b>	<p>El algoritmo tiene en cuenta el número de aviones que entra en cada sector, y calcula su mínimo y su máximo, así como la media.</p> <p>También comprueba los cambios de altitud o de dirección que el piloto necesita y comprueba si estos pueden realizarse con seguridad o no (calculando que las distancias de seguridad entre aeronaves sean respetadas).</p>
<b>Más información:</b>	<a href="http://www.fpz.unizg.hr/traffic/index.php/PROMTT/article/viewFile/849/700">http://www.fpz.unizg.hr/traffic/index.php/PROMTT/article/viewFile/849/700</a>
<b>Referencia:</b>	LUČOVNIK, Brane, et al. Theoretical Model for Determination of Airspace Capacity. <i>PROMET-Traffic&amp;Transportation</i> , 2012, vol. 14, no 2, p. 71-78.

### Solución 4.2

<b>Autores:</b>	Fanngin Liu, Minghua Hu.
<b>Fecha:</b>	2009.
<b>Algoritmo Utilizado:</b>	Directed Acyclic Graph y Algoritmo Genetico Multi-Objetivo.
<b>Explicación:</b>	<p>Este enfoque del problema de capacidad representa los sectores mediante un Directed Acyclic Graph, junto con un modelo multi-objetivo e integral de optimización. Debido a esto, se pueden analizar las relaciones de un sector a otro, algo que puede resultar útil para modelar el algoritmo que se encargará de encontrar la solución del problema.</p> <p>Para la solución del problema, los autores utilizan un algoritmo genético multi-objetivo. Según sus resultados, se demuestra que este algoritmo es capaz de encontrar buenas soluciones para el problema. Los autores explican que, de incorporarse este sistema a un modelo de distribución y regulación de la capacidad del espacio aéreo, este podría regular y optimizar los recursos de la gestión de tráfico aéreo de forma paulatina.</p>
<b>Más información:</b>	<a href="http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=4797410&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4797410">http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=4797410&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4797410</a>
<b>Referencia:</b>	LIU, Fangqin; HU, Minghua. Airspace capacity management based on control workload and coupling constraints between

	airspaces. En Computer Modeling and Simulation, 2009. ICCMS'09. International Conference on. IEEE, 2009. p. 329-334.
<b><u>Solución 4.3</u></b>	
<b>Autores:</b>	Chris R. Brinton
<b>Fecha:</b>	2008.
<b>Algoritmo Utilizado:</b>	Algoritmo de Clustering.
<b>Explicación:</b>	<p>En esta solución del problema se utiliza el algoritmo de clustering, para tratar de resolver el problema de capacidad del espacio aéreo. El algoritmo agrupa las rutas, manteniendo niveles aceptables de densidad dentro de cada sector.</p> <p>Los resultados demuestran la efectividad de la aplicación de este algoritmo para fragmentar el espacio aéreo en sectores.</p>
<b>Más información:</b>	<a href="http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=4702800&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4702800">http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=4702800&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4702800</a>
<b>Referencia:</b>	BRINTON, C. R.; PLEDGIE, S. Airspace partitioning using flight clustering and computational geometry. En <i>Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th</i> . IEEE, 2008. p. 3. B. 3-1-3. B. 3-10.
<b><u>Solución 4.4</u></b>	
<b>Autores:</b>	Zúñiga C.A., Piera M.A., Barua, O.T, Departamento de Telecomunicaciones y Ingeniería de Sistemas.
<b>Fecha:</b>	2012.
<b>Algoritmo Utilizado:</b>	Algoritmo de resolución modelado mediante una red de Petri coloreada (CPN Tools).
<b>Explicación:</b>	<p>Este enfoque de resolución del problema, se basa en detectar los momentos en que dos aeronaves se vayan a encontrar por debajo de la separación mínima establecida por ICAO. El algoritmo propuesto resuelve este tipo de conflictos, además de comprobar que las soluciones no provoquen otro conflicto a su vez.</p> <p>Este algoritmo podría ser una buena herramienta para incorporar a aquellos modelos de ATM que se basan en solucionar el problema de capacidad en sí, ya que consigue evitar conflictos y solucionarlos tratando de perder el menor tiempo posible, hecho que consigue optimizar el uso del espacio aéreo y mejora su gestión.</p>
<b>Más información:</b>	<a href="http://www.hala-sesar.net/sites/default/files/documents/p233-zuniga.pdf">http://www.hala-sesar.net/sites/default/files/documents/p233-zuniga.pdf</a>



<b>Referencia:</b>	CA, Zúñiga; PIERA, M. A.; BARUWA, O. T. Pre-tactical Trajectory De-Confliction Algorithm for Air Traffic Management. 2012.
--------------------	--

#### **Solución 4.5**

<b>Autores:</b>	Min Xue, Universidad de California.
<b>Fecha:</b>	2009.
<b>Algoritmo Utilizado:</b>	Diagrama de Veronoi junto con algoritmo de iteración profunda y algoritmo genético.
<b>Explicación:</b>	<p>En este problema, los autores utilizan el Diagrama de Veronoi para sectorizar el espacio aéreo, mientras que el algoritmo genético es utilizado para realizar una optimización multiobjetivo. Por último, el algoritmo de iteración profunda se usa para acelerar todo el proceso.</p> <p>Los autores clasifican los resultados de muy prometedores en términos de la aplicación en sectorización, ya que el método utilizado consigue balancear el tiempo de trabajo de las aeronaves presentes en el espacio aéreo de forma satisfactoria.</p>
<b>Más información:</b>	<a href="http://aviationsystems.arc.nasa.gov/publications/strategic/Xue_GC_NC_2008_02.pdf">http://aviationsystems.arc.nasa.gov/publications/strategic/Xue_GC_NC_2008_02.pdf</a>
<b>Referencia:</b>	XUE, Min. Airspace sector redesign based on Voronoi diagrams. <i>Journal of Aerospace Computing, Information, and Communication</i> , 2009, vol. 6, no 12, p. 624-634.

#### **Solución 4.6**

<b>Autores:</b>	Amitabh Basu, Joseph S. B. Mitchell, Girishkumar Sabhnani Stony Brook University
<b>Fecha:</b>	2009.
<b>Algoritmo Utilizado:</b>	Algoritmo geométrico y heurística (Binary Space Partitions y Pie-Partitions).
<b>Explicación:</b>	<p>En este enfoque de resolución, los autores modelan el problema de sectorización y lo solucionan aplicando un algoritmo geométrico y aplicando una heurística que mezcla el método de Binary Space Partitions junto con el de Pie-Partitions.</p> <p>En primera instancia, se soluciona el problema de capacidad desde la perspectiva del 1D, dando en algunas versiones del problema una solución exacta. Una vez hecho esto, se presenta el problema en 2D y se utiliza la solución obtenida en el 1D como un sub problema.</p> <p>Una vez hecho esto, se procede a simular y a probar la efectividad real del método utilizando para ello datos reales y se procede a compararlos con los métodos utilizados en la actualidad. Los resultados resultan ser muy prometedores, ya</p>

	que las heurísticas que obtienen mejores resultados muestran mejoras en la capacidad de dos a tres veces que con los métodos actuales.
<b>Más información:</b>	<a href="http://www.ams.sunysb.edu/~jsbm/papers/alnex08.pdf">http://www.ams.sunysb.edu/~jsbm/papers/alnex08.pdf</a>
<b>Referencia:</b>	BASU, Amitabh; MITCHELL, Joseph SB; SABHNANI, Girish Kumar. Geometric algorithms for optimal airspace design and air traffic controller workload balancing. <i>Journal of Experimental Algorithmics (JEA)</i> , 2009, vol. 14, p. 3.

### **Problema 5: Ground Holding Policy.**

#### **Descripción del problema:**

La congestión de los aeropuertos de todo el mundo se ha incrementado durante los últimos años, llegando a convertirse en uno de los mayores problemas y retos a los que debe enfrentarse la aviación comercial. De hecho, en la actualidad, no es raro encontrar aeropuertos cuyo ratio de operaciones/capacidad sea de 2:1 o incluso de 3.1. Como ya hemos podido ver en el problema de capacidad, el espacio aéreo se encuentra en la actualidad saturado, y constantemente se están estudiando nuevas medidas para mejorar en lo posible este problema.

Una buena respuesta para solucionar la congestión del espacio aéreo y de los aeropuertos, sería dar una solución efectiva a la política de propiedad de suelo en el aeropuerto. Las soluciones a largo plazo se basan fundamentalmente en crear más aeropuertos, o mejorar los ya existentes con más pistas de aterrizaje y estructuras adicionales para, de esta forma mejorar su capacidad actual. Esta solución no parece ser la más efectiva, ya que el tráfico aéreo continuará creciendo durante los siguientes años, y los aeropuertos nuevos no podrán absorber toda esa nueva demanda en su totalidad.

Por lo tanto, la solución a corto plazo es mejorar la capacidad de gestión de los aeropuertos, con herramientas que les permitan gestionar y optimizar mejor su capacidad, además de conseguir reducir al máximo los costes generados por los retrasos. Una pieza fundamental en este proceso es el problema del Ground Holding Policy, que se podría resumir con la siguiente pregunta: ¿Cuánto tiempo debe permanecer en el aeropuerto un avión antes de despegar? El objetivo del problema es conseguir reducir al mínimo el coste del retraso total de cada vuelo, tomando las medidas necesarias para cada uno de las aeronaves.

Debemos tener en cuenta también, que el coste de dejar esperando un aeronave en el suelo es mucho más barato que dejarla esperando en el aire. Así pues, el reto principal que se plantea con este problema sería formulado de la siguiente manera:

“Dada una red de aeropuertos determinada y un número de aeronaves, ¿cuánto tiempo y donde deben permanecer las aeronaves en el suelo para reducir al máximo el coste total de los retrasos?”

Por descontado, se trata de un problema combinatorio de difícil solución desde el punto de vista computacional, ya que la cantidad de datos relacionados entre sí a gestionar son ingentes, y aumentan a medida que más aeropuertos se incorporan al problema.

## Soluciones Propuestas

### Solución 5.1

<b>Autores:</b>	Peter B. Vranas, Dimitris J. Bertsimas, Amadeo R. Odon, Massachusetts Institute of Technology.
<b>Año:</b>	1993.
<b>Algoritmo Utilizado:</b>	Algoritmo Heurístico.
<b>Explicación:</b>	<p>Se formulan diferentes modelos y se propone solucionar el problema mediante un algoritmo heurístico. Se ofrecen datos provenientes de la simulación que demuestran unos buenos resultados aplicando diferentes parámetros de entrada. Este enfoque propone una versión estática del problema de Ground Holding Policy, es decir, que los retrasos propuestos tanto en el aire como en tierra se realizaran antes de que comiencen los vuelos y no se cambiarán durante toda la operación.</p> <p>Los resultados obtenidos son buenos y además se obtienen en un tiempo razonable, hasta con 6 aeropuertos diferentes y 3000 vuelos diarios.</p>
<b>Más información:</b>	<a href="http://www.mit.edu/~dbertsim/papers/AirTransportation/The%20multiairport%20groundholding%20problem%20in%20air%20traffic%20control.pdf">http://www.mit.edu/~dbertsim/papers/AirTransportation/The%20multiairport%20groundholding%20problem%20in%20air%20traffic%20control.pdf</a>
<b>Referencia:</b>	VRANAS, Peter B.; BERTSIMAS, Dimitris J.; ODoni, Amadeo R. The multi-airport ground-holding problem in air traffic control. <i>Operations Research</i> , 1994, vol. 42, no 2, p. 249-261.

### Solución 5.2

<b>Autores:</b>	Lili Wang, Zahoning Hang, Civil Aviation University of China.
<b>Fecha:</b>	2005.
<b>Algoritmo Utilizado:</b>	Método analizador de eventos discretos.
<b>Explicación:</b>	<p>La ventaja del método propuesto a continuación es que incluso para aeropuertos con muchos vuelos diarios, el problema puede ser resuelto de forma prácticamente instantánea por un solo ordenador.</p> <p>Esto se debe a que trata el problema desde un evento discreto, es decir que calcula los tiempos de salida, de vuelo y de llegada de un solo avión. Ya que el modelo es dinámico, este irá optimizando sus datos en función de cada aterrizaje.</p>
<b>Más información:</b>	<a href="http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.509.6034&amp;rep=rep1&amp;type=pdf">http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.509.6034&amp;rep=rep1&amp;type=pdf</a>

<b>Referencia:</b>	LILI, W. A. N. G.; ZHANG, Zhaoning. A RECURSION EVENT-DRIVEN MODEL TO SOLVE THE SINGLE AIRPORT GROUND-HOLDING PROBLEM. En <i>Proceedings of the Eastern Asia Society for Transportation Studies</i> . 2005. p. 500-506.
--------------------	---

### **Solución 5.3**

<b>Autores:</b>	Lili Wang, Air Traffic Management College Civil, Aviation University of China, Tian Jin, Zhaoning Zhang, Jimin Liu.
<b>Fecha:</b>	2010.
<b>Algoritmo Utilizado:</b>	Algoritmo heurístico.
<b>Explicación:</b>	<p>Este enfoque trata de solucionar el problema de Ground Holding Policy, utilizando un algoritmo heurístico óptimo. El algoritmo analiza los diferentes vuelos y considera distintos factores como el tiempo previsto de llegada a su destino o cómo afectaría su retraso a la solución del problema. De esta forma, otorga prioridad a algunos vuelos. A los aeropuertos de la red, se les asignan los slots según su capacidad.</p> <p>De esta forma el algoritmo trata de solucionar el problema provocando los menores retrasos posibles y optimizando el coste final de la solución. Los resultados obtenidos en la simulación son positivos y prueban la efectividad del presente algoritmo.</p>
<b>Más información:</b>	<a href="http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=5524112&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5524112">http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=5524112&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5524112</a>
<b>Referencia:</b>	WANG, Lili; ZHANG, Zhaoning; LIU, Jimin. Multiple airport ground holding dynamic model and optimal algorithm basing flight priority. En <i>Control and Automation (ICCA), 2010 8th IEEE International Conference on</i> . IEEE, 2010. p. 1081-1085.

### **Solución 5.4**

<b>Autores:</b>	C.G Panayiotou, Boston University, C.G Cassandras.
<b>Fecha:</b>	2001.
<b>Algoritmo Utilizado:</b>	Análisis Finito de Perturbación (FPA), algoritmo de control.
<b>Explicación:</b>	<p>Se trata de solucionar el problema utilizando un análisis finito de Perturbación y un algoritmo de control desarrollado por los autores. La simulación se lleva a cabo con datos reales del aeropuerto del Aeropuerto Internacional de Logan.</p> <p>El algoritmo obtiene buenos resultados, ya que consigue eliminar todas las esperas en el aire, hecho que resulta muy beneficioso en términos de coste para la solución final.</p>

<b>Más información:</b>	<a href="http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=918903&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F87%2F19873%2F00918903.pdf%3Farnumber%3D918903">http://ieeexplore.ieee.org/xpl/login.jsp?tp=&amp;arnumber=918903&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F87%2F19873%2F00918903.pdf%3Farnumber%3D918903</a>
<b>Referencia:</b>	PANAYIOTOU, Christos G.; CASSANDRAS, Christos G. A sample path approach for solving the ground-holding policy problem in air traffic control. <i>IEEE Transactions on Control Systems Technology</i> , 2001, vol. 9, no 3, p. 510-523.
<b><u>Solución 5.5</u></b>	
<b>Autores:</b>	Pulugurtha, S. and Nambisan, S.
<b>Fecha:</b>	2001.
<b>Algoritmo Utilizado:</b>	Algoritmo Genético.
<b>Explicación:</b>	<p>En este artículo se analiza la aplicabilidad de los algoritmos genéticos para resolver problemas relativos al Ground Holding Policy. Se realizan diferentes tests y pruebas para ver si, efectivamente este algoritmo es una buena solución para este tipo de problemas.</p> <p>Se observa que los resultados de las simulaciones son positivos y además, el tiempo de ejecución es bastante rápido.</p>
<b>Más información:</b>	<a href="http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2001)127%3A5(442)">http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2001)127%3A5(442)</a>
<b>Referencia:</b>	PULUGURTHA, Srinivas S.; NAMBISAN, Shashi S. Using genetic algorithms to evaluate aircraft ground holding policy under static conditions. <i>Journal of transportation engineering</i> , 2001, vol. 127, no 5, p. 433-441.
<b><u>Solución 5.6</u></b>	
<b>Autores:</b>	Lorenzo Brunetta, Guglielmo Guastalla, Lisa Navazio.
<b>Fecha:</b>	1998.
<b>Algoritmo Utilizado:</b>	Algoritmo heurístico.
<b>Explicación:</b>	Se utiliza un nuevo algoritmo heurístico que se basa en reglas de prioridad para resolver el problema. Para las simulaciones del problema, se utiliza una red de 10 aeropuertos congestionados con unos 5000 vuelos.
<b>Más información:</b>	<a href="http://link.springer.com/article/10.1023%2FA%3A1018909224543">http://link.springer.com/article/10.1023%2FA%3A1018909224543</a>
<b>Referencia:</b>	BRUNETTA, Lorenzo; GUASTALLA, Guglielmo; NAVAIZIO, Lisa. Solving the multi-airport ground holding problem. <i>Annals of operations research</i> , 1998, vol. 81, p. 271-288.

## **Problema 6: Asignación de rutas.**

### **Descripción del problema:**

El problema de creación y asignación de rutas, viene a ser el famoso problema de optimización combinatoria de enrutamiento de vehículos, pero aplicado a la aeronáutica. Su objetivo es simple: dada una red de aeropuertos y de aviones se deben asignar a cada avión, aquellas rutas que consigan el menor gasto en términos de coste y que puedan cumplir todas las condiciones iniciales propuestas. Obviamente este problema cuenta con restricciones obvias que deben ser tenidas en cuenta en su formulación, tales como el mantenimiento de las aeronaves, o los tiempos máximos de vuelo para las tripulaciones.

Encontrar algoritmos que permitan resolver este problema de forma óptima y en poco tiempo, representa un reto debido a que la naturaleza del problema en NP-Hard, es decir un tipo de problema difícil de resolver computacionalmente. Las soluciones más comunes para el problema de enrutación de vehículos, suelen basarse en heurísticas ya que funcionan bien en problemas con grandes espacios de soluciones y pueden encontrar rutas óptimas en tiempos razonables.

Este problema puede extrapolarse también a la aviación militar. Ya que el objetivo de los algoritmos que se propongan para solucionarlo debe ser el encontrar la ruta con menor coste de un punto a otro, este también debería ser capaz de evitar rutas que, por ejemplo no puedan ser sobrevoladas (ya sea por obstáculos geográficos o bien por la presencia de fuerzas hostiles en un determinado sector).

Este problema puede combinarse también con uno visto anteriormente, el problema de asignación de tripulaciones o incluso con el de asignación de flotas, ya que al fin y al cabo, todos estos problemas se dan en la fase de planificación de la operación. Algunas de las soluciones que veremos a continuación, añaden también a sus propuestas algoritmos que controlen rotaciones de tripulaciones junto con la creación de rutas para los aeronave.

## Soluciones Propuestas

### Solución 6.1

<b>Autores:</b>	Jean-François Cordeau, Goran Stojkovic, François Soumis, Jacques Desrosiers.
<b>Año:</b>	2001.
<b>Algoritmo Utilizado:</b>	Generación de columnas y Branch and Bound.
<b>Explicación:</b>	<p>En esta solución, se tratan a la vez el problema de asignación de tripulaciones junto con el problema de rutas. El problema se divide en dos partes para hacer más fácil su tratamiento; el problema principal, que sería el problema de asignación de rutas y un subproblema, asignación de tripulaciones. En este caso no se resuelve el problema de asignación de flotas, ya que se establece que todos los vuelos se hacen con un mismo tipo de avión.</p> <p>Se imponen algunas restricciones básicas para la tripulación, tales como un tiempo máximo de vuelo, así como para las aeronaves (requerimientos de mantenimiento). Los dos problemas son resueltos por generación de columnas y luego se utiliza un método heurístico de Branch and Bound para finalizar las soluciones propuestas. Se utilizan datos reales y la mayor simulación realizada se realiza con hasta 500 tramos de vuelo en un horizonte temporal de tres días.</p> <p>Los resultados obtenidos superan por un buen margen a los típicos procesos de planificación empleados por aerolíneas y, lo más sorprendente de todo es que las soluciones no siguen el típico esquema de hub-spoke.</p>
<b>Más información:</b>	<a href="https://www.semanticscholar.org/paper/Benders-Decomposition-for-Simultaneous-Aircraft-Cordeau-Stojkovic/ce867bfcd8316156582fcceb3bb0a8d2458acc75/pdf">https://www.semanticscholar.org/paper/Benders-Decomposition-for-Simultaneous-Aircraft-Cordeau-Stojkovic/ce867bfcd8316156582fcceb3bb0a8d2458acc75/pdf</a>
<b>Referencia:</b>	CORDEAU, Jean-François, et al. Benders decomposition for simultaneous aircraft routing and crew scheduling. <i>Transportation science</i> , 2001, vol. 35, no 4, p. 375-388.

### Solución 6.2

<b>Autores:</b>	Oliver Weide,, David Ryana, Matthias Ehrgott, Departamento de Ciencia e Ingeniería, Universidad de Auckland, Nueva Zelanda.
<b>Fecha:</b>	2009.
<b>Algoritmo Utilizado:</b>	Algoritmo iterativo.



<b>Explicación:</b>	<p>Este enfoque de la solución, también toma en cuenta el problema de asignación de tripulaciones junto con el problema de asignación de rutas y se aplica un algoritmo iterativo para su solución. Los autores, al igual que en la solución 1, dividen los dos problemas, pero en este caso los resuelven de forma iterativa. Se realizan iteraciones hasta que el nivel de robustez llega a su máximo, pero en cada iteración el problema de creación y asignación de rutas se resuelve primero.</p> <p>Los autores además tratan de expandir el problema, distinguiendo las tripulaciones en dos tipos, la tripulación técnica y la tripulación de atención de cabina.</p>
<b>Más información:</b>	<a href="http://eprints.lancs.ac.uk/63744/1/10.pdf">http://eprints.lancs.ac.uk/63744/1/10.pdf</a>
<b>Referencia:</b>	WEIDE, Oliver; RYAN, David; EHRGOTT, Matthias. An iterative approach to robust and integrated aircraft routing and crew scheduling. <i>Computers &amp; Operations Research</i> , 2010, vol. 37, no 5, p. 833-844.
<b><u>Solución 6.3</u></b>	
<b>Autores:</b>	Mercier, Soumis
<b>Fecha:</b>	2007.
<b>Algoritmo Utilizado:</b>	Descomposición de Benders, generación de columnas y generación de restricciones dinámicas.
<b>Explicación:</b>	<p>En este enfoque de la resolución del problema, Mercier y Soumis tratan de resolver, como los otros autores, los dos problemas a la vez y establecen que los aviones pueden salir en una ventana de tiempo de cinco minutos más tarde o más pronto, según la necesidad.</p> <p>Se establecen restricciones de igualdad en ambos problemas para asegurar que, una vez unidas, las soluciones de las tripulaciones y los aviones tengan los mismos tiempos de salida.</p> <p>La solución es un modelo mejorado al propuesto por uno de los autores anteriormente [2].</p> <p>Esta mejora no mejora la calidad de la solución, sin embargo consigue disminuir los tiempos de resolución de la simulación una media de doce veces más rápido que el anterior modelo.</p> <p>Se explica que, últimamente se ha descubierto que, pese a que resolver secuencialmente los dos problemas simplifica el proceso, el hecho de solucionar ambos problemas integrándolos en uno solo, consigue obtener mejores soluciones que la resolución secuencial.</p>

<b>Más información:</b>	<a href="https://www.researchgate.net/publication/222360140_An_integrated_aircraft_routing_crew_scheduling_and_flight_retiming_model">https://www.researchgate.net/publication/222360140_An_integrated_aircraft_routing_crew_scheduling_and_flight_retiming_model</a>
<b>Referencia:</b>	MERCIER, Anne; SOUMIS, François. An integrated aircraft routing, crew scheduling and flight retiming model. <i>Computers &amp; Operations Research</i> , 2007, vol. 34, no 8, p. 2251-2265.
<b><u>Solución 6.4</u></b>	
<b>Autores:</b>	Dr. Jenny Díaz-Ramírez, Dr. José Ignacio Huertas, Dr. Federico Trigos.
<b>Fecha:</b>	2013.
<b>Algoritmo Utilizado:</b>	Algoritmo AFS-MRA (The Airline Flight Schedule Design and Maintenance Routing Algorithm).
<b>Explicación:</b>	<p>Se propone resolver la planificación de horarios de los vuelos junto con el problema de asignación de rutas, teniendo en cuenta los tiempos relativos al mantenimiento de las aeronaves.</p> <p>Las restricciones de mantenimiento asignadas al problema no son muy restrictivas, sin embargo los autores comentan que se han usado de forma extensa en la simulación de este tipo de problemas. Estas restricciones son las siguientes:</p> <ul style="list-style-type: none"> <li>-En la flota simulada, solo existe una base donde realizar el mantenimiento.</li> <li>-Se asume que los requerimientos de mantenimiento son cumplidos enviando el avión a la base durante una noche. Para modelar la red del problema, se utiliza un método muy usado conocido como red de conectividad o “leg-on-node”.</li> </ul> <p>El problema es resuelto por el algoritmo AFS-MRA, un algoritmo que utiliza la heurística para encontrar la mejor solución, tomando los dos problemas y asignándoles las mismas restricciones de mantenimiento a cada uno. El funcionamiento del problema se puede resumir en esta figura, proporcionada por los autores, aunque se recomienda encarecidamente revisar el artículo para obtener un buen conocimiento del algoritmo y de la formulación utilizada para solucionar el problema.</p>

	<pre> <b>Begin</b>   <math>r \leftarrow 0</math>   feasible <math>\leftarrow</math> true   <math>ub_y \leftarrow 1 \quad \forall (i, j) \in A</math>   <b>while</b> <math>((\exists ub_y \neq 0 \quad \forall j : (i, j) \in E) \text{ and } (r &lt; F) \text{ and } \text{feasible} = \text{true})</math>     solve model <b>AMRP</b>.     if feasible, Let <math>y_{ij}^*</math> be the optimal solution       set <math>\Psi = \{y_{ij} : y_{ij}^* \geq 1\}</math>       <math>r \leftarrow r + 1</math>       route <math>(r) = \{i : y_{ij} \in \Psi\}</math>       <b>for all</b> <math>i \in N</math>         if <math>y_{ij} = 1</math> for some <math>j : (i, j) \in A</math>           <b>then</b> <math>U_{ij} \leftarrow 0</math> for all <math>j : (i, j) \in E</math>         <b>end if</b>       <b>end for</b>     <b>else</b>       feasible <math>\leftarrow</math> false     <b>end if</b>   <b>end while</b> <b>end</b> </pre> <p><i>Figura 3. Representación del algoritmo utilizado[37].</i></p> <p>En los resultados finales los autores observan que, si se comparan con un modelo secuencial estándar (que además no tiene en cuenta las restricciones de mantenimiento), el modelo propuesto obtiene como mínimo las mismas soluciones o mejores en todos los casos. Por otra parte, el esfuerzo en términos de computación no es muy alto ya que, con un ordenador convencional la extensión más larga utilizada (3000 vuelos y una flota de 10 aviones), la simulación ha tardado menos de 25 minutos.</p>
<b>Más información:</b>	<a href="http://eaas-journal.org/survey/userfiles/files/Aeurnautical%20engineering%203.pdf">http://eaas-journal.org/survey/userfiles/files/Aeurnautical%20engineering%203.pdf</a>
<b>Referencias:</b>	DÍAZ-RAMÍREZ, Jenny; HUERTAS, José Ignacio; TRIGOS, Federico. Simultaneous schedule design & routing with maintenance constraints for single fleet airlines. <i>International Journal of Engineering</i> , 2013, vol. 2, no 2, p. 2305-8269.
<b>Solución 6.5</b>	
<b>Autores:</b>	Michelle Dunbar, Gary Froyland , Cheng-Lung Wu.
<b>Fecha:</b>	2014.

<b>Algoritmo Utilizado:</b>	Algoritmo heurístico.
<b>Explicación:</b>	<p>Esta solución es la ampliación de una resolución previa de los autores [1], donde se integra un algoritmo que trata de disminuir la propagación de retrasos cuando un vuelo es retrasado, evitando en la medida de lo posible, que este retraso se contagie a los demás vuelos.</p> <p>En este trabajo, se proponen dos nuevos algoritmos que resuelven los problemas de asignación de rutas y asignación de tripulaciones, pero que además tratan de evitar la propagación de los retrasos. Además, se añade una heurística adicional que trabajo junto con estos dos algoritmos y que se encarga de reprogramar vuelos que resultan inconvenientes para la solución, asignarles una solución y reducir el coste final que un retraso pueda tener en la solución.</p> <p>Los resultados obtenidos son buenos, ya que se demuestra que el algoritmo consigue reducir la propagación de los retrasos un 14% más que la anterior solución ofrecida.</p>
<b>Más información:</b>	<a href="http://www.optimization-online.org/DB_FILE/2013/03/3801.pdf">http://www.optimization-online.org/DB_FILE/2013/03/3801.pdf</a>
<b>Referencia:</b>	DUNBAR, Michelle; FROYLAND, Gary; WU, Cheng-Lung. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. <i>Computers &amp; Operations Research</i> , 2014, vol. 45, p. 68-86.
<b><u>Solución 6.6</u></b>	
<b>Autores:</b>	Torsten Reiners.
<b>Fecha:</b>	2012.
<b>Algoritmo Utilizado:</b>	Algoritmo Genético.
<b>Explicación:</b>	En esta solución, el autor junta el problema de asignación de rutas junto con el problema de asignación de flota, y encuentra la solución a partir de un algoritmo genético. En el problema también se tienen en cuenta las pausas de las aeronaves por mantenimiento.
<b>Más información:</b>	<a href="https://www.computer.org/csdl/proceedings/hicss/2012/4525/00/4525b267.pdf">https://www.computer.org/csdl/proceedings/hicss/2012/4525/00/4525b267.pdf</a>
<b>Referencia:</b>	REINERS, Torsten, et al. Integrated aircraft scheduling problem: An auto-adapting algorithm to find robust aircraft assignments for large flight plans. En <i>System Science (HICSS), 2012 45th Hawaii International Conference on</i> . IEEE, 2012. p. 1267-1276.

## 7. Explicación algoritmos.

---

En este apartado, se introducirán los algoritmos más utilizados en la resolución de los problemas citados en el apartado 5. Se ofrecerá una pequeña explicación sobre ellos y enlaces a documentos donde se describe el algoritmo con más detalle, en caso de que el lector pueda estar interesado en ello.

### 7.1 Algoritmos genéticos.

Los algoritmos genéticos se basan en el proceso de evolución de los organismos vivos y en el principio de selección natural. Estos algoritmos suelen utilizarse en problemas que no tienen un algoritmo especializado y donde no se pueden aplicar algoritmos exactos debido al tiempo que estos requieren.

En la naturaleza, generaciones de individuos de la misma especie compiten por conseguir recursos limitados presentes en la naturaleza, como la comida, territorios etc. Y pese contadas excepciones, aquellos individuos con las mejores capacidades y sus mejores adaptaciones genéticas consiguen más recursos (y por tanto, consiguen sobrevivir y reproducirse, mientras que otros individuos menos efectivos, no). Así pues, a medida que vaya pasando el tiempo solo aquellas especies con los mejores genes sobrevivirán.

Este algoritmo funciona de una forma muy similar a la que sigue el principio genético de la selección natural. Se trabaja con muchas soluciones diferentes factibles (que representan a los individuos). El algoritmo asigna a cada solución posible, una puntuación según su efectividad. Cuanto mayor sea esta puntuación, más posibilidades tendrá esta solución de “reproducirse” y juntar sus genes con otras soluciones también efectivas. De esta forma se conseguirá una nueva generación de soluciones posibles, con los mejores genes de la anterior generación.

Después de realizar una serie de ciclos, el algoritmo encontrará una solución final que contendrá los mejores genes de las anteriores generaciones. Se ha demostrado que, a pesar de no encontrar la solución óptima, este algoritmo consigue soluciones notables en términos de efectividad y en un tiempo.

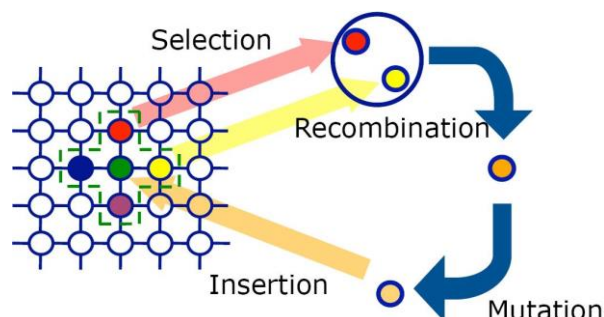


Figura 4. *Proceso algoritmo genético simplificado [IV]*

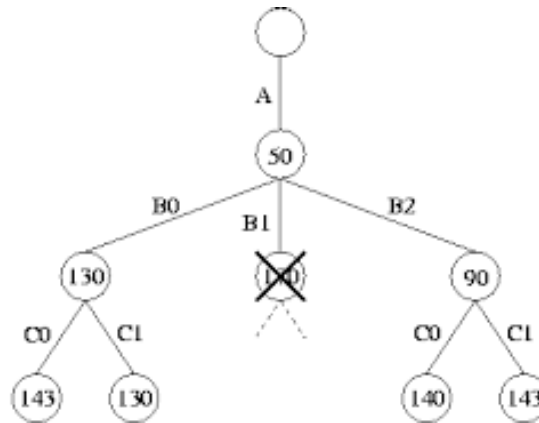
**Más información:**

<http://web.cecs.pdx.edu/~mm/cs410-510-winter-2005/ga-tutorial.pdf>

**7.2 Branch and Bound.**

El algoritmo de Branch and Bound se representa como un árbol, cuyas ramas representan las diferentes soluciones que tiene el problema dado. Estas “ramas”, no serían una solución final, sino más bien un camino por el cual el algoritmo sigue investigando nuevas soluciones. El mismo algoritmo se encarga de comprobar si las soluciones que están dando las diferentes ramas están siendo buenas o no, y en caso de que no lo sean, se encarga de podar esa rama (detiene la búsqueda de soluciones a través de ese camino concreto). De esta forma, se consigue ahorrar muchos recursos y tiempo, ya que estamos acotando el espacio de búsqueda de soluciones.

Como puede suponerse, este algoritmo no busca encontrar la solución óptima al problema, sino encontrar una solución factible y en un tiempo razonable.



*Figura 5. Representación Gráfica Árbol del Algoritmo [V].*

**Más información:**

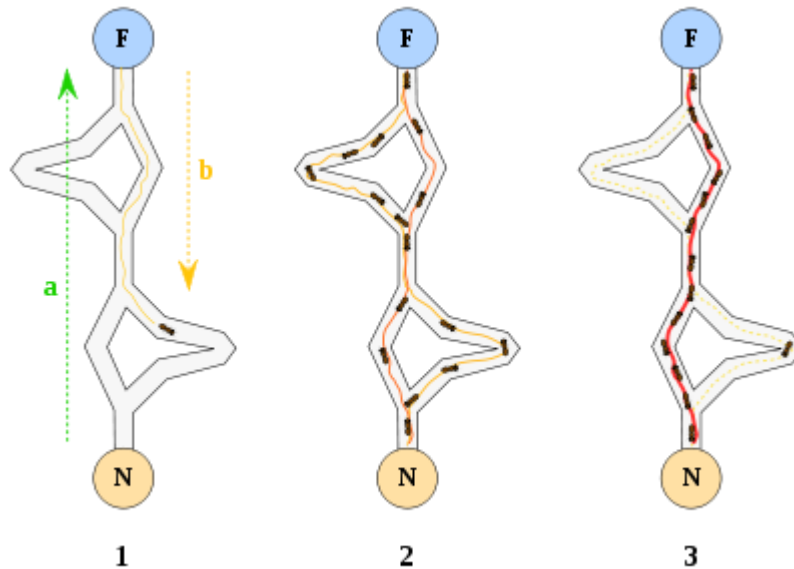
<http://www.imada.sdu.dk/Employees/jbj/heuristikker/TSPtext.pdf>

**7.3 Algoritmo de optimización de la colonia de hormigas.**

Este algoritmo basa su funcionamiento en la habilidad natural de las hormigas para encontrar los caminos más cortos de fuentes de alimentos a sus hormigueros. Mientras realizan el camino de la fuente de comida al hormiguero, algunas especies de hormigas depositan feromonas, una sustancia química por el camino. Se ha demostrado que las hormigas se mueven de forma aleatoria si no encuentran ningún rastro de feromonas, pero que las probabilidades de escoger un camino determinado aumentan según la concentración de feromonas depositada.

Además, las feromonas tienden a desvanecerse con el tiempo. Esto significa que los caminos largos tendrán, a la larga menos posibilidades de ser seguidos

por las hormigas y que, finalmente por probabilidad y repetición, todas seguirán un único camino, el más corto hacia el alimento.



*Figura 6. Representación proceso de búsqueda [VI].*

Este tipo de estrategia natural se ha extrapolado a la resolución de problemas, creando este algoritmo muy útil para la resolución de problemas computacionalmente difíciles de solucionar.

#### **Más información:**

[https://es.wikipedia.org/wiki/Algoritmo\\_de\\_la\\_colonia\\_de\\_hormigas](https://es.wikipedia.org/wiki/Algoritmo_de_la_colonia_de_hormigas)

### **7.4 Simulación de recocido.**

El nombre de este algoritmo está basado en la técnica del recocido, muy utilizada en todo el mundo. Consiste principalmente en calentar un material (como el metal o la cerámica) y luego dejar que se enfríe. Esto provoca que los materiales cambien sus propiedades físicas por otras más deseables.

El algoritmo es similar al método de Monte Carlo (de hecho, está basado en una adaptación del mismo, Metropolis-Hastings). El algoritmo empieza con un determinado estado, llamado  $s$ . Mediante otro proceso, crea un estado anexo del actual, llamado  $s'$ . Entonces, el algoritmo realiza una comparación de los dos estados y escoge uno de los dos. La probabilidad de que el algoritmo escoja un estado peor del que hay actualmente, se basa en la siguiente fórmula:

$$P(\Delta f, T) = e^{-\Delta f/T}$$

Donde  $\Delta f$  corresponde a la diferencia de evaluación (la calificación que el algoritmo impone a cada uno de los dos estados, en función de lo buena que es su solución), entre  $s$  i  $s'$ .  $T$  corresponde a la temperatura, y aquí es donde podemos ver su similitud con el proceso de recocido. Al iniciar el problema, la variable  $T$  tiene su máximo valor, de forma que es menos probable que el algoritmo escoja un mejor al que ya tiene. Sin embargo, a medida que el algoritmo ejecuta iteraciones, el valor de  $T$  va bajando de forma gradual (esto simula el enfriamiento). Así pues, las probabilidades que el algoritmo escoja un estado mejor al que ya tiene aumentan. Una vez la temperatura llega a cero, el algoritmo solo aceptará aquellos nuevos estados que proporcionen mejores soluciones. Las iteraciones se repetirán hasta que se alcance un tiempo determinado (fijado con antelación) u otras condiciones.

Este método, tiene la particularidad de que al principio de su ejecución, explora una gran parte del espacio de búsqueda. Pero a medida que la temperatura desciende, el algoritmo va reduciendo cada vez más su espacio de búsqueda, hasta que finalmente se centra en pequeños espacios que contienen buenas soluciones y acaba al encontrar un mínimo local en esos espacios.

**Más información:**

[https://es.wikipedia.org/wiki/Algoritmo\\_de\\_recocido\\_simulado](https://es.wikipedia.org/wiki/Algoritmo_de_recocido_simulado)

### **7.5 Algoritmo de Generación de columnas.**

Algoritmo muy utilizado para resolver problemas lineales con una gran cantidad de variables y pocas restricciones. Este algoritmo trata con problemas que son muy difíciles de tratar computacionalmente, ya que el número de variables que contienen, los hace muy difíciles de resolver computacionalmente en un tiempo razonable. De esta forma, el algoritmo toma el problema inicial (conocido como problema maestro), y genera un problema maestro restringido, un problema que cumple las condiciones del inicial y donde se generan solo aquellas columnas que se van a necesitar. Esto supone un ahorro de tiempo muy considerable, ya que no estamos explorando todo el espacio de soluciones del problema. Una vez se tiene el problema maestro restringido, el algoritmo lo resuelve y comprueba si la solución es óptima. En caso de que no lo sea, se añade una nueva columna al problema restringido y se vuelve a solucionar. Esta iteración se repite hasta lograr una solución óptima.

**Más información:**

<https://aprendeenlinea.udea.edu.co/revistas/index.php/ingenieria/article/view/17937/15406>

### **7.6 Algoritmo de evolución diferencial.**

El algoritmo de evolución diferencial es un método de optimización parecido al algoritmo genético. Al igual que él, sus soluciones son combinadas entre sí y mutan para conseguir nuevas soluciones que tengan algo de las anteriores. Su principal característica es que utiliza los llamados vectores de prueba, que se comparan con las soluciones obtenidas por el algoritmo.



Aquel que tenga el valor más alto de desempeño (es decir, el que ofrezca una mejor solución para el problema), será el que “sobrevivirá” y por tanto pasará a la siguiente iteración del algoritmo, donde se seguirá mutando y reproduciendo con otras soluciones. Como el algoritmo genético, este algoritmo termina cuando una o varias condiciones se cumplen (número de iteraciones máximo, valor mínimo de solución, tiempo transcurrido etc).

**Más información:**

[https://es.wikipedia.org/wiki/Evoluci%C3%B3n\\_diferencial](https://es.wikipedia.org/wiki/Evoluci%C3%B3n_diferencial)

### **7.7 Algoritmo adaptativo.**

Un algoritmo adaptativo es un tipo de algoritmo que modifica su comportamiento durante su ejecución, adaptándose a cambios en el entorno del problema. Estos algoritmos son capaces de ajustar las actividades que realizan para lograr conseguir la mejor solución posible a medida que las variables del problema varían. Un ejemplo de la aplicación de este tipo de algoritmos puede verse aplicada en los buscadores actuales de internet. Estos algoritmos son capaces de “aprender” de las anteriores búsquedas del usuario para conseguir mejores resultados en futuras búsquedas.

**Más información:** [https://en.wikipedia.org/wiki/Adaptive\\_algorithm](https://en.wikipedia.org/wiki/Adaptive_algorithm)

### **7.8 Algoritmo Heurístico.**

La heurística es una técnica utilizada la optimización matemática y en la ciencia computacional que se utiliza cuando los métodos clásicos para resolver problemas no pueden encontrar soluciones en un tiempo competitivo. Esto se consigue sacrificando alguno de los factores básicos de la solución (como precisión o que la solución sea optima) por velocidad de ejecución.

De esta forma, el objetivo de un algoritmo basado en la heurística es el de hallar una solución para el problema en cuestión en un plazo de tiempo razonable. No se garantiza que la solución obtenida sea la mejor solución para el problema, pero al menos el problema podrá resolverse con cierta seguridad. Al poder resolver el problema de forma más rápida, se pueden ejecutar varias simulaciones para comparar las soluciones y quedarse con la mejor de ellas.

**Más información:**

[https://en.wikipedia.org/wiki/Heuristic\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Heuristic_(computer_science))

## **7.9 Descomposición de Benders.**

La descomposición de Benders, es un algoritmo muy utilizado que usa como principio la idea “*divide and conquer*”. Este principio consiste en dividir un problema en sub-problemas de una dificultad menor para resolverlos uno por uno, y así conseguir resolver el problema. Este algoritmo permite resolver problemas lineales muy grandes y que otros algoritmos no pueden resolver de forma adecuada.

En la descomposición de Benders, las variables que existen en el problema a resolver se dividen en dos sets diferentes. El problema inicial se resuelve utilizando solo las variables del primer set y los valores para el segundo conjunto de variables se deciden mediante otro sub-problema. Este sub-problema determina, además, si las decisiones de la primera etapa son factibles o no y en este último caso, se añaden los llamados “Cortes de Bender” al problema principal. Este proceso se sigue realizando en forma de ciclo hasta que no se pueden añadir más cortes.

Este proceso, que soluciona el problema añadiendo soluciones al problema inicial para resolver la búsqueda de la solución óptima se llama “generación de filas”.

### **Más información:**

[https://www.me.utexas.edu/~jensen/ORMM/methods/unit/integer/subunits/teach\\_int/benders.pdf](https://www.me.utexas.edu/~jensen/ORMM/methods/unit/integer/subunits/teach_int/benders.pdf)

## 8. Tabla de algoritmos utilizados.

---

En este apartado se presentarán los diferentes algoritmos utilizados para resolver los problemas combinatorios presentados, ordenados de forma cronológica.

Problema	Algoritmo utilizado	Año
Ground holding policy	Algoritmo Heurístico.	1993
Asignación de tripulaciones	Algoritmo Genético híbrido (con heurística de búsqueda local).	1996
Ground holding policy	Algoritmo Heurístico.	1998
Asignación de flotas	Simulación de Recocido (SA).	1999
Asignación de flotas	Programación entera mixta y reducción de coeficientes.	1999
Asignación de tripulaciones	División de problema complejo en sub problemas de menor dificultad, Algoritmo Genético.	2000
Capacidad del espacio aéreo	Modelo Teórico algoritmo propio.	2000
Gestión de ingresos	Algoritmo Adaptativo.	2000
Asignación de tripulaciones	Algoritmo Génético (GraGA).	2001
Ground holding policy	Análisis Finito de Perturbación (FPA), Algoritmo de Control.	2001
Ground holding policy	Algoritmo Genético.	2001
Asignación de rutas.	Generación de columnas y Branch Bound.	2001
Gestión de ingresos	Algoritmo basado en la programación dinámica aproximada.	2003
Asignación de tripulaciones	Descomposición de Benders, generación de columnas y algoritmo de Branch-and-Bound.	2005

<b>Problema</b>	<b>Algoritmo utilizado</b>	<b>Año</b>
Ground holding policy	Método analizador de eventos discretos.	2005
Asignación de flotas	Modelo matemático estructura de red multi-artículo no lineal y concepto de homogeneidad, aproximación heurística.	2006
Gestión de ingresos	Algoritmos online.	2006
Asignación de flotas	Optimización de la colonia de hormigas.	2007
Asignación de rutas.	Descomposición de Benders, generación de columnas y generación de restricciones dinámicas.	2007
Capacidad del espacio aéreo	Algoritmo de Clustering.	2008
Asignación de rutas.	Algoritmo Iterativo.	2009
Capacidad del espacio aéreo	Diagrama de Veronoi junto con algoritmo de iteración profunda y algoritmo genético.	2009
Capacidad del espacio aéreo	Directed Acyclic Graph y Algoritmo Genetico Multi-Objetivo.	2009
Capacidad del espacio aéreo	Algoritmo Geométrico y heurística (Binary Space Partitions y Pie-Partitions).	2009
Ground holding policy	Algoritmo Heurístico.	2010
Gestión de ingresos	Algoritmo de aproximación estocástico.	2010
Asignación de tripulaciones	Algoritmo de evolución diferencial (personalizado).	2010

<b>Problema</b>	<b>Algoritmo utilizado</b>	<b>Año</b>
Asignación de tripulaciones	Algoritmo de optimización la colonia de hormigas.	2011
Capacidad del espacio aéreo	Algoritmo de resolución modelado mediante una red de Petri coloreada (CPN Tools).	2012
Gestión de ingresos	Algoritmo Genético.	2012
Asignación de rutas.	Algoritmo Genético.	2012
Asignación de rutas.	Algoritmo AFS-MRA (The Airline Flight Schedule Design and Maintenance Routing Algorithm).	2013
Asignación de flotas	Algoritmo Genético.	2013
Gestión de ingresos	Algoritmo Genético. Matlab.	2014
Gestión de ingresos	Algoritmos Genéticos.	2014
Asignación de rutas.	Algoritmo Heurístico.	2014

## 9. Conclusiones

---

Como ha podido observarse a lo largo de este proyecto, los problemas de optimización combinatoria están muy presentes en el mundo aeronáutico y su correcta gestión y tratamiento por parte de las aerolíneas, puede suponer la diferencia entre un modelo de negocio exitoso o el hundimiento de una compañía aérea.

En primer lugar, podemos observar que en la gran mayoría de las soluciones propuestas, los autores buscan soluciones cercanas a las óptimas y no encuentran la solución óptima del problema en cuestión. Así pues, pese a los avances en potencia de cómputo, vemos que los problemas combinatorios siguen siendo muy complicados de resolver. Debido a su inmenso espacio de soluciones, los algoritmos que los tratan deben enfocar estos problemas de forma inteligente, para tratar de hallar una solución factible y que ofrezca buenos resultados, todo ello en un tiempo razonable.

En segundo lugar, se ha observado que los sistemas con los que cuentan las aerolíneas para afrontar este tipo de problemas están desfasados, ya que en muchos de los documentos consultados, los autores realizan simulaciones donde comparan las soluciones ofrecidas por sus enfoques con las de los sistemas de las compañías. En la mayoría de los casos, las soluciones dadas por el enfoque de los autores superaban en calidad a las del sistema de las compañías, obteniendo estos en el mejor de los casos un empate puntual en el resultado.

Una buena explicación a este hecho sería el coste (en tiempo y dinero) de desarrollar e introducir un nuevo sistema de gestión en una compañía aérea, ya que se debe pasar todo un proceso de formación y adaptación muy delicado, cosa que puede explicar el “conformismo” de las aerolíneas en este sentido.

Sin embargo, los resultados de invertir y aplicar en sistemas que resuelvan y mejoren la gestión de este tipo de problemas se ha demostrado muy ventajosa para las compañías que han tomado el riesgo de implementar e investigar nuevos sistemas, como en el ya descrito caso de American Airlines, con su sistema de gestión de ingresos.

En tercer lugar es interesante mencionar que pese en las soluciones encontradas suelen tomarse datos reales, en la mayoría de los artículos analizados los problemas son llevados a cabo por universidades e integrantes de las mismas. En contadas ocasiones se habla de proyectos llevados a cabo por aerolíneas, gobiernos u organizaciones del sector. Resulta difícil obtener información específica a cerca de nuevos algoritmos o algoritmos actuales utilizados por aerolíneas, y esa podría ser una buena dirección para ampliar el presente trabajo.

En cuarto lugar, cabría destacar la gran variedad de algoritmos que se utilizan para resolver problemas combinatorios, como podemos ver en la tabla de algoritmos final. Para el total de 36 soluciones recopiladas en este proyecto, se han utilizado una gran variedad algoritmos diferentes, así como diversos planteamientos, herramientas para modelar y diferentes enfoques para cada problema en cuestión. Pese a la gran variedad de los resultados obtenidos, podemos ver como el

algoritmo genético es el más escogido por los autores para enfrentar problemas de optimización combinatoria. Este algoritmo ha sido elegido, parcial o totalmente un total de 9 veces en 36 soluciones, y es mayoritariamente utilizado en soluciones desarrolladas en los últimos cuatro años.

Como se puede observar en la tabla final de algoritmos, las diferentes soluciones del problema abarcan un periodo de tiempo extenso (21 años), lo que nos da una idea del interés por desarrollar y encontrar soluciones efectivas a este tipo de problemas. Consultando los apartados de simulación de los diferentes artículos se puede apreciar como la escala de los problemas va creciendo de forma continuada a durante los años.

Por último, cabría mencionar la dificultad de encontrar artículos completos de resolución de problemas combinatorios, ya que muchos de ellos se exponen en conferencias y eventos y el acceso a los mismos está restringido.

En general, los objetivos de este trabajo se han cumplido satisfactoriamente; se ha conseguido encontrar un buen número de soluciones para seis problemas de optimización combinatoria, así como proveer de referencias a cualquier interesado en la materia, para poder acceder a los documentos originales. El presente documento puede resultar de mucha ayuda a personas interesadas en este tipo de problemas.

Como ya se ha mencionado, un buen trabajo a desarrollar en el futuro sería el de investigar que algoritmos o sistemas utilizan las aerolíneas actualmente y realizar comparaciones con sistemas desarrollados por autores individuales o incluso con los de otras aerolíneas.

## 10. Bibliografía.

---

### 10.1 Artículos

[1] M. Dunbar, G. Froyland, and C. L. Wu. Robust Airline Schedule Planning: Minimizing Propagated Delay in an Integrated Routing and Crewing Framework. *Transportation Science*, 46(2):204–216, 2012.

[2] Mercier A, Cordeau J-F, Soumis F. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research* 2005;32:1451–76.

[3] KERATI, Sangcham, et al. A heuristic genetic algorithm approach for the airline crew scheduling problem. *En Workshop on Multiple Objective Metaheuristics, Paris. 2002.*

[4] OZDEMIR, H. Timucin; MOHAN, Chilukuri K. Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences*, 2001, vol. 133, no 3, p. 165-173.

[5] MERCIER, Anne; CORDEAU, Jean-François; SOUMIS, François. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 2005, vol. 32, no 6, p. 1451-1476.

[6] LEVINE, David. Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations Research*, 1996, vol. 23, no 6, p. 547-558.

[7] SANTOSA, Budi, et al. Using differential evolution method to solve crew rostering problem. *Applied Mathematics*, 2010, vol. 1, no 04, p. 316.

[8] DENG, Guang-Feng; LIN, Woo-Tsong. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, 2011, vol. 38, no 5, p. 5787-5793.

[10] BÉLANGER, Nicolas, et al. Weekly airline fleet assignment with homogeneity. *Transportation Research Part B: Methodological*, 2006, vol. 40, no 4, p. 306-318.

[11] SUHL, Uwe H.; SUHL, Leena M. Solving Airline-fleet Scheduling Problems with Mixed-integer Programming. *En Operational Research in Industry*. Palgrave Macmillan UK, 1999. p. 135-156.

[12] GÖTZ, S., et al. Solving the weekly fleet assignment problem for large airlines. *En Proceedings of the III Metaheuristic International Conference*. 1999.

[13] MARTÍN, Francisco Javier Diego; MANTECA, José Ángel González; ARIAS, Javier Carrasco. Análisis del problema de asignación de flotas flexible con ventanas de tiempo discretas, duración variable, preferencias de salida y



relaciones de precedencia. Optimización por colonias de hormigas. *Dirección y Organización*, 2008, no 35, p. 61-68.

[14] LI, Yaohua; TAN, Na. Study on Fleet Assignment Problem Model and Algorithm. *Mathematical Problems in Engineering*, 2013, vol. 2013.

[15] "International Journey of advanced Research in Computer Science and Software Engineering". Volumen 4, problema 1. ISSN: 2277 128X

[16] BERTSIMAS, Dimitris; POPESCU, Ioana. Revenue management in a dynamic network environment. *Transportation science*, 2003, vol. 37, no 3, p. 257-277.

[17] LAN, Yingjie, et al. Revenue management with limited demand information. *Management Science*, 2008, vol. 54, no 9, p. 1594-1609.

[18] VOCK, Sebastian; ENZ, Steffen; CLEOPHAS, Catherine. Genetic algorithms for calibrating airline revenue management simulations. En *Proceedings of the Winter Simulation Conference 2014*. IEEE, 2014. p. 264-275.

[19] ETEBARI, F.; AAGHAIE, A.; KHOSHALHAN, F. A Genetic Algorithm for Choice-Based Network Revenue Management. *Iranian Journal of Operations Research*, 2011, vol. 3, no 1, p. 89-103.

[20] VAN RYZIN, Garrett; MCGILL, Jeff. Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. *Management Science*, 2000, vol. 46, no 6, p. 760-775.

[21] KUNNUMKAL, Sumit; TOPALOGLU, Huseyin. A stochastic approximation algorithm for making pricing decisions in network revenue management problems. *Journal of Revenue and Pricing Management*, 2010, vol. 9, no 5, p. 419-442.

[22] LUČOVNIK, Brane, et al. Theoretical Model for Determination of Airspace Capacity. *PROMET-Traffic&Transportation*, 2012, vol. 14, no 2, p. 71-78.

[23] LIU, Fangqin; HU, Minghua. Airspace capacity management based on control workload and coupling constraints between airspaces. En *Computer Modeling and Simulation*, 2009. ICCMS'09. International Conference on. IEEE, 2009. p. 329-334.

[24] BRINTON, C. R.; PLEDGIE, S. Airspace partitioning using flight clustering and computational geometry. En *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*. IEEE, 2008. p. 3. B. 3-1-3. B. 3-10.

[25] CA, Zúñiga; PIERA, M. A.; BARUWA, O. T. Pre-tactical Trajectory De-Confliction Algorithm for Air Traffic Management. 2012.

- [26] XUE, Min. Airspace sector redesign based on Voronoi diagrams. *Journal of Aerospace Computing, Information, and Communication*, 2009, vol. 6, no 12, p. 624-634.
- [27] BASU, Amitabh; MITCHELL, Joseph SB; SABHNANI, Girish Kumar. Geometric algorithms for optimal airspace design and air traffic controller workload balancing. *Journal of Experimental Algorithmics (JEA)*, 2009, vol. 14, p. 3.
- [28] VRANAS, Peter B.; BERTSIMAS, Dimitris J.; ODoni, Amedeo R. The multi-airport ground-holding problem in air traffic control. *Operations Research*, 1994, vol. 42, no 2, p. 249-261.
- [29] LILI, W. A. N. G.; ZHANG, Zhaoning. A RECURSION EVENT-DRIVEN MODEL TO SOLVE THE SINGLE AIRPORT GROUND-HOLDING PROBLEM. En *Proceedings of the Eastern Asia Society for Transportation Studies*. 2005. p. 500-506.
- [30] WANG, Lili; ZHANG, Zhaoning; LIU, Jimin. Multiple airport ground holding dynamic model and optimal algorithm basing flight priority. En *Control and Automation (ICCA), 2010 8th IEEE International Conference on*. IEEE, 2010. p. 1081-1085.
- [31] PANAYIOTOU, Christos G.; CASSANDRAS, Christos G. A sample path approach for solving the ground-holding policy problem in air traffic control. *IEEE Transactions on Control Systems Technology*, 2001, vol. 9, no 3, p. 510-523.
- [32] PULUGURTHA, Srinivas S.; NAMBIAN, Shashi S. Using genetic algorithms to evaluate aircraft ground holding policy under static conditions. *Journal of transportation engineering*, 2001, vol. 127, no 5, p. 433-441.
- [33] BRUNETTA, Lorenzo; GUASTALLA, Guglielmo; NAVAIZIO, Lisa. Solving the multi-airport ground holding problem. *Annals of operations research*, 1998, vol. 81, p. 271-288.
- [34] CORDEAU, Jean-François, et al. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 2001, vol. 35, no 4, p. 375-388.
- [35] WEIDE, Oliver; RYAN, David; EHRGOTT, Matthias. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 2010, vol. 37, no 5, p. 833-844.
- [36] MERCIER, Anne; SOUMIS, François. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 2007, vol. 34, no 8, p. 2251-2265.
- [37] DÍAZ-RAMÍREZ, Jenny; HUERTAS, José Ignacio; TRIGOS, Federico. Simultaneous schedule design & routing with maintenance constraints for single

fleet airlines. *International Journal of Engineering*, 2013, vol. 2, no 2, p. 2305-8269.

[38] DUNBAR, Michelle; FROYLAND, Gary; WU, Cheng-Lung. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers & Operations Research*, 2014, vol. 45, p. 68-86.

[39] REINERS, Torsten, et al. Integrated aircraft scheduling problem: An auto-adapting algorithm to find robust aircraft assignments for large flight plans. *EnSystem Science (HICSS), 2012 45th Hawaii International Conference on. IEEE*, 2012. p. 1267-1276.

[40] Abdelgany Ahmed; Abdelgan Khaled. (2009). *Modeling Applications in the Airline Industry*.

[41] Barnhart, Cynthia; Smith Barney. (2012) *Quantitative Problem Solving Methods in the Airline Industry*.

[42] Bazargan, Massoud. (2010). *Airline Operations and Scheduling*.

[43] Walsh R. Conor. (2011). *Airline Industry: Strategies, Operations and Safety*.

[44] Graham, Anne. (2009). *Managing Airports*.

## **10.2 Webs**

[I] <http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/problema-del-agente-viajero-tsp/>

[II] [http://www.fomento.gob.es/mfom/lang\\_castellano/default.htm](http://www.fomento.gob.es/mfom/lang_castellano/default.htm)

[III] <https://desdelacabinadevuelo.com/2015/12/23/buenas-expectativas-en-el-transporte-aereo-para-el-2016/>

[IV] <http://santacenero.blogspot.com.es/2015/05/mi-camino-hacia-el-aprendizaje-de.html>

[V] <https://www.mcs.anl.gov/~itf/dbpp/text/node21.html>

[VI] [https://es.wikipedia.org/wiki/Algoritmo\\_de\\_la\\_colonia\\_de\\_hormigas](https://es.wikipedia.org/wiki/Algoritmo_de_la_colonia_de_hormigas)

[VII] <http://web.cecs.pdx.edu/~mm/cs410-510-winter-2005/ga-tutorial.pdf>

[VIII] [https://es.wikipedia.org/wiki/Algoritmo\\_de\\_recocido\\_simulado](https://es.wikipedia.org/wiki/Algoritmo_de_recocido_simulado)

[IX] <https://aprendeonlinea.udea.edu.co/revistas/index.php/ingenieria/article/view/17937/15406>

- [X] [https://es.wikipedia.org/wiki/Evoluci%C3%B3n\\_diferencial](https://es.wikipedia.org/wiki/Evoluci%C3%B3n_diferencial)
- [XI] [https://en.wikipedia.org/wiki/Adaptive\\_algorithm](https://en.wikipedia.org/wiki/Adaptive_algorithm)
- [XI] [https://en.wikipedia.org/wiki/Heuristic\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Heuristic_(computer_science))
- [XII] [https://www.me.utexas.edu/~jensen/ORMM/methods/unit/integer/subunits/each\\_int/benders.pdf](https://www.me.utexas.edu/~jensen/ORMM/methods/unit/integer/subunits/each_int/benders.pdf)
- [XIII] [https://www.researchgate.net/publication/228774764\\_A\\_heuristic\\_genetic\\_algorithm\\_approach\\_for\\_the\\_airline\\_crew\\_scheduling\\_problem](https://www.researchgate.net/publication/228774764_A_heuristic_genetic_algorithm_approach_for_the_airline_crew_scheduling_problem)
- [XIV] <http://www.pcporoje.com/filedata/644547.pdf>
- [XV] [https://www.researchgate.net/publication/222403543\\_A\\_computational\\_study\\_of\\_Benders\\_decomposition\\_for\\_the\\_integrated\\_aircraft\\_routing\\_and\\_crew\\_scheduling\\_problem](https://www.researchgate.net/publication/222403543_A_computational_study_of_Benders_decomposition_for_the_integrated_aircraft_routing_and_crew_scheduling_problem)
- [XVI] [http://140.221.6.23/pub/tech\\_reports/reports/P557.pdf](http://140.221.6.23/pub/tech_reports/reports/P557.pdf)
- [XVII] <http://www.scirp.org/journal/PaperInformation.aspx?PaperID=2910>
- [XVIII] <http://www.sciencedirect.com/science/article/pii/S0957417410011978>
- [XIX] [https://www.researchgate.net/profile/Jacques\\_Desrosiers/publication/223805452\\_Weekly\\_airline\\_fleet\\_assignment\\_with\\_homogeneity/links/0fcfd5086aa67b7a49000000.pdf](https://www.researchgate.net/profile/Jacques_Desrosiers/publication/223805452_Weekly_airline_fleet_assignment_with_homogeneity/links/0fcfd5086aa67b7a49000000.pdf)
- [XX] [http://link.springer.com/chapter/10.1057%2F9780230372924\\_7](http://link.springer.com/chapter/10.1057%2F9780230372924_7)
- [XXI] <http://www.revistadyo.com/index.php/dyo/article/view/57/57>
- [XXII] <http://www.hindawi.com/journals/mpe/2013/581586/>
- [XXIII] [http://www.ijarcse.com/docs/papers/Volume\\_4/1\\_January2014/V4I1-0366.pdf](http://www.ijarcse.com/docs/papers/Volume_4/1_January2014/V4I1-0366.pdf)
- [XXIV] <http://www.mit.edu/~dbertsim/papers/Revenue%20Management/Revenue%20Management%20in%20a%20Dynamic%20Network%20Environment.pdf>
- [XXV] <https://www.gsb.stanford.edu/sites/default/files/documents/CompetitiveAnalysisofOnlineAlgorithmsinRevenueManagementAModelingPerspective.pdf>
- [XXVI] [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7019894&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D7019894](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7019894&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D7019894)
- [XXVII] <http://www.iors.ir/journal/article-1-137-fa.pdf>

[XXVIII] [https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/3921/air\\_line\\_seat\\_protection.pdf](https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/3921/air_line_seat_protection.pdf)

[XXIX] [https://people.orie.cornell.edu/huseyin/publications/pricing\\_network.pdf](https://people.orie.cornell.edu/huseyin/publications/pricing_network.pdf)

[XXX] <http://www.fpz.unizg.hr/traffic/index.php/PROMTT/article/viewFile/849/700>

[XXXI] [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4797410&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D4797410](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4797410&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4797410)

[XXXII] [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4702800&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D4702800](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4702800&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4702800)

[XXXIII] <http://www.hala-sesar.net/sites/default/files/documents/p233-zuniga.pdf>

[XXXIV] [http://aviationsystems.arc.nasa.gov/publications/strategic/Xue\\_GNC\\_2008\\_02.pdf](http://aviationsystems.arc.nasa.gov/publications/strategic/Xue_GNC_2008_02.pdf)

[XXXV] <http://www.ams.sunysb.edu/~jsbm/papers/alnex08.pdf>

[XXXVI] <http://www.mit.edu/~dbertsim/papers/AirTransportation/The%20multi-airport%20groundholding%20problem%20in%20air%20traffic%20control.pdf>

[XXXVII] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.509.6034&rep=rep1&type=pdf>

[XXXVIII] [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5524112&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D5524112](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5524112&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5524112)

[XXXIX] <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=918903&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F87%2F19873%2F00918903.pdf%3Farnumber%3D918903>

[XL] [http://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(2001\)127%3A5\(442\)](http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2001)127%3A5(442))

[XLI] <http://link.springer.com/article/10.1023%2FA%3A1018909224543>

[XLII] <https://www.semanticscholar.org/paper/Benders-Decomposition-for-Simultaneous-Aircraft-Cordeau-Stojkovic/ce867bfcd8316156582fcceb3bb0a8d2458acc75/pdf>

[XLIII] <http://eprints.lancs.ac.uk/63744/1/10.pdf>

[XLIV] [https://www.researchgate.net/publication/222360140\\_An\\_integrated\\_aircraft\\_routing\\_crew\\_scheduling\\_and\\_flight\\_retiming\\_model](https://www.researchgate.net/publication/222360140_An_integrated_aircraft_routing_crew_scheduling_and_flight_retiming_model)

[XLV] <http://eaasjournal.org/survey/userfiles/files/Aeronautical%20engineering%203.pdf>

[XLVI] [http://www.optimization-online.org/DB\\_FILE/2013/03/3801.pdf](http://www.optimization-online.org/DB_FILE/2013/03/3801.pdf)

[XLVII] <https://www.computer.org/csdl/proceedings/hicss/2012/4525/00/4525b267.pdf>

Firma Alumno:

Duany.R