Algoritmos Exactos y Metaheurísticas

Introducción al problema de planificación de aterrizajes

Gabriel Castillo

20 de junio de 2022

Evaluación (No borrar)

Resumen	(5 puntos):	________
Introducción	(10 puntos):	________
Definición del Problema	(15 puntos):	________
Estado del Arte	(25 puntos):	________
Bibliografía	(5 puntos):	________
	Nota Final :	________

Resumen

El tráfico de aviones, su calendarización y la planificación de vuelos, ha sido siempre un eje central de costos que paga una empresa dentro del rubro de transportes aéreos. A priori, una mayor cantidad de vuelos disponibles por día se logra traducir en un mayor bien económico para los propietarios y de servicio para los consumidores, es por ello que se ha incentivado la resolución de un problema de optimización que logre encontrar dentro de un mar de posibilidades la mejor composición de viajes, tal que los costos atribuidos a la mala planificación sean mínimos.

1. Introducción

En este informe se estudiará uno de los principales problemas que las aerolíneas deben enfrentar en el día a día, el cual consiste en encontrar una óptima calendarización y planificación del aterrizaje y despegue de aviones dada una serie de restricciones a seguir tanto en tierra como en aire. El problema surge debido al coste operacional que significa aterrizar un avión una vez despegado, puesto que de depender solo de la planificación inicial (obviando lo ocurrido en el aire) la resolución sería sencilla. Cuando un avión debe aterrizar en una pista se enfrenta a una extensa cantidad de problemas a resolver, uno de estos son las turbulencias ocasionadas por la nave cuando ésta está preparando su llegada a tierra, de venir otro avión cerca las turbulencias generadas podrían afectar la integridad de la nave posterior, por lo que existe una ventana de tiempo a considerar que debe respetar un mínimo factible entre aterrizajes.

Se procederá a explicar detalladamente el problema mencionado, sus variantes, así como también se describirá el modelo matemático que lo rige con su objetivo a optimizar y algunas de las soluciones mejor trabajadas hasta la fecha.

2. Definición del Problema

Corresponde a un problema de tipo discreto, con principios en la matemática combinatoria.El problema básico consiste en hallar un horario óptimo para el aterrizaje de los aviones, así como también que aviones aterrizarán en cada pista (caso dinámico). Debido a lo explicado anteriormente, no es posible aterrizar un avión tras otro sin que esto traiga problemas, es por esto que se define el aterrizaje según ventanas de tiempo detalladas posteriormente. El rango de las ventanas se definirá según un tiempo más temprano de aterrizaje y un tiempo más tardío.

2.1. Notación empleada

La notación utilizada para definir el problema es la siguiente.

- P : Número de aviones
- $Wait$: Espera en aire (tiempo de agendamiento)
- E_i : Tiempo más temprano de llegada posible, corresponderá al tiempo de aterrizaje tras viajar a su velocidad máxima.
- L_i : Tiempo más tardío, corresponderá al tiempo límite que tendrá el avión para aterrizar si es que hace el uso más eficiente del combustible posible.
- : T_i : Tiempo objetivo (target time) corresponde al tiempo "favorito"de aterrizaje del avión y varía según su tipo (cargo, transporte, etc). Este tiempo está calculado por economía, optimibilidad en el uso de combustible, etc. $E_i \leq T_i \leq L_i$
- S_{ij} : Separación mínima de tiempo entre la llegada de un avión i y el siguiente j, tal que $S_{ij} \geq 0$ ya que 2 aviones no pueden llegar al mismo tiempo, además $i \neq j$.
- g_i : Costo añadido si un avión i llega antes de su tiempo objetivo $g_i \geq 0$.
- h_i : Costo añadido si un avión i llega después de su tiempo objetivo $h_i \geq 0$.
- A_i : Tiempo de llegada del avión i a la unidad de control aéreo.

Variables de decisión

- x_i : Tiempo de llegada del avión i
- α_i : Qué tan antes aterriza el avión con respecto a su tiempo objetivo T_i
- β_i : Qué tan después en tiempo aterriza el avión con respecto a su tiempo objetivo T_i
- δ_{ij} : Variable binaria que será 1 si es que el avión i llega antes del avión j, 0 si no.

En la formulación básica y estática del problema sugiere que los tiempos T_i, E_i, L_i sean números entero. También el problema está restringido a la existencia de una única pista de aterrizaje, un modelo más dinámico será presentado más adelante.

2.2. Módelo de representación del problema

Para el problema básico, la función objetivo queda definida según:

$$Min \sum_{i=1}^{P}(g_i\alpha_i + h_i\beta_i) \tag{1}$$

Cuyo objetivo será el de reducir los costos adquiridos debido a un aterrizaje atrasado o adelantado.

2.3. Restricciones

$$E_i \leq x_i \leq L_i \tag{2}$$

Esta restricción se utiliza para restringir la posibilidad de que un avión llegue fuera de su ventana de tiempo, puesto que una solución factible del problema es aquella que logra agendar cada aterrizaje respetando cada ventana.

$$\begin{gathered}\delta_{ij} + \delta_{ji} = 1 \\ i = 1, 2, 3...P, j = 1, 2, 3...P\end{gathered} \tag{3}$$

Básicamente se imposibilita un error con principios en su lógica, si un avión i llega antes que un avión j no es posible que j haya llegado antes que i.

La diferencia mínima en el aterrizaje entre dos aviones secuenciales S_{ij} debe siempre cumplirse, sin embargo dentro del abanico de posibilidades para dos ventanas vecinas, dígase [30,60], [90,100] podría ocurrir que 2 aterrizajes secuenciales transgredan S_{ij}. A modo de ejemplo considerar $S_{ij} = 40$, existe la posibilidad de que el avión i llegue en un tiempo de aterrizaje $x_i = 60$ y otro avión j en un tiempo de $x_j = 90$, si se considera entonces $x_j - x_i = 30$, existe una violación en la factibilidad del problema puesto que $30 < S_{ij}$ y no se respeta la diferencia mínima entre aterrizajes. Para solucionar esto se consideran los siguientes conjuntos.

- U = Conjunto de pares (i, j) para los cuales no existe certeza sobre si el avión i llega antes que j.

- V = Conjunto de (i, j) para los que existe certeza de que el avión i llega antes que j´, pero se desconoce si la diferencia mínima entre aterrizajes es respetada.

- W = Conjunto de pares (i,j) para los cuales i llegará antes que j y además estará asegurado el cumplimiento de S_{ij}

La definición formal del conjunto W puede expresarse matemáticamente como:

$$\begin{aligned} W = [(i, j) \mid L_i < E_j \ \wedge \ L_i + S_{ij} \leq E_j] \\ i = 1, 2...P \ j = 1, 2...P \ j \neq i \end{aligned} \tag{4}$$

Lo que nos está diciendo (4) es que toda pareja de aviones (i,j) cumplirá con que el tiempo más temprano de aterrizaje para el avión j posterior a i, el tiempo de aterrizaje del avión j será obligatoriamente mayor que el tiempo más tardío de aterrizaje del avión i. Además $L_i + S_{ij} \leq E_j$ asegura el cumplimiento de la diferencia mínima entre aterrizajes.

Análogamente podemos definir al conjunto V como:

$$\begin{aligned} V = [(i, j) \ \mid \ L_i < E_j \ \wedge \ L_i + S_{ij} > E_j] \\ i = 1, 2...P \ j = 1, 2...P \ j \neq i \end{aligned} \tag{5}$$

Similar al conjunto anterior, salvo que se imposibilita una transgresión en el tiempo mínimo a esperar.
Finalmente se define U como:

$$\begin{aligned} [(i, j) \mid i = 1, 2...P \ j = 1, 2...P \ i \neq j \\ E_j \leq E_i \leq L_j \ \vee E_j \leq L_i \leq L_j \\ \vee E_i \leq E_j \leq L_i \ \vee \ E_i \leq L_j \leq L_i] \end{aligned} \tag{6}$$

Permitiendo todo tipo de combinaciones.
A sabiendas de esto tenemos:

$$\delta_{ij} = 1 \ \forall (i, j) \ \in W \cup V \tag{7}$$

Restricción interpolada de las propiedades de los conjuntos V, W que por definición aseguran que el avión i llegará antes que el avión j.

Se empieza entonces a restringir las variables dentro de los conjuntos que permiten errores, para V tenemos:

$$x_j \geq x_i + S_{ij} \ \forall (i, j) \in V \tag{8}$$

Cuya expresión establece que el tiempo S_{ij} debe respetarse dentro del conjunto. Tenemos que para cada par (i, j) la diferencia en la llegada $x_j - x_i \geq S_{ij}$, por lo que se resguarda la

diferencia mínima necesaria.
Finalmente se restringe el conjunto U de la siguiente forma:

$$x_j \geq x_i + S_{ij} - M\delta_{ji}\, \forall (i,j)\ \in U \tag{9}$$

Esta restricción tiene como objetivo inactivar o descartar una posible combinación dentro del conjunto. La expresión $\delta_{ji} = 1$ significa que el avión j llegó antes que el avión i, sin embargo no es posible entonces que el tiempo de aterrizaje x_j sea mayor que el tiempo de aterrizaje del avión anterior x_i más la diferencia entre aterrizajes "S_{ij}" , esta última asumida en el caso de que i hubiese llegado antes que j contradictoria a δ_{ji}. A priori es posible plantear (8) sólo porque es sabido que siempre se cumplirá que i llegará antes que j y que esto se cumple para todo par dentro del conjunto V. Pero en U esta situación no se puede obviar y se descarta en (9) utilizando un número entero M lo suficientemente grande para que la expresión no tenga sentido. Cabe destacar que si $\delta_{ji} = 0$, tenemos la ecuación (8).
Aunque esto puede solucionar el problema, se debe relajar.[el] problema a uno de programación lineal, consideremos M como $L_i + S_{ij} - E_j$, tras un juego matemático es posible definir la restricción anterior como:

$$x_j \geq E_j + (x_i - L_i) \tag{10}$$

Se deben restringir las demás variables, dígase α_i, β_i, x_i
Para α_i equivalente al costo por llegar antes del tiempo objetivo se propone que el costo sea mayor o igual a la diferencia entre su tiempo objetivo y su tiempo de llegada.

$$\begin{gathered} \alpha_i \geq T_i - x_i \\ i = 1,2,3...P \end{gathered} \tag{11}$$

Además se acota entre la no negatividad y el costo mínimo que puede tomar puesto que el tiempo más temprano corresponde a E_i.

$$\begin{gathered} 0 \leq \alpha_i \leq T_i - E_i \\ i = 1,2,3...P \end{gathered} \tag{12}$$

Con un objetivo similar y de manera análoga se restringe el costo por llegar tarde como:

$$\begin{gathered} \beta_i \geq x_i - T_i \\ i = 1,2,3...P \end{gathered} \tag{13}$$

$$0 \leq \beta_i \leq L_i - T_i \tag{14}$$

Finalmente el rango de valores para x_i se restringue según:

$$x_i = T_i + \alpha_i + \beta_i \tag{15}$$

Esta expresión es sumamente lógica, puesto que de no llegar el avión atrasado ni adelantado $\alpha_i = \beta_i = 0$ y el tiempo de llegada cumpliría el tiempo objetivo, de llegarse tarde el tiempo de llegada sería $T_i + \beta_i$ y de llegar temprano $T_i + \alpha_i$.

3. Algunas variantes

Para el caso estático manifestado con anterioridad, pueden haber algunas modificaciones dentro de sus restricciones o en la misma función objetivo según corresponda, una de estas es:

$$min\ max[x_i \mid i = 1, 2...P] \tag{16}$$

El objetivo de este cambio en la función objetivo es el de lograr aterrizar el último avión lo antes posible. En la práctica y para algunas situación posibles en la vida real, puede ser conveniente representar el problema de esta forma, por ejemplo, en situaciones dónde exista demasiado tráfico sería más adecuado sacrificar la minimización del costo y enfocar el problema hacía una minimización en los tiempos de llegada.

Otra variación en el problema es que alguna de las funciones de costo tenga asociada una función por tramos, dígase si tras atrasarse después de una determinada hora el costo variara según otra función lineal. Para incorporar esta versión se considera b_i como el punto quiebre entre el costo por llegar tarde, tal que $T_i \leq b_i \leq L_i$. Sea H_i el nuevo costo tras superar el tiempo de llegada b_i, de forma que $H_i \geq 0$. Este cambio se incorpora definiendo un $\theta_i \geq 0$ tal que:

$$\theta_i = x_i - b_i\ i = 1, 2, 3...P \tag{17}$$

y agregando a la función objetivo:

$$\sum_{i=1}^{P}(H_i \theta_i) \tag{18}$$

Esta corresponde a una formulación general de agregar funciones por tramos en los costos, se puede aplicar también para costos por aterrizaje más temprano y, además, puede haber más de un punto quiebre.

3.1. Módelo dinámico

La sutil y gran diferencia entre el problema estático presentado anteriormente y este es la cantidad de pistas de aterrizaje disponibles entre las cuales el avión podrá decidir aterrizar, en general la notación y la mayoría de restricciones son las mismas, pero existen variaciones. En los parámetros

- s_{ij} Separación entre dos aterrizajes para aviones (i, j) en **la misma** pista.
- t_{ij} Separación entre dos aterrizajes para aviones (i, j) en **diferentes** pistas.
- R Cantidad de pistas de aterrizaje

En las variables de decisión tenemos los siguientes cambios.

- $\gamma_{ir} = 1$ si es que el avión i está agendado para aterrizar en la pista r, 0 si es que no lo está.

- $\rho_{ij} = 1$ si es que el avión i aterriza en la misma pista que el avión j, 0 si no lo hace.

Cambios presentes en las restricciones.

$$\begin{aligned}(x_j - x_i) \geq s_{ij}\rho_{ij} + t_{ij}(1 - \rho_{ij}) - M\delta_{ji} \\ i, j = 1, 2, 3...P\end{aligned} \tag{19}$$

Esta restricción expresa que, de aterrizar en una misma pista de aterrizaje 2 aviones, deberá cumplirse un mínimo tiempo de espera s_{ij}, mientras que por otro lado, si 2 aviones aterrizan en pistas de aterrizaje diferentes, deberá cumplirse un mínimo tiempo de aterrizaje t_{ij}

$$\begin{aligned}\rho_{ij} \geq \gamma_{ir} + \gamma_{jr} - 1 \\ i, j = 1, 2, 3...Pr = 1, 2, 3...R\end{aligned} \tag{20}$$

Vinculación entre las variables ρ_{ij}, γ_{ir}

4. Estado del Arte

Este problema maneja múltiples versiones su planteamiento, alguna de ellas son las que utilizan un modelo entero-mixto, otras buscan relajar el problema hacia una formulación lineal o entera, algunas buscan enfocar la solución en los tiempos de aterrizaje, otras en los costos. Varían también las restricciones, algunas acotan las ventanas de tiempo con la finalidad de obtener una solución óptima, pero para combinaciones más discretas, otras restringen la cantidad de aeropuertos o pistas de aterrizaje. Existen también las que buscan dar una categoría más real al problema incluyendo aún más restricciones que las formalmente definidas en el problema estándar.

A pesar de todo eso, las soluciones más fuertemente trabajadas son aquellas que involucran el uso de múltiples pistas de aterrizaje (dinámicas) dado que se asemeja más a lo cotidiano.

Dentro de las resoluciones y planteamientos más importantes hasta la fecha tenemos la de Beasley [4] cuyo modelo solución permite resolver de manera **óptima** un problema con 50 aviones y múltiples pistas de aterrizaje. También está una solución empleada por Pinol y Beasley [6] que permitió resolver de manera eficiente, mas no óptima el problema con múltiples pistas de aterrizaje y cerca de 500 aviones utilizando un algoritmo poblacional conocido como "Scatter search" en conjunto con algoritmos binomiales. Se ha logrado resolver el modelo para versiones puntuales y más acotadas del problema, como por ejemplo la de Bauerle [3] que tras utilizar teoría de colas logró resolver instancias de 75 aviones y 2 pistas. Otro impresionante planteamiento es el entregado por Artiouchine [2] cuyo modelo permite ser resuelto en tiempos polinomiales para algunas instancias específicas, el planteamientos utilizado es uno similar al propuesto más arriba en el sentido de que usa un modelo entero mixto, pero limita fuertemente la variabilidad de las ventanas de tiempo y otras restricciones vinculadas.

Se puede observar que se han desarrollado soluciones al problema utilizando algoritmos exactos y, en otros casos, heurísticos. Este problema al ser considerado NP difícil manifiesta vicisitudes en la complejidad operacional que define las restricciones, esto ocurre hasta para instancias mediana-pequeñas, por lo que a pesar de los esfuerzos por conseguir soluciones óptimas utilizando heurísticas o búsquedas completas se ha empezado a trabajar el problema utilizando metaheurísticas. Estos últimos modelos solución no entregan una respuesta óptima necesariamente, aunque son herramientas poderosas que pueden entregar soluciones más que aceptables, incluso [1] a través del uso de meta heurísticas ha logrado dar una solución razonable en tiempo para instancias de hasta 500 aviones, esto a través de un algoritmo titulado " **A simulated annealing based-meta-heuritic for aircraft landing problem**" que asegura resoluciones cercanas a la de un óptimo global.

5. Representación

Se utiliza el lenguaje python, una de las facilidades que implica la utilización de este lenguaje es la de la flexibilidad con la que se puede representar un grafo a través de una lista de listas. Los parámetros del problema se representan según:

- Lista de cantidad de aviones y tiempo de espera en aire $[P, Wait]$
- Grafo de aviones (lista de listas): Dónde cada nodo representa $[A_i, E_i, T_i, L_i]$
- Grafo de costos (lista de listas): Dónde cada nodo representa $[g_i, h_i]$
- Grafo de espera mínima (lista de listas): Dónde cada nodo representa $[S_{ij}]$

Las soluciones se representa de la siguiente manera [5]:

Aircrafts (i)	1	2	3	4	5	6	7	8	9	10
Landing times (t_i)	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}

Figura 1: Modelo de solución

Donde i corresponde al índice de la lista, en este caso el avión, y t_i el tiempo en que aterriza.

6. Funcionamiento

Para el caso con una sola pista de aterrizaje, la idea general de este algoritmo es la de subdividir el problema en instancias más pequeñas y aplicar Hill climbing sobre ellas, estas soluciones preliminares condicionarán los tiempos de aterrizaje de aviones posteriores según las restricciones del modelo lo señalan.

Lo primero que se hace es aplicar un algoritmo de First Come First Served (FCFS) para reordenar la información según los tiempos de llegada al radar de la torre de control, la finalidad de esto es la de crear una primera vecindad con los aviones que llegan primero.

```
Antes de ordenar :  [[24, 129, 155, 559], [120, 195, 258, 744], [14, 89, 100, 510], [21, 96, 106, 521], [16, 110, 123, 555], [45, 120, 135, 576]]
Después de ordenar :  [[14, 89, 100, 510], [16, 110, 123, 555], [21, 96, 106, 521], [24, 129, 155, 559], [45, 120, 135, 576], [120, 195, 258, 744]]
```

Figura 2: Modelo de solución

Puesto que en los parámetros iniciales del problema se fija un tiempo de espera en aire $Wait$, cuando un avión i entra en el radar de la torre de control, se retiene durante este tiempo a la espera de más aviones, en el código, se crea un arreglo **combinaciones** con el índice de los aviones que llegan en el intervalo $A_i + Wait$.

Para la subdivisión de combinaciones se utilizan las dimensiones 1,2,3. Esto indicará el tamaño del vecindario a explorar y sus operaciones. Si el tamaño del primer índice de **combinaciones** es de tamaño 3, el primer vecindario a explorar lo hará un algoritmo de Hill climbing de dimensión 3, generado desde el tiempo mínimo de aterrizaje (Velocidad máxima) en que los 3 aviones respetan las restricciones. De la misma forma ocurre con la dimensión 2, y para 1 simplemente se aterriza el avión en su tiempo óptimo de llegada, si esto es posible. De llegar más de 3 aviones en un mismo intervalo se subdivide la lista en listas de tamaño 3, 2 y 1 hasta lograr representar su dimensión inicial, luego el razonamiento anterior es el mismo.

```
[[0, 1, 2, 3], [1, 2, 3], [2, 3], [3], [4], [5]]
[[0, 1, 2], [3], [4], [5]]
```

Figura 3: Ejemplo de las combinaciones

```
Vecindario inicial [[76, 158], [76, 157], [75, 158], [75, 157]]
```

Figura 4: Ejemplo de un vecindario tamaño 2

```
Antes de ordenar :  [[54, 129, 155, 559], [120, 195, 258, 744], [14, 89, 98, 510], [21, 96, 106, 521], [35, 110, 123, 555], [45, 120, 135, 576], [49, 124, 138, 577]]
Después de ordenar :  [[14, 89, 98, 510], [21, 96, 106, 521], [35, 110, 123, 555], [45, 120, 135, 576], [49, 124, 138, 577], [54, 129, 155, 559], [120, 195, 258, 744]]
```

Figura 5: Antes y después de ordenar

6.1. Hill climbing

6.1.1. Operaciones

Se recibe una combinación de tiempos de entrada de dimensión 3 o 2. Las operaciones a realizar para un arreglo de entrada de tamaño 3 son:

- +1,+1,+1
- +1,0,+1
- +1,0,0
- +1,0,-1
- +1,-1,+1
- +1,-1,0
- +1,-1,-1
- 0,+1,+1
- 0,+1,0
- 0,+1,-1
- 0,0,+1
- 0,0,0
- 0,0,-1
- 0,-1,+1
- 0,-1,0
- 0,-1,-1
- 0, -1, -1
- -1,+1,+1
- -1,+1,0
- -1,+1,-1
- -1,0,+1
- -1,0,0

- -1,0,-1
- -1,-1,+1
- -1,-1,0
- -1,-1,-1

Estas combinaciones generarán un nuevo vecindario de soluciones que se evaluarán posteriormente en la función objetivo, cuando eso ocurra se seleccionará la solución con menor costo y se repetirá el proceso. Todas las soluciones del vecindario están subscritas a las restricciones de la definición del problema, por lo que aquellas combinaciones que salgan de la ventana no se agregarán al vecindario. El procedimiento es análogo para un arreglo de entrada de tamaño 2.

Tras escoger una combinación óptima se agrega la configuración al arreglo de salida, para los aviones que faltan su nuevo tiempo mínimo corresponderá al tiempo de llegada del último avión más S_{ij}, esto es una forma indirecta de decir que un grupo de aviones llegó antes. Una vez terminado este proceso, se elimina de la lista de aviones por agendar, esta es **combinaciones** y se continúan generando vecindarios para los demás aviones.

7. Resultado preliminares

File name	N	Optimal	Gurobi		
			z	t (secs)	Error %
Airland1	10	700	700	0.26	0
Airland2	15	1480	1480	0.85	0
Airland3	20	820	820	0.40	0
Airland4	20	2520	2520	5.50	0
Airland5	20	3100	3100	13.75	0
Airland6	30	24442	24442	1.82	0
Airland7	44	1550	1550	2.50	0
Airland8	50	1950	1950	4.62	0
Airland9	100	5611.70	5635.23	5985	0.42
Airland10	150	12292.20	13130.19	6881	6.81
Airland11	200	12418.32	13032.62	2101	4.95
Airland12	250	16122.18	16677.01	1054	3.44
Airland13	500	37064.11	40193.09	7168	8.44

Table 4: Computational results of Gurobi Optimizer

Figura 6: Resultados Gurobi

[5] Esta tabla de resultados corresponde a la resolución del modelo matemático con Gurobi,éste último es un conjunto de módulos utilizado para la optimización, es posible utilizar Gurobi con python. Si se implementa el modelo según lo representado en este documento los valores serán muy cercanos al óptimo global del problema. En este documento se utilizaron los sets Airland1, Airland2, Airland. N corresponde al número de aviones, **z** al valor de la evaluación de la configuración en la función objetivo.

8. Análisis de resultados

Estos resultados en algunos casos pueden significar valores óptimos, mientras que en otros simplemente un valor bajo. La idea utilizada asegura un óptimo global para las subdivisiones del problema en dimensión 3, 2 y 1, esto pues el vecindario se calcula en todas las direcciones a seguir, eligiendo la mejor como si se tratara de un gradiente. Sin embargo, se deja entrever

Resultados hill climbing	
N	Valor solución
10	1240.0
15	5620.0
20	45320.0

Cuadro 1: Resultados hill climbing

que una solución para una división por separado no necesariamente posee sinergia con las demás y las limitaciones en las subdivisiones pueden llegar a restringir el decrecimiento en combinaciones posteriores. Para instancias muy pequeñas, es posible encontrar un óptimo global o en un principio, valores muy pequeños, pero a medida las instancias van creciendo el resultado es cada vez menos favorable. No obstante, la resolución utilizada permite dar un buen pie de inicio para aplicar otra resolución sobre la solución encontrada.

9. Conclusión y trabajo a futuro

Un posible trabajo a futuro sobre esta metodología es la reutilización de la solución obtenida para dar un buen pie de inicio en la aplicación de un nuevo modelo solución. Esta resolución al menos plantea la obtención de valor pequeño cada vez más preciso conforme se disminuye el tamaño de las instancias. Por otro lado, es posible mejorar lo anteriormente planteado tras incorporar la subdivisión de combinaciones tal que estas incorporen divisiones que sean mayores a 3, esto asegurará la obtención de valores cada vez menores, pero también retrasará el tiempo de resolución, pues la complejidad de esta solución es polinomial, limitada principalmente por la dimensión de los vecindarios generados.

Otro posible trabajo a futuro puede ser el de la incorporación de esta resolución al problema con más de una pista de aterrizaje, acá una vez obtenido un mínimo en una división del problema, si los tiempos de aterrizaje condicionan el valor de llegada de aviones posteriores estos podrían agendarse a una nueva pista, esto podría asegurar la independencia de cada subdivisión, que es la principal limitante de este modelo.

Referencias

[1] Leila Moslemi Naeni Amir Salehipour Mohammad Modarres. “An efficent hybrid metaheuristic for aircraft landing problem”. En: *Computer and operations research* 40 (2013).

[2] Durr C. Artiouchine K Baptiste P. “Runway sequencing with holding patterns.” En: *European Journal of Operational Research.* 189 (2008).

[3] Kolonko M. Bauerle N Engelhardt-Funke O. “On the waiting time of arriving aircrafts and the capacity of airports with one or two runways.” En: *European Journal of Operational Research.* 177 (2007).

[4] John E Beasley y col. “Scheduling aircraft landings—the static case”. En: *Transportation science* 34.2 (2000), págs. 180-197.

[5] Anastasia Kyriakidou. “A metaheuristic approach for solving the aircraft landing schedule problem”. Tesis doct. Department of applied informatics University of Macedonia, jul. de 2020.

[6] Beasley JE. Pinol H. “Scatter Search and Bionomic Algorithms for the aircraft landing problem.” En: *European Journal of Operational Research.* 171 (2006).