

# Trabajo Fin de Máster

## Máster en Ingeniería Industrial

### Optimización estática de la secuencia de aterrizajes en entornos con varias pistas

Autor: José Luis García Calderón

Tutor: Alejandro Escudero Santana

**Dpto. de Organización Industrial y Gestión de  
Empresas II**  
**Escuela Técnica Superior de Ingeniería**

Sevilla, 2019







Trabajo Fin de Máster  
Máster en Ingeniería Industrial

# **Optimización estática de la secuencia de aterrizajes en entornos con varias pistas**

Autor:

José Luis García Calderón

Tutor:

Alejandro Escudero Santana

Profesor Contratado Doctor

Dpto. de Organización Industrial y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019



# Agradecimientos

---

Este proyecto es el fruto de siete largos años de un intenso y continuado trabajo y sacrificio personal, que me ha permitido plasmar el aprendizaje adquirido durante el grado y el máster.

A mi familia por la confianza y cariño que me demuestran día a día.

A D. Alejandro Escudero Santana por hacer posible la realización de este trabajo, su paciencia y tiempo para limar tantos detalles, que me ha ido proponiendo y que han mejorado considerablemente el resultado final de este trabajo.

*José Luis García Calderón*

*Sevilla, 2019*



# Resumen

---

Este proyecto se ocupa de la resolución del problema de secuenciar los aterrizajes de un conjunto de aviones en un aeropuerto (ALP) considerando un entorno estático.

En los aeropuertos es habitual que el gestor aeroportuario se enfrente a problemas de capacidad que se agravan en determinadas épocas del año, días concretos o rangos horarios. Esta circunstancia se manifiesta en forma de retrasos generalizados, ya que, cuando en un aeropuerto congestionado es necesario modificar el horario de un vuelo, reajustar la planificación hace que muchos otros se vean afectados.

Aumentar la capacidad de los aeropuertos (pistas de aterrizajes y puertas de embarque) se ha convertido en un factor limitante a la hora de hacer frente a la creciente demanda de vuelos en la actualidad. Uno de los principales factores que determinan el rendimiento y eficiencia de las pistas de aterrizaje son los criterios de separación requeridos entre los aterrizajes y despegues de los aviones. Debido a su complejidad, es muy difícil encontrar la solución óptima al problema en la mayoría de los casos.

El colapso no afecta solamente a la infraestructura del aeropuerto, sino también en los espacios para estacionar coches, en las zonas reservadas para taxis, en la necesidad de contar con conexiones ferroviarias y, sobre todo, en que las terminales cuenten con suficientes *slots* para que las aeronaves puedan operar. Sin embargo, otro problema y de mayor relevancia es la saturación de las rutas. En puntos del planeta como el Atlántico Norte o el Sudeste Asiático se viven auténticas autopistas de aviones, cuyo número crece año a año y puede ocasionar retrasos y cancelaciones.

Para obtener soluciones al problema presentado, se utilizan los algoritmos de recocido simulado (SA) y búsqueda en entornos variables (VNS). Para comparar el desempeño de ambas técnicas se resuelven cinco problemas de dimensiones 10, 15, 20, 30 y 50 aviones, respectivamente.



# Abstract

---

This project deals with solving the problem of sequencing the landings of a set of airplanes at an airport (ALP) considering a static environment.

It is usual for the airport manager to face capacity problems that worsen at certain times of the year, specific days or time ranges. This circumstance manifests itself in the form of generalized delays, since when a congested airport needs to modify the schedule of a flight, readjusting the planning causes many others to be affected.

Increasing the capacity of airports (runways and boarding gates) has become a limiting factor when it comes to coping with the growing demand for flights today. One of the main factors that determine the performance and efficiency of runways are the separation criteria required between aircraft landings and takeoffs. Due to its complexity, it is very difficult to find the optimal solution to the problem in most cases.

The collapse does not only affect the infrastructure of the airport, but also in spaces for parking cars, in areas reserved for taxis, in the need to have rail connections and, above all, in which the terminals have enough slots so that the aircraft can operate. However, another problem and of greater relevance is the saturation of the routes. In areas of the planet such as the North Atlantic or Southeast Asia real motorways of airplanes live, whose number grows every year and can cause delays and cancellations.

To obtain solutions to the presented problem, the Simulated Annealing (SA) and Variable Neighbourhood Search (VNS) algorithms are used. To compare the performance of both techniques, five problems of dimensions 10, 15, 20, 30 and 50 aircraft, respectively, are solved.



# Índice

---

<b>Agradecimientos</b>	<b>IV</b>
<b>Resumen</b>	<b>VI</b>
<b>Abstract</b>	<b>VIII</b>
<b>Índice</b>	<b>X</b>
<b>índice de Tablas</b>	<b>XII</b>
<b>Índice de Figuras</b>	<b>XIII</b>
<b>1 INTRODUCCIÓN Y OBJETIVOS</b>	<b>1</b>
1.1 <i>Objetivos</i>	2
1.2 <i>Estructura del documento</i>	3
<b>2 ANTECEDENTES</b>	<b>4</b>
2.1 <i>Gestión del Tráfico Aéreo</i>	5
2.2 <i>Espacio aéreo</i>	5
2.2.1 <i>Reglas de vuelo</i>	6
2.2.2 <i>Divisiones del espacio aéreo</i>	6
2.2.3 <i>Clases de espacio aéreo</i>	8
2.2.4 <i>Servicios de Gestión del Tránsito Aéreo (ATS)</i>	9
<b>3 DEFINICIÓN DEL PROBLEMA</b>	<b>11</b>
3.1 <i>Aterrizaje de aviones</i>	11
3.1.1 <i>Patrón de tráfico de los aeropuertos</i>	11
3.1.2 <i>Maniobra de aterrizaje</i>	14
3.2 <i>Formulación matemática clásica</i>	15
3.2.1 <i>Variables</i>	16
3.2.2 <i>Restricciones</i>	16
3.2.3 <i>Objetivo</i>	17
3.3 <i>Ejemplo conceptual</i>	18
<b>4 ESTADO DEL ARTE</b>	<b>20</b>
<b>5 MÉTODOS DE RESOLUCIÓN</b>	<b>26</b>
5.1 <i>Revisión de las metodologías de resolución</i>	26
5.1.1 <i>Algoritmos exactos</i>	27
5.1.2 <i>Heurísticas</i>	27
5.1.3 <i>Metaheurísticas</i>	27
5.2 <i>Recocido simulado</i>	28
5.3 <i>Búsqueda en Vecindario Variables (VNS)</i>	31
5.4 <i>Adaptación al problema</i>	32
5.4.1 <i>Recocido Simulado</i>	32
5.4.2 <i>Búsqueda en entornos variables</i>	33

<b>6</b>	<b>ANALISIS DE RESULTADOS</b>	<b>40</b>
6.1	<i>Calibrado de los métodos</i>	40
6.1.1	Recocido Simulado	40
6.1.2	Búsqueda en entornos variables	43
6.2	<i>Resultados computacionales</i>	45
6.2.1	airland1	46
6.2.2	airland2	52
6.2.3	airland6	54
6.3	<i>Análisis de resultados</i>	56
<b>7</b>	<b>CONCLUSIONES</b>	<b>57</b>
7.1	<i>Líneas de investigación futura</i>	57
<b>8</b>	<b>REFERENCIAS</b>	<b>58</b>
	<b>ANEXO II. BATERÍA DE PROBLEMAS</b>	<b>61</b>

# ÍNDICE DE TABLAS

---

TABLA 1. CLASIFICACIÓN DEL ESPACIO AÉREO (FERNANDO, S. F.) .....	8
TABLA 2. CLASIFICACIÓN DE LOS VIRAJES .....	12
TABLA 3. RESTRICCIONES DEL MODELO MATEMÁTICOS.....	16
TABLA 4. INTERVALOS DE TIEMPO PARA EL ATERRIZAJE DE AVIONES .....	18
TABLA 5. TIEMPOS DE SEPARACIÓN ENTRE CADA PAR DE AVIONES.....	19
TABLA 6. EJEMPLO CONCEPTUAL.....	19
TABLA 7. CLASIFICACIÓN DE LAS REFERENCIAS BIBLIOGRÁFICAS .....	21
TABLA 8. CLASIFICACIÓN DE LOS ARTÍCULOS SEGÚN EL ENTORNO DEL PROBLEMA .....	24
TABLA 9. CLASIFICACIÓN DE LOS ARTÍCULOS SEGÚN EL TIPO DE METODOLOGÍA .....	24
TABLA 10. CLASIFICACIÓN DE LOS ARTÍCULOS SEGÚN EL NÚMERO DE PISTAS DE ATERRIZAJE .....	25
TABLA 11. ANALOGÍA DEL RECOCIDO SIMULADO .....	32
TABLA 12. VECINDARIOS PARA LAS VERSIONES CON UNA Y VARIAS PISTAS DE ATERRIZAJE .....	34
TABLA 13. PARÁMETROS CALIBRADOS.....	40
TABLA 14. VALORES ANALIZADOS DE LOS PARÁMETROS .....	41
TABLA 15. CALIBRADO DEL ALGORITMO SA .....	41
TABLA 16. PARÁMETROS DEL ALGORITMO SA .....	43
TABLA 17. CALIBRADO DEL ALGORITMO VNS PARA UNA Y VARIAS PISTAS DE ATERRIZAJE .....	43
TABLA 18. ORDEN DE LOS VECINDARIOS.....	44
TABLA 19. RESULTADOS COMPUTACIONALES.....	45

# ÍNDICE DE FIGURAS

FIGURA 1. EVOLUCIÓN DEL MERCADO AÉREO INTERNACIONAL (IATA).....	2
FIGURA 2. VÓRTICES DE LAS ESTELAS DE UN AVIÓN .....	4
FIGURA 3. REGLAS DE VUELO .....	6
FIGURA 4. REGIONES DEL ESPACIO AÉREO (FERNANDO, 2019) .....	7
FIGURA 5. FIR EN ESPAÑA (AENA) .....	7
FIGURA 6. PATRÓN DE VUELO EN UN AEROPUERTO.....	11
FIGURA 7. VIRAJE DE UN AVIÓN .....	12
FIGURA 8. FUERZAS ACTUANTES DURANTE EL PROCESO DE VIRAJE.....	13
FIGURA 9. INDICADOR DE VIRAJE .....	14
FIGURA 10. FASES DEL ATERRIZAJE. («INTRODUCCIÓN AL MANUAL DE VUELO», S. F.) .....	14
FIGURA 11. COSTES DE PENALIZACIÓN .....	18
FIGURA 12. METODOLOGÍAS DE RESOLUCIÓN. FUENTE: ANBUUDAYASANKAR, S. P, GANESH, K. Y MOHAPATRA, S.....	26
FIGURA 13. FUNCIÓN MULTIMODAL. COMPORTAMIENTO DE HEURÍSTICAS EN ÓPTIMOS LOCALES .....	27
FIGURA 14. <i>OPTIMIZATION BY SIMULATED ANNEALING</i> . FUENTE: KIRKPATRICK <i>ET AL.</i> (1983).....	28
FIGURA 15. PARÁMETROS DEL RECOCIDO SIMULADO.....	30
FIGURA 16. DIAGRAMA DE FLUJO DEL RECOCIDO SIMULADO.....	30
FIGURA 17. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 10, R = 1} .....	46
FIGURA 18. REPRESENTACIÓN GRÁFICA DE SOLUCIÓN SA. {P = 10, R = 1} .....	46
FIGURA 19. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 10, R = 1}.....	47
FIGURA 20. REPRESENTACIÓN GRÁFICA DE SOLUCIÓN VNS. {P = 10, R = 1}.....	47
FIGURA 21. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 10, R = 2} .....	48
FIGURA 22. REPRESENTACIÓN GRÁFICA SOLUCIÓN SA. {P = 10, R = 2} .....	48
FIGURA 23. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 10, R = 2}.....	49
FIGURA 24. REPRESENTACIÓN GRÁFICA DE SOLUCIÓN VNS. {P = 10, R = 2}.....	49
FIGURA 25. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 10, R = 3} .....	50
FIGURA 26. REPRESENTACIÓN GRÁFICA SOLUCIÓN SA. {P = 10, R = 3} .....	50
FIGURA 27. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 10, R = 3}.....	51
FIGURA 28. REPRESENTACIÓN GRÁFICA DE SOLUCIÓN VNS. {P = 10, R = 3}.....	51
FIGURA 29. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 15, R = 1} .....	52
FIGURA 30. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 15, R = 1}.....	52
FIGURA 31. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 15, R = 2} .....	52
FIGURA 32. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 15, R = 2}.....	53
FIGURA 33. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 15, R = 3} .....	53
FIGURA 34. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 15, R = 3}.....	53
FIGURA 35. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 30, R = 1} .....	54
FIGURA 36. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 30, R = 1}.....	54
FIGURA 37. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 30, R = 2}.....	55
FIGURA 38. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 30, R = 2}.....	55
FIGURA 39. EVOLUCIÓN DE LA SOLUCIÓN SA. {P = 30, R = 3}.....	55
FIGURA 40. EVOLUCIÓN DE LA SOLUCIÓN VNS. {P = 30, R = 3}.....	55



# 1 INTRODUCCIÓN Y OBJETIVOS

---

En los aeropuertos es habitual que el gestor aeroportuario se enfrente a problemas de capacidad que se agravan en determinadas épocas del año, días concretos o rangos horarios. Esta circunstancia se manifiesta en forma de retrasos generalizados, ya que, cuando en un aeropuerto congestionado es necesario modificar el horario de un vuelo, reajustar la planificación hace que muchos otros se vean afectados.

La capacidad de las infraestructuras es un factor inversamente proporcional al tiempo que las aeronaves están en el aeropuerto. Por ello, interesa que entren el máximo número de aviones y salgan lo antes posible (rotación). Esto se traduce en un flujo constante de entradas y salidas que permite mantener los accesos, así como las pistas de aterrizaje y otros recursos implicados, ocupados el mínimo tiempo posible.

En 1999, Reynolds-Feighan y Button observaron un crecimiento alarmante del tráfico aéreo en muchos aeropuertos de la Unión Europea, que ya venía apreciándose desde años anteriores, provocando que muchas de estas operaciones aéreas se realizasen fuera de los horarios establecidos y con retrasos medios superiores a veinte minutos. Esto suponía un gran aumento de los costes para las compañías aéreas, principalmente por alterar los parámetros de consumo de combustible y velocidad eficientes condicionados por las ventanas temporales asignada a cada tipo de avión, refiriéndose el tiempo mínimo de dicho intervalo al instante en que aterrizaría el avión si se acercase a su máxima velocidad y, el tiempo máximo correspondería con la velocidad más eficiente y económica a un consumo de combustible (velocidad de crucero).

Eurocontrol tasó estos costes derivados de la congestión aérea entre 3,4 y 5,7 billones de ecus<sup>1</sup> anuales en Europa y alrededor de 4500 billones de dólares anuales en Estados Unidos (ATAG, 2000), costes que demuestran por su magnitud, la gravedad de dicho problema.

La Comisión Europea ha centrado sus esfuerzos prioritariamente en el control del tráfico aéreo, esto es, en la fase de vuelo de los aviones. No obstante, no ha abordado en su totalidad la solución de los problemas de saturación de los aeropuertos en las fases de despegue y aterrizaje, proponiendo soluciones particulares para determinados aeropuertos como la ampliación de infraestructuras existentes y algún que otro cambio en las reglas de despegue y aterrizajes que tienen establecidas las compañías aéreas.

El número de vuelos de pasajeros y aerolíneas de carga sigue creciendo en los últimos años, tendencia que se espera continúe en el futuro. La Asociación de Transporte Aéreo Internacional (IATA) prevé que para el 2037 el tráfico de pasajeros llegará a los 8.200 millones de personas (actualmente se encuentra en los 4.000 millones). Es como si todas las personas que actualmente habitan el planeta, en algún momento del año, viajaran en avión. Para las próximas dos décadas se ha estimado una tasa anual compuesta de crecimiento (CAGR) del 3.5%, lo que implica que el número de pasajeros se duplique respecto los niveles actuales.

---

<sup>1</sup> La Unidad Monetaria Europea (*European Currency Unit*) se creó el 13 de marzo de 1979 para la contabilidad interna de la Comunidad Económica europea (hoy Unión Europea). Se trataba de una "canasta o cesta de monedas" compuesta por la suma de los valores de doce de las quince monedas comunitarias.

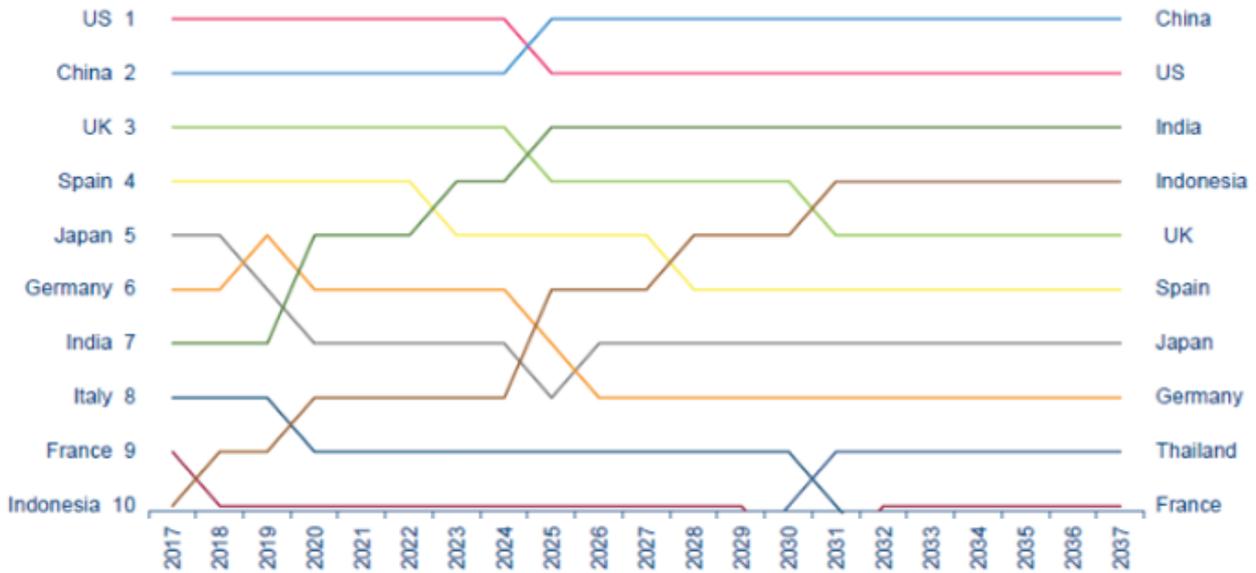


Figura 1. Evolución del Mercado Aéreo Internacional (IATA)

En Asia, ocho de las once terminales más importantes están al límite de su capacidad según la asociación Asian Sky Group en un informe publicado en 2017. De los 1.017 aeropuertos de la región, el de Hong Kong (con más de 70 millones de pasajeros anuales) es el más comprometido en cuanto operatividad, y los de Beijing, Manila y Singapur también tendrán problemas a corto plazo. En España, Aena tiene estudiado y previsto un plan estratégico en el cual se estipulan las ampliaciones a las terminales de Barajas (Madrid) y El Prat (Barcelona), que además de mayor capacidad para los pasajeros también proyecta conexiones con las líneas de tren de alta velocidad (Chuet-Missé, s. f.).

Este problema se encuadra en la clase NP-hard, es decir, no admite soluciones exactas eficientes.

“En la teoría de la complejidad computacional, la clase de complejidad NP-hard contiene los problemas H tales que todo problema L en NP puede ser transformado polinomialmente en H. Esta clase puede ser descrita como aquella que contiene a los problemas de decisión que son como mínimo tan difíciles como un problema de NP. Esta afirmación se justifica porque si se encuentra un algoritmo A que permita resolver uno de los problemas H de NP-hard en tiempo polinómico, entonces es posible construir un algoritmo que trabaje en tiempo polinómico para cualquier problema de NP ejecutando primero la reducción de este problema en H y luego ejecutando el algoritmo A” («NP-hard», 2018).

Cuando el número de variables y restricciones aumentan considerablemente, es obvio que también lo hace el espacio de soluciones posibles y, por tanto, la rutina de búsqueda con el objeto de localizar la mejor no es factible en estos casos.

## 1.1 Objetivos

En este proyecto se persigue, como primer objetivo analizar el problema de optimización de la secuencia de aterrizajes de un conjunto de aviones y, por otra parte, realizar una comparativa entre diversas metaheurísticas en la resolución de este problema, para así poder establecer una guía orientativa hacia la aplicabilidad real en futuras investigaciones.

## 1.2 Estructura del documento

Con el fin de cumplir con estos objetivos, se tratará lo siguiente:

- En el segundo capítulo se describe la situación actual de la gestión del tráfico aéreo y sus principales características, así como la justificación de las restricciones asociadas al problema.
- Para contextualizar el trabajo, en el tercer apartado se detalla el proceso de aterrizaje y se define la formulación matemática del problema.
- En el cuarto capítulo se realiza una revisión del estado del arte del problema y técnicas de resolución conocidas hasta la fecha.
- En el quinto capítulo se presentan teóricamente algunos de los métodos de resolución que se pueden aplicar a este tipo de problemas de optimización, centrandos los esfuerzos en los algoritmos de recocido simulado (SA) y búsqueda en entornos variables (VNS), que son los seleccionados para resolver el problema propuesto.
- En el sexto capítulo se analizan los resultados obtenidos a partir de un conjunto de problemas y son comparados con el fin de determinar qué metodología se ajusta mejor al problema estudiado.
- Por último, en el séptimo capítulo se exponen las conclusiones obtenidas.

## 2 ANTECEDENTES

Aumentar la capacidad de los aeropuertos (pistas de aterrizajes y puertas de embarque) se ha convertido en un factor limitante a la hora de hacer frente a la creciente demanda de vuelos en la actualidad. Uno de los principales factores que determinan el rendimiento y eficiencia de las pistas de aterrizaje son los criterios de separación requeridos entre los aterrizajes y despegues de los aviones. Debido a su complejidad, es muy difícil encontrar la solución óptima al problema en la mayoría de los casos.

Desde que fuese publicado el artículo de Blumsein (1959) donde se estimaba la capacidad o ratio de servicio de una pista de aterrizaje restringida por criterios de separación tanto espaciales como temporales, ha habido una gran variedad de estudios sobre la optimización de la capacidad de los aeropuertos. Aunque un gran volumen de los artículos de investigación que abordan este problema ha sido publicado durante las últimas tres décadas, aún no se ha desarrollado un método que haya sido implementado en la vida real. La razón podría deberse a que estos métodos relajan, incluso podrían descartar, restricciones operativas críticas (seguridad, mayor foco de atención en el problema de decisión frente al problema de control, etc), tiempos de ejecución inasumibles, estudio de un entorno estático frente a entornos dinámicos, ignorar los requisitos o necesidades de los interesados (optimización con un solo objetivo) o depender de características o funciones propias de un aeropuerto particular.

Las soluciones se deben obtener de forma rápida para que le resulte de utilidad a los controladores del tráfico aéreo. Debido a su complejidad, los métodos heurísticos que incluyen algoritmos de búsqueda local pueden ser más apropiados que los métodos enumerativos, como la programación dinámica, más exigente desde el punto de vista computacional.

El problema de encontrar una secuencia de aterrizajes óptima o ALP por sus siglas en inglés, tiene una gran importancia y múltiples aplicaciones en el área del transporte. Consiste en programar y asignar un tiempo de aterrizaje y una pista a cada avión que entra en el radar, minimizando los costes y sujeto a diversas restricciones operacionales. Los costes asociados varían según el tiempo de aterrizaje asignado (ver fig. 11). Las restricciones hacen referencia a la separación requerida entre aterrizajes consecutivos dependiente del tipo de avión y a la ocupación de pista. Para comprender más fácilmente la necesidad de incorporar estas restricciones al problema, se debe abordar el fenómeno de los torbellinos o vórtices de las estelas de los aviones.

La vorticidad es una magnitud física que cuantifica la rotación local de un fluido. En el caso de los aviones, esta propiedad se manifiesta en la formación de vórtices, que es un flujo en rotación espiral con trayectorias de corriente cerradas y dependiente del tipo y tamaño del avión. El flujo de la capa límite que se forma sobre las alas en un momento determinado llega a la punta del ala y se desprende envolviéndose en sendos vórtices, uno detrás de cada ala. La importancia de este fenómeno radica en que, si un avión de menor tamaño entra en un vórtice de un avión mayor puede hacerle perder el control. Mientras que esta situación durante el vuelo es muy poco probable, el tiempo de separación entre aviones durante el despegue y aterrizaje viene determinado por el tiempo necesario para que desaparezca la estela del avión precedente.



Figura 2. Vórtices de las estelas de un avión

Resultado de los vórtices de las estelas de los aviones, en noviembre del año 2008, un learjet 45 entró en la turbulencia de estela producida por un avión pesado Boeing 767 durante su aproximación al Aeropuerto Internacional de la Ciudad de México (AICM), siendo desestabilizado y colisionando en una zona financiera de la ciudad.

La Organización de Aviación Civil Internacional (OACI) considera cuatro categorías de aviones en función de la estela de aire turbulento que genera a su paso, clasificación muy importante desde el punto de vista aeroportuario y del control aéreo por lo comentado en el párrafo anterior. Además, cabe señalar que para realizar esta clasificación se utiliza el concepto del peso máximo al despegue, también conocido por sus siglas en inglés MTOW (*Maximum Take-Off Weight*), que hace referencia al máximo peso para el cual el avión está certificado para volar, incluyendo el peso del aeroplano en vacío operativo, el 100% de carga útil y un 100% de combustible. A continuación, se indican las diferentes clases de aviones:

- L (*Light*): MTOW < 7000 kg
- M (*Medium*): 7000 < MTOW < 136000 kg
- H (*Heavy*): MTOW > 136000 kg
- S (*Super*): exclusivo del Airbus A380.

El método más simple y utilizado por los controladores aéreos para realizar esta secuencia en una única pista de aterrizaje es asignar los tiempos según orden de llegada, comúnmente conocido como *First-Come-First-Served* (FCFS). No obstante, se ha demostrado que esta técnica rara vez proporciona las mejores soluciones en términos de capacidad o retrasos. En Dear (1976) se introdujo el concepto *Constrained Position Shifting* (CPS), Cambio de Posición Restringida traducido al castellano, para el problema ALP para evitar que las posiciones de los aviones en la secuencia final de aterrizaje difiriesen más de un número de posiciones prefijadas (MPS) del orden establecido por la técnica FCFS.

## 2.1 Gestión del Tráfico Aéreo

En las siguientes líneas se describe brevemente la historia y funciones del Control del Tráfico Aéreo, también conocido como ATC (del inglés *Air Traffic Control*). Se presenta en los países partícipes de la convención que tuvo lugar en la ciudad de Chicago en 1944, entre ellos España. Fruto de esta convocatoria salió el Convenio sobre Aviación Civil Internacional, conocido como el Convenio de Chicago, el cual estableció las bases del Derecho Aéreo y en esencia controla el tráfico aéreo, evitando colisiones, proporciona información a los pilotos y asegura los servicios de socorro y salvamento.

Este convenio es el que actualmente regula la aviación civil internacional en todo el mundo.

A medida que la aviación se fue desarrollando, los vuelos fueron cada vez más largos, el desarrollo de aplicaciones útiles y la cantidad de aeronaves fue en aumento. Esto condujo a la necesidad de establecer alguna forma de comunicación entre los pilotos de las aeronaves y el personal en tierra. Para enviar mensajes a las aeronaves se utilizaban señales pintadas en el suelo, banderas, luces, etc. En cambio, para comunicarse con tierra, las aeronaves utilizaban maniobras, señales luminosas, sonoras o visuales.

La revolución definitiva en el control del tráfico aéreo tuvo lugar durante la Segunda Guerra Mundial, periodo en el cual la aviación experimentó los mayores avances de su historia, con el uso de la radio como medio de comunicación y el radar para conocer la ubicación.

## 2.2 Espacio aéreo

El espacio aéreo hace referencia a todo lo que hay al aire libre, desde el nivel del suelo o del mar hasta los límites de la atmósfera, dividido según acuerdos internacionales, para organizar y regular su uso por las aeronaves según unas normas consensuadas. En España, AESA asume las competencias en materia de espacio aéreo.

## 2.2.1 Reglas de vuelo

Existen dos tipos de reglas de vuelo: VFR (*Visual Flight Rules*) e IFR (*Instrumental Flight Rules*)

“Las Reglas de Vuelo Visual son un conjunto de normas que establecen las condiciones suficientes para que el piloto pueda dirigir su aeronave, navegar y mantener la separación de seguridad con cualquier obstáculo con la única ayuda de la observación visual” («Regla de Vuelo Visual - EcuRed», s. f.).

El piloto con habilitación VFR vuela siguiendo los aspectos característicos del terreno y se mantiene distante visualmente de las nubes y de los otros aviones. El vuelo debe realizarse en condiciones VMC (*Visual Meteorological Conditions*), que dependen del espacio aéreo en que se encuentre el avión. Además, el piloto se hace responsable de las incidencias que puedan presentarse en vuelo.

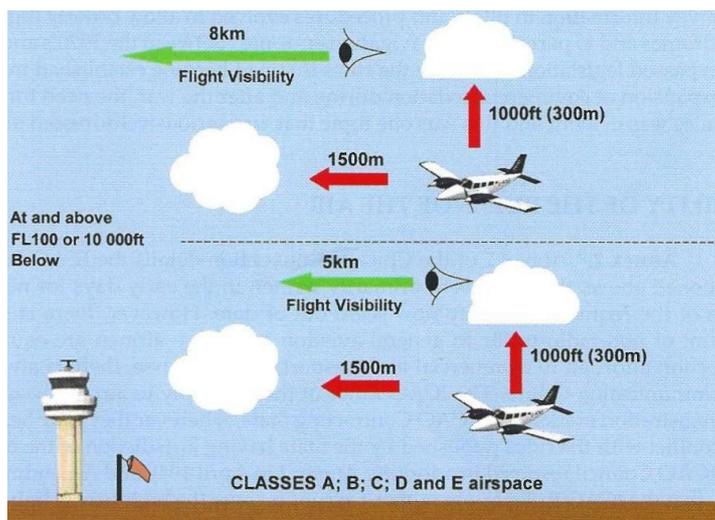


Figura 3. Reglas de vuelo

En cambio, IFR son el conjunto de normas y procedimientos que regulan el vuelo de aeronaves con base en el uso de instrumentos para la navegación, lo cual implica que no necesariamente debe tener contacto visual con el terreno. El piloto debe seguir las autorizaciones de control y notificar su paso por determinados puntos. Puede volar con condiciones meteorológicas difíciles y adversas.

## 2.2.2 Divisiones del espacio aéreo

OACI agrupa las regiones con necesidades de gestión del espacio aéreo similares, asignando a cada zona un código.

- ASIA: Asia
- PAC: Pacífico
- NAM: Norte América
- CAR: América Central
- SAM: Sur América
- NAT: Atlántico Norte
- EUR: Europa
- AFI: África
- MID: Oriente Medio

Dentro de cada una de estas zonas, el espacio aéreo se divide en regiones de información de vuelo, es decir, demarcaciones espaciales para la gestión, el control y la organización del tráfico aéreo. Cada una de ellas, a su vez, se divide en dos sectores:

- FIR (*Flight Information Region*): abarca desde el suelo hasta el nivel de vuelo FL 245
- UIR (*Upper Information Region*): se extiende desde el límite superior del FIR hasta el FL 460

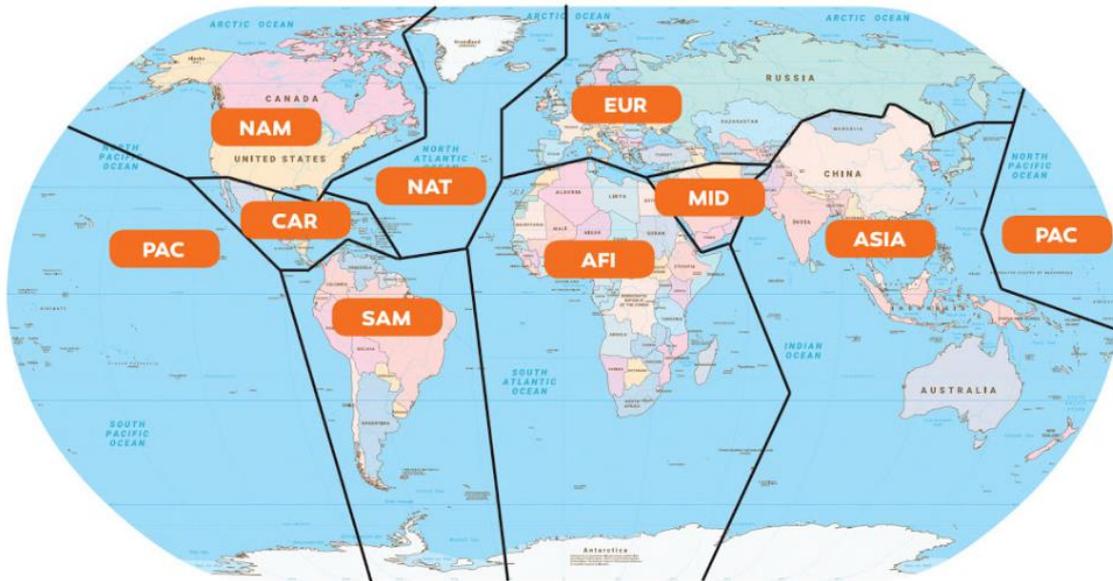


Figura 4. Regiones del espacio aéreo (Fernando, 2019)

De forma general, estas regiones suelen coincidir con los límites de cada país o nación. Un país puede tener varias regiones de información, como es el caso de España que cuenta con tres: FIR/UIR Madrid, FIR/UIR Barcelona y FIR/UIR Canarias. Se puede observar en la siguiente imagen:

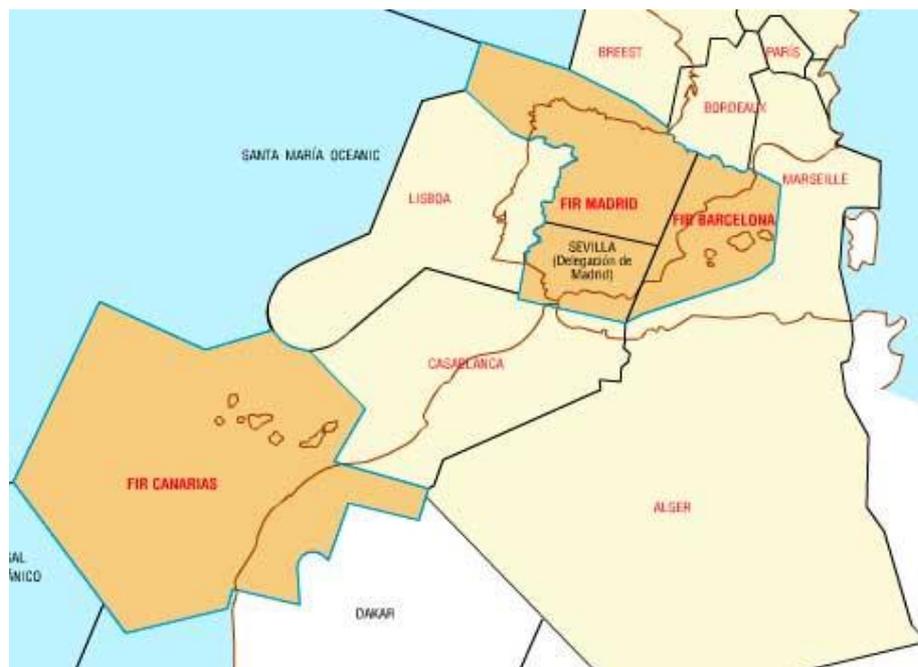


Figura 5. FIR en España (AENA)

### 2.2.3 Clases de espacio aéreo

El espacio aéreo en España se clasifica de acuerdo con lo establecido en el Anexo 11 de OACI, “Servicios de Tránsito Aéreo”. El espacio aéreo con servicio de tránsito aéreo se clasifica en espacio aéreo controlado y espacio aéreo no controlado. Estos, a su vez, se subdividen en siete clases (A, B, C, D, E, F y G) en base a las reglas que rigen cada una de ellas y los servicios facilitados. A continuación, se describen con mayor nivel de detalle:

- “Espacio aéreo controlado: comprende las áreas de control, las aerovías y las zonas de control y, en función del tipo de vuelo y los servicios de tránsito aéreo facilitados, se clasifica en clase A, B, C, D y E.
- Espacio aéreo no controlado: agrupa el resto del espacio aéreo ATS y, en función del tipo de vuelo y los servicios de tránsito aéreo facilitados, se clasifica en clase F y G”, (Fernando, s. f.).

Existe, además, la denominada capa de libre circulación, ajena a cualquiera zona de control o de tránsito de aeródromo, que se extiende desde el suelo o el agua hasta 300 metros (1000 pies) de altura sobre el terreno o 900 metros (3000 pies) de altitud sobre el nivel del mar, lo que resulte más alto. En esta zona se puede volar libremente, cumpliendo con la normativa vigente en cada caso.

Tabla 1: Clasificación del espacio aéreo. Fuente: (Fernando, s. f.)

Clase	Tipo de vuelo	Separación proporcionada	Servicios suministrados	Limitación de velocidad	Autorización ATC	Requiere comunicación por radio
A	IFR	Todas las aeronaves	Servicio de control tránsito aéreo	No	Sí	Continua en ambos sentidos
B	IFR					
	VFR	Todas las aeronaves				
C	IFR	IFR - IFR IFR - VFR	Información	250 kt IAS por debajo de 3050 m AMSL		
	VFR	VFR - IFR				
D	IFR	IFR - IFR	Control			
	VFR	Ninguna	Información de IFR y VFR			
E	IFR	IFR - IFR	Control	250 kt IAS por debajo de 3050 m AMSL	Sí	
	VFR	Ninguna	Información		No	No
F	IFR	IFR - IFR	Asesoramiento e información	250 kt IAS por debajo de 3050 m AMSL	No	Continua
	VFR	Ninguna	Información		No	No

G	IFR			250 kt IAS por debajo de 3050 m AMSL	No	Continua
	VFR				No	No

## 2.2.4 Servicios de Gestión del Tránsito Aéreo (ATS)

Los servicios de tránsito aéreo (ATS) comprenden el servicio de control de tránsito aéreo (ATC), el servicio de información de vuelo (FIS) y el servicio de alerta. Sus objetivos son el anticiparse y evitar colisiones entre aeronaves u obstáculos dentro del área de su responsabilidad, acelerar y coordinar el tráfico aéreo, asesorar e informar a las aeronaves y asistir en caso de requerir la aeronave de ayuda de búsqueda y salvamento.

### 2.2.4.1 Servicio de Control de Tránsito Aéreo (ATC)

Se encarga de controlar y coordinar el tráfico aéreo existente en espacios aéreos controlados. En función del tipo de control que realicen, aeronaves controladas o espacio afectado se puede establecer la siguiente clasificación:

1. Servicio de control de área (ACC): indican a los pilotos la ruta más eficiente para llegar a su destino. Se aseguran de que los aviones utilicen el espacio aéreo del país al que pertenecen, y que mantienen la distancia de seguridad entre sí.
2. Servicio de control de aproximación (APP): autorizan al piloto el acercamiento al aeropuerto, lo mantienen informado sobre el estado y localización de la pista, y sobre las condiciones climáticas. Si hay un retraso, el controlador de aproximación debe ordenar al piloto que entre en la zona de espera. Esto implica que el avión debe volar a diferentes alturas, moviéndose de forma gradual en dirección descendente, hasta que sea seguro acercarse más a la pista.
3. Servicio de control de aeródromo (TWR): trabajan en la parte superior de la torre de control, en la Sala de Control Visual desde la que pueden visualizar el aeródromo en todos sus ángulos. Indican a los pilotos las pistas que deben utilizar y se aseguran de que los sistemas de orientación luminosos funcionan correctamente.

### 2.2.4.2 Servicio de Información de vuelo (FIS)

Es un servicio de información pertinente para llevar a cabo el vuelo de forma eficiente y segura. Sus funciones principales están publicadas en la documentación de la OACI e incluyen el suministro de información con respecto a los siguientes aspectos:

- Condiciones meteorológicas
- Información sobre los cambios en la disponibilidad de los servicios de radionavegación
- Información sobre cambios en el estado de los aeródromos y de sus instalaciones asociadas
- Información sobre globos libres no tripulados
- Información relativa a la actividad volcánica y liberación en la atmósfera de materiales radiactivos o sustancias tóxicas
- Y cualquier otra información que pueda afectar a la seguridad operacional

### **2.2.4.3 Servicio de Alerta**

Es el responsable de alertar a los organismos pertinentes las aeronaves que necesitan ayuda, así como colaborar con ellos en caso de ser necesario. El servicio de alerta se preocupa de la evolución del vuelo de las aeronaves, establece las medidas oportunas para su auxilio y alerta y establece las comunicaciones pertinentes con el organismo competente para indicar la posición del avión.

# 3 DEFINICIÓN DEL PROBLEMA

El problema consiste en decidir los tiempos de aterrizaje para un conjunto de aviones de forma que se cumplan unas determinadas restricciones temporales. El modelo utilizado es válido tanto para problemas que estudian los aterrizajes y despegues conjuntamente en la misma pista, así como los aterrizajes únicamente.

Con el objeto de entender y profundizar en este problema, en los siguientes epígrafes se describe de forma detallada el proceso de aterrizaje.

## 3.1 Aterrizaje de aviones

En la gestión de un aeropuerto, las fases de despegue y aterrizaje centran el foco de atención de pasajeros y controladores ya que comprenden una serie de actuaciones críticas dirigidas a conseguir unas condiciones óptimas de velocidad requeridas, bien para despegar, o en otro caso, para aterrizar.

En estas dos fases es donde mayor número de accidentes se producen. Por ello, es fundamental el conocimiento de todas las fuerzas y variables que intervienen en el avión.

Como se ha citado en capítulos anteriores, en este trabajo se centran los esfuerzos en el aterrizaje. Dada la gran dificultad de esta operación, tiene que estar completamente supervisada y controlada. Los aviones se aproximan al aeropuerto guiados por los controladores del tráfico aéreo, generándose un flujo de aviones hacia la pista. Éstos controlan los movimientos de los aviones, manteniendo contacto visual, por radio y por radar durante la aproximación a la zona de aterrizaje.

### 3.1.1 Patrón de tráfico de los aeropuertos

Un avión puede, de forma general, elegir el itinerario más conveniente, siempre respetando las normativas vigentes, pero si el tráfico está congestionado lo único que puede hacer es reducir la velocidad. En estas situaciones, para solucionar el problema, es necesario el establecimiento de patrones y procedimientos, tanto de tráfico como de control, que debe conocer y aplicar el piloto siempre que sea necesario.

En la siguiente figura se puede observar el circuito básico de vuelo en un aeródromo, dependiendo su orientación de la pista de aterrizaje:

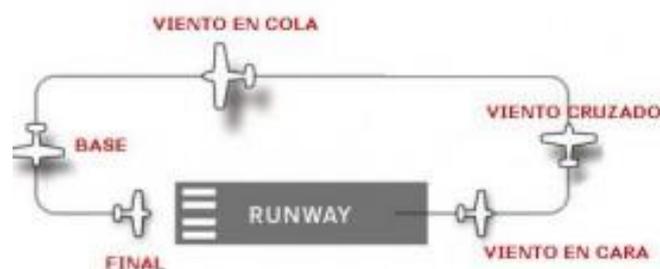


Figura 6. Patrón de vuelo en un aeropuerto

- A. Viento en cara: Este tramo corresponde a la fase de despegue, esto es, la trayectoria de vuelo paralela a la pista de aterrizaje.
- B. Viento cruzado: es la trayectoria de vuelo perpendicular a la pista de aterrizaje, a partir del final del tramo anterior (viraje a la izquierda de 90 grados). Recibe este nombre porque, generalmente, el viento incide en el costado del avión. De forma general, al comienzo de este tramo el aeroplano sigue ascendiendo tras el despegue.
- C. Viento en cola: Tramo paralelo a la pista, pero de sentido contrario al despegue o aterrizaje.
- D. Base: Perpendicular a la pista y de sentido opuesto al tramo de viento cruzado. En este tramo, se ajusta la velocidad para realizar el viraje al tramo final a una velocidad de planeo adecuada, siendo el piloto quien de forma aproximada y en virtud de su experiencia, calcula a la distancia que debe comenzar este viraje para que, el avión esté por encima de la prolongación del eje de la pista de aterrizaje a su terminación.
- E. Final: Tramo correspondiente a la trayectoria de aterrizaje, con la dirección y sentido de la pista.

De forma general, los vientos dominantes soplan aproximadamente desde la misma dirección. Previo a la construcción de una pista se estudian dichos vientos para orientarla de la mejor forma posible y así favorecer las operaciones de despegue y aterrizaje. Ahora bien, es inevitable en determinados momentos que no se cumplan las condiciones previstas y haya que realizar cambios en las operaciones.

En el circuito estándar los virajes se realizan hacia la izquierda y, siempre que esto sea posible, los aeródromos deben seguir este patrón. Sin embargo, esto no implica que todos los circuitos estén trazados de esta forma, por ejemplo, si a un lado de la pista hay una población cercana y en el otro un área deshabitada, se priorizarán las maniobras en esta última zona por seguridad.

### 3.1.1.1 Aerodinámica en viraje

Un viraje se realiza inclinando las alas en la dirección del giro deseado. Los virajes se clasifican en función de su grado de inclinación como se indica a continuación:

Tabla 2  
*Clasificación de los virajes*

Suave	Medio	Escarpado
0 -20°	20°- 45°	> 45°

En la siguiente imagen se puede observar cada una de las situaciones descritas en la clasificación anterior junto con la maniobra que debe realizar el piloto:

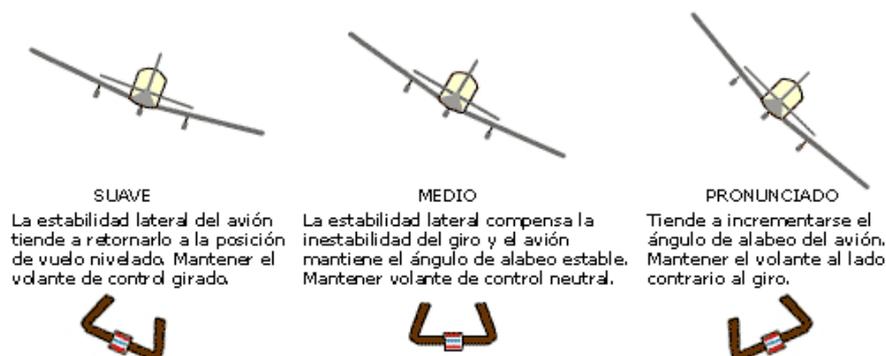


Figura 7. Viraje de un avión

Todo viraje conlleva la pérdida del equilibrio inicial de fuerzas por la aparición de otras que afectan tanto a la sustentación como al factor de carga del avión. En la Fig. 8 se representan las fuerzas a las que se ve sometido el avión en un viraje: peso  $W$ , sustentación  $L$  y fuerza de inercia  $F_i$  (fuerza centrífuga).

A iniciar el viraje, la fuerza de sustentación se descompone en dos vectores: el vector vertical o sustentación útil, y el vector horizontal. Esto implica que la componente vertical pierda efectividad por la merma que supone la aparición de la componente horizontal, restando capacidad de sustentación. Por otra parte, la fuerza de la gravedad se descompone en otros dos vectores: el peso y la fuerza centrífuga. El vector resultante se denomina factor de carga.

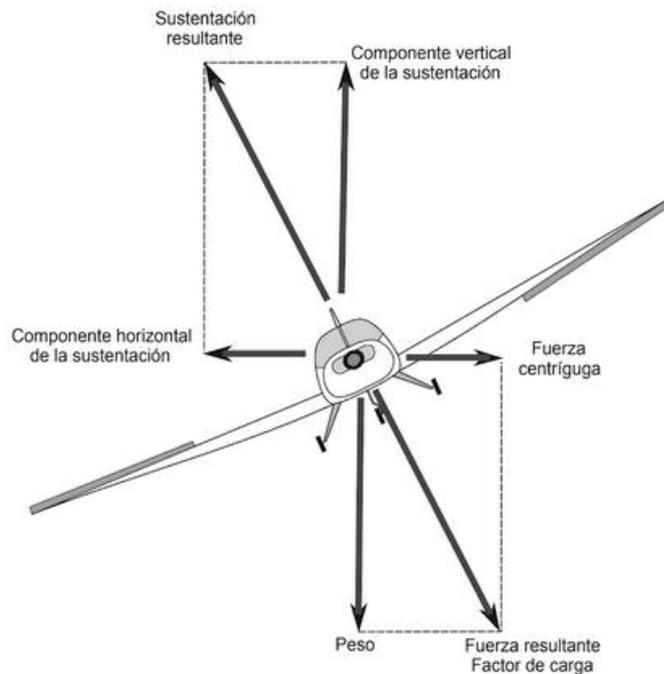


Figura 8. Fuerzas actuantes durante el proceso de viraje

La fuerza de inercia se origina en el movimiento circular durante el giro, con una aceleración centrípeta de valor:

$$a_N = \frac{V^2}{R}$$

Siendo  $V$  la velocidad y  $R$  el radio de la circunferencia.

La fuerza de inercia tiene por valor:  $F_i = m \cdot a_N$ , siendo  $m$  la masa del avión, de forma que:

$$F_i = \frac{W}{g} \cdot a_N; \quad F_i = \frac{W}{g} \cdot \frac{V^2}{R},$$

Por tanto, las proyecciones de las fuerzas sobre los ejes horizontal y vertical e igualándolas permiten obtener las siguientes fórmulas, siendo  $\varphi$  el ángulo de inclinación de las alas respecto a la horizontal:

$$L \cdot \cos \varphi = W$$

$$L \cdot \operatorname{sen} \varphi = F_i$$

Como el coseno de  $\varphi$  es siempre menor de 1, la fuerza de sustentación que necesita el avión en un viraje es mayor que su peso y proporcional al incremento de dicho ángulo.

La resistencia inducida será mayor que la que existiría en vuelo horizontal y rectilíneo para el mismo peso, es decir, la potencia necesaria será superior a la del vuelo horizontal. Luego si se mantiene la velocidad constante será necesario aumentar la potencia, y si no se varía la potencia disminuirá la velocidad. Debido a que el valor de la resistencia inducida es el que predomina a velocidades pequeñas, los virajes cerrados pueden suponer un

aumento considerable de la potencia necesaria para mantener la altitud. Es por ello por lo que se deben evitar los virajes escarpados inmediatamente después del despegue, aproximación, o cualquier situación crítica en la que la potencia no pueda ser aumentada en lo que un viraje exigiría, por ejemplo, por el fallo o mal funcionamiento del motor. Además, cabe mencionar que, en ese tipo de condiciones de vuelo, se está próximo a la velocidad de pérdida (Carmona, 2015).

La ejecución correcta de un viraje se realiza coordinando el movimiento de alabeo (palanca) con el de guiñada (pedales) para mantener equivalentes las fuerzas que producen el viraje, contrarrestando efectos como el originado por la ‘guiñada adversa’, fenómeno producido por el aumento de la resistencia inducida en el plano de mayor sustentación a consecuencia del alabeo, forzando al avión a girar en sentido contrario al del viraje.

La coordinación del viraje se puede observar a través del indicador de viraje y alabeo o ‘la bola’, que es el instrumento que informa sobre el equilibrio de las fuerzas centrífuga y centrípeta que afectan al aeroplano durante la maniobra. Si el viraje es coordinado, esta bola permanece en el centro.



Figura 9. Indicador de viraje

### 3.1.2 Maniobra de aterrizaje

El aterrizaje es la fase con la que concluye el vuelo, momento que requiere de una gran habilidad por parte del piloto para posar de forma suave el aeroplano en tierra. Se debe prestar especial atención a la velocidad a la que se realiza esta maniobra, puesto que una diferencia significativa entre la adoptada por el piloto y la recomendada por el fabricante puede conllevar consecuencias catastróficas, por inestabilidades o incremento de la carrera posterior de aterrizaje.

Se pueden distinguir las siguientes fases en la maniobra del aterrizaje:

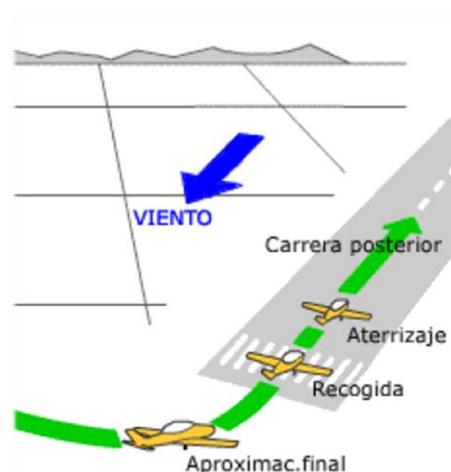


Figura 10. Fases del aterrizaje. («Introducción al Manual de Vuelo», s. f.)

#### 3.1.2.1 Aproximación final

Tras completar el giro de base, el avión desciende siguiendo en línea recta el eje de la pista hacia el punto teórico de contacto, normalmente en el primer tercio de la pista. Si se elige un punto muy cercano se tiende a realizar recogida con demasiada altura con el riesgo de provocar un aterrizaje brusco. Si por el contrario el punto de contacto es muy bajo, se sobrevuela demasiada pista pudiendo quedar poco espacio para la carrera de aterrizaje.

Las variables críticas y, por ello, necesario controlarlas en esta fase del vuelo son la velocidad, altura y distancia al punto de contacto. Puede ser necesario ligeros incrementos de potencia, si el avión está a menos altitud de la esperada, para mantener el ángulo de descenso adecuado a la velocidad de aproximación. Con carácter general, el equilibrio entre ellas debe ser tal que permita al avión alcanzar la pista planeando con motor a ralentí.

### **3.1.2.2 Recogida**

En esta fase se modifica de forma suave y paulatina la trayectoria de vuelo mantenida durante la aproximación hacia una trayectoria paralela a pista. La altura evoluciona desde los 3-5 metros iniciales hasta el contacto de las ruedas con la pista. Durante el tiempo que el avión permanece volando paralelo a la pista con motor a ralentí, este va perdiendo velocidad progresivamente disminuyendo la sustentación, y por tanto descendiendo hasta tomar tierra. Con el objeto de suavizar dicho contacto, durante este descenso final se debe aumentar gradualmente el ángulo de ataque, lo que produce un aumento de sustentación que retrasa al máximo el momento del contacto con tierra.

### **3.1.2.3 Contacto**

Es el momento del aterrizaje. En el caso de aviones con tren convencional se deben posar las tres ruedas a la vez, procurando que la mayor parte de la energía del contacto sea absorbida por el tren principal.

### **3.1.2.4 Carrera posterior**

Carrera en la que el avión disipa la fuerza de inercia acumulada. Esta fase finaliza cuando se requiere la acción del motor para continuar el desplazamiento.

## **3.2 Formulación matemática clásica**

Cuando un avión entra en el rango del radar del aeropuerto recibe instrucciones de la torre de control, que son un tiempo y una pista donde aterrizar. Este tiempo debe estar comprendido entre los instantes más temprano y más tardío de aterrizaje. El primero de ellos corresponde al instante en que podría aterrizar el avión si volase a su máxima velocidad. Por otro lado, el tiempo más tardío hace referencia al máximo retraso en que podría incurrir el avión disminuyendo su velocidad y/o tomando rutas alternativas. Perteneciente a este intervalo, existe un tiempo deseado y objetivo que corresponde al instante de aterrizaje si el avión volase a su velocidad de crucero. De esta forma, es evidente que aterrizar antes o después de este instante implica unos costes (directos e indirectos) a la compañía aérea. Además, entre cada par de aviones se debe respetar una distancia de seguridad. Estas distancias se traducen restricciones temporales asignando velocidades fijas de aterrizaje, que dependen del tipo de avión. Por tanto, tiene que transcurrir un tiempo mínimo entre el aterrizaje de un avión y los aviones sucesivos, sea o no en la misma pista de aterrizaje.

En esta sección se presenta una formulación lineal entera mixta para el problema de secuenciación de los aterrizajes (ALP) con múltiples pistas en un entorno estático, basado en la propuesta en Beasley *et al.* (2000).

### 3.2.1 Variables

Se define:

- $P$  Número de aviones
- $R$  Número de pistas de aterrizaje
- $E_i$  Tiempo más temprano de aterrizaje para el avión  $i$
- $L_i$  Tiempo más tardío de aterrizaje para el avión  $i$
- $[E_i, L_i]$  Ventana temporal del avión  $i$ , donde  $E_i \leq T_i \leq L_i$
- $T_i$  Tiempo objetivo de aterrizaje para el avión  $i$
- $S_{ij}$  Tiempo de separación requerido entre el aterrizaje del avión  $i$  y el aterrizaje del avión  $j$  en una misma pista de aterrizaje.
- $s_{ij}$  Tiempo de separación requerido entre los aterrizajes del avión  $i$  y el avión  $j$  en pistas diferentes.
- $g_i$  Coste de penalización ( $\geq 0$ ) por unidad de tiempo si el avión aterriza antes que  $T_i$
- $h_i$  Coste de penalización ( $\geq 0$ ) por unidad de tiempo si el avión aterriza después de  $T_i$

En el caso estático, el número de aviones  $P$  y la información relativa a cada uno de ellos es conocido desde el instante inicial.

Las variables de decisión propias del modelo son:

- $x_i$  Tiempo de aterrizaje del avión  $i$
- $\alpha_i$  Indica el grado de antelación del aterrizaje del avión  $i$  respecto  $T_i$
- $\beta_i$  Indica el retraso del aterrizaje del avión  $i$  respecto  $T_i$
- $\delta_{ij} = \begin{cases} 1 & \text{si el avión } i \text{ aterriza antes que el avión } j \\ 0 & \text{en otro caso} \end{cases}$
- $z_{ij} = \begin{cases} 1 & \text{si el avión } i \text{ y } j \text{ aterrizan en la misma pista } (i, j \in P; i \neq j) \\ 0 & \text{en otro caso} \end{cases}$
- $y_{jr} = \begin{cases} 1 & \text{si el avión } j \text{ aterriza en la pista } r \text{ } (j \in P; r \in R) \\ 0 & \text{en otro caso} \end{cases}$

### 3.2.2 Restricciones

Tabla 3

*Restricciones del modelo matemático*

Restricción	Descripción
$E_i \leq T_i \leq L_i \quad \forall i \in P; i \neq j$	El tiempo de aterrizaje de cada avión debe pertenecer al intervalo de su ventana temporal
$\delta_{ij} + \delta_{ji} \leq 1 \quad \forall i, j \in P$	Indica que para cada par de aviones $(i, j)$ , el avión $i$ aterriza antes que el avión $j$ o viceversa.
$\sum_{r=1}^R y_{ir} = 1 \quad \forall i \in P; r \in R$	Todo avión aterriza en una pista únicamente
$z_{ij} = z_{ji} \quad \forall i, j \in P; i \neq j$	Se verifica si los aviones $i$ y $j$ aterrizan en la misma pista

$z_{ij} \geq y_{ir} + y_{jr} - 1 \quad \forall i, j \in P; i \neq j$	Si los aviones $i$ y $j$ aterrizan en la misma pista, se fuerza a la variable $z_{ij}$ tomar el valor 1
$x_j \geq x_i + S_{ij} \cdot z_{ij} + (L_i + S_{ij} - E_j) \cdot \delta_{ji} \quad \forall i, j \in P; i \neq j$	Restricción de separación temporal
$\alpha_i \geq T_i - x_i \quad \forall i \in P$	Valores límites de $\alpha_i$
$0 \leq \alpha_i \leq T_i - E_i \quad \forall i \in P$	
$\beta_i \geq x_i - T_i \quad \forall i \in P$	Valores límites de $\beta_i$
$0 \leq \beta_i \leq L_i - T_i \quad \forall i \in P$	
$x_i = T_i - \alpha_i + \beta_i \quad \forall i \in P$	Relación entre el tiempo de aterrizaje y las variables $\alpha_i$ y $\beta_i$
$\delta_{ij}, z_{ij}, y_{jr} \quad \forall i, j \in P; i \neq j$	Variables binarias

---

### 3.2.3 Objetivo

Por último, para completar el modelo matemático queda por definir la función objetivo. En el control del tráfico aéreo la desviación frente al tiempo de aterrizaje objetivo es un factor clave. Sin embargo, es importante tener la capacidad de adaptar nuestro objetivo en función a ciertos condicionantes como son la densidad del tráfico aéreo o las condiciones atmosféricas. Debido a esto, se presentan dos funciones objetivo diferentes basadas en la desviación respecto el tiempo objetivo.

#### 3.2.3.1 Objetivo no lineal

Este objetivo utiliza una función no lineal basada en la diferencia entre el tiempo programado de aterrizaje y el tiempo deseado.

Se define:

$$d_i = x_i - T_i \quad i = 1, \dots, P$$

Si la desviación es positiva implica que el avión aterriza después del tiempo objetivo, situación no deseable y penalizada. Si por el contrario la desviación es negativa, el avión aterrizaría antes de  $T_i$ , escenario ideal. De esta forma se define la siguiente función objetivo:

$$\text{Maximizar} \sum_{i=1}^P D_i \quad D_i = \begin{cases} -d_i^2 & \text{si } d_i \geq 0 \\ +d_i^2 & \text{en otro caso} \end{cases}$$

#### 3.2.3.2 Objetivo lineal

Este objetivo penaliza la desviación respecto al tiempo objetivo utilizando sendos costes  $g_i$  y  $h_i$ . En la siguiente figura se puede observar la variación del coste en función del instante en que aterrice el avión  $i$ .

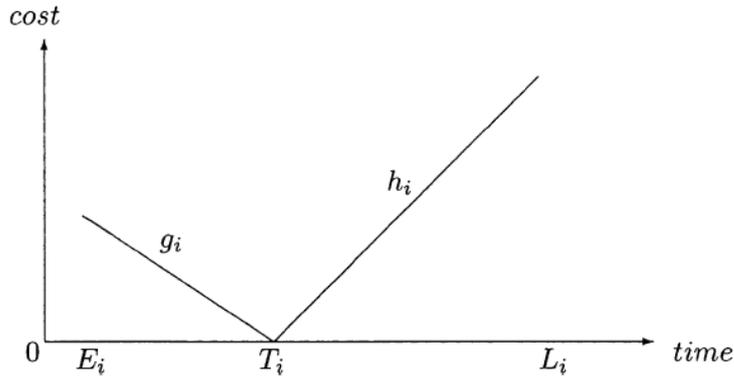


Figura 11. Costes de penalización

El coste que se presenta en este modelo no es el incurrido en el aterrizaje de un avión, sino el coste adicional cuando el aterrizaje no tiene lugar en  $T_i$ . Se define:

$$\alpha_i = \max(0, T_i - x_i) \quad i = 1, \dots, P$$

$$\beta_i = \max(0, x_i - T_i) \quad i = 1, \dots, P$$

El coste correspondiente al aterrizaje del avión  $i$  se puede expresar como  $\alpha_i g_i + \beta_i h_i$ . Si el avión aterriza en  $T_i$  incurre en un coste igual a cero, tal y como puede comprobarse en la imagen anterior. El objetivo es:

$$\text{Minimizar} \sum_{i=1}^P \alpha_i g_i + \beta_i h_i$$

### 3.3 Ejemplo conceptual

Se considera el ejemplo ilustrado en el archivo de texto “Airland1” disponible en “OR-Library” (<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>). En las dos siguientes tablas adjuntas, se observan los datos proporcionados en este ejemplo, que son los tiempos de aterrizaje objetivos ( $T_i$ ), más temprano ( $E_i$ ) y más tardío ( $L_i$ ) para cada avión y los tiempos mínimos de separación ( $S_{ij}$ ) entre cada par de aviones.

Tabla 4  
Intervalos de tiempo para el aterrizaje de los aviones

Avión	$E_i$	$T_i$	$L_i$
1	129	155	559
2	195	258	744
3	89	98	510
4	96	106	521
5	110	123	555
6	120	135	576
7	124	138	577
8	126	140	573
9	135	150	591
10	160	180	657

Tabla 5  
*Tiempos de separación entre cada par de aviones*

i	1	2	3	4	5	6	7	8	9	10
1	-	3	15	15	15	15	15	15	15	15
2	3	-	15	15	15	15	15	15	15	15
3	15	15	-	8	8	8	8	8	8	8
4	15	15	8	-	8	8	8	8	8	8
5	15	15	8	8	-	8	8	8	8	8
6	15	15	8	8	8	-	8	8	8	8
7	15	15	8	8	8	8	-	8	8	8
8	15	15	8	8	8	8	8	-	8	8
9	15	15	8	8	8	8	8	8	-	8
10	15	15	8	8	8	8	8	8	8	-

A la vista de los datos, una posible solución ( $\mu$ ) sería obtener la secuencia de aterrizaje ordenando crecientemente los tiempos de aterrizaje objetivos. En la tabla x, se puede observar los aviones que pueden aterrizar en su tiempo objetivo, representados por las celdas marcadas en verde. En cambio, las celdas resaltadas muestran que, aun aterrizando en el intervalo temporal permitido ( $E_i \leq x_i \leq L_i$ ), se retrasan respecto al tiempo objetivo la cantidad indicada en la fila inferior.

Tabla 6: *Ejemplo conceptual*

$\mu$	3	4	5	6	7	8	9	1	10	2
$T_i$	98	106	123	135	138	140	150	155	180	258
$x_i$	98	106	123	135	143	151	159	174	189	258
$\beta_i$					5	11	9	19	9	

# 4 ESTADO DEL ARTE

---

En este capítulo se centran los esfuerzos en revisar la literatura que aborda el problema de secuenciar y asignar tiempos de aterrizaje a un conjunto de aeronaves. El objetivo de este estudio es avanzar y ampliar los conocimientos sobre las metodologías, así como permitir realizar una comparativa sólida y exhaustiva de los diferentes artículos. La búsqueda se enfocó especificando las siguientes palabras clave o *keywords*: ‘*aircraft landing scheduling*’, ‘*air traffic control*’, ‘*sequence-dependent scheduling*’, ‘*runway operations*’, ‘*Aircraft landing*’, ‘*Airline industry*’ en diferentes bases de datos (*Web of Science*, *SciVerse Scopus*, *IEEE Xplore*, etc), permitiendo abarcar los artículos más destacados de revistas y conferencias internacionales. Resultado de este proceso de búsqueda, se seleccionaron 28 artículos que abarcan un periodo de tiempo de 42 años desde 1976 hasta 2018

Dada la complejidad del problema y su aplicabilidad en situaciones reales, se requieren métodos capaces de obtener buenas soluciones en un tiempo admisible. Por este motivo, los métodos exactos se han ido dejando de lado progresivamente, y las últimas investigaciones se centran en métodos heurísticos, metaheurísticas principalmente.

A continuación, se presenta una tabla donde se resume y clasifican los artículos seleccionados en orden cronológico, de forma que se puede observar cómo han ido evolucionando las investigaciones sobre este problema a lo largo del tiempo. Cada artículo se documenta en detalle en este trabajo de manera que sirva de guía informativa para futuros investigadores o interesados en esta área. Las tablas sucesivas se presentan con el objeto de complementar y profundizar el análisis realizado en la Tabla 1.

Tras su revisión se identificaron los siguientes 6 criterios para ilustrar íntegramente el contenido: ‘Tipo de metodología’, ‘Tipo de problema de optimización’, ‘Función objetivo’, ‘Entorno’, ‘Modelo de optimización’ y ‘Número de pistas de aterrizaje’. Se explican a continuación:

- ‘Tipo de metodología’ y ‘Modelo de optimización’ destaca el carácter exacto, aproximado o híbrido del método aplicado para resolver el problema.
- ‘Tipo de problema de optimización’ divide los artículos en aquellos que trabajan con modelos de optimización de objetivo único y multi-objetivos. ‘Función objetivo’ describe el objetivo que se persigue. Este puede ser minimizar el *makespan* o tiempo de aterrizaje del último avión, el coste total de la desviación.
- ‘Entorno’ y ‘Número de pistas de aterrizaje’ permiten clasificar estos artículos según estudien la secuenciación de los aterrizajes en un entorno estático o dinámico y si hay disponible una o varias pistas, respectivamente.
- Los modelos de optimización considerados son: *Job Shop Scheduling Model (JSS)*, *Mixed Integer Programming Model (MIP)*, *Dynamic Programming Model (DP)*, *Quadratic Integer Model (QM)*.

Paper	Tipo de metodología	Tipo de problema de optimización	Función objetivo	Entorno	Modelo de optimización	Número de pistas de aterrizaje
[4] Dear, R. (1976)		Múltiples objetivos	<i>Minimizing total delay/Minimizing the maximum delay/Maximizing throughput</i>	Dinámico	<i>DP CPS</i>	Única
[7] Abela et al. (1995)	Exacto	Objetivo único	<i>Minimize the total cost of penalties (LO)</i>	Estático	<i>MIP</i>	Única
[5] Ernst et al. (1999)	Exacto	Objetivo único	<i>Minimize the total cost of penalties (LO)</i>	Estático	<i>JSP</i>	Única/ Múltiples
[1] Beasley et al. (2000)	Exacto	Objetivo único	<i>Minimising total cost, linearly related to deviation from its target time</i>	Estático	<i>MIP</i>	Única/Múltiples
[20] Pinol et al. (2004)	Heurístico	Objetivo único	<i>Linear/Non-linear objective</i>	Estático	<i>MIP</i>	Múltiples
[2] Beasley et al. (2004)	Heurística	Objetivo único	<i>Minimising total cost, linearly related to deviation from its target time</i>	Dinámico	<i>MIP Displacement problem</i>	Múltiples
[12] Balakrishnan et al. (2006)	Aproximado	Objetivo único	<i>Minimize the makespan</i>	Estático	<i>DP</i>	Única
[9] Moser and Hendtlass (2007)	Aproximado	Objetivo único	<i>Minimise the penalties for early/late landings</i>	Dinámico	<i>MIP</i>	Única
[10] Yu et al. (2009)	Aproximado	Objetivo único	<i>Minimize the total cost of penalties</i>	Dinámico	<i>MIP</i>	Única
[13] Bencheikh et al. (2009)	Híbrido			Estático	<i>JSP</i>	Múltiples
[6] Farah et al. (2011)	Aproximado	Objetivo único	<i>Minimize the total delay cost</i>	Estático	<i>QIM</i>	Única
[22] Bencheikh et al. (2011)	Heurístico	Objetivo único	<i>Linear/Non-linear objective</i>	Estático	<i>NM</i>	Múltiples

[24] Dhouib (2011)	Híbrido	Objetivo único	<i>minimize the total (weighted) deviation from the target landing time for each plane</i>	Estático	MIP	Múltiples
[8] Awasthi et al. (2013)	Exacto	Objetivo único	<i>Optimize landing sequences and landing times for a set of airplanes</i>	Estático	MIP	Única/Múltiples
[14] Bencheikh et al. (2013)	Híbrido	Objetivo único	<i>Minimize the total cost of penalties (LO)</i>	Estático	MIP	Múltiples
[15] Salehipour et al. (2013)	Híbrido	Objetivo único	<i>Minimize the total cost of penalties (LO)</i>	Estático	MIP	Única/Múltiples
[16] Xie et al. (2013)	Híbrido	Objetivo único	<i>minimizes the total cost of landing deviation from target time</i>	Estático	MIP	Única/Múltiples
[11] Vadlamani et al. (2014)	Aproximado	Objetivo único	<i>minimize the average delay</i>	Estático	MIP	Única
[23] Feng Xiao-rong et al. (2014)	Heurístico	Objetivo único	<i>Minimize total delay time</i>	Dinámico	DP (multi-nest)	Única
[28] Kendall et al. (2014)	Híbrido	Objetivo único	<i>minimize the total penalty cost by generating, for a given set of aircrafts, the best landing sequence on given runways and assigning a landing time for each aircraft</i>	Estático	MIP	Múltiples
[21] Lieder et al. (2015)	Exacto	Objetivo único	<i>Minimize the total delay cost</i>	Estático	DP	Múltiples
[17] Bencheikh et al. (2016)	Híbrido	Objetivo único	<i>minimize the total penalty cost and the displacement function which corresponds to the total move of the current solution from the previous one</i>	Dinámico	MIP+GR	Múltiples

[18] Girish (2016)	Híbrido	Objetivo único	<i>Minimize the total cost of penalties (LO)</i>	Estático	<i>MIP</i>	Única/Múltiples
[19] Salehipour (2018)	Exacto	Objetivo único	<i>Minimize the total cost of landing deviation from the target landing times.</i>	Dinámico	<i>DP</i>	Única/Múltiples
García, J. L., (2019)	Metaheurístico	Objetivo único	<i>Minimize the total delay cost</i>	Estático	<i>MIP</i>	Única

Tabla 7

*Clasificación de las referencias bibliográficas*

Profundizando en el análisis anterior, se presentan tres tablas que evalúan estos artículos según los criterios considerados más reveladores. Se puede observar que la mayoría centran sus esfuerzos en estudiar el entorno estático con múltiples pistas de aterrizaje o, en primer lugar, analizar el modelo de una única pista para, posteriormente, ampliar y proponer una metodología que permita resolver el modelo de múltiples pistas de aterrizaje. Según la Tabla 7, los artículos más recientes se focalizan mayoritariamente en metaheurísticas híbridas, ratificando su mejor desempeño teórico respecto los métodos exactos y aproximados.

Tabla 8  
Clasificación de los artículos según el entorno del problema

Entorno	Artículos	Número (%)
Estático	[1] Beasley et al. (2000), [5] Ernst et al. (1999), [6] Farah et al. (2011), [7] Abela et al. (1995), [8] Awasthi et al. (2013), [11] Vadlamani et al. (2014), [12] Balakrishnan et al. (2006), [13] Bencheikh et al. (2009), [14] Bencheikh et al. (2013), [15] Salehipour et al. (2013), [16] Xie et al. (2013), [18] Girish (2016), [20] Pinol et al. (2004), [21] Lieder et al. (2015), [22] Bencheikh et al. (2011), [24] Dhouib (2011), [28] Kendall et al. (2014).	17 (70.83)
Dinámico	[2] Beasley et al. (2004), [4] Dear, R. (1976), [9] Moser and Hendtlass (2007), [10] Yu et al. (2009), [17] Bencheikh et al. (2016), [19] Salehipour (2018), [23] Feng Xiao-rong et al. (2014)	7 (29.17)

Tabla 9  
Clasificación de los artículos según el tipo de metodología

Tipo de metodología	Artículos	Número (%)
Exacta	[1] Beasley et al. (2000), [5] Ernst et al. (1999), [7] Abela et al. (1995), [8] Awasthi et al. (2013), [19] Salehipour (2018), [21] Lieder et al. (2015).	6 (26.09)
Aproximada	[2] Beasley et al. (2004), [6] Farah et al. (2011), [9] Moser and Hendtlass (2007), [10] Yu et al. (2009), [11] Vadlamani et al. (2014), [12] Balakrishnan et al. (2006), [20] Pinol et al. (2004), [22] Bencheikh et al. (2011), [23] Feng Xiao-rong et al. (2014).	8 (34.78)
Híbrida	[13] Bencheikh et al. (2009), [14] Bencheikh et al. (2013), [15] Salehipour et al. (2013), [16] Xie et al. (2013), [17] Bencheikh et al. (2016), [18] Girish (2016), [24] Dhouib (2011), [28] Kendall et al. (2014).	9 (39.13)

Tabla 10

Clasificación de los artículos según el número de pistas de aterrizaje

Número de pistas de aterrizaje	Artículos	Número (%)
Única	[4] Dear, R. (1976), [6] Farah et al. (2011), [7] Abela et al. (1995), [9] Moser and Hendtlass (2007), [10] Yu et al. (2009), [11] Vadlamani et al. (2014), [12] Balakrishnan et al. (2006), [23] Feng Xiao-rong et al. (2014).	8 (33.3)
Múltiples	[2] Beasley et al. (2004), [13] Bencheikh et al. (2009), [14] Bencheikh et al. (2013), [17] Bencheikh et al. (2016), [20] Pinol et al. (2004), [21] Lieder et al. (2015), [22] Bencheikh et al. (2011), [24] Dhouib (2011), [28] Kendall et al. (2014).	9 (37.5)
Única/Múltiples	[1] Beasley et al. (2000), [5] Ernst et al. (1999), [8] Awasthi et al. (2013), [15] Salehipour et al. (2013), [16] Xie et al. (2013), [18] Girish (2016), [19] Salehipour (2018).	7 (29.2)

# 5 MÉTODOS DE RESOLUCIÓN

En la gran mayoría de los problemas reales no existen métodos o algoritmos exactos que localicen el óptimo. Esta imposibilidad nos lleva a utilizar técnicas aproximadas o heurísticas que, aunque no se alcance la solución óptima, al menos se pueda obtener una buena solución en un tiempo razonable.

Existen muchos algoritmos aproximados que proporcionan buenos resultados para estos problemas de optimización combinatoria, como el expuesto en el capítulo anterior. En los siguientes epígrafes, se describen estos procedimientos centrandos los esfuerzos en el recocido simulado y la búsqueda en vecindarios variables, que son los seleccionados para resolver el problema planteado en este proyecto. Cabe mencionar que, para implementar estos métodos, se codifica en Python utilizando el software JetBrains PyCharm Edu 2017.3.

## 5.1 Revisión de las metodologías de resolución

Existen varias clasificaciones para los algoritmos. Una primera clasificación sería: algoritmos deterministas y no deterministas en función de si existe o no aleatoriedad que pudiera aparecer en el proceso de búsqueda de una solución.

En función de la “precisión” del algoritmo, se presenta la siguiente clasificación:

- Algoritmos exactos: siempre generan una solución óptima
- Algoritmos aproximados: las soluciones obtenidas cumplen con el margen o porcentaje establecido del óptimo
- Algoritmos heurísticos: no hay garantías de optimalidad, pero al mismo tiempo, en general, los tiempos de ejecución son menores.

En la siguiente figura se puede observar con mayor claridad lo explicado anteriormente:

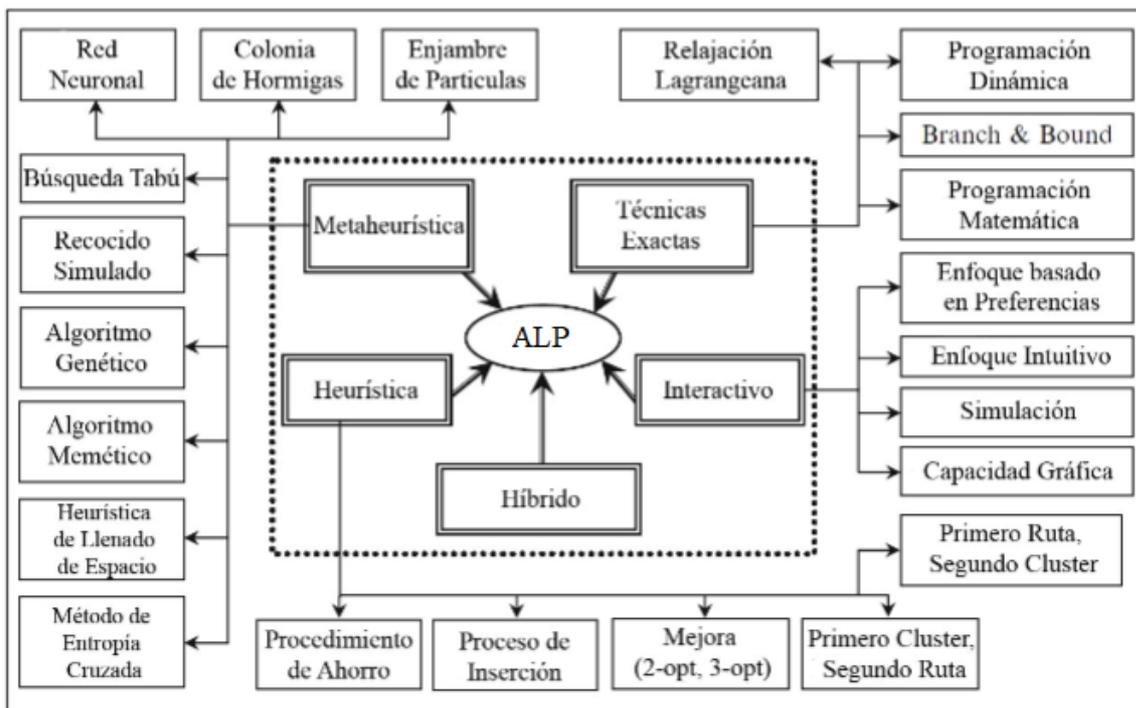


Figura 12. Metodologías de resolución. Fuente: Anbuudayasankar, S. P, Ganesh, K. y Mohapatra, S.

En los siguientes epígrafes, se detalla con mayor nivel de detalle cada uno de estos algoritmos.

### 5.1.1 Algoritmos exactos

Estos algoritmos permiten obtener soluciones admisibles (enteras) formulando modelos de programación lineal.

La lógica de estos métodos sigue el esquema en árbol, interpretando así el problema inicial como la raíz, y enumerando implícitamente las soluciones a través de la división del espacio y de la búsqueda de subconjuntos de soluciones. Tras acotar el problema, el algoritmo explora los subconjuntos de soluciones que son representadas como las ramas del árbol, para así descartar aquellas ramas por las que no se obtiene mejor solución que la encontrada hasta el momento. Las ramas que no han sido descartadas vuelven a dividirse y se repite el procedimiento de nuevo, para llegar así a la iteración en la que no se obtengan mejores soluciones.

Uno de los inconvenientes de estos algoritmos es que pueden ser muy lentos. En este caso se pueden utilizar algoritmos aproximados o, si siguen siendo demasiado lentos, los algoritmos heurísticos.

### 5.1.2 Heurísticas

Las heurísticas, aunque no aseguran encontrar la solución óptima, sí permiten conseguir buenas soluciones en menores tiempos de ejecución. Esto es especialmente útil en el caso de no existir métodos exactos, o los tiempos de ejecución para obtener la solución óptima son inadmisibles. Además, pueden utilizarse para generar una solución inicial en la aplicación conjunta con otros algoritmos.

El gran problema que presentan estos algoritmos es la dificultad de huir de los óptimos locales. En la Figura 12 se muestra este fenómeno para una vecindad<sup>2</sup> de un algoritmo con una rutina de búsqueda local.

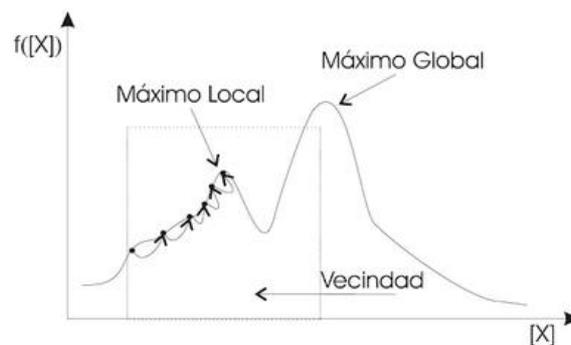


Figura 13. Función multimodal. Comportamiento de heurísticas en óptimos locales

Para solucionar el problema que presentan los algoritmos heurísticos para escapar de óptimos locales se utilizan otras metodologías que solventan este inconveniente en la medida de lo posible, que son las metaheurísticas y que poseen mecanismos que orientan a las heurísticas para evitar el problema mencionado.

### 5.1.3 Metaheurísticas

En situaciones que no existan procedimientos fiables para resolver un problema, ya sea por su complejidad o falta de conocimientos e investigaciones, se pueden utilizar metaheurísticas, que es una estrategia (heurística) general que imitan fenómenos observables en la naturaleza y que parecen tener una lógica artificial.

Las técnicas metaheurísticas no garantizan encontrar el valor óptimo, aunque permiten obtener buenas

<sup>2</sup> Conjunto de soluciones o entorno de la solución que realiza el recorrido

soluciones en tiempos de ejecución razonables. Partiendo de soluciones no óptimas, se pueden obtener otras que cumplan algún tipo de criterio en la búsqueda de otra mejor. Se repite el proceso a partir de esta nueva solución, y cuando se verifique la condición de parada, finaliza el proceso.

Entre otros, se pueden destacar los siguientes:

- Algoritmos genéticos o evolutivos bordan con éxito la convergencia prematura teniendo en cuenta, al mismo tiempo, una serie de soluciones.
- Optimización basados en el comportamiento de sistemas descentralizados (auto - organizados). Dos de los enfoques de mayor éxito son la *optimización de las colonias de hormigas* (ACO) y *optimización de enjambre de partículas* (PSO). En ACO, hormigas artificiales constituyen soluciones desplazándose en el gráfico del problema y modificándolo de tal manera que las futuras hormigas puedan construir mejores soluciones. PSO se utiliza para resolver problemas en los que se pueda representar una solución mejor como un punto o una superficie en un espacio n-dimensional. La principal ventaja de estas técnicas es que son muy eficaces frente al problema local de los óptimos.
- Recocido Simulado, presentado en 1983, utiliza un enfoque similar al de *Hill climbing*, pero en ocasiones acepta soluciones de peor calidad que la actual.
- Búsqueda Tabú extiende la idea de evitar el óptimo local mediante el uso de estructuras de memoria. Este algoritmo prohíbe la repetición de movimientos que se han hecho recientemente.

## 5.2 Recocido simulado

El Recocido Simulado, *simulated annealing en inglés* (en adelante, SA), fue introducido en el campo de la optimización computacional por Kirkpatrick *et al.* (1983). Se trata de una de las metaheurísticas clásicas, variante de la Búsqueda Local que permite movimientos ascendentes para escapar de óptimos locales. Su nombre proviene del proceso de recocido<sup>3</sup> del acero y las cerámicas para obtener material cristalizado.



Figura 14. *Optimization by Simulated Annealing*. Fuente: Kirkpatrick *et al.* (1983)

Se basa en la similitud que puede haber entre el proceso termodinámico mencionado anteriormente con un proceso de optimización combinatoria.

<sup>3</sup> Enfriamiento de un material a un "estado fundamental", un estado de mínima energía. Tomar un material muy caliente y enfriarlo lentamente para que se solidifique despacio en el estado de mínima energía, evitando que se endurezca en alguna condición subóptima.

Dado un estado  $i$  con energía  $E_i$ ,

Generar un nuevo estado  $j$  mediante un mecanismo de perturbación (pequeña distorsión del estado  $i$ ).

Calcular la energía del nuevo estado  $E_j$ .

Si  $(E_j - E_i) \leq 0$

Se acepta el estado  $j$  como nuevo estado

Sino,

Aceptar el estado con probabilidad  $\exp\left(\frac{E_i - E_j}{k_B \cdot T}\right)$

El algoritmo de recocido simulado se puede ver como una iteración de algoritmos de Metrópolis. La idea principal de esta técnica se puede observar en la tabla anterior (Metropolis et al. 1953) donde  $T$  denota la temperatura y  $k_B$  la constante de Boltzmann.

El funcionamiento de este algoritmo es generar un vecino, calcular su energía (i.e., evaluar su coste mediante la función objetivo o de evaluación en problemas de optimización) y se acepta dicho vecino si disminuye la energía o aceptarlo con mayor energía con cierta probabilidad. Se comprueba que repitiendo este proceso durante muchas iteraciones se puede llegar a lo que se conoce como un equilibrio térmico.

Para evitar caer en óptimos locales, el algoritmo permite aceptar soluciones peores según una determinada probabilidad, que va disminuyendo mientras aumenta el número de iteraciones o  $T$ . Este parámetro empieza con valores altos y tiende asintóticamente a cero.

Como ocurre en el proceso de recocido, el éxito de esta heurística radica en la velocidad con la que se realice el enfriamiento del sistema. Hay que buscar un equilibrio en dicha velocidad de enfriamiento, los cambios bruscos le restan eficacia como se puede observar en infinidad de procesos naturales.

Kirkpatrick *et al.* (1983) utilizan el siguiente modelo exponencial para realizar el proceso de enfriamiento:

$$T_{k+1} = T_0 \cdot \alpha^k$$

Con  $\alpha = 0.90$  y  $T_0$  la temperatura inicial. Es decir,

$$T_{k+1} = \alpha \cdot T_k$$

Este modelo suele ser el más común para realizar el enfriamiento al aplicar el algoritmo de SA en una gran variedad de problemas de optimización por su facilidad y eficacia. Según Kim *et al.* (2002) hay dos modelos ampliamente aceptados para llevar a cabo el enfriamiento de la temperatura en el algoritmo SA.; el primero de ellos el modelo logarítmico:

$$T_k = \frac{T_0 \cdot \log(k_0)}{\log(k + k_0)} \quad k_0 > 1$$

El segundo, el modelo geométrico  $T_k = \alpha \cdot T_{k-1}$  con  $0 < \alpha < 1$ . El primero de ellos garantiza la convergencia, pero es muy lento en el proceso; por esta razón, el modelo geométrico es el más utilizado en la práctica.

Además, otros modelos de enfriamiento de uso menos frecuentes que el anterior, se enuncian a continuación:

- (Hayek, 1988) y (Geman & Geman, 1984) implementaron el siguiente modelo logarítmico:

$$T_{k+1} = \frac{c}{\log(k + 1)}$$

- (Aarts & Van Laarhoven, 1985)

$$T_{k+1} = T_k \cdot \left\{ 1 + \frac{T_k \cdot \ln(1 + \delta)}{3 \cdot \sigma_k} \right\}^{-1}$$

Siendo  $\delta$  una constante cercana a cero y  $\sigma_k$  la desviación estándar de los valores que toma la función objetivo en la  $k$ -ésima cadena de Markov.

- (Lundy & Mees, 1986)

$$T_{k+1} = \frac{T_k}{1 + \beta \cdot T_k}$$

- (Osman & Christofides, 1994), basados en el modelo anterior, propusieron:

$$T_{k+1} = \frac{T_k}{1 + \beta_k \cdot T_k}, \text{ tal que } \beta_k = \frac{T_0 - T_f}{(\alpha + \gamma \cdot \sqrt{k}) \cdot T_0 \cdot T_f}$$

Donde  $\alpha$  y  $\gamma$  son constantes.

- (Atiqullah, 2004)

$$T_k = T_0 \cdot a^{-\left[\frac{k}{f \cdot k_{m\acute{a}x}}\right]^b},$$

Donde  $a$  y  $f$  son parámetros de control y  $k_{m\acute{a}x}$  es el máximo número de cadenas que serán ejecutadas.

Otros parámetros característicos son el tiempo de equilibrio térmico o la longitud de la cadena  $L(T)$ , que determina el número de iteraciones usando la misma temperatura, y la función de enfriamiento que establece cómo se reduce el valor de la temperatura.

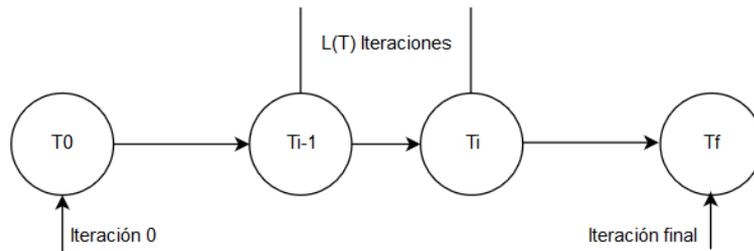


Figura 15. Parámetros del recocido simulado

El diagrama de flujo del algoritmo se presenta en la Fig. 16.

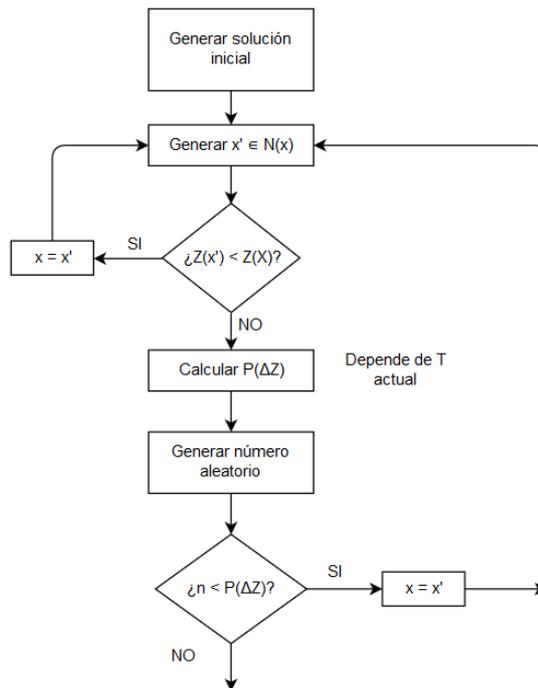


Figura 16. Diagrama de flujo del recocido simulado

En el algoritmo hay un conjunto de parámetros o variables de control que dependen del problema al que se aplique la metaheurística. Estas decisiones requeridas se clasifican en dos conjuntos:

- Las relativas a los parámetros del proceso, que son la temperatura inicial  $T_0$ , tiempo de equilibrio térmico  $L$ , función de enfriamiento  $\alpha(t)$  y condición de parada.
- Y las relativas al problema concreto: estructura de la solución, solución inicial  $s_0$ , el coste asociado y la estructura del vecindario  $N(s_0)$ .

Seleccionar de forma correcta los parámetros del algoritmo es la parte del diseño más ardua y se debe realizar mediante una exhaustiva experimentación. La elección del vecindario es fundamental para un buen funcionamiento del algoritmo, en el que se limite el número de iteraciones para alcanzar una solución partiendo de otra cualquiera. De esta forma, el proceso converge al margen de cual sea la solución inicial  $s_0$ .

Se demuestra la optimalidad del SA en la literatura. No obstante, la rapidez de convergencia queda condicionada por la selección de los parámetros. Es evidente que, cuantas más soluciones sean evaluadas, mayor será la probabilidad de encontrar la solución óptima, implicando un mayor coste computacional.

### 5.3 Búsqueda en Vecindario Variables (VNS)

La búsqueda en vecindario variable (en inglés *Variable Neighborhood Search*) es una estrategia basada en modificar dinámicamente las estructuras de vecindario. Se trata de un procedimiento muy genérico, ya que permite un alto grado de libertad en su implementación. Fue ideada por Hansen y Mladenovic (1999), basada en ir cambiando sistemáticamente la exploración del vecindario, para huir de mínimos locales.

Se definen una sucesión de vecindarios  $N_1(s), N_2(s), \dots, N_k(s)$ , escogidos aleatoriamente. En cada iteración, se parte de una solución  $s'$  del vecindario actual  $N_k(s)$  y se mejora mediante un procedimiento de búsqueda local que proporciona un nuevo mínimo local  $s''$ . Si es menor que el anterior, es decir,  $f(s'') \leq f(s)$ ,  $s''$  se convierte en la solución actual y la rutina de búsqueda sigue desde  $N_1(s'')$ ; en caso contrario, dicha rutina continua desde  $N_{k+1}(s)$  en la siguiente iteración.

```

BEGIN
/* Nk: k=1, ..., kmax, estructuras de vecindarios */
/* Solución inicial, S.
kmáx estructuras de vecindarios
Repeat
k ← 1
While (k ≤ kmáx)
    S' = shaking(S,N)
    S'* = LocalSearch(S')
    If f(S' *) ≤ f(S)
        S ← S' *
        N = 1
    Else
        k += 1
Until
Condición de parada

```

## 5.4 Adaptación al problema

### 5.4.1 Recocido Simulado

En la Tabla 11, se describe la analogía existente entre el proceso de recocido utilizado en termodinámica y la heurística que se basa en dicho proceso que permite resolver problemas de optimización combinatoria.

Tabla 11: *Analogía del recocido simulado*

TERMODINÁMICA	OPTIMIZACIÓN
Estado del sistema físico	Solución factible
Configuración de mínima energía interna	Solución óptima
Energía del estado	Coste de la solución
Temperatura	Parámetro de control

#### 5.4.1.1 Representación de la solución

La representación elegida para las soluciones del problema de una pista de aterrizaje es:

$$s = [y_1, y_2, \dots, y_p]$$

Siendo  $y_i$  la proporción ( $0 \leq y_i \leq 1$ ) del intervalo  $[E_i, L_i]$  que transcurre previo al aterrizaje del avión  $i$ . Matemáticamente se puede expresar de la siguiente forma:

$$y_i = \frac{x_i - E_i}{L_i - E_i}$$

El tiempo de aterrizaje asignado al avión  $i$  se calcula mediante la siguiente expresión:

$$x_i = E_i + y_i(L_i - E_i)$$

Por ejemplo, si  $[E_i, L_i]$  es igual a  $[150, 300]$  (expresados en segundos) e  $y_i = 0.45$ , entonces:

$$x_i = 150 + 0.45 \cdot (300 - 150) = 217.5$$

$$y_i = \frac{217.5 - 150}{300 - 150} = 0.45$$

Esta forma de representar las soluciones presenta la propiedad de satisfacer automáticamente la restricción asociada a las ventanas temporales. Por ejemplo, suponiendo que  $P = 3$  y las ventanas temporales son  $[100, 300]$ ,  $[200, 1500]$  y  $[50, 200]$  para los aviones 1, 2 y 3, respectivamente. Si los valores  $y_i$  son  $[0.45, 0.21, 0.71]$ , entonces los tiempos de aterrizajes para cada avión serían:

$$x_1 = 100 + 0.45 \cdot (300 - 100) = 190$$

$$100 < 190 < 300 \quad \checkmark$$

$$x_2 = 200 + 0.21 \cdot (1500 - 200) = 473$$

$$200 < 473 < 1500 \quad \checkmark$$

$$x_3 = 50 + 0.71 \cdot (200 - 50) = 156.5$$

$$50 < 156.5 < 200 \quad \checkmark$$

La función objetivo corresponde al objetivo lineal explicado en el apartado 3.2.3.2.

En el caso que se aborde la versión del problema con múltiples pistas de aterrizaje, las soluciones se representan de la siguiente forma:

$$s = [1, (z, y_z), (v, y_v), \dots, (v, y_v), 2, (i, y_i), \dots, (j, y_j), r, \dots, (l, y_l)]$$

Siendo los elementos de tipo entero las pistas de aterrizaje, y los vectores, los aviones. De forma que, los aviones ubicados detrás del entero  $i$ , indica que aterrizan en dicha pista de aterrizaje.

#### 5.4.1.2 Temperatura

La temperatura es un parámetro de control particular del algoritmo que permite aceptar con una determinada probabilidad peores soluciones para evitar caer en mínimos locales. Se elige una temperatura inicial y la posibilidad de elegir una solución mala decrece con la disminución de la temperatura.

#### 5.4.1.3 Longitud de la cadena

En este proyecto se considera la longitud de la cadena  $L(T)$  que es una constante en la ejecución del algoritmo. Dicho parámetro indica el número de soluciones que son evaluadas en cada iteración realizando búsquedas más exhaustivas para cada temperatura.

#### 5.4.1.4 Condición de parada

Se establece la condición de parada propia de la heurística SA, esto es, la temperatura final  $t_f$ , que marca el nivel final de energía que se puede alcanzar.

### 5.4.2 Búsqueda en entornos variables

La representación de las soluciones es la misma que la utilizada en la implementación del SA, esto es,

$$s = [y_1, y_2, \dots, y_p]$$

Siendo  $y_i$  la proporción ( $0 \leq y_i \leq 1$ ) del intervalo  $[E_i, L_i]$  que transcurre previo al aterrizaje del avión  $i$ . Matemáticamente se puede expresar de la siguiente forma:

$$y_i = \frac{x_i - E_i}{L_i - E_i}$$

#### 5.4.2.1 Vecindarios

Para generar los vecindarios  $N(x)$  en cada iteración se han empleado técnicas como el intercambio de posiciones, inserción y asignación de nuevos valores a los tiempos de aterrizaje. Además, cabe mencionar que está permitido trabajar con soluciones no factibles.

En la tabla 12, se ilustra con mayor nivel de detalle los vecindarios implementados en el VNS tomando de ejemplo las soluciones,

Una pista: 

0.25	0.41	0.50	0.64	0.17
------	------	------	------	------

Varias pistas: 

[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
-----------------------------------	------------------------	------------------------

Tabla 12: *Vecindarios para las versiones con una y varias pistas de aterrizaje*

<b>UNA PISTA DE ATERRIZAJE</b>																															
N	Descripción																														
1	<p><u>Incremento de los tiempos de aterrizaje del avión <math>i</math> (<math>i = 1, \dots, P</math>)</u></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> <b>for</b> i=1, ..., P   s = current_solution[:]   s[i] *= (1+random(0,1)) <b>end_for</b>                     </pre> </div> <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;"><b><math>0.25 \cdot (1 + \text{random}(0, 1))</math></b></td> <td style="text-align: center;">0.41</td> <td style="text-align: center;">0.50</td> <td style="text-align: center;">0.64</td> <td style="text-align: center;">0.17</td> </tr> <tr> <td style="text-align: center;">0.25</td> <td colspan="2" style="text-align: center;"><b><math>0.41 \cdot (1 + \text{random}(0, 1))</math></b></td> <td style="text-align: center;">0.50</td> <td style="text-align: center;">0.64</td> <td style="text-align: center;">0.17</td> </tr> <tr> <td style="text-align: center;">0.25</td> <td style="text-align: center;">0.41</td> <td colspan="2" style="text-align: center;"><b><math>0.50 \cdot (1 + \text{random}(0, 1))</math></b></td> <td style="text-align: center;">0.64</td> <td style="text-align: center;">0.17</td> </tr> <tr> <td style="text-align: center;">0.25</td> <td style="text-align: center;">0.41</td> <td style="text-align: center;">0.50</td> <td colspan="2" style="text-align: center;"><b><math>0.64 \cdot (1 + \text{random}(0, 1))</math></b></td> <td style="text-align: center;">0.17</td> </tr> <tr> <td style="text-align: center;">0.25</td> <td style="text-align: center;">0.41</td> <td style="text-align: center;">0.50</td> <td style="text-align: center;">0.64</td> <td colspan="2" style="text-align: center;"><b><math>0.17 \cdot (1 + \text{random}(0, 1))</math></b></td> </tr> </table> <p>Nota: Verificar que <math>x_i \leq LT_i, i = 1, \dots, P</math></p>	<b><math>0.25 \cdot (1 + \text{random}(0, 1))</math></b>		0.41	0.50	0.64	0.17	0.25	<b><math>0.41 \cdot (1 + \text{random}(0, 1))</math></b>		0.50	0.64	0.17	0.25	0.41	<b><math>0.50 \cdot (1 + \text{random}(0, 1))</math></b>		0.64	0.17	0.25	0.41	0.50	<b><math>0.64 \cdot (1 + \text{random}(0, 1))</math></b>		0.17	0.25	0.41	0.50	0.64	<b><math>0.17 \cdot (1 + \text{random}(0, 1))</math></b>	
<b><math>0.25 \cdot (1 + \text{random}(0, 1))</math></b>		0.41	0.50	0.64	0.17																										
0.25	<b><math>0.41 \cdot (1 + \text{random}(0, 1))</math></b>		0.50	0.64	0.17																										
0.25	0.41	<b><math>0.50 \cdot (1 + \text{random}(0, 1))</math></b>		0.64	0.17																										
0.25	0.41	0.50	<b><math>0.64 \cdot (1 + \text{random}(0, 1))</math></b>		0.17																										
0.25	0.41	0.50	0.64	<b><math>0.17 \cdot (1 + \text{random}(0, 1))</math></b>																											

Incremento de los tiempos de aterrizaje del avión  $i$  ( $i = 1, \dots, P$ ), manteniendo los cambios realizados en los aviones previos

```
s = current_solution[:]
for i=1,...,P
    s[i] *= (1+random(0,1))
end_for
```

Si  $n = (1 + \text{random}(0,1))$ , entonces:

2

<b><math>0.25 \cdot n</math></b>	0.41	0.50	0.64	0.17
<b><math>0.25 \cdot n</math></b>	<b><math>0.41 \cdot n</math></b>	0.50	0.64	0.17
<b><math>0.25 \cdot n</math></b>	<b><math>0.41 \cdot n</math></b>	<b><math>0.50 \cdot n</math></b>	0.64	0.17
<b><math>0.25 \cdot n</math></b>	<b><math>0.41 \cdot n</math></b>	<b><math>0.50 \cdot n</math></b>	<b><math>0.64 \cdot n</math></b>	0.17
<b><math>0.25 \cdot n</math></b>	<b><math>0.41 \cdot n</math></b>	<b><math>0.50 \cdot n</math></b>	<b><math>0.64 \cdot n</math></b>	<b><math>0.17 \cdot n</math></b>

Nota: Verificar que  $x_i \leq LT_i$ ,  $i = 1, \dots, P$

Decrementos de los tiempos de aterrizaje del avión  $i$  ( $i = 1, \dots, P$ )

```
for i=1,...,P
    s = current_solution[:]
    s[i] *= (1-random(0,1))
end_for
```

3

<b><math>0.25 \cdot (1 - \text{random}(0,1))</math></b>		0.41	0.50	0.64	0.17
0.25	<b><math>0.41 \cdot (1 - \text{random}(0,1))</math></b>		0.50	0.64	0.17
0.25	0.41	<b><math>0.50 \cdot (1 - \text{random}(0,1))</math></b>		0.64	0.17
0.25	0.41	0.50	<b><math>0.64 \cdot (1 - \text{random}(0,1))</math></b>		0.17
0.25	0.41	0.50	0.64	<b><math>0.17 \cdot (1 - \text{random}(0,1))</math></b>	

Nota: Verificar que  $x_i \geq ET_i$ ,  $\forall i = 1, \dots, P$

Decremento de los tiempos de aterrizaje del avión  $i$  ( $i = 1, \dots, P$ ), manteniendo los cambios realizados en los aviones previos

```
s = current_solution[:]
for i=1,...,P
    s[i] *= (1-random(0,1))
end_for
```

Si  $n = (1 - \text{random}(0,1))$ , entonces:

4

$0.25 \cdot n$	0.41	0.50	0.64	0.17
$0.25 \cdot n$	$0.41 \cdot n$	0.50	0.64	0.17
$0.25 \cdot n$	$0.41 \cdot n$	$0.50 \cdot n$	0.64	0.17
$0.25 \cdot n$	$0.41 \cdot n$	$0.50 \cdot n$	$0.64 \cdot n$	0.17
$0.25 \cdot n$	$0.41 \cdot n$	$0.50 \cdot n$	$0.64 \cdot n$	$0.17 \cdot n$

Nota: Verificar que  $x_i \geq ET_i$ ,  $i = 1, \dots, P$

Asignar el tiempo de aterrizaje objetivo/deseado a cada avión  $i$ ,  $i = 1, \dots, P$

```
for i=1,...,P
    s[i] = TT[i]
end_for
```

5

$\frac{TT_1 - ET_1}{LT_1 - ET_1}$		0.41	0.50	0.64	0.17
0.25	$\frac{TT_2 - ET_2}{LT_2 - ET_2}$	0.50	0.64	0.17	
0.25	0.41	$\frac{TT_3 - ET_3}{LT_3 - ET_3}$	0.64	0.17	
0.25	0.41	0.50	$\frac{TT_4 - ET_4}{LT_4 - ET_4}$	0.17	
0.25	0.41	0.50	0.64	$\frac{TT_5 - ET_5}{LT_5 - ET_5}$	

Intercambio

```

for i=1,...,P
    for j=i+1,...,P
        s[i] ↔ s[j]
    end_for_i
end_for_i

```

6

Por ejemplo, el vecino que resulta de intercambiar las posiciones 1 y 3 es:

0.50	0.41	0.25	0.64	0.17
------	------	------	------	------

Insercción

```

for i=1,...,P-1
    for j=i+1,...,P
        Copiar valor s[i] e intercalar en posición j
    end_for_i
end_for_i

```

7

Por ejemplo, el siguiente vecino ha sido obtenido insertando el segundo elemento en la última posición:

0.25	0.50	0.64	0.17	0.41
------	------	------	------	------

**VARIAS PISTAS DE ATERRIZAJE**

Intercambio del parámetro  $y_i$  entre aviones que aterrizan en la misma pista

1	[(1, <b>0.35</b> ), (3, <b>0.40</b> ), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
	[(1, <b>0.78</b> ), (3, 0.35), (6, <b>0.40</b> )]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
	[(1, 0.40), (3, <b>0.78</b> ), (6, <b>0.35</b> )]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
	[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, <b>0.85</b> ), (7, <b>0.17</b> )]	[(2, 0.61), (4, 0.93)]
...			

2	<p><u>Insertar cada avión en el resto de las pistas</u></p> <table border="1" data-bbox="245 322 1008 376"> <tr> <td>[(3, 0.35), (6, 0.78)]</td> <td>[(1, 0.40), (5, 0.85), (7, 0.17)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 405 1008 459"> <tr> <td>[(3, 0.35), (6, 0.78)]</td> <td>[(5, 0.85), (7, 0.17)]</td> <td>[(1, 0.40), (2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 488 1008 542"> <tr> <td>[(1, 0.40), (6, 0.78)]</td> <td>[(3, 0.35), (5, 0.85), (7, 0.17)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 571 1008 624"> <tr> <td>[(1, 0.40), (6, 0.78)]</td> <td>[(5, 0.85), (7, 0.17)]</td> <td>[(3, 0.35), (2, 0.61), (4, 0.93)]</td> </tr> </table> <p>...</p>	[(3, 0.35), (6, 0.78)]	[(1, 0.40), (5, 0.85), (7, 0.17)]	[(2, 0.61), (4, 0.93)]	[(3, 0.35), (6, 0.78)]	[(5, 0.85), (7, 0.17)]	[(1, 0.40), (2, 0.61), (4, 0.93)]	[(1, 0.40), (6, 0.78)]	[(3, 0.35), (5, 0.85), (7, 0.17)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (6, 0.78)]	[(5, 0.85), (7, 0.17)]	[(3, 0.35), (2, 0.61), (4, 0.93)]
[(3, 0.35), (6, 0.78)]	[(1, 0.40), (5, 0.85), (7, 0.17)]	[(2, 0.61), (4, 0.93)]											
[(3, 0.35), (6, 0.78)]	[(5, 0.85), (7, 0.17)]	[(1, 0.40), (2, 0.61), (4, 0.93)]											
[(1, 0.40), (6, 0.78)]	[(3, 0.35), (5, 0.85), (7, 0.17)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (6, 0.78)]	[(5, 0.85), (7, 0.17)]	[(3, 0.35), (2, 0.61), (4, 0.93)]											
3	<p><u>Incremento de los tiempos de aterrizaje del avión <math>i</math>, <math>i = 1, \dots, P</math></u></p> <p>Si <math>n = (1 + \text{random}(0,1))</math>, entonces:</p> <table border="1" data-bbox="245 860 1008 913"> <tr> <td>[(1, 0.40 · n), (3, 0.35), (6, 0.78)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 943 1008 996"> <tr> <td>[(1, 0.40), (3, 0.35 · n), (6, 0.78)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 1025 1008 1079"> <tr> <td>[(1, 0.40), (3, 0.35), (6, 0.78 · n)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 1108 1008 1162"> <tr> <td>[(1, 0.40), (3, 0.35), (6, 0.78)]</td> <td>[(5, 0.17 · n), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <p>...</p> <p>Nota: Verificar que <math>x_i \leq LT_i</math>, <math>i = 1, \dots, P</math></p>	[(1, 0.40 · n), (3, 0.35), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35 · n), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35), (6, 0.78 · n)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, 0.17 · n), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
[(1, 0.40 · n), (3, 0.35), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35 · n), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35), (6, 0.78 · n)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, 0.17 · n), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
4	<p><u>Decrementos de los tiempos de aterrizaje del avión <math>i</math>, <math>i = 1, \dots, P</math></u></p> <p>Si <math>n = (1 - \text{random}(0,1))</math>, entonces:</p> <table border="1" data-bbox="245 1480 1008 1534"> <tr> <td>[(1, 0.40 · n), (3, 0.35), (6, 0.78)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 1563 1008 1617"> <tr> <td>[(1, 0.40), (3, 0.35 · n), (6, 0.78)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 1646 1008 1700"> <tr> <td>[(1, 0.40), (3, 0.35), (6, 0.78 · n)]</td> <td>[(5, 0.17), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <table border="1" data-bbox="245 1729 1008 1783"> <tr> <td>[(1, 0.40), (3, 0.35), (6, 0.78)]</td> <td>[(5, 0.17 · n), (7, 0.85)]</td> <td>[(2, 0.61), (4, 0.93)]</td> </tr> </table> <p>...</p> <p>Nota: Verificar que <math>x_i \geq ET_i</math>, <math>i = 1, \dots, P</math></p>	[(1, 0.40 · n), (3, 0.35), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35 · n), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35), (6, 0.78 · n)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]	[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, 0.17 · n), (7, 0.85)]	[(2, 0.61), (4, 0.93)]
[(1, 0.40 · n), (3, 0.35), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35 · n), (6, 0.78)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35), (6, 0.78 · n)]	[(5, 0.17), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											
[(1, 0.40), (3, 0.35), (6, 0.78)]	[(5, 0.17 · n), (7, 0.85)]	[(2, 0.61), (4, 0.93)]											

5	<p><u>Asignar el tiempo de aterrizaje objetivo/deseado a cada avión <math>i</math>, <math>i = 1, \dots, P</math></u></p> <table border="1" data-bbox="280 322 986 389"> <tr> <td><math>[(1, \frac{TT_1-ET_1}{LT_1-ET_1}), (3, 0.35), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 416 986 483"> <tr> <td><math>[(1, 0.40), (3, \frac{TT_3-ET_3}{LT_3-ET_3}), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 510 986 577"> <tr> <td><math>[(1, 0.40), (3, 0.35), (6, \frac{TT_6-ET_6}{LT_6-ET_6})]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 604 986 672"> <tr> <td><math>[(1, 0.40), (3, 0.35), (6, 0.78)]</math></td> <td><math>[(5, \frac{TT_5-ET_5}{LT_5-ET_5}), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <p>...</p>	$[(1, \frac{TT_1-ET_1}{LT_1-ET_1}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, 0.40), (3, \frac{TT_3-ET_3}{LT_3-ET_3}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, 0.40), (3, 0.35), (6, \frac{TT_6-ET_6}{LT_6-ET_6})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, 0.40), (3, 0.35), (6, 0.78)]$	$[(5, \frac{TT_5-ET_5}{LT_5-ET_5}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$
$[(1, \frac{TT_1-ET_1}{LT_1-ET_1}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, 0.40), (3, \frac{TT_3-ET_3}{LT_3-ET_3}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, 0.40), (3, 0.35), (6, \frac{TT_6-ET_6}{LT_6-ET_6})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, 0.40), (3, 0.35), (6, 0.78)]$	$[(5, \frac{TT_5-ET_5}{LT_5-ET_5}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
6	<p><u>Aumentar los tiempos de aterrizajes de cada avión <math>i</math>, manteniendo los cambios previos realizados</u></p> <p>Si <math>n = (1 + \text{random}(0,1))</math>, entonces:</p> <table border="1" data-bbox="280 936 1005 987"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1014 1046 1066"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1093 1088 1144"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1171 1134 1223"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]</math></td> <td><math>[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <p>...</p> <p>Nota: Verificar que <math>x_i \leq LT_i</math>, <math>i = 1, \dots, P</math></p>	$[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$
$[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
7	<p><u>Reducir los tiempos de aterrizajes de cada avión <math>i</math>, manteniendo los cambios previos realizados</u></p> <p>Si <math>n = (1 - \text{random}(0,1))</math>, entonces:</p> <table border="1" data-bbox="280 1592 1005 1644"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1671 1046 1722"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1749 1088 1800"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]</math></td> <td><math>[(5, 0.17), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <table border="1" data-bbox="280 1827 1134 1879"> <tr> <td><math>[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]</math></td> <td><math>[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]</math></td> <td><math>[(2, 0.61), (4, 0.93)]</math></td> </tr> </table> <p>...</p> <p>Nota: Verificar que <math>x_i \geq ET_i</math>, <math>i = 1, \dots, P</math></p>	$[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$	$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$
$[(1, \mathbf{0.40 \cdot n}), (3, 0.35), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, 0.78)]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, 0.17), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											
$[(1, \mathbf{0.40 \cdot n}), (3, \mathbf{0.35 \cdot n}), (6, \mathbf{0.78 \cdot n})]$	$[(5, \mathbf{0.17 \cdot n}), (7, 0.85)]$	$[(2, 0.61), (4, 0.93)]$											

# 6 ANALISIS DE RESULTADOS

En este capítulo se simula en el entorno de programación JetBrains PyCharm los dos métodos establecidos para generar las soluciones al problema planteado. Se analizan cinco escenarios diferentes (ver anexo II), experimentando para, de esta forma validar y comparar los dos métodos de resolución implementados.

Estas técnicas se comparan en base al retraso total de los aviones y los tiempos de ejecución incurridos hasta la obtención de buenas soluciones. Para llevar a cabo esta comparación, se debe tener en cuenta que los parámetros propios de cada técnica influyen significativamente en las soluciones obtenidas, así como en los tiempos computacionales.

## 6.1 Calibrado de los métodos

El recocido simulado y la búsqueda en vecindades variables han sido aplicados satisfactoriamente en numerosos problemas complejos de optimización combinatoria. Esto se consigue escogiendo la representación de las soluciones más adecuada según las características del problema, así como las estructuras que generan soluciones vecinas y la selección de los valores de los parámetros utilizados en los algoritmos.

Tabla 13: *Parámetros calibrados*

<b>Recocido Simulado (SA)</b>	<b>Búsqueda en entornos variables (VNS)</b>
<ul style="list-style-type: none"><li>• <math>T_0</math></li><li>• A</li><li>• L</li><li>• <math>T_f</math></li></ul>	<ul style="list-style-type: none"><li>• Orden y selección de los vecindarios (k)</li></ul>

Seleccionar los valores más apropiados de estos parámetros tiene gran repercusión en las soluciones obtenidas y en el tiempo de ejecución de ambos algoritmos. Sin embargo, pese a la importancia que presenta este ajuste para entender el desempeño de la heurística, las referencias literarias que afrontan este problema son escasas, y en su mayoría estos valores se obtienen mediante una experimentación exhaustiva, requiriendo tiempos de computación muy elevados, además del esfuerzo y la continua intervención humana durante el proceso de diseño. No obstante, con la orientación y los conocimientos sobre las características particulares del problema que se está resolviendo, se obtienen buenos resultados con un número de iteraciones y tiempos razonables.

Para ambos métodos se experimentó con un problema reducido de 10 aviones y dos pistas de aterizajes con el objeto de acelerar el proceso de calibrado y reducir el tiempo de ejecución.

### 6.1.1 Recocido Simulado

Para obtener los valores de los parámetros se realizaron diversas experimentaciones combinando los valores representados en la Tabla 14, determinando la mejor opción en función del valor promedio.

Tabla 14  
Valores analizados de los parámetros

$T_0$	$\alpha$	L	$T_f$
20	0.8	5	10
50	0.9	10	5
100	0.95	20	1
200	0.97	50	0.1
500	0.99	100	0.05

Tabla 15  
Proceso de calibrado del algoritmo SA

Id	$T_0$	$\alpha$	L	$T_f$	$FO_{\min}$	$FO_{\min}$
					Una pista de aterrizaje (20 iteraciones)	Varias pistas de aterrizaje (15 iteraciones)
1	20	0.8	5	10	29389	-
2	20	0.8	5	5	33086	-
3	20	0.8	5	1	20440	-
4	20	0.8	5	0.1	4500	-
5	20	0.8	5	0.05	1931	-
6	20	0.8	10	5	24969	-
7	20	0.8	10	1	15591	-
8	20	0.8	10	0.1	2620	-
9	20	0.8	10	0.05	1900	-
10	20	0.8	20	0.1	1190	-
11	20	0.8	20	0.05	1180	-
12	20	0.8	50	0.1	1450	-
13	20	0.8	50	0.05	1160	-
14	20	0.8	100	0.1	920	-
15	20	0.8	100	0.05	910	-
16	20	0.9	5	0.05	1590	-

17	20	0.9	10	0.05	890	-
18	20	0.9	20	0.05	1140	-
19	20	0.9	50	0.05	1790	-
20	20	0.9	100	0.05	1030	-
21	20	0.95	10	0.05	3640	-
22	20	0.95	100	0.05	700	-
23	20	0.99	10	0.05	1570	-
24	20	0.99	100	0.05	760	-
25	20	0.97	100	0.05	990	-
26	50	0.8	100	0.05	970	168
27	50	0.9	100	0.05	940	111
28	50	0.95	100	0.05	820	129
29	50	0.97	100	0.05	1090	-
30	50	0.99	100	0.05	1010	-
31	100	0.9	100	0.05	810	124
32	100	0.95	100	0.05	1010	126
33	100	0.97	100	0.05	1000	145
34	100	0.99	100	0.05	910	140
35	200	0.9	100	0.05	930	120
36	200	0.95	100	0.1	760	172
37	200	0.95	100	0.05	880	175
38	200	0.97	100	0.05	880	123
39	200	0.99	100	0.05	710	143
40	500	0.9	100	0.05	1030	120
41	500	0.95	100	0.05	700	139
42	500	0.97	100	0.05	970	128
43	500	0.99	100	0.05	880	125

---

Según los resultados obtenidos, aquellas combinaciones que alcanzaron valores de la función objetivo inferiores a 1000 (u.m.) se analizaron con mayor nivel de detalle (valor promedio) para las dos versiones del problema estudiadas, obteniendo los siguientes valores:

Tabla 16  
*Parámetros del algoritmo recocido simulado*

Parámetro	Una pista de aterrizaje	Varias pistas de aterrizaje
$T_0$	500	200
$\alpha$	0.97	0.97
L	100	100
$T_f$	0.05	0.05

### 6.1.2 Búsqueda en entornos variables

Los vecindarios definidos para resolver el problema ALP se han descrito en 5.4.2.1. En este apartado se busca determinar la secuencia que permita obtener las mejores soluciones.

Tabla 17  
*Proceso de calibrado del algoritmo VNS para una y varias pistas de aterrizaje*

Id	Orden de los vecindarios	$FO_{\min}$	
		Una pista de aterrizaje (15 iteraciones)	Orden de los vecindarios Varias pistas de aterrizaje (15 iteraciones)
1	4,3,2,1,7,6,5	2200	1,2,3,4,5,6,7,8
2	3,4,1,2,7,6,5	1260	8,2,3,4,5,6,7,1
3	1,3,2,4,5,7	1670	2,3,7,8,4,6,1,5
4	1,3,2,4,7,5	1630	2,4,6,8,3,7,1,5
5	1,3,2,4,5,6	1110	5,4,6,8,3,7,1,2
6	1,3,2,4,6,5	1430	2,6,4,8,7,3,1,5
7	3,4,1,2,7,5	2050	2,4,6,5,3,7,1,8
8	3,4,1,2,5,7	1610	2,3,6,8,4,7,1,5
9	2,4,3,1,5	2100	2,3,7,8,4,6,1,5
10	3,4,1,2,5	7480	2,3,7,8,4,6,5

11	1,2,3,4,5	5530	2,3,6,8,4,7,5	210
12	5,2,3,4,1	8570	2,3,6,8,4,7,1	1423
13	1,5,3,2,6	2130	5,2,3,6,8,4,7	3189
14	5,1,3,2,6	2610	3,4,2,6,7,8,5	321
15	1,5,3,6	1330	2,3,6,4,7,5	1727
16	1,3,5,6	1180	3,7,2,6,4,5	4007
17	2,4,6,5	8430		
18	1,4,6,5	4790		
19	1,3,6,5	1540		
20	1,3,6,2	2540		
21	1,3,6,2,5	1120		
22	1,3,6,4,5	1460		
23	1,3,2,6,5	1920		
24	1,3,2,4,5	1830		
25	1,3,2,5,4	1680		
26	1,3,2,5,4,7	1190		
27	1,3,2,4,5,7	3730		
28	1,3,2,5,7,4	1870		
29	1,3,2,4,5,6,7	1440		
30	1,3,2,4,5,7,6	1970		
31	1,3,2,4,6,7,5	1610		

Obteniéndose las mejores soluciones para las secuencias de estructuras de vecindades indicadas en la Tabla 19.

Tabla 18  
*Orden de los vecindarios*

Una pista de aterrizaje	Varias pistas de aterrizaje
1,3,2,4,5,6	2,3,7,8,4,6,1,5

## 6.2 Resultados computacionales

Las pruebas realizadas han sido ejecutadas en un ordenador con sistema operativo Windows de 64 bits, procesador Intel® Core™ i7-6700HQ, CPU 2.60 GHz y 8.00 GB (RAM).

Tabla 19  
*Resultados computacionales*

	Número de aviones	Número de pistas de aterrizaje	Mejor valor conocido	SA			VNS		
				Valor	Tiempo (s)	Gráfica	Valor	Tiempo (s)	Gráfica
airland1	10	1	700	700	1.58	Fig. 17	1120	0.09	Fig. 19
		2	90	108.8	3.12	Fig. 21	110.4	0.54	Fig. 23
		3	0	0	7.74	Fig. 25	0	0.75	Fig. 27
airland2	15	1	1480	1540	3.89	Fig. 29	2300	0.24	Fig. 30
		2	210	250	14.55	Fig. 31	300	1.53	Fig. 32
		3	0	10	17.03	Fig. 33	60	1.97	Fig. 34
airland3	20	1	820	880	8.47	-	2380	0.91	-
		2	60	200	21.55	-	320	5.69	-
		3	0	172.7	13.66	-	140	5.37	-
airland6	30	1	24442	24442.0	6.61	Fig. 35	24442.0	1.86	Fig. 36
		2	554	1268	36.44	Fig. 37	2514	7.16	Fig. 38
		3	0	300	42.60	Fig. 39	748	7.48	Fig. 40
airland8	50	1	1950	4500	296.50	-	6550	21.46	-
		2	135	3715	117.74	-	1265	176.93	-
		3	0	3840	117.26	-	1055	232.08	-

## 6.2.1 airland1

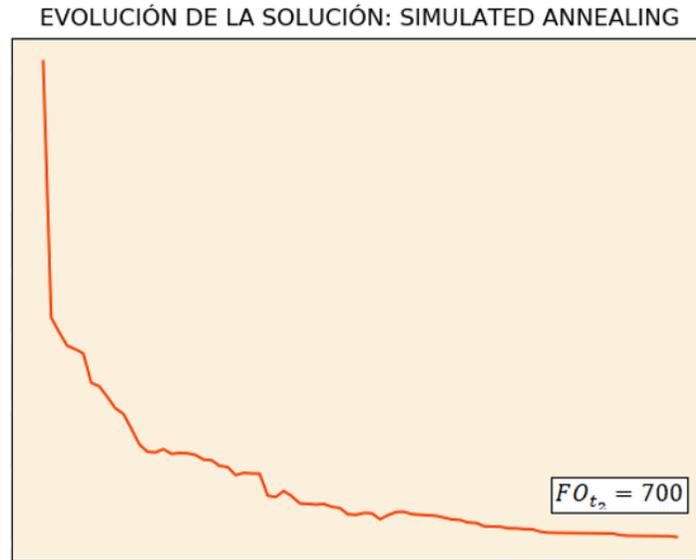


Figura 17. Evolución de la solución SA.  $\{P = 10, r = 1\}$

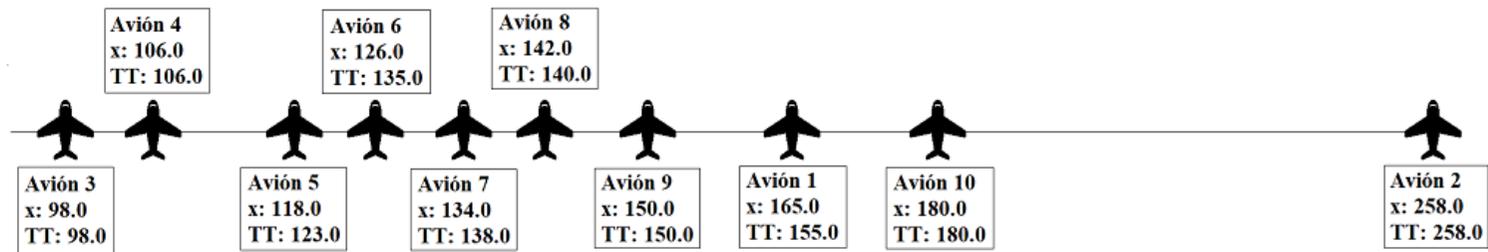


Figura 18. Representación gráfica de solución SA.  $\{P = 10, r = 1\}$

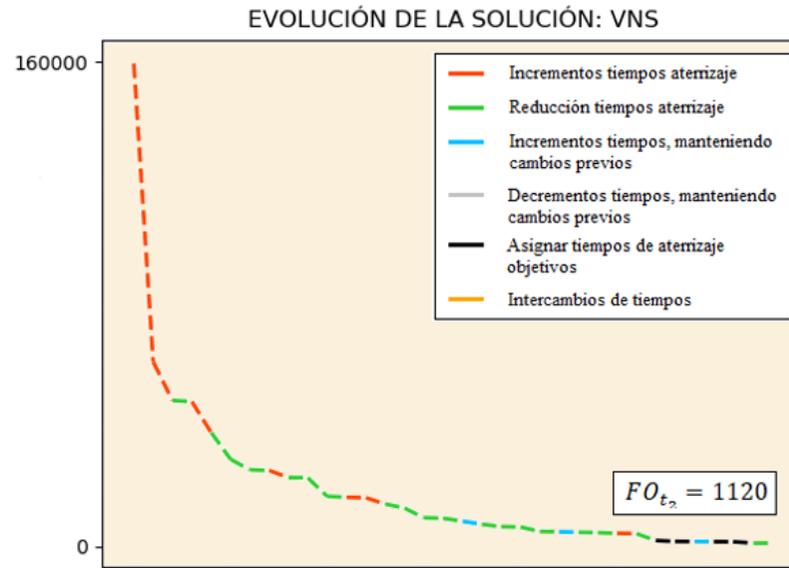


Figura 19. Evolución de la solución VNS.  $\{P = 10, r = 1\}$

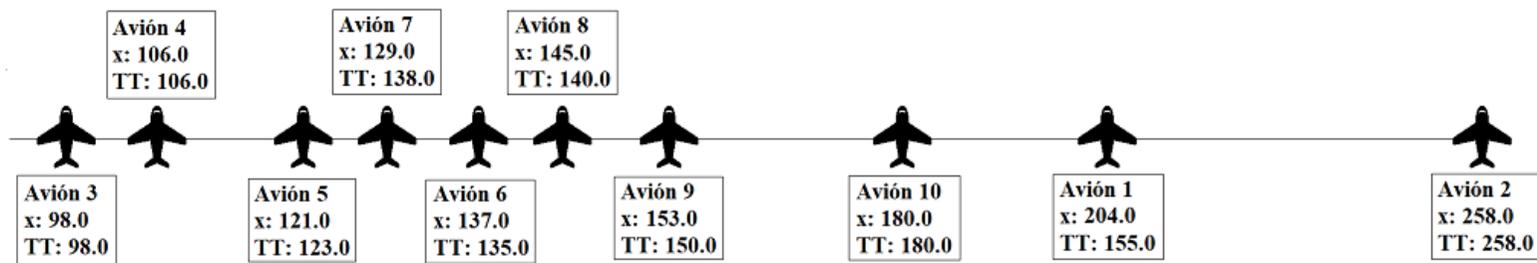


Figura 20. Representación gráfica de solución VNS.  $\{P = 10, r = 1\}$

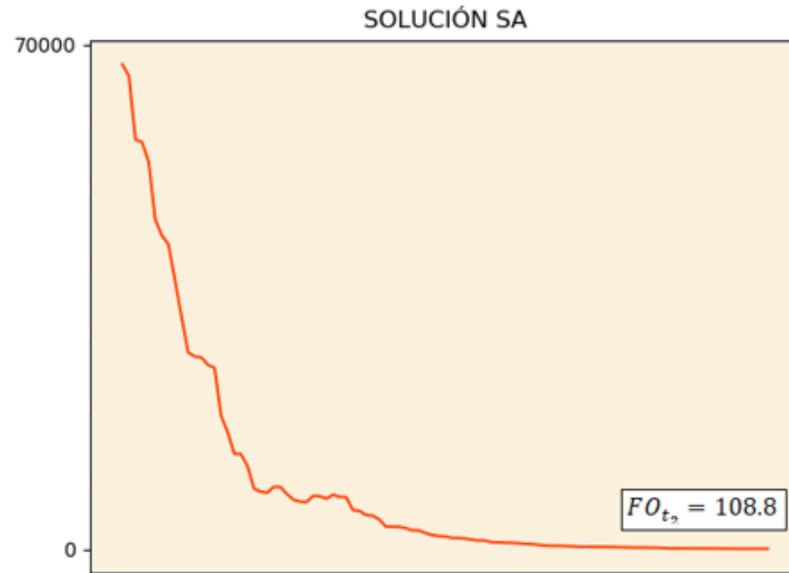


Figura 21. Evolución de la solución SA.  $\{P = 10, r = 2\}$

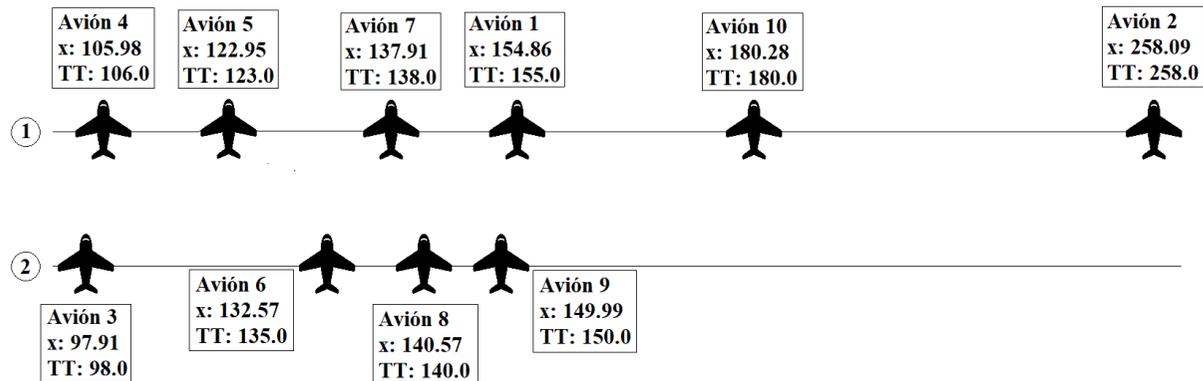


Figura 22. Representación gráfica solución SA.  $\{P = 10, r = 2\}$

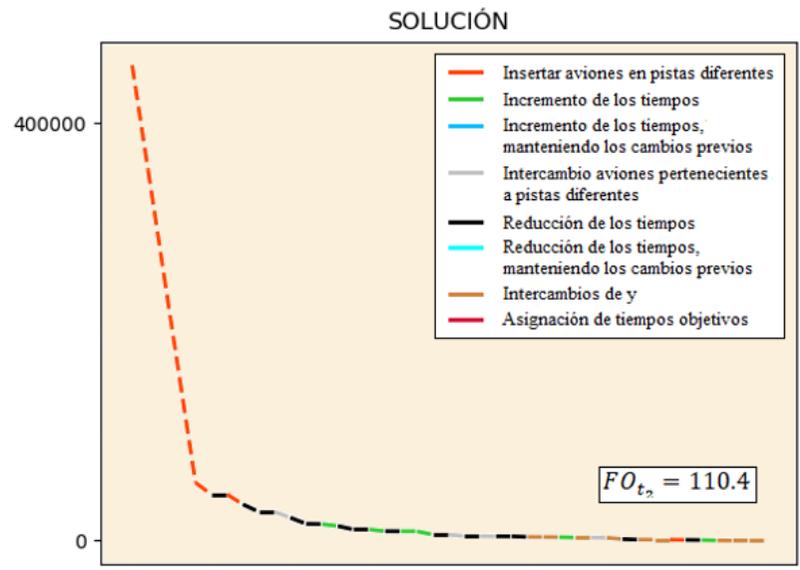


Figura 23. Evolución de la solución VNS. {P = 10, r = 2}

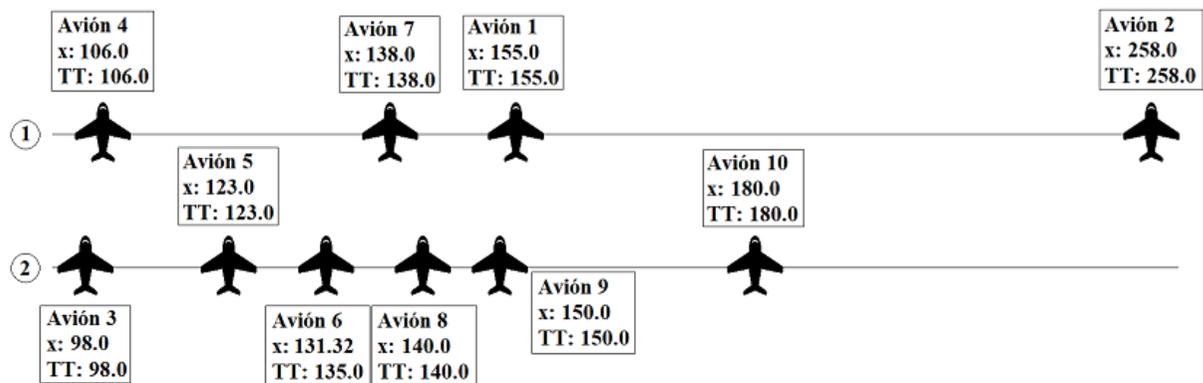


Figura 24. Representación gráfica de solución VNS. {P = 10, r = 2}

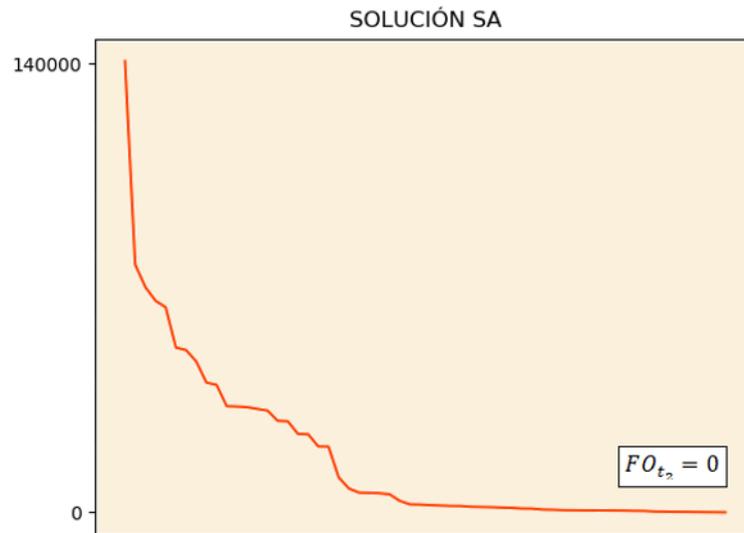


Figura 25. Evolución de la solución SA.  $\{P = 10, r = 3\}$

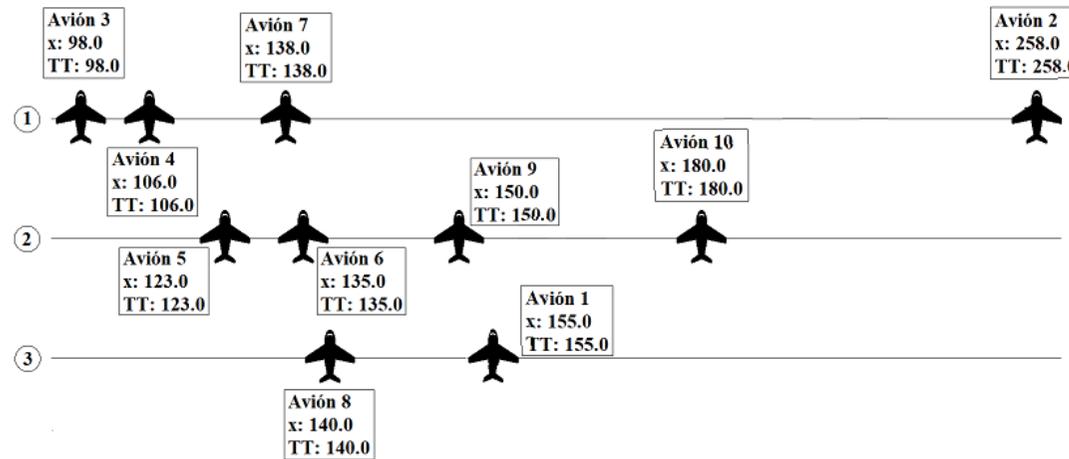


Figura 26. Representación gráfica solución SA.  $\{P = 10, r = 3\}$

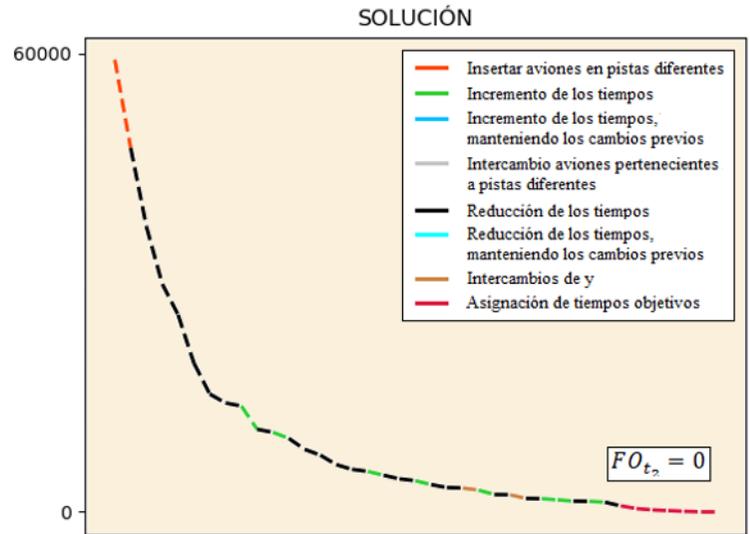


Figura 27. Evolución de la solución VNS.  $\{P = 10, r = 3\}$

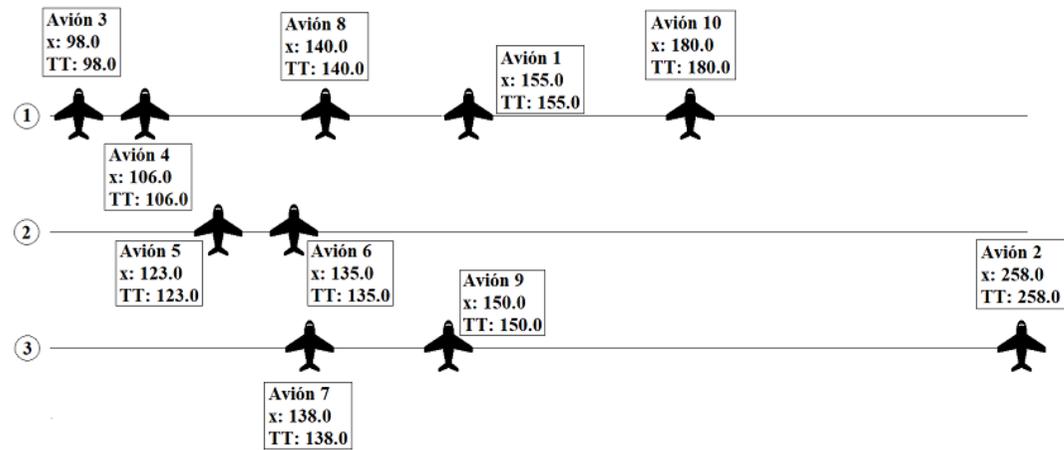


Figura 28. Representación gráfica de solución VNS.  $\{P = 10, r = 3\}$

## 6.2.2 airland2

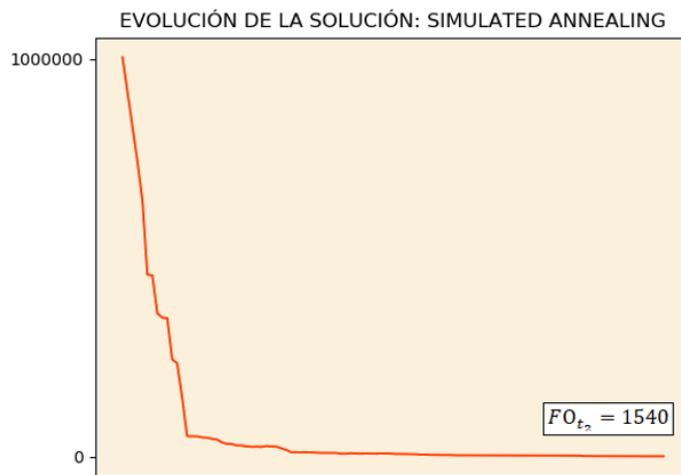


Figura 29. Evolución de la solución SA.  $\{P = 15, r = 1\}$

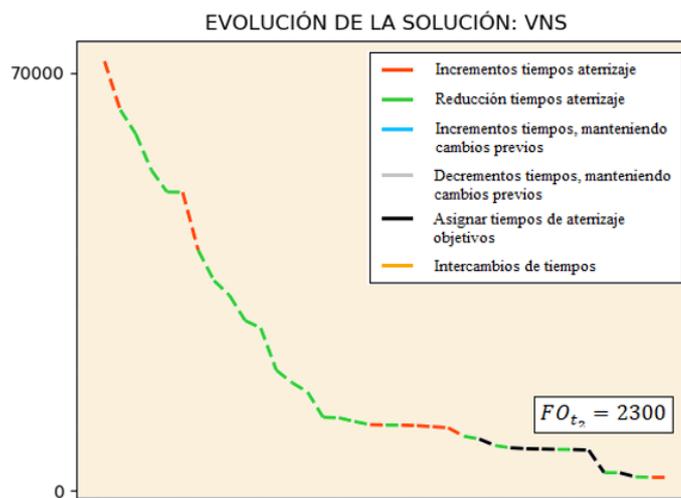


Figura 30. Evolución de la solución VNS.  $\{P = 15, r = 1\}$

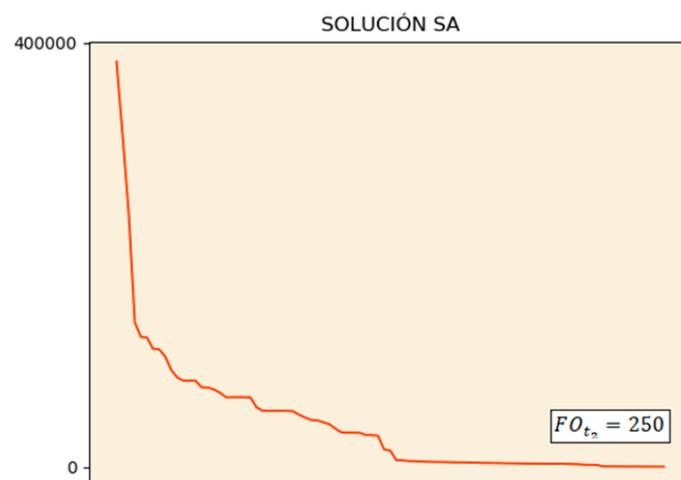


Figura 31. Evolución de la solución SA.  $\{P = 15, r = 2\}$

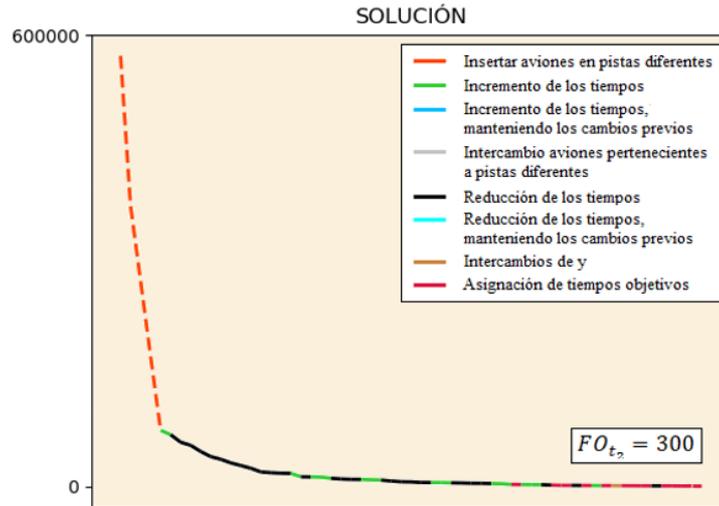


Figura 32. Evolución de la solución VNS.  $\{P = 15, r = 2\}$

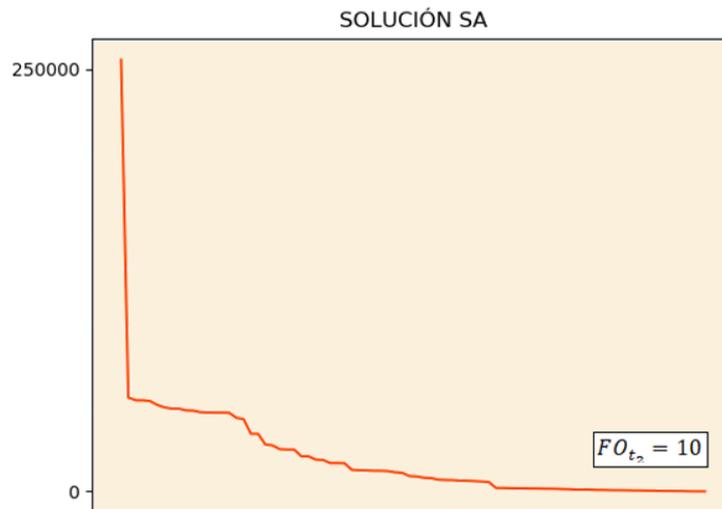


Figura 33. Evolución de la solución SA.  $\{P = 15, r = 3\}$

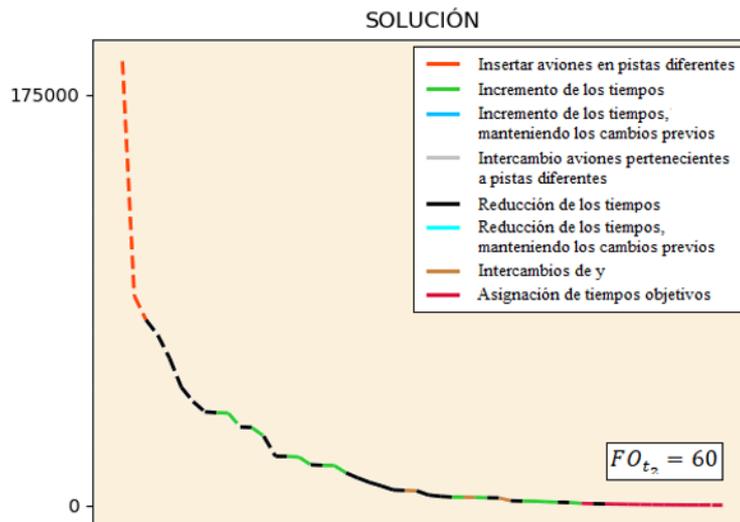


Figura 34. Evolución de la solución VNS.  $\{P = 15, r = 3\}$

### 6.2.3 irland6

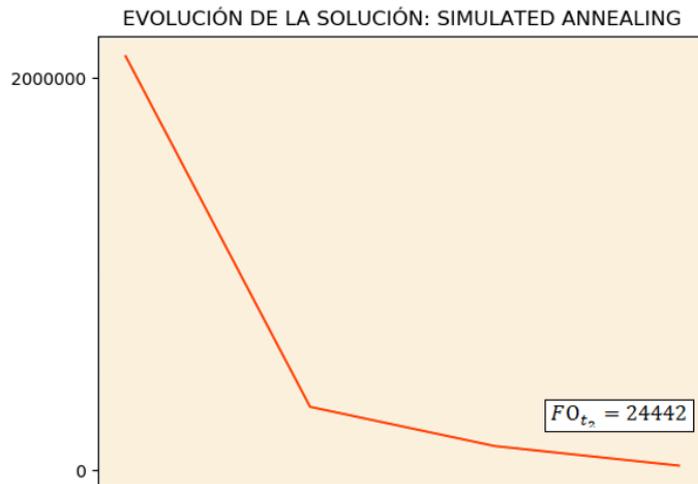


Figura 35. Evolución de la solución SA.  $\{P = 30, r = 1\}$

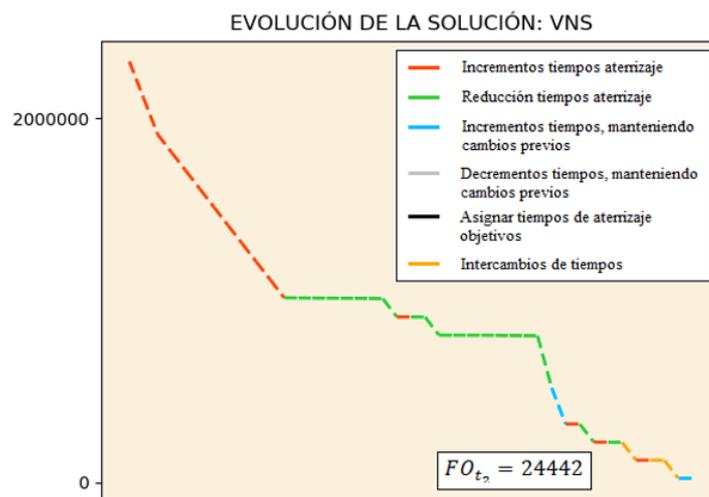


Figura 36. Evolución de la solución VNS.  $\{P = 30, r = 1\}$

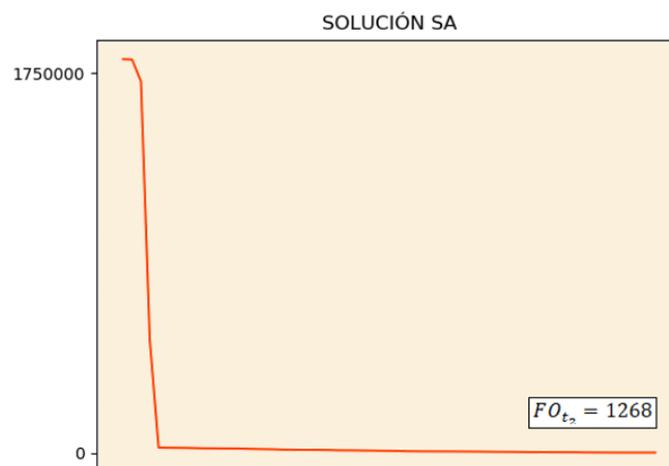


Figura 37. Evolución de la solución SA.  $\{P = 30, r = 2\}$

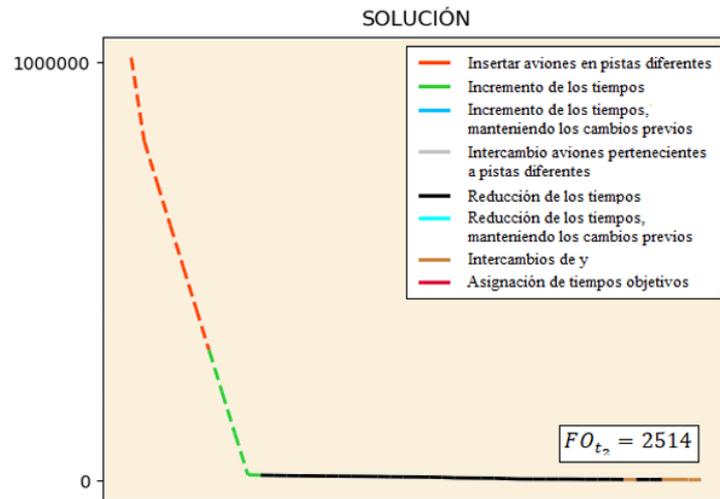


Figura 38. Evolución de la solución VNS.  $\{P = 30, r = 2\}$

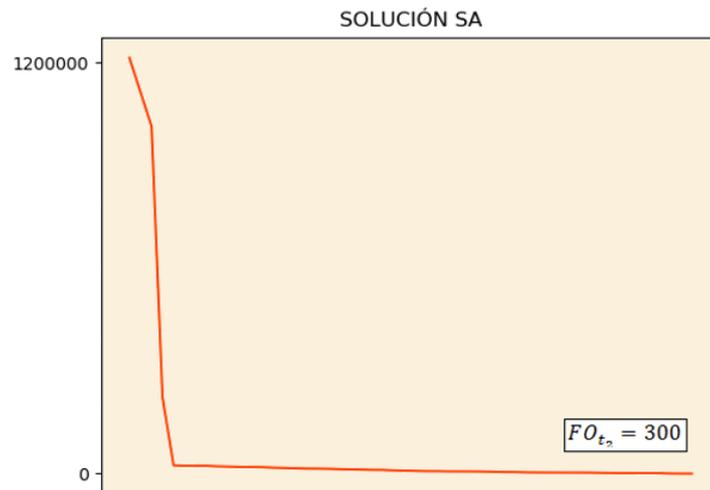


Figura 39. Evolución de la solución SA.  $\{P = 30, r = 3\}$

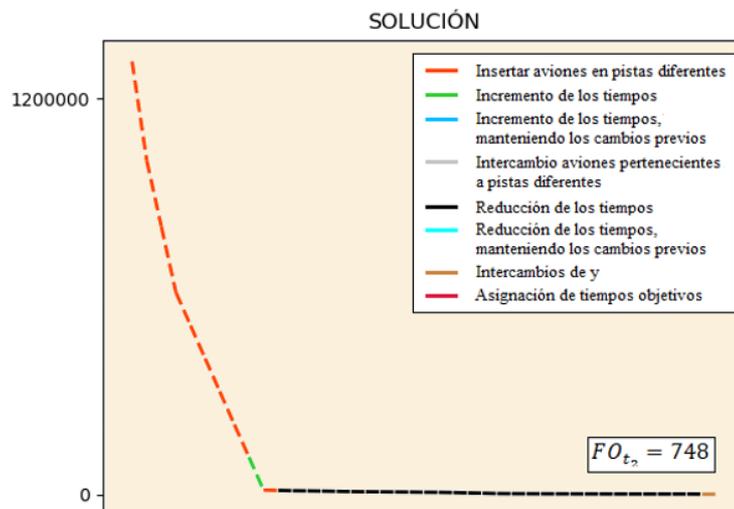


Figura 40. Evolución de la solución VNS.  $\{P = 30, r = 3\}$

### 6.3 Análisis de resultados

Habiendo resuelto ambas variantes del problema ALP mediante los algoritmos de recocido simulado y búsqueda en entornos variables, se procede en este apartado a realizar una comparativa entre ambas técnicas.

En cuanto a la duración de la ejecución del código de ambos algoritmos, se observa que, en un principio, el tiempo computacional del VNS es menor respecto del SA para cualquier número de pistas de aterrizaje. Sin embargo, en el caso de 50 aviones es el SA la técnica que presenta tiempos inferiores para dos y tres pistas de aterrizaje. Por otra parte, como es obvio, ambos coinciden en que, a mayor número de aviones, mayor es el tiempo computacional.

Cabe mencionar que la importancia de cada vecindario puede variar según las características del problema en particular. Por ejemplo, el caso de 30 aviones con una pista de aterrizaje, el valor óptimo se obtiene asignando a cada avión los tiempos más tardíos de aterrizaje ( $x_i = LT_i$ ). De esta forma, el vecindario que incrementa los tiempos manteniendo los cambios realizados en iteraciones anteriores es bastante eficaz en la búsqueda de la mejor solución.

Además, como se puede observar en la Tabla 20, los resultados más parecidos a los mejores valores conocidos hasta el momento se obtienen para los problemas con menor número de aviones. Esto es debido, principalmente, a que el calibrado de los parámetros y orden de los vecindarios se ha realizado para diez aviones y con un número limitado de simulaciones.

Ambos métodos presentan ventajas e inconvenientes. Como se ha mencionado anteriormente, de forma general, el VNS resuelve el problema en un menor tiempo computacional y el SA, por su parte, presenta mejores valores de la función objetivo. Así, en situaciones reales se debe discernir entre la mayor relevancia de la rapidez computacional o mejorar el valor de la función objetivo. Lo más adecuado es intentar lograr un equilibrio entre ambos, sin penalizar el tiempo de ejecución y al mismo tiempo, no sacrificar buenas soluciones.

# 7 CONCLUSIONES

---

El objeto que persigue este proyecto es resolver uno de los principales problemas presentes en los aeropuertos en la actualidad, que son la congestión del tráfico aéreo y los retrasos de los vuelos. La preocupación existente en los aeropuertos para abordar el aumento de la demanda de vuelos agiliza la búsqueda de nuevos métodos y técnicas que permitan gestionar los recursos disponibles (utilización de las pistas de aterrizaje y coordinación de los despegues y aterrizajes) de la forma más eficiente posible.

El problema de secuenciar los aterrizajes de los aviones, conocido como *Aircraft Landing Problem*, se ha modelado como un problema de optimización mixta-entera (MIP) que tiene por función objetivo minimizar la desviación total de los tiempos de aterrizaje respecto los tiempos objetivo.

Tomando como referencia el aeropuerto de Sevilla, que proporciona información de las llegadas en intervalos de 3h se ha considerado analizar problemas con 10, 15, 20, 30 y 50 aviones en cada intervalo de tiempo objeto de estudio.

A continuación, se resume las diferentes etapas seguidas para la realización de este proyecto:

- En los primeros capítulos se introduce el problema ALP, su importancia en el día a día y el porqué es interesante y necesario analizar diferentes metodologías para resolverlo. Además, se incluye un estado del arte referente al problema y técnicas aplicadas hasta la fecha.
- Se presenta una formulación matemática que modela el ALP en un entorno estático con múltiples pistas de aterrizaje, basado en la propuesta en Beasley *et al.* (2000).
- Posteriormente, se detallan las características propias de los algoritmos utilizados para la resolución del ALP, el recocido simulado y la búsqueda en entornos o vecindades variables, con el objeto de entender cómo funcionan y su adaptación al problema.
- Por último, se ejecutan ambos métodos para resolver cinco problemas diferentes, realizando una comparativa de los resultados obtenidos a fin de determinar el método que mejor se adapte al problema planteado.

Una vez conocidos los resultados, se comprueba que ambos métodos son eficientes y permiten obtener soluciones muy parecidas a las mejores conocidas hasta la fecha. En cuanto a los tiempos de ejecución, son relativamente bajos y el coste computacional del SA es mayor con respecto al VNS. Por tanto, se concluye que, tanto el método del recocido simulado como la búsqueda en entornos variables son igualmente válidos en la resolución del problema porque se obtienen soluciones eficientes sin elevar excesivamente el coste computacional.

## 7.1 Líneas de investigación futura

La necesidad de seguir profundizando en esta línea de investigación, tanto de manera teórica como la validación práctica de los modelos es evidente.

En este proyecto se han centrado los esfuerzos en dar solución al problema de secuenciar los aterrizajes de los aviones en un entorno estático, esto es, conociendo de partida los datos referentes al número de pistas de aterrizaje, aviones y tiempos asociados a cada uno de ellos. Por otro lado, se ha resuelto el problema considerando únicamente la llegada de aviones (aterrizajes), cuando existen aeropuertos que gestionan los aterrizajes y los despegues conjuntamente en las mismas pistas de aterrizaje. Por tanto, una mejora para futuras investigaciones es afrontar este problema en un entorno dinámico.

Además, abrir la posibilidad de probar otros nuevos vecindarios o mejorar los existentes con el fin de mejorar la búsqueda de la solución óptima. Siguiendo esta línea, se puede mencionar, por ejemplo, la mejora en el proceso de obtención de la solución inicial mediante la implementación de algún procedimiento.

# 8 REFERENCIAS

---

- [1] Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D., 2000, "Scheduling Aircraft Landings – The Static Case", *Transportation Science*, Vol. 34 (2), pp. 180-197.
- [2] Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., Abramson, D., 2004, "Displacement problem and dynamically scheduling aircraft landings", *Journal of the Operational Research Society*, Vol. 55, pp. 54-64.
- [3] A. Blumstein, "The landing Capacity of a Runway", *Operations Research*, vol. 7(6), pp. 752-763, 1959.
- [4] R. Dear, "The Dynamic Scheduling of Aircraft in the Near Terminal Area", R76-9, Flight Transportation Laboratory, MIT, USA, Tech. Rep., 1976.
- [5] Ernst, A. T., Krishnamoorthy, M., Storer, R. H., 1999, "Heuristic and Exact Algorithms for Scheduling Aircraft Landings", *Networks*, Vol. 34 (3), pp. 229-241
- [6] Farah, I., Kansou, A., Yassine, A., Galinho, T., 2011, "Ant Colony Optimization for aircrafts landings", *2011 4<sup>th</sup> International Conference on Logistics*, Hammamet, Tunisia, pp. 44-49.
- [7] Abela, J., Abramson, D., Krishnamoorthy, M., A. De Silva, Mills, G., 1995, "Computing Optimal Schedules for Landing Aircraft", *12<sup>th</sup> National Conference of the Australian Society for Operations Research*, Adelaide.
- [8] Awasthi, A., Kramer, O., Lässig, J., 2013, "Aircraft Landing Problem: Efficient Algorithm for a Given Landing Sequence", *IEEE 16<sup>th</sup> International Conference on Computational Science and Engineering*, Sydney, NSW, Australia.
- [9] Irene Moser and Tim Hendtlass, 2007, "Solving Dynamic Single-Runway Aircraft Landing Problems with Extremal Optimisation", *2007 IEEE Symposium on Computational*.
- [10] Shenpeng Yu, Xianbin Cao, Maobin Hu, Wenbo Du, Jun Zhang, 2009, "A real-time schedule method for aircraft landing scheduling problem based on cellular automaton", *Genetic and Evolutionary Computation Conference*, Shanghai, China.
- [11] Vadlamani, S., Hosseini, S., 2014, "A novel heuristic approach for solving aircraft landing problem with single runway", *Journal of Air Transport Management*, Vol. 40, pp. 144-148.
- [12] Balakrishnan, H., Chandran, B., 2006, "Scheduling Aircraft Landings under Constrained Position Shifting", *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
- [13] Bencheikh, G., Boukachour, J., 2009, "Hybrid method for aircraft landing scheduling based on a Job Shop formulation", *International Journal of Computer Science and Network Security*, Vol. 9, pp. 78-88.
- [14] Bencheikh, G., El Khoukhi, F., Boudebous, D., 2013, "Hybrid algorithms for the multiple runway aircraft landing problem", *International Journal of Computer Science and Applications*, Vol. 10(2), pp. 53-71.
- [15] Salehipour, A., Modarres, M., Moslemi Naeni, L., 2013, "An efficient hybrid meta-heuristic for aircraft landing problem", *Computers & Operations Research*, Vol. 40, pp. 207-213.
- [16] Xie, J., Zhou, Y., Zheng, H., 2013, "A hybrid metaheuristic for multiple runways aircraft landing problem based on bat algorithm" *Journal of Applied Mathematics*, Vol. 2013.
- [17] Bencheikh, G., Boukachour, J., El Hilali Alaoui, A., 2016, "A memetic algorithm to solve the dynamic multiple runway aircraft landing problem", *Journal of King Saud University - Computer and Information Sciences*, Vol. 28, pp. 98-109.
- [18] B. S. Girish, B. S., 2016, "An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem", *Applied Soft Computing*, Vol. 44, pp. 200–221.

- [19] Salehipour, A., 2018, “Algorithms for Single -and multiple- Runway Aircraft Landing Problem”.
- [20] Piinol, H., Beasley, J. E., 2004, “Scatter Search and bionomic Algorithms for the aircraft landing problem”, *European Journal of Operational Research*, Vol. 171(2006), pp. 439-462.
- [21] Lieder, A., Briskorn, D., Stolletz, R., 2015, “A dynamic programming approach for the aircraft landing problem with aircraft classes”, *European Journal of Operational Research*, Vol. 243(1), pp. 61-69.
- [22] Bencheikh, G., Boukachour, J., El Hilali Alaoui, A., 2011, “Improved Ant Colony Algorithm to Solve the Aircraft Landing Problem”, *International Journal of Computer Theory and Engineering*, Vol. 3(2).
- [23] Feng Xiao-rong, Feng Xing-jie, Liu-dong, 2014, “A Dynamic Multi-nest Ant Colony Algorithm for Aircraft Landing Problem”, *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 12(3), pp. 2196-2202.
- [24] Dhouib, S., 2011, “A Multi Start Adaptive Variable Neighborhood Search Metaheuristic for the Aircraft Landing Problem”, *2011 4<sup>th</sup> International Conference on Logistics*, Hammamet, Tunisia.
- [25] Nasser R. Sabar, Graham Kendall, 2014, “Aircraft Landing Problem using Hybrid Differential Evolution and Simple Descent Algorithm”, *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China.
- [26] Beasley, J. E., Sonander, J., Havelock, P., 2001, “Scheduling aircraft landings at London Heathrow using a population heuristic”, *Journal of the Operational Research Society*, Vol. 52, pp. 483-493.
- [27] Abdullah, O. S., Abdullah, S., Sarim, H. M., Ahmad Nazri, M. Z., 2018, “Optimization Approaches in aircraft landing problems”, *Sci. Int. (Lahore)*, Vol. 30(3), pp. 455-459.
- [28] J. Beasley, “OR-library: Distributing test problems by electronic mail,” *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.
- [29] Hansen, P., Mladenovic, N., 1999, “Variable Neighborhood Search”, *Computers & Operations Research*, Vol. 24(11), pp. 1097-1100.
- [30] ATAG (Air Transport Action Group), 2000, “The Economic Benefits of Air Transport”, [www.atag.org](http://www.atag.org)
- [31] Reynolds-Feighan, A., Button, K. J., 1999, “An Assessment of the Capacity and Congestion Levels at European Airport”, *Journal of Air Transport Management*, Vol. 5(3), pp. 113-134.
- [32] Ginés de Rus Mendoza (dir.), “La política de transporte europea: el papel del análisis económico”, Fundación BBVA, 2006.
- [33] Fernando, B. S. (s. f.). *Manual de radiofonista para pilotos de drones RPAS*. Ediciones Paraninfo, S.A.
- [34] Espinosa, R. (2019). *Manual del piloto de ultraligero*. España: Paraninfo.
- [35] P. Hansen and N. Mladenovic, 1999, *An introduction to variable neighborhood search, Metaheuristics: Advances and trends in local search paradigms for optimization*, 30, pp. 433–458. Kluwer.
- [36] Kim, D., Kim, K., Jang, W. y Chen, F., (2002), “Unrelated parallel machine scheduling with setup times using simulated annealing”, *Robotics and Computer Integrated Manufacturing*, 18, pp. 223-231
- [37] Aarts, E. y Korst, J., 1990, “Simulated annealing and boltzmann machines: A stochastic approach to combinatorial optimization and neural computing”, John Wiley & Sons, Chichester.
- [38] Aarts, E. y Van Laarhoven, P., 1985, “Statistical cooling: a general approach to combinatorial optimization problems”, *Philips Journal of Research*, 40 (1), pp. 193-226.
- [39] Atiqullah, M. M. (2004). “An Efficient Simple Cooling Schedule for Simulated Annealing”, En A. Laganá, M. L. Gavrilova, V. Kumar, Y. Mun, C. J. K. Tan, & O. Gervasi (Eds.), *Computational Science and Its Applications – ICCSA 2004* (pp. 396-404). Springer Berlin Heidelberg.
- [40] Cohn, H., & Fielding, M. (1999). Simulated Annealing: Searching for an Optimal Temperature Schedule. *SIAM Journal on Optimization*, 9(3), pp. 779-802.
- [41] Geman, S., & Geman, D. (s. f.). *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*. *IEEE transactions on pattern analysis and machine intelligence*. Vol. PAMI-6(6), Noviembre 1984.
- [42] Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, Vol.

13(2), pp. 311-329.

[43] Lundy, S. & Mees, A. (1986). Convergence of an annealing algorithm. *Mathematical programming*, Vol. 34(1), pp. 111-124.

[44] Osman, I. & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, Vol. 1(3), pp. 317-336.

# ANEXO II. BATERÍA DE PROBLEMAS

Los problemas que se resuelven en este proyecto se corresponden con los archivos de texto *airland1*, *airland2*, *airland3*, *airland6* y *airland8* disponibles en *OR-Library*, que es una colección de datos de un conjunto de problemas de optimización combinatoria que permite testear el desempeño de diversas técnicas de resolución. Este recurso puede localizarse en la dirección web: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/>

El formato de estos ficheros se explica a continuación:

Número de aviones (P)

Para cada avión  $i$  ( $i = 1..P$ ):

Instante que entra en rango del radar, Tiempo más temprano de aterrizaje, Tiempo de aterrizaje objetivo, tiempo más tardío de aterrizaje, coste de penalización por unidad de tiempo si el aterrizaje tiene lugar antes del tiempo deseado, coste de penalización por unidad de tiempo si el aterrizaje tiene lugar después del tiempo objetivo.

Para cada avión  $j$  ( $j = 1..P$ ): tiempo de separación requerido entre los aterrizajes de los aviones  $i$ ,  $j$  ( $i$  aterriza antes que el avión  $j$ )

A modo de ejemplo, se ilustra el contenido del archivo *airland8.txt*:

```
50
0 75 82 486 30.00 30.00
99999 3 8 15 8 15 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
28 103 117 565 30.00 30.00
15 15 15 99999 8 15 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
82 157 197 628 10.00 10.00
3 99999 8 15 8 15 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
126 201 261 735 10.00 10.00
8 8 8 8 99999 15 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
59 134 160 561 10.00 10.00
8 8 99999 15 8 15 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
20 95 106 524 30.00 30.00
15 15 15 15 15 99999 3 8
8 3 3 8 8 3 3 15
15 8 8 15 15 3 8 8
8 3 3 8 3 15 15 3
8 15 15 3 8 3 15 8
15 15 3 3 15 15 3 8
15 3
110 185 229 664 10.00 10.00
```

3 3 3 3 3 3 99999 8	175 250 316 790 10.00 10.00
8 3 3 8 8 3 3 15	3 3 3 3 3 3 3 3
15 8 8 15 15 3 8 8	3 3 3 3 3 99999 3 15
8 3 3 8 3 15 15 3	15 8 8 15 15 3 8 8
8 15 15 3 8 3 15 8	8 3 3 8 3 15 15 3
15 15 3 3 15 15 3 8	8 15 15 3 8 3 15 8
15 3	15 15 3 3 15 15 3 8
	15 3
23 98 108 523 30.00 30.00	
8 8 8 8 8 8 8 99999	
8 3 3 8 8 3 3 15	139 214 258 688 10.00 10.00
15 8 8 15 15 3 8 8	3 3 3 3 3 3 3 3
8 3 3 8 3 15 15 3	3 3 3 3 3 3 99999 15
8 15 15 3 8 3 15 8	15 8 8 15 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8
42 117 132 578 30.00 30.00	15 3
8 8 8 8 8 8 8 8	
99999 3 3 8 8 3 3 15	235 310 409 967 10.00 10.00
15 8 8 15 15 3 8 8	15 15 15 15 15 15 15 15
8 3 3 8 3 15 15 3	15 15 15 15 15 15 15 99999
8 15 15 3 8 3 15 8	15 8 8 15 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8
42 117 130 569 30.00 30.00	15 3
3 3 3 3 3 3 3 3	
3 99999 3 8 8 3 3 15	194 269 338 818 10.00 10.00
15 8 8 15 15 3 8 8	15 15 15 15 15 15 15 15
8 3 3 8 3 15 15 3	15 15 15 15 15 15 15 15
8 15 15 3 8 3 15 8	99999 8 8 15 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8
57 132 149 615 30.00 30.00	15 3
3 3 3 3 3 3 3 3	
3 3 99999 8 8 3 3 15	162 237 287 726 10.00 10.00
15 8 8 15 15 3 8 8	8 8 8 8 8 8 8 8
8 3 3 8 3 15 15 3	8 8 8 8 8 8 8 8
8 15 15 3 8 3 15 8	8 99999 8 15 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8
39 114 126 551 30.00 30.00	15 3
8 8 8 8 8 8 8 8	
8 8 8 99999 8 3 3 15	69 144 160 607 30.00 30.00
15 8 8 15 15 3 8 8	8 8 8 8 8 8 8 8
8 3 3 8 3 15 15 3	8 8 8 8 8 8 8 8
8 15 15 3 8 3 15 8	8 8 99999 15 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8
186 261 336 834 10.00 10.00	15 3
8 8 8 8 8 8 8 8	
8 8 8 8 99999 3 3 15	76 151 169 624 30.00 30.00
15 8 8 15 15 3 8 8	15 15 15 15 15 15 15 15
8 3 3 8 3 15 15 3	15 15 15 15 15 15 15 15
8 15 15 3 8 3 15 8	15 15 15 99999 15 3 8 8
15 15 3 3 15 15 3 8	8 3 3 8 3 15 15 3
15 3	8 15 15 3 8 3 15 8
	15 15 3 3 15 15 3 8

15 3  
 504 579 628 866 15.00 15.00  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 99999 3 8 8  
 8 3 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 257 332 425 799 30.00 30.00  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 99999 8 8  
 8 3 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 150 225 333 712 25.00 25.00  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 99999 8  
 8 3 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 101 176 269 610 10.00 10.00  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 99999  
 8 3 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 203 278 398 721 15.00 15.00  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 99999 3 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 220 295 378 781 30.00 30.00  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 99999 3 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 250 325 426 954 20.00 20.00  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 99999 8 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 15 15 3 3 15 15 3 8  
 15 3  
 259 334 498 897 20.00 20.00  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 99999 3 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 287 362 521 1012 15.00 15.00  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 99999 15 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 421 496 634 1023 20.00 20.00  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 99999 15 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 400 475 674 976 15.00 15.00  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 99999 3  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 254 329 512 863 30.00 30.00  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 3  
 3 3 3 3 3 3 3 99999  
 8 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 281 356 524 963 25.00 25.00  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 8 8 8 8 8 8 8 8  
 99999 15 15 3 8 3 15 8  
 15 15 3 3 15 15 3 8  
 15 3  
 296 371 562 872 10.00 10.00  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15  
 15 15 15 15 15 15 15 15

15 99999 15 3 8 3 15 8  
15 15 3 3 15 15 3 8  
15 3

236 311 412 896 15.00 15.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 99999 3 8 3 15 8  
15 15 3 3 15 15 3 8  
15 3

417 492 717 1231 30.00 30.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 99999 8 3 15 8  
15 15 3 3 15 15 3 8  
15 3

396 471 564 1013 20.00 20.00  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 99999 3 15 8  
15 15 3 3 15 15 3 8  
15 3

360 435 572 964 20.00 20.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 99999 15 8  
15 15 3 3 15 15 3 8  
15 3

386 461 654 1036 15.00 15.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 99999 8  
15 15 3 3 15 15 3 8  
15 3

416 491 725 1098 20.00 20.00  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 99999  
15 15 3 3 15 15 3 8  
15 3

437 512 732 1127 15.00 15.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15

15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
99999 15 3 3 15 15 3 8  
15 3

439 514 763 1134 30.00 30.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 99999 3 3 15 15 3 8  
15 3

223 298 401 815 25.00 25.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 99999 3 15 15 3 8  
15 3

244 319 450 827 10.00 10.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 99999 15 15 3 8  
15 3

254 329 456 927 15.00 15.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 99999 15 3 8  
15 3

346 421 673 1042 30.00 30.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 99999 3 8  
15 3

289 364 527 958 20.00 20.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 99999 8  
15 3

307 382 577 916 20.00 20.00  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8

8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 8  
8 8 8 8 8 8 8 99999  
15 3

302 377 468 894 15.00 15.00  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
15 15 15 15 15 15 15 15  
99999 3

136 211 345 828 20.00 20.00  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 3 3 3 3 3 3 3  
3 99999