

ALGORITMOS EXACTOS Y METAHEURÍSTICAS: TAREA 2

Integrantes

José Martín Berríos Piña

Renato Óscar Benjamín Contreras Carvajal

Índice

1. Introducción	3
2. Escenario: 1 Pista	3
2.1. Algoritmos Greedy (1 Pista)	4
2.1.1. Greedy Determinista (1 Pista)	4
2.1.2. Greedy Estocástico (1 Pista)	5
2.2. GRASP + Hill Climbing (1 Pista, Base Greedy Determinista)	7
2.2.1. GRASP con Greedy Determinista como base y HC Alguna-Mejora (1 Pista)	7
2.3. GRASP + Hill Climbing (1 Pista, Base Greedy Estocástico con Componente de Restart ILS)	11
2.3.1. GRASP con Greedy Estocástico, HC Alguna-Mejora y Búsqueda Local Iterada con Reinicios (1 Pista)	11
2.4. Simulated Annealing (1 Pista, Greedy Determinista)	16
2.4.1. Diseño del Algoritmo Simulated Annealing (Determinista, 1 Pista)	17
2.4.2. Experimentación y Resultados (1 Pista, Greedy Determinista)	17
2.5. Simulated Annealing (1 Pista, Greedy Estocástico)	23
2.5.1. Diseño del Algoritmo y Metodología Experimental (Estocástico, 1 Pista)	23
2.5.2. Experimentación y Resultados (1 Pista, Greedy Estocástico)	24
3. Escenario: 2 Pistas	28
3.1. Algoritmos Greedy (2 Pistas)	29
3.1.1. Greedy Determinista (2 Pistas)	29
3.1.2. Greedy Estocástico (2 Pistas)	30
3.2. GRASP + Hill Climbing (2 Pistas, Base Greedy Determinista)	32
3.2.1. GRASP con Greedy Determinista como base y HC Alguna-Mejora (2 Pistas)	32
3.3. GRASP + Hill Climbing (2 Pistas, Base Greedy Estocástico con Componente de Restart ILS)	36
3.3.1. GRASP con Greedy Estocástico, HC Alguna-Mejora y Búsqueda Local Iterada con Reinicios (2 Pistas)	36
3.4. Simulated Annealing (2 Pistas, Greedy Determinista)	40
3.4.1. Diseño del Algoritmo Simulated Annealing (Determinista, 2 Pistas)	41
3.4.2. Experimentación y Resultados (2 Pistas, Greedy Determinista)	41
3.5. Simulated Annealing (2 Pistas, Greedy Estocástico)	46
3.5.1. Diseño del Algoritmo y Metodología Experimental (Estocástica, 2 Pistas)	46
3.5.2. Experimentación y Resultados (2 Pistas, Greedy Estocástico)	46
4. Análisis Comparativo y Conclusiones	52
4.1. Análisis del Rendimiento de los Algoritmos	52
4.1.1. Algoritmos Greedy: Una Base Constructiva	52

4.1.2. GRASP con Hill Climbing: El Poder de la Búsqueda Local	52
4.1.3. Simulated Annealing: Exploración Robusta del Espacio de Soluciones	53
4.2. Impacto del Número de Pistas	54
4.3. Influencia de los Parámetros y Estrategias de Inicialización	54
4.4. Conclusión	55
5. Referencias	55

1. Introducción

La gestión eficiente del tráfico aéreo se ha convertido en un pilar fundamental para la operación segura y optimizada de los aeropuertos a nivel mundial. Con una densidad de vuelos en constante aumento, la coordinación precisa de los aterrizajes de aeronaves es crucial no solo para garantizar la seguridad operacional, sino también para maximizar la eficiencia y minimizar los costos asociados a retrasos y desviaciones de los planes de vuelo. Este desafío se conoce comúnmente como el Problema de Secuenciación de Aterrizajes de Aeronaves (Aircraft Landing Problem - ALP).

El problema central abordado en este trabajo consiste en determinar una secuencia óptima de aterrizaje para un conjunto dado de D aeronaves. Cada aeronave k posee una ventana de tiempo definida por un tiempo de aterrizaje más temprano (E_k), un tiempo preferente (P_k) y un tiempo más tardío (L_k). El objetivo es asignar un tiempo de aterrizaje real a cada aeronave dentro de su ventana $[E_k, L_k]$, de tal manera que se minimice una función de costo total, la cual penaliza las desviaciones respecto al tiempo preferente P_k . Adicionalmente, se deben respetar estrictamente los tiempos mínimos de separación (τ_{ij}) entre el aterrizaje de una aeronave i y la siguiente aeronave j . La complejidad de este problema se incrementa al considerar la capacidad de las pistas y la posibilidad de operar en escenarios con una o múltiples pistas.

El presente informe detalla el diseño, implementación y evaluación experimental de diversas estrategias algorítmicas para abordar el Problema de Secuenciación de Aterrizajes. Se exploran desde enfoques constructivos simples, como los algoritmos Greedy en sus variantes determinista y estocástica, hasta metaheurísticas más sofisticadas, incluyendo GRASP (Greedy Randomized Adaptive Search Procedure) combinado con Búsqueda Local mediante Hill Climbing, y Simulated Annealing. Estos algoritmos se aplican y analizan en dos escenarios operativos distintos: uno con una única pista de aterrizaje disponible y otro con dos pistas, permitiendo así evaluar el impacto de la capacidad adicional en la complejidad y la calidad de las soluciones obtenidas.

El documento se estructura de la siguiente manera: la Sección 2 describe la aplicación y los resultados de los algoritmos mencionados para el escenario de una única pista. De forma análoga, la Sección 3 presenta el análisis correspondiente al escenario de dos pistas. Posteriormente, la Sección 4 ofrece un análisis comparativo global del rendimiento de todas las estrategias implementadas, discutiendo sus fortalezas, debilidades y la influencia del número de pistas, para finalmente extraer conclusiones sobre las técnicas más efectivas para el problema estudiado. El taller completo se puede encontrar en el Github: <https://github.com/renato357/Tarea-2-A-y-M-JB-RC>.

2. Escenario: 1 Pista

En este escenario, todas las aeronaves deben ser asignadas a una única pista de aterrizaje. Se aplicarán y evaluarán las tres estrategias algorítmicas propuestas: Algoritmos Greedy, GRASP + Hill Climbing, y Simulated Annealing.

2.1. Algoritmos Greedy (1 Pista)

Los algoritmos Greedy construyen una solución de manera iterativa, seleccionando en cada paso la opción que parece ser la más ventajosa localmente, con el objetivo de alcanzar una buena solución global. Para el problema de la secuenciación de aterrizajes, esto se traduce en decidir qué avión programar a continuación y asignarle un tiempo de aterrizaje, basándose en un criterio específico que busca optimizar algún aspecto inmediato del problema. A continuación, se describen las variantes determinista y estocástica de esta estrategia para el escenario de una única pista.

2.1.1. Greedy Determinista (1 Pista)

Descripción: El algoritmo Greedy Determinista para una pista única establece un orden de aterrizaje iniciando el tiempo del sistema T con el valor E_k (tiempo más temprano) mínimo entre los aviones pendientes. En cada ciclo, se identifica globalmente el avión no programado con el E_k más bajo. Este avión sirve como referencia para calcular un instante mínimo $T_{\min_objetivo} = \max(E_k, \text{Objetivo}, T_{\text{ultimo aterrizaje}} + \tau_{\text{ultimo, Objetivo}})$, considerando su E_k y la separación τ_{ij} con el último avión aterrizado. Si T es menor que este $T_{\min_objetivo}$ y este último es factible (no excede L_k del avión de referencia), T se actualiza. Luego, se evalúan todos los aviones pendientes para aterrizar en el T actual, verificando que T esté dentro de su ventana $[E_k, L_k]$ y que se respete la separación τ_{ij} . Entre los candidatos válidos, se selecciona el que posea el menor E_k , usando L_k , P_k y el ID como criterios de desempate sucesivos. El avión elegido se asigna a T y se retira de la lista de pendientes. Si no hay candidatos para el T actual, se avanza T al próximo evento significativo o de forma incremental. El proceso continúa hasta programar todos los aviones o agotar las iteraciones, siempre validando las restricciones operativas.

Resultados: La tabla 1 muestra el costo total y el orden de aterrizaje de los aviones obtenido por el algoritmo Greedy Determinista para cada caso de prueba en el escenario de 1 pista.

Caso de Prueba	Costo Total Obtenido	Orden de Aterrizaje (Secuencia de IDs)
Case 1	4600.00	3, 4, 5, 6, 8, 7, 9, 1, 10, 14, 13, 2, 12, 11, 15
Case 2	7140.00	1, 6, 8, 4, 12, 10, 9, 11, 3, 19, 20, 2, 7, 5, 15, 18, 14, 13, 17, 16
Case 3	1550.00	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44
Case 4	41325.91	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

Tabla 1: Resultados del Greedy Determinista (1 Pista)

Análisis: Los costos totales del algoritmo Greedy Determinista para una pista presentan una marcada fluctuación entre los casos. Aunque un mayor número de aviones, como en el Caso 4 (100 aviones, costo 41325.91), tiende a generar costos más altos, el Caso 3 (44 aviones) es una excepción notable, con un costo de 1550.00, significativamente menor que los Casos 1 (15 aviones, costo 4600.00) y 2 (20 aviones, costo 7140.00). Esta disparidad resalta cómo las características particulares de cada conjunto de datos —tales como la severidad de las penalizaciones por desviación (C_k, C'_k), la amplitud de las ventanas de aterrizaje ($L_k - E_k$), o la rigidez de los tiempos de separación τ_{ij} — influyen directamente en el desempeño del método. El Caso 3, por ejemplo, inicia con una aeronave cuyas ventanas de tiempo y preferente son cero, facilitando un comienzo sin penalización que podría propagar un efecto beneficioso. Por el contrario, la naturaleza inflexible de una estrategia determinista puede resultar en costos elevados cuando las decisiones iniciales, enfocadas en el E_k más bajo, provocan conflictos o desviaciones costosas más adelante. La secuencia de aterrizaje resultante es fija y no permite la exploración de alternativas que pudieran ser globalmente superiores.

2.1.2. Greedy Estocástico (1 Pista)

Descripción: El enfoque Greedy Estocástico para una pista comparte la estructura base del método determinista en cuanto al manejo del tiempo del sistema T y la identificación de aviones candidatos aptos para aterrizar en T (cumpliendo E_k, L_k y τ_{ij}). Su principal punto de divergencia yace en la selección entre múltiples candidatos factibles. En vez de una regla de desempate fija, se incorpora un elemento de azar: para cada candidato se calcula el costo de penalización si aterrizara en el T actual. Luego, se asigna un peso a cada uno, inversamente proporcional a dicho costo (peso $\approx 1/\text{costo}$), de manera que opciones de menor costo inmediato tengan mayor probabilidad de ser

elegidas. La selección final del avión se realiza aleatoriamente, ponderada por estos pesos. Si todos los costos de los candidatos son excesivamente altos o idénticos, la elección se hace de manera uniforme. Este procedimiento, repetido para diferentes semillas de aleatoriedad, busca diversificar las secuencias de aterrizaje, explorando caminos que el determinista omitiría y aumentando la probabilidad de hallar soluciones de menor costo global. Cada ejecución con una semilla puede requerir varios intentos internos si las primeras trayectorias aleatorias no logran programar todos los aviones de forma factible.

Resultados: Se realizaron 10 ejecuciones del algoritmo Greedy Estocástico para cada caso de prueba en el escenario de 1 pista, utilizando las siguientes semillas predefinidas: 42, 123, 7, 99, 500, 777, 2024, 1, 100, 314. Los resultados se resumen en las tablas 2 a 5. El 'Costo Det.' corresponde al valor obtenido por el Greedy Determinista para el mismo caso.

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
4600.00	4550.00	5800.00	5003.00	475.85

Tabla 2: Resumen Resultados Greedy Estocástico (1 Pista) - Case 1

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
7140.00	7240.00	9760.00	8121.00	828.33

Tabla 3: Resumen Resultados Greedy Estocástico (1 Pista) - Case 2

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
1550.00	1880.00	3006.00	2148.20	349.81

Tabla 4: Resumen Resultados Greedy Estocástico (1 Pista) - Case 3

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
41325.91	39158.82	41363.62	40322.84	845.31

Tabla 5: Resumen Resultados Greedy Estocástico (1 Pista) - Case 4

Análisis: La incorporación de aleatoriedad en el algoritmo Greedy permite la exploración de un espectro más diverso de soluciones. Las 10 ejecuciones con distintas semillas para cada caso evidencian una variabilidad en los costos finales. Para el Caso 1, la mejor solución estocástica (4550.00) superó ligeramente a la determinista (4600.00), si bien algunas semillas condujeron a costos mayores (hasta 5800.00). En el Caso 2, el determinista se mantuvo superior (7140.00) frente al mejor resultado estocástico (7240.00), y el promedio de este último (8121.00) fue aún más elevado. El Caso 3 es particular: el determinista (1550.00) fue notablemente mejor que cualquier solución estocástica encontrada (mejor estocástica: 1880.00), lo que sugiere que la estructura de esta instancia

se alinea bien con la lógica determinista o que la aleatorización implementada no fue suficiente para superar ese óptimo local. En contraste, para el Caso 4, el enfoque estocástico demostró su valor al obtener un mejor resultado (39158.82) que el determinista (41325.91), y un costo promedio también inferior.

La dispersión de los resultados, cuantificada por la desviación estándar y el rango entre el mejor y peor costo, subraya que la calidad de las soluciones del Greedy Estocástico es sensible a la secuencia de decisiones aleatorias. Es de notar que en el Caso 3, ciertas semillas (como la 777 y 314) requirieron un número elevado de intentos (608 y 161 respectivamente) para generar una solución factible, indicando una mayor dificultad para encontrar caminos viables en ese espacio de búsqueda con esas semillas. Esta necesidad de múltiples intentos internos resalta la complejidad que algunas configuraciones de datos pueden presentar a un proceso estocástico simple. Aunque la diversificación es una ventaja, la capacidad del Greedy Estocástico para superar consistentemente al determinista depende de la instancia del problema y de la efectividad de la aleatorización para guiar la búsqueda hacia regiones prometedoras del espacio de soluciones.

2.2. GRASP + Hill Climbing (1 Pista, Base Greedy Determinista)

GRASP (Greedy Randomized Adaptive Search Procedure) es una metaheurística iterativa que consta de dos fases principales: una fase de construcción para generar una solución factible y una fase de búsqueda local para intentar mejorarla. En esta sección, se describe una implementación de GRASP donde la fase de construcción utiliza el algoritmo Greedy Determinista y la fase de búsqueda local se realiza mediante un algoritmo Hill Climbing (HC) del tipo Alguna-Mejora.

2.2.1. GRASP con Greedy Determinista como base y HC Alguna-Mejora (1 Pista)

Descripción del Algoritmo:

Fase de Construcción: La solución inicial para el proceso GRASP se construye utilizando el algoritmo Greedy Determinista (priorizando el menor E_k), tal como se describió en la Sección 2.1.1. Este enfoque produce una única solución determinista de buena calidad como punto de partida. En una implementación GRASP más canónica, esta fase de construcción suele incorporar aleatorización para generar diversas soluciones iniciales (por ejemplo, mediante una Lista de Candidatos Restringidos - RCL), pero para esta variante determinista, se utiliza la salida directa del Greedy.

Fase de Búsqueda Local (Hill Climbing Alguna-Mejora): Una vez obtenida la solución del Greedy Determinista, se aplica un algoritmo de Hill Climbing (HC) para intentar mejorarla. Se ha optado por la variante **Alguna-Mejora (First Improvement)** para la búsqueda local. La justificación para esta elección radica en su eficiencia computacional por iteración en comparación con Mejor-Mejora (Best Improvement), ya que Alguna-Mejora acepta el primer vecino que ofrece una reducción en el costo, sin necesidad de explorar todo el vecindario.

El vecindario para Hill Climbing se define mediante un operador de movimiento simple:

- **Operador de Movimiento:** Para cada avión en la solución actual, se exploran sistemáticamente nuevos tiempos de aterrizaje. Estos tiempos candidatos incluyen:

1. El tiempo preferente del avión (P_k).
2. Tiempos alrededor del tiempo de aterrizaje actual del avión (t_{actual}), explorando deltas de $\pm\delta, \pm2\delta, \dots, \pm4\delta$, donde δ es un paso de tiempo predefinido (ej. ‘paso_tiempo_delta / 2’).
3. Los tiempos de aterrizaje más temprano (E_k) y más tardío (L_k) del avión.

Todos los tiempos candidatos deben estar dentro de la ventana $[E_k, L_k]$ del avión.

Para cada avión, se prueban estos tiempos candidatos en un orden determinista. Si al mover un avión a un nuevo tiempo se genera una solución vecina *factible* y con un *costo total menor* que la solución actual, esta nueva solución se acepta inmediatamente (estrategia Alguna-Mejora) y el proceso de exploración del vecindario se reinicia desde esta nueva solución.

Criterio de Parada para Hill Climbing: La fase de Hill Climbing se detiene cuando se completa una exploración de todos los posibles movimientos para todos los aviones (según la definición del vecindario) sin encontrar ninguna mejora en el costo total de la solución, o si se alcanza un número máximo de iteraciones de HC predefinido. La solución resultante es el óptimo local encontrado por HC.

Resultados y Análisis (1 Pista, GRASP+HC Determinista): Se aplicó el procedimiento GRASP con construcción Greedy Determinista seguido de Hill Climbing Alguna-Mejora a los cuatro casos de prueba para el escenario de una pista. Los resultados, comparando el costo de la solución inicial del Greedy Determinista con el costo final obtenido después de la búsqueda local, se presentan en la Tabla 6.

Caso de Prueba	Costo Greedy Det.	Costo Final HC	Mejora Absoluta	Mejora Relativa (%)
Case 1 (15 aviones)	4600.00	1970.00	2630.00	57.17 %
Case 2 (20 aviones)	7140.00	2060.00	5080.00	71.15 %
Case 3 (44 aviones)	1550.00	1550.00	0.00	0.00 %
Case 4 (100 aviones)	41325.91	22585.94	18739.97	45.35 %

Tabla 6: Resultados de GRASP + HC Alguna-Mejora (1 Pista, Greedy Determinista)

- **Case 1:** El costo se redujo de 4600.00 a 1970.00, lo que representa una mejora del 57.17 %. La Figura 1 muestra la convergencia de Hill Climbing, donde se observa una rápida disminución del costo en las primeras iteraciones de la búsqueda local.
- **Case 2:** Se observó la mayor mejora relativa, con una reducción del costo del 71.15 %, pasando de 7140.00 a 2060.00. El gráfico de convergencia (Figura 2) indica que Hill Climbing fue capaz de encontrar una secuencia de movimientos ventajosos para reestructurar significativamente la solución inicial.

- **Case 3:** En esta instancia, Hill Climbing no logró encontrar ninguna mejora sobre la solución del Greedy Determinista, manteniendo el costo en 1550.00. Esto sugiere que la solución Greedy para este caso es un óptimo local con respecto al vecindario definido por el operador de HC, o que la naturaleza de la instancia es tal que las mejoras locales son difíciles de encontrar desde ese punto de partida. La Figura 3 muestra una línea plana, indicando la ausencia de mejora.
- **Case 4:** Para el caso más grande, con 100 aviones, Hill Climbing logró una reducción considerable del costo, de 41325.91 a 22585.94 (una mejora del 45.35 %). El gráfico de convergencia (Figura 4) muestra un descenso más gradual pero sostenido, lo que es esperable en espacios de búsqueda más grandes y complejos.

En general, la fase de búsqueda local mediante Hill Climbing Alguna-Mejora fue efectiva para refinar las soluciones proporcionadas por el Greedy Determinista, excepto en el Case 3. La magnitud de la mejora varía según la instancia, lo que indica que la 'calidad' de la solución Greedy inicial y la estructura del paisaje de soluciones influyen en la capacidad de HC para encontrar mejores óptimos locales. El uso de Alguna-Mejora permite una exploración rápida, aunque no garantiza alcanzar el mejor vecino en cada paso, puede ser una estrategia eficaz cuando se combina con múltiples reinicios o una fase de construcción aleatorizada (que se explorará en la sección de GRASP Estocástico).

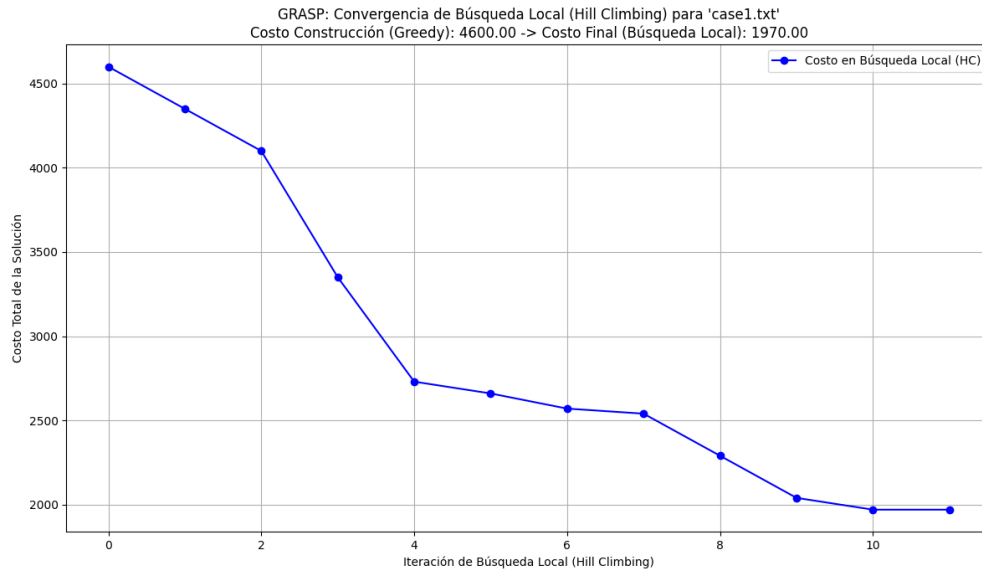


Figura 1: GRASP: Convergencia de Búsqueda Local (Hill Climbing Alguna-Mejora) para Case 1 (1 Pista, Greedy Determinista).

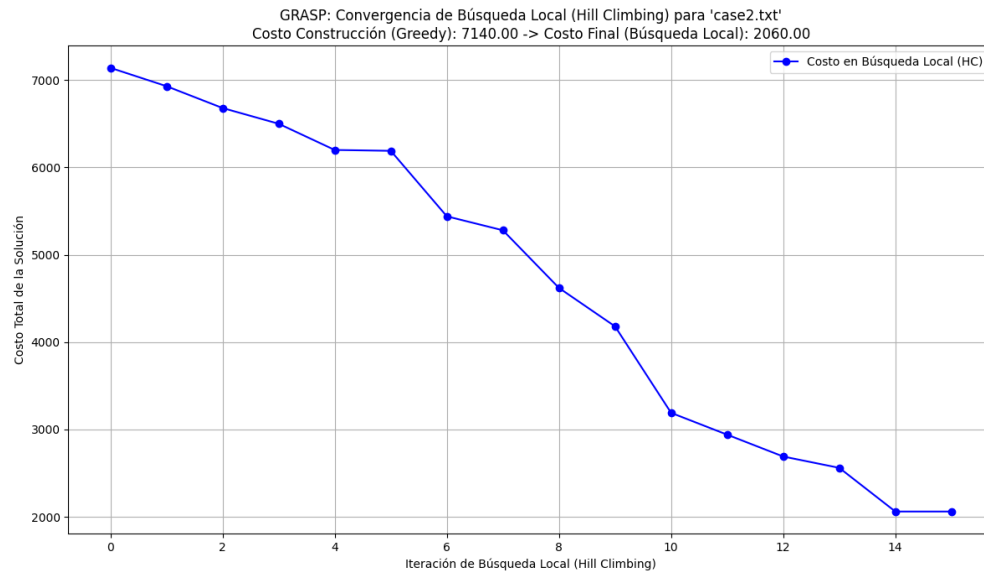


Figura 2: GRASP: Convergencia de Búsqueda Local (Hill Climbing Alguna-Mejora) para Case 2 (1 Pista, Greedy Determinista).

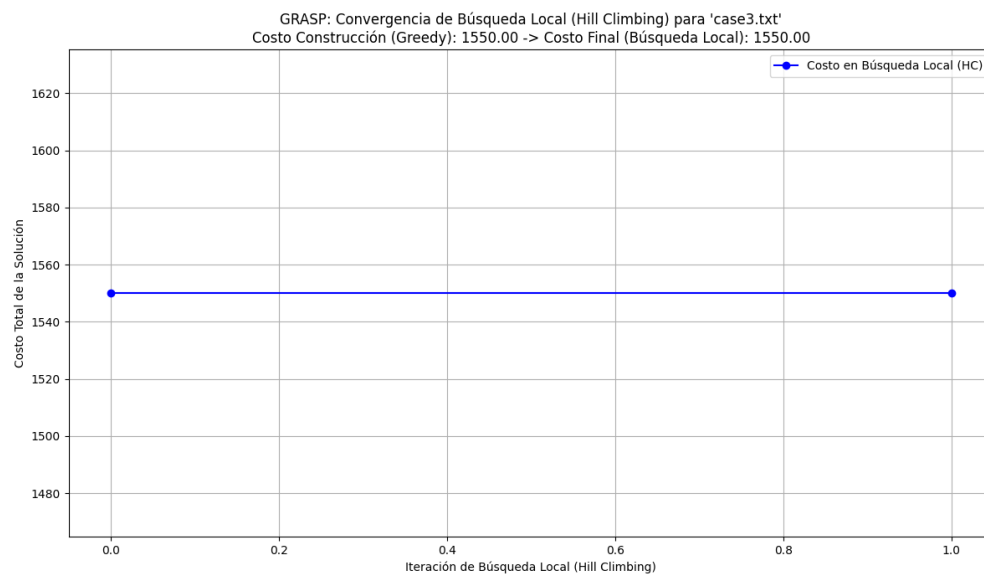


Figura 3: GRASP: Convergencia de Búsqueda Local (Hill Climbing Alguna-Mejora) para Case 3 (1 Pista, Greedy Determinista).

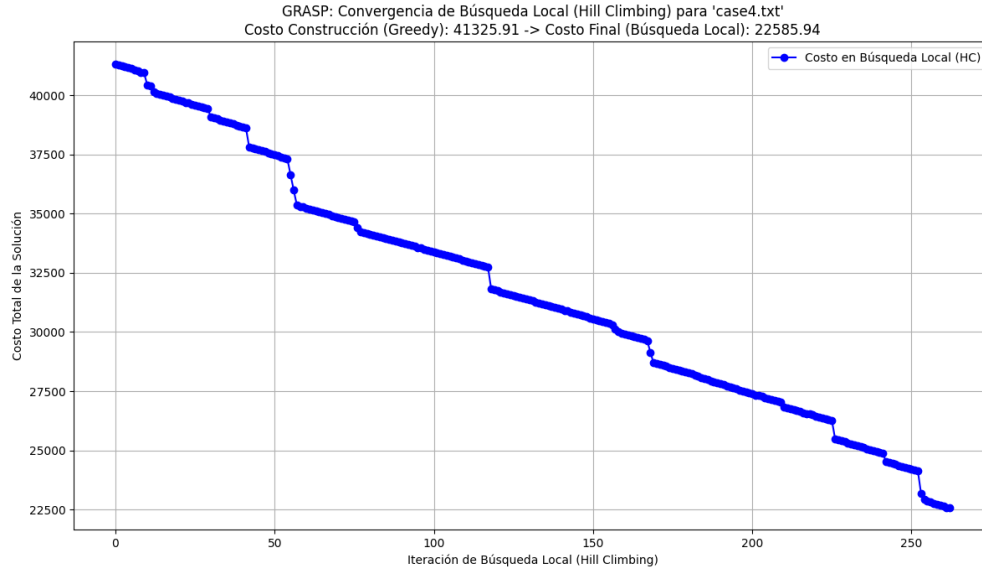


Figura 4: GRASP: Convergencia de Búsqueda Local (Hill Climbing Alguna-Mejora) para Case 4 (1 Pista, Greedy Determinista).

2.3. GRASP + Hill Climbing (1 Pista, Base Greedy Estocástico con Componente de Restart ILS)

En esta implementación de GRASP (Greedy Randomized Adaptive Search Procedure), se busca potenciar la exploración del espacio de soluciones combinando una fase de construcción aleatorizada con una búsqueda local mejorada. Se utiliza el Greedy Estocástico para generar diversos puntos de partida, y a cada uno se le aplica un Hill Climbing (HC) Alguna-Mejora. De manera para cumplir con el requisito de un componente de 'RESTART', se integra una Búsqueda Local Iterada (ILS) después del HC inicial. Esta ILS consiste en realizar múltiples perturbaciones de la solución actual seguidas de nuevas aplicaciones de HC, lo que permite 'reiniciar' la búsqueda desde diferentes puntos del vecindario para escapar de óptimos locales. Todo este proceso (Construcción Estocástica + HC Inicial + ILS con sus reinicios internos) se repite para 10 semillas distintas del Greedy Estocástico, seleccionando al final la mejor solución global encontrada.

2.3.1. GRASP con Greedy Estocástico, HC Alguna-Mejora y Búsqueda Local Iterada con Reinicios (1 Pista)

Descripción del Algoritmo:

El algoritmo GRASP implementado para esta sección sigue una estructura iterativa y multi-arranque, por lo que su funcionamiento sería de la siguiente forma:

1. **Múltiples Puntos de Partida (Arranques Globales GRASP):** El proceso GRASP completo se ejecuta 10 veces de manera independiente. Cada una de estas 10 ejecuciones

comienza con una fase de construcción diferente.

2. **Fase de Construcción Aleatorizada (Greedy Estocástico) por Arranque:** Para cada una de las 10 ejecuciones GRASP, la solución inicial se construye utilizando el algoritmo Greedy Estocástico (descrito en la Sección 2.1.2). A cada una de estas 10 construcciones se le asigna una semilla de aleatoriedad distinta de un conjunto predefinido (42, 123, 7, 99, 500, 777, 2024, 1, 100, 314). Esto asegura que cada uno de los 10 arranques de GRASP comience desde un punto potencialmente diferente en el espacio de soluciones.
3. **Fase de Búsqueda Local Inicial (Hill Climbing Alguna-Mejora):** Inmediatamente después de que el Greedy Estocástico construye una solución inicial para un arranque GRASP, se aplica un algoritmo de Hill Climbing (HC) del tipo **Alguna-Mejora (First Improvement)**. La justificación y el mecanismo de este HC (exploración de vecindario basada en cambios de tiempo de aterrizaje) son los mismos que los detallados en la Sección 2.2.1. Esta fase refina la solución Greedy Estocástica inicial hasta alcanzar un óptimo local.
4. **Componente de Restart mediante Búsqueda Local Iterada (ILS):** A la solución obtenida tras el HC inicial (dentro de una ejecución GRASP particular), se le aplica un procedimiento de Búsqueda Local Iterada (ILS). Este es el componente de ‘RESTART’ solicitado. La ILS funciona de la siguiente manera:
 - Se toma la mejor solución encontrada hasta el momento en la ejecución GRASP actual.
 - **Perturbación:** Esta solución se perturba realizando un número específico (‘num_switches_perturbacion’) de intercambios de tiempos de aterrizaje entre pares de aviones seleccionados al azar. Se verifica que la solución perturbada siga siendo factible.
 - **Nueva Búsqueda Local (HC):** A la solución perturbada (si es factible) se le aplica nuevamente el mismo algoritmo Hill Climbing Alguna-Mejora.
 - Este ciclo de perturbación y HC se repite un número de veces definido por el parámetro ‘num_restarts_ils’. Cada repetición actúa como un ‘reinicio’ de la búsqueda local desde un nuevo punto (perturbado), permitiendo al algoritmo escapar de óptimos locales previamente encontrados.
 - Si alguna de estas nuevas búsquedas locales encuentra una solución mejor que la mejor conocida para la ejecución GRASP actual, esta última se actualiza.

Resultados y Análisis (1 Pista, GRASP+HC Estocástico con Restart ILS): Se realizaron 10 ejecuciones independientes del proceso GRASP descrito. Dentro de cada una de estas 10 ejecuciones, la Búsqueda Local Iterada (ILS) se configuró con ‘num_restarts_ils = 5’ (es decir, 5 ciclos de perturbación y nueva aplicación de HC) y ‘num_switches_perturbacion = 3’.

La Tabla 7 resume los costos de las soluciones iniciales (Greedy Estocástico) y los costos finales obtenidos por el proceso completo GRASP+HC+ILS para cada una de las 10 semillas. También se indica el mejor resultado global encontrado para cada caso de prueba.

Métrica Case 1	Semilla del Greedy Estocástico Inicial										Mejor Global
	42	123	7	99	500	777	2024	1	100	314	
Costo Inicial (Greedy Sto.)	4600	4550	4660	4600	4840	4790	5800	5620	5030	5140	-
Costo Final (GRASP+HC+ILS)	2040	2100	2100	2100	2090	2420	3110	2700	2100	2700	2040
Métrica Case 2	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto.)	7240	8540	7320	8680	9760	7380	7740	7320	8690	8540	-
Costo Final (GRASP+HC+ILS)	1680	2660	1880	4130	3240	1900	2310	2250	1790	3850	1680
Métrica Case 3	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto.)	1990	2054	2218	1944	1944	2448	2054	1944	1880	3006	-
Costo Final (GRASP+HC+ILS)	1990	2054	2218	1944	1944	2448	2054	1944	1880	2722	1880
Métrica Case 4	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto.)	39159	40073	39805	39613	41206	39913	41364	40943	39953	41201	-
Costo Final (GRASP+HC+ILS)	21491	22352	23633	20059	22894	24310	24225	20819	24940	23431	20059

Tabla 7: Resultados de GRASP + HC Alguna-Mejora con Restart ILS (1 Pista, Base Greedy Estocástico, 10 Ejecuciones GRASP)

- **Case 1:** La mejor solución global (costo 2040.00) se consiguió partiendo de la solución Greedy Estocástica generada con la semilla 42 (costo inicial 4600.00). Es interesante notar que este costo es ligeramente superior al obtenido aplicando solo HC a la solución del Greedy Determinista (1970.00). Esto sugiere que, para esta instancia, la calidad de la solución única del Greedy Determinista era un punto de partida muy bueno que el HC simple pudo explotar eficientemente. La Figura 5 ilustra la convergencia de las 10 ejecuciones GRASP.
- **Case 2:** Para este caso, la mejor solución final de 1680.00 (originada por la semilla Greedy 42, costo inicial 7240.00) es mejor que el resultado del GRASP con base determinista (2060.00). Esto destaca cómo la diversificación en los puntos de partida del Greedy Estocástico, combinada con los reinicios de la ILS, puede llevar a regiones de solución superiores. La Figura 6 muestra las trayectorias.
- **Case 3:** El mejor resultado global fue de 1880.00 (semilla Greedy 100, que ya tenía ese costo). Para la mayoría de las semillas, el proceso GRASP+HC+ILS no mejoró las soluciones iniciales del Greedy Estocástico. Ninguna ejecución alcanzó el costo de 1550.00 obtenido por el Greedy Determinista. Esto es una particularidad de esta instancia, lo cual puede ser un óptimo local del que no se logró escapar. Ver Figura 7.
- **Case 4:** La instancia más grande se benefició notablemente de este enfoque estocástico. El mejor costo final fue de 20058.73 (semilla Greedy 99, costo inicial 39613.05), superando el resultado del GRASP de base determinista (22585.94). La Figura 8 muestra la variabilidad y la convergencia de las diferentes ejecuciones GRASP.

El componente de ‘RESTART’ implementado a través de la Búsqueda Local Iterada (ILS) dentro de cada ejecución GRASP juega un papel crucial. Al perturbar la solución actual y reiniciar la búsqueda local desde ese nuevo punto (‘num_restarts_ils = 5’ veces), se aumenta la capacidad del algoritmo para escapar de óptimos locales que un simple HC podría no superar. Combinado con los

múltiples arranques globales (10 semillas Greedy), esta estrategia explora el espacio de soluciones de manera más exhaustiva.

La efectividad del GRASP con base estocástica y reinicios ILS es particularmente evidente en los casos más complejos (Case 2 y Case 4), donde la diversificación en la construcción y la capacidad de escapar de óptimos locales mediante la ILS permitieron encontrar soluciones superiores a las de un enfoque determinista más directo.

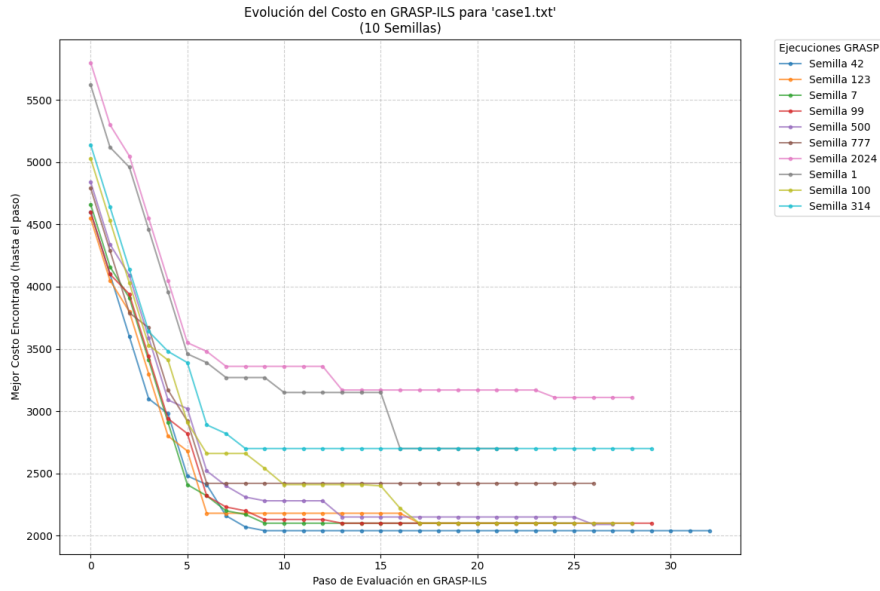


Figura 5: GRASP+ILS con Restart Interno: Evolución del Mejor Costo para Case 1 (1 Pista, 10 Semillas Greedy).

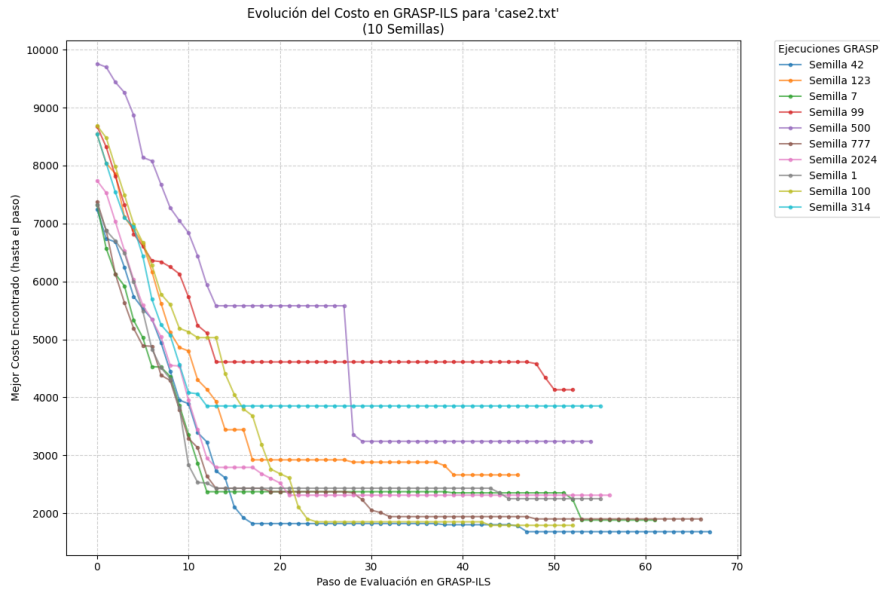


Figura 6: GRASP+ILS con Restart Interno: Evolución del Mejor Costo para Case 2 (1 Pista, 10 Semillas Greedy).

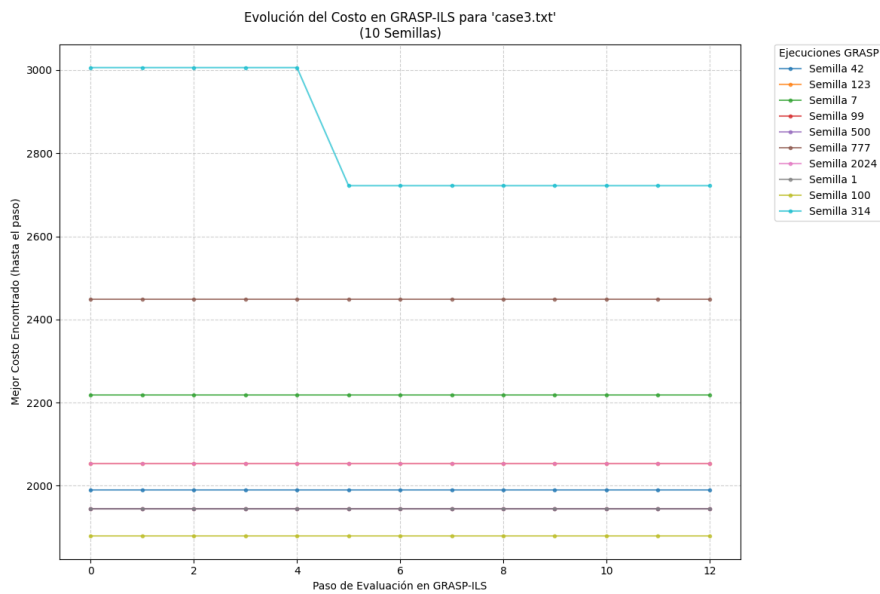


Figura 7: GRASP+ILS con Restart Interno: Evolución del Mejor Costo para Case 3 (1 Pista, 10 Semillas Greedy).

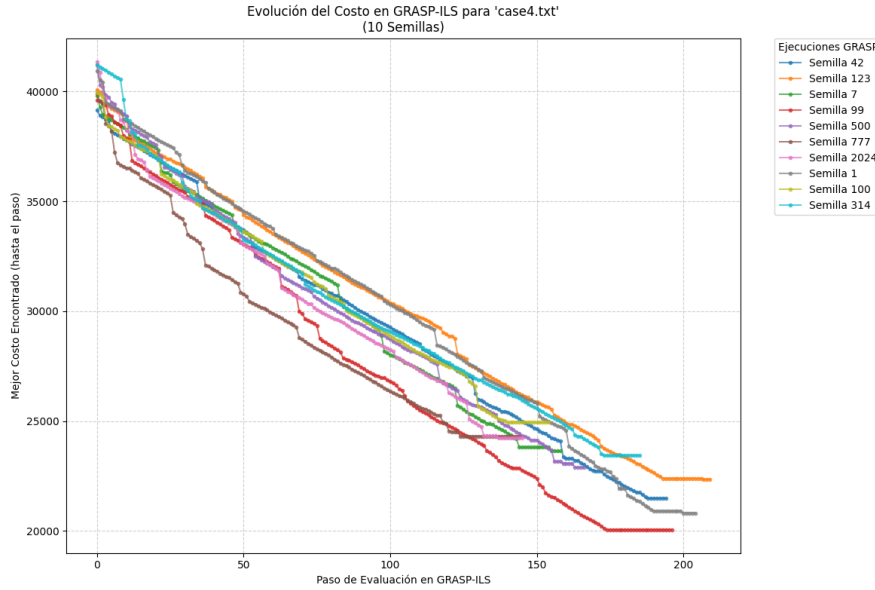


Figura 8: GRASP+ILS con Restart Interno: Evolución del Mejor Costo para Case 4 (1 Pista, 10 Semillas Greedy).

2.4. Simulated Annealing (1 Pista, Greedy Determinista)

El algoritmo de Simulated Annealing (SA) es una metaheurística de optimización global que explora el espacio de soluciones para encontrar un buen mínimo, inspirada en el proceso físico de recocido de metales. Su fortaleza radica en la capacidad de aceptar soluciones peores de forma probabilística, lo que le permite escapar de óptimos locales.

Para abordar la optimización de la secuencia de aterrizajes, se optó por implementar Simulated Annealing en lugar de Tabu Search. Esta elección se fundamenta en varias consideraciones. Primeramente, la principal ventaja de SA para este problema es su inherente capacidad probabilística para aceptar movimientos que empeoran la solución actual, lo cual es especialmente útil en paisajes de soluciones complejos como el Problema de Secuenciación de Aterrizajes (ALP), donde los óptimos locales pueden ser numerosos. Si bien Tabu Search también posee mecanismos para este fin, como la lista tabú, la gestión de dicha lista (su tamaño y la duración del estado tabú de un movimiento) puede requerir una calibración más fina y dependiente de la instancia para ser consistentemente efectiva, introduciendo así parámetros adicionales cuya sintonización puede ser compleja.

En segundo lugar, se consideró la simplicidad relativa en la implementación y conceptualización de los parámetros clave de SA. Aunque SA cuenta con parámetros cruciales como la temperatura inicial, el esquema de enfriamiento y el número de iteraciones por temperatura, estos están bien establecidos en la literatura y su comprensión se ve facilitada por la analogía física del recocido. La aleatoriedad inherente a SA puede, en ciertos escenarios, simplificar la exploración del espacio de soluciones en comparación con la gestión más determinista y memorística que caracteriza a una lista tabú, sobre todo cuando se definen vecindarios de búsqueda complejos.

Adicionalmente, los operadores de vecindad diseñados para este problema —que incluyen el cambio del tiempo de aterrizaje de un avión, el intercambio de tiempos entre dos aviones y la reinserción de un avión en la secuencia— se integran de manera natural y directa en el marco de trabajo de Simulated Annealing. La evaluación de una solución vecina y la subsecuente decisión de aceptarla se basan fundamentalmente en la variación del costo y en la temperatura actual del sistema, sin necesidad de consultar un historial complejo de movimientos previos más allá de la mejor solución encontrada hasta el momento.

Finalmente, la robustez de Simulated Annealing ha sido comprobada en una amplia gama de problemas de optimización combinatoria. Esto incluye numerosos problemas de secuenciación y scheduling, los cuales comparten similitudes estructurales y desafíos con el ALP. Aunque Tabu Search es indudablemente una metaheurística poderosa y efectiva, se estimó que la flexibilidad intrínseca, el probado mecanismo de escape de óptimos locales de SA y una parametrización que puede resultar más intuitiva para una primera aproximación metaheurística avanzada, lo convertían en una elección adecuada y prometedora para este estudio.

En esta sección, se detalla la implementación de Simulated Annealing para el problema de secuenciación de aterrizajes en una única pista. Crucialmente, **la solución inicial para SA se obtiene del algoritmo Greedy Determinista** (descrito en la Sección 2.1.1). Adicionalmente, para las experimentaciones presentadas aquí, todas las ejecuciones de SA con diferentes configuraciones de parámetros utilizan una **semilla de aleatoriedad global fija**, con el objetivo de analizar el impacto de los parámetros de SA de manera controlada.

2.4.1. Diseño del Algoritmo Simulated Annealing (Determinista, 1 Pista)

Solución Inicial: Como se mencionó, la solución de partida para el algoritmo Simulated Annealing es la secuencia de aterrizaje y los tiempos asociados generados por el algoritmo Greedy Determinista (ver Sección 2.1.1).

2.4.2. Experimentación y Resultados (1 Pista, Greedy Determinista)

Se ejecutó el algoritmo Simulated Annealing utilizando la solución del Greedy Determinista como punto de partida. Para cada caso de prueba, se evaluaron 5 configuraciones de parámetros distintas, manteniendo una semilla de aleatoriedad global ('SEMILLA_GLOBAL_REPLICACION = 42') para todas las ejecuciones de SA, con el fin de asegurar la replicabilidad y permitir una comparación justa del efecto de los parámetros de SA. Los parámetros específicos para cada configuración fueron:

- **Config1_HighT_SlowCool:** $T_{\text{inicial}} = 100000,0$, $T_{\text{final}} = 0,1$, $\alpha = 0,995$, $Iter_{\text{temp}} = 100$, $P_{\text{single_move}} = 0,5$, $P_{\text{swap_times}} = 0,3$.
- **Config2_MedT_MedCool:** $T_{\text{inicial}} = 50000,0$, $T_{\text{final}} = 0,1$, $\alpha = 0,99$, $Iter_{\text{temp}} = 150$, $P_{\text{single_move}} = 0,4$, $P_{\text{swap_times}} = 0,3$.
- **Config3_LowT_FastCool_MoreReinsert:** $T_{\text{inicial}} = 10000,0$, $T_{\text{final}} = 0,01$, $\alpha = 0,98$, $Iter_{\text{temp}} = 200$, $P_{\text{single_move}} = 0,3$, $P_{\text{swap_times}} = 0,3$.

- **Config4_HighIter_BalancedOps:** $T_{\text{inicial}} = 50000,0$, $T_{\text{final}} = 0,1$, $\alpha = 0,99$, $Iter_{\text{temp}} = 300$, $P_{\text{single_move}} = 0,33$, $P_{\text{swap_times}} = 0,33$.
- **Config5_FocusSingleMove:** $T_{\text{inicial}} = 50000,0$, $T_{\text{final}} = 0,1$, $\alpha = 0,99$, $Iter_{\text{temp}} = 150$, $P_{\text{single_move}} = 0,7$, $P_{\text{swap_times}} = 0,15$.

La probabilidad del operador ‘Reinsert Plane’ es $1 - P_{\text{single_move}} - P_{\text{swap_times}}$ en cada caso.

Los resultados obtenidos para cada caso de prueba y cada configuración se resumen en la Tabla 8. El costo inicial corresponde al obtenido por el Greedy Determinista.

Configuración SA	Case 1 (Costo Inicial: 4600.00)	Case 2 (Costo Inicial: 7140.00)	Case 3 (Costo Inicial: 1550.00)	Case 4 (Costo Inicial: 41325.91)
Config1_HighT_SlowCool	1620.00	820.00	1550.00	10645.66
Config2_MedT_MedCool	1540.00	880.00	1550.00	9955.89
Config3_LowT_FastCool_MoreReinsert	1540.00	1210.00	1550.00	10132.76
Config4_HighIter_BalancedOps	1510.00	820.00	1550.00	8236.25
Config5_FocusSingleMove	1770.00	840.00	1550.00	11093.12

Tabla 8: Resultados del Simulated Annealing (1 Pista, Greedy Determinista, Semilla Global SA: 42)

A continuación, se presenta un análisis individual para cada caso de prueba, junto con los gráficos de evolución del mejor costo encontrado por el algoritmo SA partiendo de la solución Greedy Determinista.

Análisis para Case 1:

- Costo Inicial (Greedy Determinista): 4600.00
- Mejor Costo SA: 1510.00 (obtenido con Config4_HighIter_BalancedOps)
- Observaciones: Todas las configuraciones de SA lograron una mejora sustancial respecto a la solución Greedy inicial. La ‘Config4_HighIter_BalancedOps’, con una temperatura inicial media, enfriamiento medio, un mayor número de iteraciones por temperatura y probabilidades balanceadas para los operadores de vecindad, fue la que encontró la mejor solución. Las configuraciones ‘Config2’ y ‘Config3’ también mostraron un desempeño competitivo, alcanzando un costo de 1540.00. La ‘Config1’ con temperatura inicial alta y enfriamiento lento también fue efectiva, aunque ligeramente superada. La ‘Config5’, que prioriza el operador ‘SingleMove’, obtuvo el peor resultado entre las configuraciones de SA, aunque aun así significativamente mejor que el Greedy. El gráfico de evolución (Figura 9) muestra cómo las distintas configuraciones exploran el espacio de soluciones. Se observa que ‘Config4’ y ‘Config2’ descienden rápidamente a regiones de bajo costo.

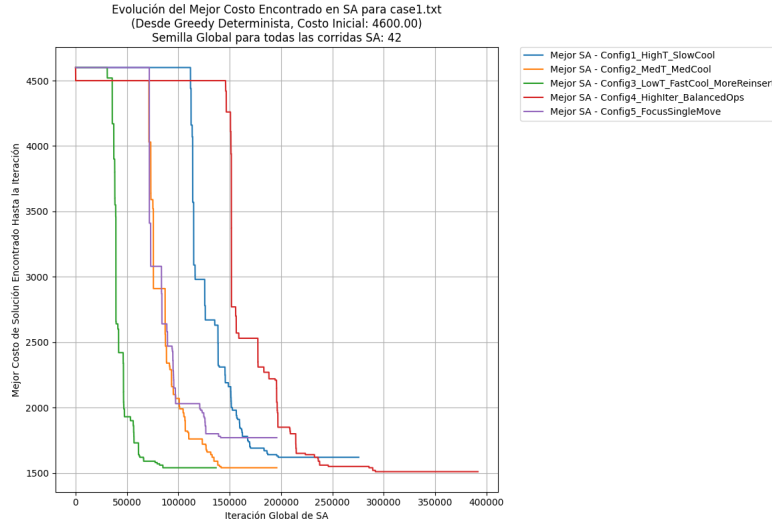


Figura 9: Evolución del mejor costo encontrado por SA para Case 1 (1 Pista, Greedy Det., Semilla SA 42).

Análisis para Case 2:

- Costo Inicial (Greedy Determinista): 7140.00
- Mejor Costo SA: 820.00 (obtenido con Config1_HighT_SlowCool y Config4_HighIter_BalancedOps)
- Observaciones: En este caso, Simulated Annealing demuestra una capacidad de mejora aún más drástica, reduciendo el costo en casi un 90 %. Las configuraciones ‘Config1’ (alta temperatura inicial, enfriamiento lento) y ‘Config4’ (iteraciones altas, operadores balanceados) alcanzaron el mismo costo óptimo de 820.00. ‘Config5’ (foco en ‘SingleMove’) también fue muy efectiva, llegando a 840.00. ‘Config2’ (temperatura y enfriamiento medios) le sigue de cerca con 880.00. La ‘Config3’ (baja temperatura inicial, enfriamiento rápido) fue la menos efectiva, aunque todavía logró una mejora considerable. La Figura 10 ilustra que las configuraciones ‘Config1’ y ‘Config4’ convergen rápidamente a soluciones de muy bajo costo. La alta temperatura inicial de ‘Config1’ parece haber sido beneficiosa para una exploración amplia inicial.

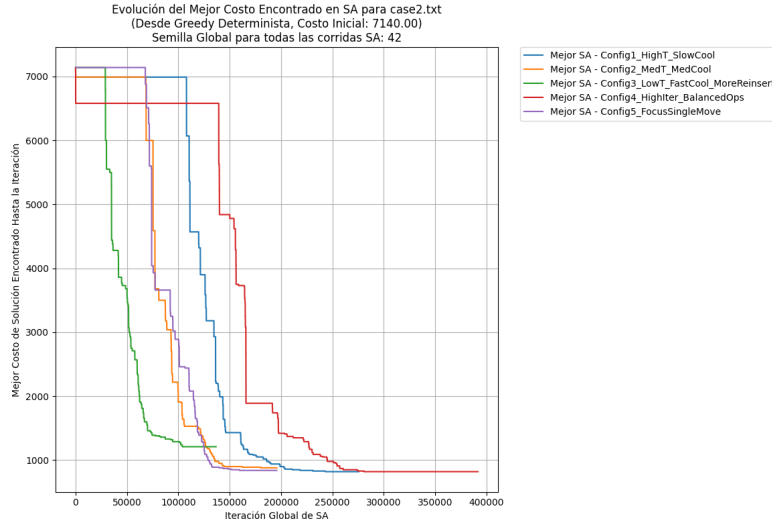


Figura 10: Evolución del mejor costo encontrado por SA para Case 2 (1 Pista, Greedy Det., Semilla SA 42).

Análisis para Case 3:

- Costo Inicial (Greedy Determinista): 1550.00
- Mejor Costo SA: 1550.00 (obtenido con todas las configuraciones)
- Observaciones: Para el Case 3, ninguna de las configuraciones de Simulated Annealing logró mejorar la solución inicial proporcionada por el Greedy Determinista. El costo se mantuvo en 1550.00 para todas las ejecuciones de SA. Esto sugiere que la solución Greedy para este caso particular podría ser un óptimo local muy fuerte o incluso el óptimo global, o que la parametrización y los operadores de SA no fueron adecuados para encontrar mejoras en este paisaje de soluciones específico. Como se observa en la Figura 11, las líneas de costo permanecen planas, indicando que SA no pudo escapar de la solución inicial o encontrar una mejor. La naturaleza de los datos de este caso (por ejemplo, el primer avión con $E_k = P_k = L_k = 0$) podría estar contribuyendo a esta estabilidad.

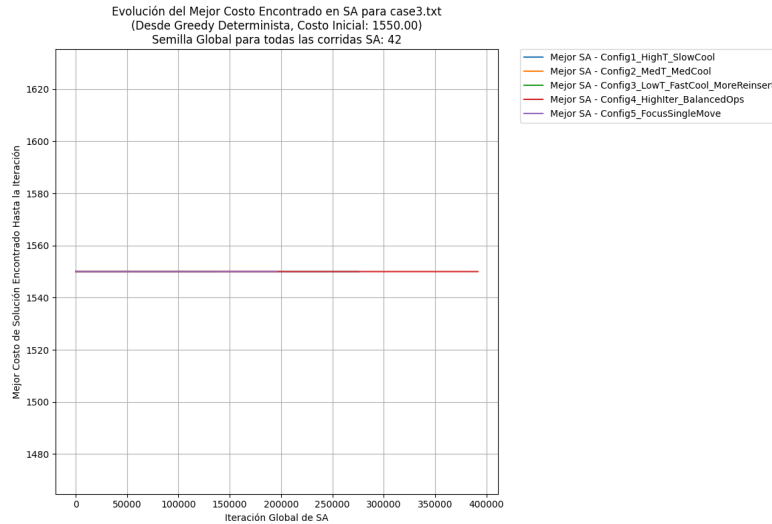


Figura 11: Evolución del mejor costo encontrado por SA para Case 3 (1 Pista, Greedy Det., Semilla SA 42).

Análisis para Case 4:

- Costo Inicial (Greedy Determinista): 41325.91
- Mejor Costo SA: 8236.25 (obtenido con Config4_HighIter_BalancedOps)
- Observaciones: Similar al Case 2, SA logra una reducción de costo buena para el Case 4 (aproximadamente un 80 %). La ‘Config4_HighIter_BalancedOps’ vuelve a destacarse como la mejor, beneficiándose de un mayor número de iteraciones por temperatura y un uso equilibrado de los operadores de vecindad. Le sigue ‘Config2’ (9955.89) y ‘Config3’ (10132.76), mostrando que diferentes combinaciones de temperatura, enfriamiento e iteraciones pueden ser efectivas. ‘Config1’ (10645.66) y ‘Config5’ (11093.12) también mejoraron enormemente la solución Greedy, aunque no tanto como las otras. La Figura 12 muestra una convergencia pronunciada para todas las configuraciones, especialmente para ‘Config4’, que parece explorar más agresivamente y encontrar mejores soluciones más rápidamente. La complejidad del Case 4 (100 aviones) resalta la capacidad de SA para navegar espacios de búsqueda grandes.

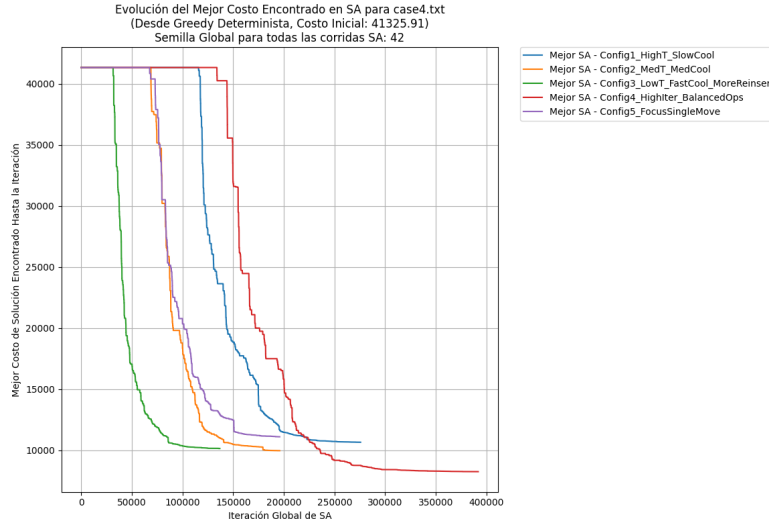


Figura 12: Evolución del mejor costo encontrado por SA para Case 4 (1 Pista, Greedy Det., Semilla SA 42).

Discusión General de Parámetros (1 Pista, Greedy Determinista): A partir de los resultados de las 5 configuraciones aplicadas sobre la solución inicial del Greedy Determinista y utilizando una semilla global fija para SA, se observa que la ‘Config4_HighIter_BalancedOps’ (temperatura inicial media, tasa de enfriamiento media, un número elevado de iteraciones por temperatura y una distribución balanceada de la probabilidad entre los operadores de vecindad) fue la más robusta, obteniendo los mejores resultados en Case 1 y Case 4, y empatando en el mejor resultado para Case 2. Esto sugiere que permitir un mayor número de exploraciones en cada nivel de temperatura (‘iter_per_temp = 300’) es beneficioso, especialmente en problemas más grandes.

La ‘Config1_HighT_SlowCool’ también mostró un buen rendimiento, especialmente en Case 2, donde igualó el mejor costo. Una temperatura inicial alta y un enfriamiento lento permiten una exploración más amplia y una mayor probabilidad de escapar de óptimos locales al principio del proceso.

La ‘Config3_LowT_FastCool_MoreReinsert’ (temperatura inicial baja, enfriamiento rápido) fue consistentemente la menos efectiva o una de las menos efectivas en los casos donde SA encontró mejoras (Case 1, 2, 4). Una temperatura inicial baja podría restringir demasiado la exploración inicial, y un enfriamiento rápido podría llevar a una convergencia prematura.

El caso excepcional fue Case 3, donde ninguna configuración de SA mejoró la solución Greedy Determinista. Esto podría indicar que la solución Greedy ya es de muy alta calidad para esa instancia específica, o que el vecindario y los parámetros de SA no fueron capaces de encontrar una ruta de escape hacia una mejor solución desde ese punto de partida.

La semilla global fija (‘SEMILLA_GLOBAL_REPLICACION = 42’) para las ejecuciones de SA fue fundamental para aislar el efecto de los diferentes conjuntos de parámetros de SA, asegurando que las diferencias observadas en los resultados se debieran a las estrategias de enfriamiento y exploración y no a variaciones en las secuencias de números aleatorios entre configuraciones.

2.5. Simulated Annealing (1 Pista, Greedy Estocástico)

En esta variante de la experimentación con Simulated Annealing (SA), el objetivo es evaluar cómo se desempeña el algoritmo cuando parte de soluciones iniciales generadas por el método Greedy Estocástico (descrito en la Sección 2.1.2). El Greedy Estocástico, al introducir aleatoriedad, tiene la capacidad de generar un conjunto diverso de soluciones iniciales, las cuales pueden estar en diferentes regiones del espacio de búsqueda. Al aplicar SA a cada una de estas, se busca explorar si alguna de estas combinaciones conduce a soluciones finales de mejor calidad en comparación con partir de una única solución determinista.

2.5.1. Diseño del Algoritmo y Metodología Experimental (Estocástico, 1 Pista)

Soluciones Iniciales Múltiples: Se utilizan las 10 soluciones generadas por el algoritmo Greedy Estocástico (una para cada una de las 10 semillas predefinidas: 42, 123, 7, 99, 500, 777, 2024, 1, 100, 314) como puntos de partida para SA. Esto significa que para cada caso de prueba, se ejecutarán 10 series de experimentos de SA, cada serie partiendo de una solución Greedy Estocástica diferente.

Configuraciones de Simulated Annealing: Para cada una de las 10 soluciones iniciales Greedy Estocásticas, se aplican otras 5 configuraciones de parámetros de SA a las que se detallaron en la sección anterior (cuando se partió del Greedy Determinista). Estas son:

- **Config1_Mod_ExploitFocus_From_C2:** $T_{\text{inicial}} = 20000,0$, $T_{\text{final}} = 1,0$, $\alpha = 0,97$, $Iter_{\text{temp}} = 150$, $P_{\text{single_move}} = 0,7$, $P_{\text{swap_times}} = 0,1$.
- **Config2_Fast_LowIter_MedT:** $T_{\text{inicial}} = 20000,0$, $T_{\text{final}} = 1,0$, $\alpha = 0,97$, $Iter_{\text{temp}} = 50$, $P_{\text{single_move}} = 0,4$, $P_{\text{swap_times}} = 0,3$.
- **Config3_VeryFast_VLowIter_LowT:** $T_{\text{inicial}} = 5000,0$, $T_{\text{final}} = 1,0$, $\alpha = 0,95$, $Iter_{\text{temp}} = 30$, $P_{\text{single_move}} = 0,5$, $P_{\text{swap_times}} = 0,25$.
- **Config4_HighIter_BalancedOps:** $T_{\text{inicial}} = 50000,0$, $T_{\text{final}} = 0,1$, $\alpha = 0,99$, $Iter_{\text{temp}} = 300$, $P_{\text{single_move}} = 0,33$, $P_{\text{swap_times}} = 0,33$.
- **Config5_Fast_RapidCool_MedIter:** $T_{\text{inicial}} = 30000,0$, $T_{\text{final}} = 0,5$, $\alpha = 0,96$, $Iter_{\text{temp}} = 70$, $P_{\text{single_move}} = 0,6$, $P_{\text{swap_times}} = 0,2$.

A cada ejecución de SA (combinación de una solución Greedy Estocástica inicial y una configuración de SA) se le asigna una semilla única y determinista para el propio SA, con el fin de asegurar la reproducibilidad de esa ejecución específica. Por ejemplo, la solución Greedy de la semilla 42, al ser procesada por la Config1 de SA, usará una semilla SA_1; al ser procesada por la Config2 de SA, usará una semilla SA_2, y así sucesivamente. Similarmente, la solución Greedy de la semilla 123, al ser procesada por Config1 de SA, usará una semilla SA_X, etc. Esto asegura que cada ejecución individual SA(Greedy_seed_X, Config_Y) sea reproducible.

Mecanismos de SA: La lógica interna del algoritmo Simulated Annealing (generación de vecinos, función de evaluación, esquema de enfriamiento, criterio de aceptación y condición de parada) es la misma que la descrita en la Sección 2.4.1.

2.5.2. Experimentación y Resultados (1 Pista, Greedy Estocástico)

Para cada uno de los 4 casos de prueba, se generaron 10 soluciones iniciales mediante el Greedy Estocástico (utilizando las 10 semillas especificadas). Cada una de estas 10 soluciones sirvió como punto de partida para 5 ejecuciones de Simulated Annealing, cada una con una de las 5 configuraciones de parámetros de SA definidas. Esto resulta en un total de $10 \times 5 = 50$ ejecuciones de SA por cada caso de prueba.

Dada la gran cantidad de ejecuciones, se presenta un resumen de los resultados. Para cada caso de prueba, se reporta:

- El costo promedio de las 10 soluciones iniciales del Greedy Estocástico.
- El mejor costo encontrado por el Greedy Estocástico (entre las 10 semillas).
- El mejor costo final obtenido por SA después de procesar todas las combinaciones (mejor resultado entre las 50 ejecuciones de SA).
- La semilla del Greedy Estocástico y la configuración de SA que condujeron a este mejor resultado global de SA.

La Tabla 9 resume estos hallazgos.

Métrica	Case 1	Case 2	Case 3	Case 4
Costo Promedio Inicial (Greedy Sto.)	5003.00	8121.00	2148.20	40322.84
Mejor Costo Inicial (Greedy Sto.)	4550.00 (Semilla 123)	7240.00 (Semilla 42)	1880.00 (Semilla 100)	39158.82 (Semilla 42)
Mejor Costo Final SA (desde Greedy Sto.)	1500.00	820.00	1550.00	7797.66
Semilla Greedy Inicial para Mejor SA	123	99	Varias*	314
Config. SA para Mejor SA	Config4	Config4	Config4	Config4
Semilla SA Específica para Mejor SA	2400	4400	(e.g., 1400, 2400, etc.)	10400
Costo Inicial Greedy Específico del Mejor SA	4550.00	8680.00	(e.g., 1990.00 de Sem. 42)	41201.24

Tabla 9: Resumen de Resultados: Simulated Annealing (1 Pista, Greedy Estocástico)

A continuación, se muestra para cada caso de prueba un gráfico que ilustra la evolución del costo de SA para la mejor combinación encontrada (es decir, la solución inicial del Greedy Estocástico con su semilla específica, y la configuración de SA que, aplicada a esa partida, produjo el mejor costo global para ese caso de prueba).

Análisis para Case 1 (Estocástico):

- Mejor Costo Inicial (Greedy Estocástico): 4550.00 (con semilla Greedy 123).
- Mejor Costo Final SA: 1500.00.
- Detalles de la Mejor Ejecución SA: Partiendo de la solución Greedy Estocástica (semilla 123, costo 4550.00), la ‘Config4_HighIter_BalancedOps’ de SA (con semilla SA 2400) logró el costo de 1500.00.
- Observaciones: Partir de una de las mejores soluciones del Greedy Estocástico (4550.00 vs 4600.00 del determinista) permitió a SA alcanzar un costo final ligeramente mejor que cuando se partió del Greedy Determinista (1500.00 vs 1510.00). Esto subraya el beneficio de tener un buen punto de partida, aunque diversificado. La Figura 13 muestra la convergencia de las 5 configuraciones de SA para esta mejor solución inicial Greedy.

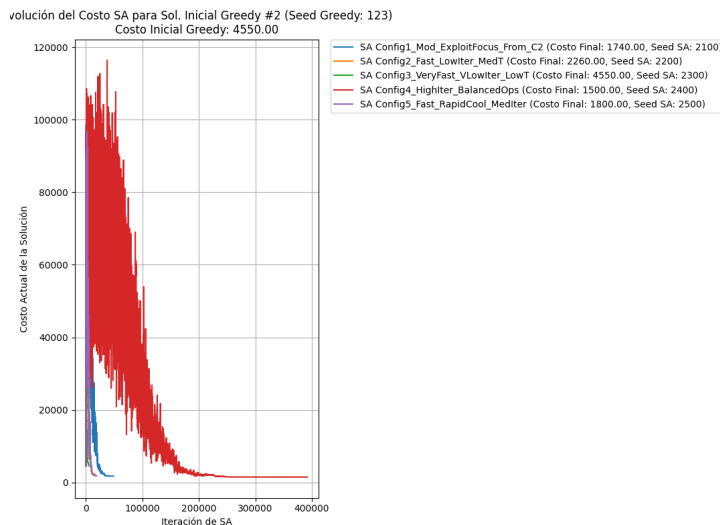


Figura 13: Evolución del costo SA para Case 1, partiendo de la Solución Inicial Greedy Estocástico #2 (Semilla Greedy: 123, Costo Inicial: 4550.00).

Análisis para Case 2 (Estocástico):

- Mejor Costo Inicial (Greedy Estocástico): 7240.00 (con semilla Greedy 42).
- Mejor Costo Final SA: 820.00.
- Detalles de la Mejor Ejecución SA: Partiendo de la solución Greedy Estocástica generada con la semilla 99 (costo inicial 8680.00), la ‘Config4_HighIter_BalancedOps’ de SA (con semilla SA 4400) alcanzó el costo de 820.00.

- Observaciones: Interesantemente, la mejor solución SA no provino de la mejor solución inicial Greedy Estocástica (7240.00), sino de una con costo inicial mayor (8680.00). Esto demuestra la capacidad de SA para mejorar incluso puntos de partida que no son los mejores localmente, y la importancia de la exploración diversificada que ofrece el Greedy Estocástico. El costo final de 820.00 iguala el mejor obtenido cuando se partió del Greedy Determinista. La Figura 14 muestra la evolución.

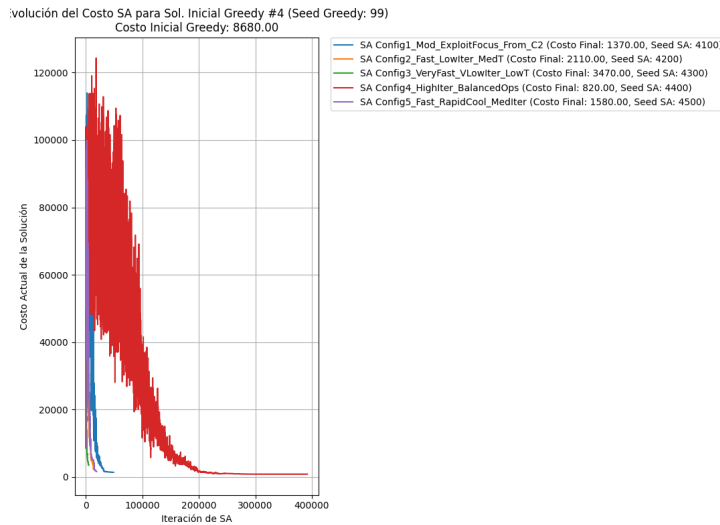


Figura 14: Evolución del costo SA para Case 2, partiendo de la Solución Inicial Greedy Estocástico #4 (Semilla Greedy: 99, Costo Inicial: 8680.00).

Análisis para Case 3 (Estocástico):

- Mejor Costo Inicial (Greedy Estocástico): 1880.00 (con semilla Greedy 100).
- Mejor Costo Final SA: 1550.00.
- Detalles de la Mejor Ejecución SA: Múltiples soluciones iniciales del Greedy Estocástico, al ser procesadas por 'Config4_HighIter_BalancedOps' de SA, convergieron al costo de 1550.00. Por ejemplo, partiendo de la solución Greedy (semilla 42, costo 1990.00), la Config4 de SA (semilla SA 1400) llegó a 1550.00.
- Observaciones: Similar a la experimentación con partida determinista, SA no logró mejorar el costo de 1550.00 para el Case 3, incluso partiendo de diversas soluciones estocásticas (cuyos costos eran peores que 1550.00). Esto refuerza la hipótesis de que 1550.00 es un óptimo local muy robusto o el óptimo global para esta instancia. La Figura 15 muestra la evolución para una de estas combinaciones.

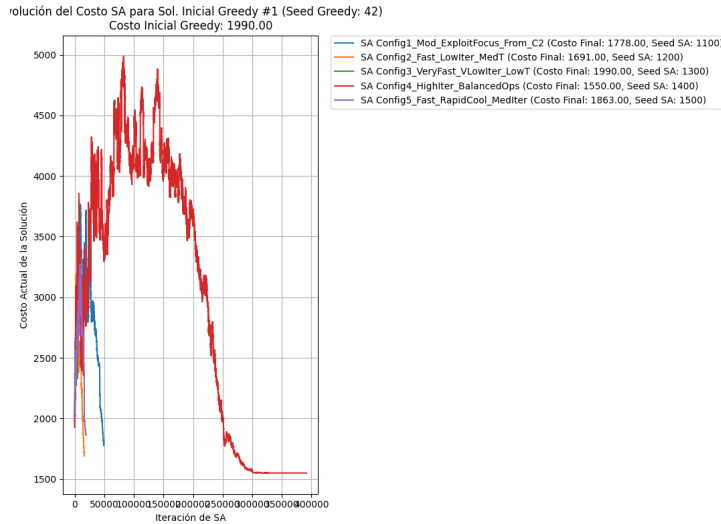


Figura 15: Evolución del costo SA para Case 3, partiendo de la Solución Inicial Greedy Estocástico #1 (Semilla Greedy: 42, Costo Inicial: 1990.00).

Análisis para Case 4 (Estocástico):

- Mejor Costo Inicial (Greedy Estocástico): 39158.82 (con semilla Greedy 42).
- Mejor Costo Final SA: 7797.66.
- Detalles de la Mejor Ejecución SA: Partiendo de la solución Greedy Estocástica (semilla 314, costo inicial 41201.24), la ‘Config4_HighIter_BalancedOps’ de SA (con semilla SA 10400) logró el costo de 7797.66.
- Observaciones: En este caso complejo, utilizar una partida del Greedy Estocástico permitió a SA encontrar una solución notablemente mejor (7797.66) que cuando se partió de la única solución del Greedy Determinista (mejor SA allí fue 8236.25). Nuevamente, la mejor solución final de SA no necesariamente partió de la mejor solución inicial del Greedy Estocástico, resaltando la sinergia entre la diversificación del Greedy Estocástico y la capacidad de búsqueda intensiva del SA. La Figura 16 muestra la evolución.

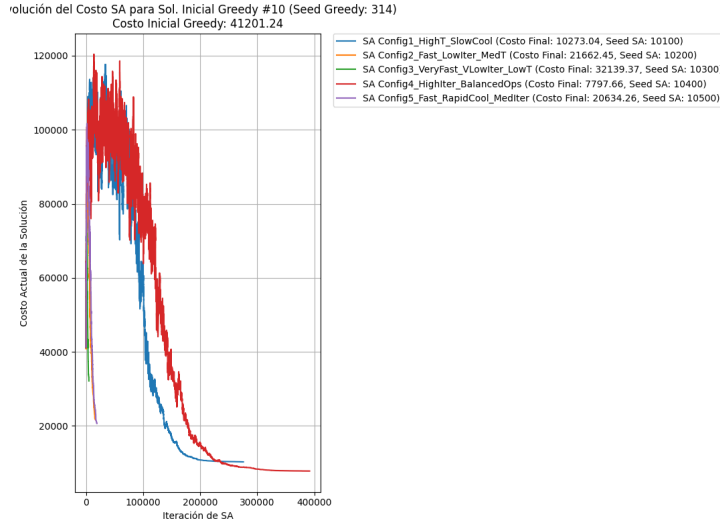


Figura 16: Evolución del costo SA para Case 4, partiendo de la Solución Inicial Greedy Estocástico #10 (Semilla Greedy: 314, Costo Inicial: 41201.24).

Discusión General de Parámetros (1 Pista, Greedy Estocástico): Al utilizar soluciones del Greedy Estocástico como puntos de partida, Simulated Annealing demostró ser capaz de alcanzar costos finales muy competitivos. De manera consistente, la ‘Config4_HighIter_BalancedOps’ de SA emergió como la más efectiva para refinar estas soluciones iniciales, logrando el mejor costo final SA en todos los casos de prueba.

Es notable que para el Case 1 y el Case 4, partir de una solución del Greedy Estocástico (aunque no siempre la de menor costo inicial) y luego aplicar SA, condujo a mejores resultados finales que aplicando SA sobre la solución única del Greedy Determinista. Esto sugiere que la diversidad introducida por el Greedy Estocástico puede ofrecer puntos de entrada más ventajosos para la búsqueda intensiva de SA, permitiéndole alcanzar regiones del espacio de soluciones que no eran accesibles desde la solución Greedy Determinista.

Para el Case 2, el mejor resultado de SA fue el mismo independientemente de si se partió de la solución determinista o de una de las estocásticas. Para el Case 3, el SA no pudo mejorar la marca de 1550.00 en ninguna de las dos estrategias de inicialización.

La estrategia de ejecutar múltiples instancias de Greedy Estocástico para generar diversos puntos de partida para una metaheurística más intensiva como SA parece ser una técnica prometedora, especialmente para los casos más complejos, ya que aumenta la probabilidad de encontrar una solución inicial que esté en la cuenca de atracción de un óptimo global de alta calidad.

3. Escenario: 2 Pistas

En este escenario, las aeronaves pueden ser asignadas a cualquiera de las dos pistas disponibles. Se replicará la estructura de algoritmos y análisis del escenario de 1 pista, adaptando las lógicas para considerar la disponibilidad de dos pistas.

3.1. Algoritmos Greedy (2 Pistas)

Al igual que en el escenario de una pista, la estrategia Greedy busca tomar decisiones localmente óptimas. Con dos pistas, la decisión no solo implica qué avión programar y cuándo, sino también en qué pista hacerlo.

3.1.1. Greedy Determinista (2 Pistas)

Descripción: El algoritmo Greedy Determinista para el escenario de dos pistas busca en cada paso la mejor oportunidad de aterrizaje inmediata entre todos los aviones pendientes y ambas pistas. Para cada avión aún no programado, se evalúa su aterrizaje en la Pista 0 y en la Pista 1. El tiempo de aterrizaje efectivo (T_{efectivo}) para un avión k en una pista p se determina como el mayor valor entre su tiempo más temprano E_k y, si la pista p ya tiene operaciones, el tiempo de aterrizaje del último avión en p más el tiempo de separación τ_{ij} requerido. Solo se consideran opciones válidas aquellas donde T_{efectivo} no excede el tiempo más tardío L_k del avión.

De entre todas las combinaciones factibles (avión, pista, T_{efectivo}) identificadas en una iteración, el algoritmo selecciona la 'mejor' opción global. Esta selección se basa en un criterio lexicográfico: se prefiere la opción con el menor T_{efectivo} ; los empates se resuelven secuencialmente priorizando el menor E_k , luego el menor L_k , seguido por el menor P_k , después el menor ID del avión, y finalmente, el menor índice de la pista (Pista 0 antes que Pista 1 si todo lo demás es idéntico). Una vez seleccionada la mejor opción, el avión correspondiente se asigna a la pista y tiempo elegidos, y se retira de la lista de pendientes. El proceso continúa hasta que todos los aviones han sido programados o no se encuentran más opciones factibles.

Resultados: La tabla 10 muestra el costo total y el orden de aterrizaje global de los aviones obtenido por el algoritmo Greedy Determinista para cada caso de prueba en el escenario de 2 pistas.

Caso de Prueba	Costo Total Obtenido	Orden de Aterrizaje (Secuencia de IDs)
Case 1	6690.00	3, 4, 5, 6, 8, 7, 9, 1, 10, 14, 13, 2, 12, 11, 15
Case 2	9550.00	1, 6, 8, 4, 12, 10, 9, 11, 3, 19, 20, 2, 7, 5, 15, 18, 14, 13, 17, 16
Case 3	7639.00	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44
Case 4	51775.43	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

Tabla 10: Resultados del Greedy Determinista (2 Pistas)

Análisis: Al operar con dos pistas, el algoritmo Greedy Determinista busca asignar cada avión a la pista y al tiempo que resulten en el aterrizaje más temprano posible. Los costos totales obtenidos para este escenario son de 6690.00 para el Caso 1, 9550.00 para el Caso 2, 7639.00 para el Caso 3, y un considerable 51775.43 para el Caso 4.

Similar al escenario de una pista, la variabilidad en los costos entre los casos sugiere una fuerte dependencia de las características de la instancia. El Caso 4, con la mayor cantidad de aviones, sigue presentando el costo más alto. La disponibilidad de una segunda pista introduce una mayor flexibilidad para programar aterrizajes en paralelo, lo que podría mitigar algunos de los conflictos de separación que surgen con una sola pista. Sin embargo, la estrategia Greedy sigue tomando decisiones basadas en la optimización local (el aterrizaje más próximo factible) sin una visión global de cómo estas decisiones afectan el costo total acumulado. La secuencia de aterrizaje (el orden de los IDs) es el resultado de estas elecciones secuenciales y la asignación a la pista que ofrece la 'mejor' opción inmediata.

3.1.2. Greedy Estocástico (2 Pistas)

Descripción: La variante estocástica del algoritmo Greedy para dos pistas introduce aleatoriedad en múltiples puntos para fomentar la exploración de diversas secuencias de aterrizaje. Al inicio de cada ciclo de decisión, la lista de aviones pendientes de programar es barajada. Luego, para cada avión en este orden aleatorio y para cada una de las dos pistas, se evalúa su factibilidad. Si un avión puede iniciar su aterrizaje en una pista (considerando su E_k y la separación τ_{ij} con el último avión en esa pista si se aterrizara en el tiempo base del sistema T), se le permite un margen de tiempo adicional aleatorio, $S_a = \text{random.randint}(0, K_{\text{MAX_SLACK}})$, para proponer un $T_{\text{propuesto}} = T + S_a$. Este

$T_{\text{propuesto}}$ debe respetar el L_k del avión. Todas las combinaciones válidas (avión, pista, $T_{\text{propuesto}}$) forman una lista de candidatos factibles.

Si solo existe un candidato factible, este es seleccionado. Si hay múltiples, se calcula el costo de penalización para cada uno en su respectivo $T_{\text{propuesto}}$. Se asignan pesos inversamente proporcionales a estos costos (menor costo, mayor peso, usando $1/(\text{costo} + \epsilon)$), y se elige un candidato mediante una selección aleatoria ponderada. En ausencia de una clara preferencia por costo (pesos muy bajos o iguales), se recurre a una selección uniforme. El avión, pista y tiempo seleccionados se registran, y el avión se considera programado. Si no se puede programar ningún avión en la iteración actual, el tiempo base T del sistema se avanza al próximo momento factible más temprano o de forma incremental. Cada una de las 10 ejecuciones requeridas utiliza una semilla distinta para el generador de números aleatorios, buscando producir diferentes soluciones.

Resultados: Se realizaron 10 ejecuciones del algoritmo Greedy Estocástico para cada caso de prueba en el escenario de 2 pistas, utilizando las semillas predefinidas: 42, 123, 7, 99, 500, 777, 2024, 1, 100, 314. Las tablas 11 a 14 resumen los estadísticos de estas ejecuciones. El 'Costo Det.' corresponde al valor del Greedy Determinista (2 Pistas).

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
6690.00	4980.00	5780.00	5492.00	302.99

Tabla 11: Resumen Resultados Greedy Estocástico (2 Pistas) - Case 1

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
9550.00	7940.00	8440.00	8192.00	169.03

Tabla 12: Resumen Resultados Greedy Estocástico (2 Pistas) - Case 2

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
7639.00	7432.00	7507.00	7471.30	24.07

Tabla 13: Resumen Resultados Greedy Estocástico (2 Pistas) - Case 3

Costo Det.	Mejor Costo Sto.	Peor Costo Sto.	Promedio Sto.	Desv. Estándar Sto.
51775.43	50894.90	51331.42	51160.35	174.75

Tabla 14: Resumen Resultados Greedy Estocástico (2 Pistas) - Case 4

Análisis: La implementación del Greedy Estocástico para dos pistas introduce elementos aleatorios tanto en el orden de evaluación de los aviones como en la selección de un ligero margen de tiempo adicional para el aterrizaje. Esto permite una exploración más amplia del espacio de soluciones en comparación con su contraparte determinista.

Al observar los resultados de las 10 ejecuciones por caso, se destaca que para todos los casos (1, 2, 3 y 4), el mejor costo obtenido por el Greedy Estocástico fue consistentemente inferior al costo del Greedy Determinista para dos pistas. Por ejemplo, en el Caso 1, el estocástico logró un costo mínimo de 4980.00 frente a los 6690.00 del determinista; similarmente, en el Caso 4, la mejora fue de 50894.90 (estocástico) contra 51775.43 (determinista). Esto sugiere que la capacidad de diversificar las asignaciones de aviones a pistas y los ligeros ajustes en los tiempos de aterrizaje, aunque guiados por una heurística local, pueden efectivamente descubrir configuraciones más económicas.

La variabilidad, indicada por la desviación estándar y la diferencia entre el mejor y peor costo, es presente pero parece ser relativamente contenida en casos como el Caso 3 (DE: 24.07) en comparación con, por ejemplo, el Caso 1 (DE: 302.99). Esto podría implicar que para ciertas estructuras de problemas (como el Caso 3), aunque el estocástico mejora al determinista, las diferentes trayectorias aleatorias tienden a converger a soluciones de calidad similar. En contraste, para el Caso 1, la aleatoriedad parece tener un impacto más disperso. La estrategia de barajar los aviones pendientes antes de evaluar candidatos y la adición de un intervalo aleatorio al tiempo de aterrizaje parece ser componentes efectivos para escapar de las secuencias rígidas impuestas por el método puramente determinista, llevando a mejoras promedio en el costo total para la mayoría de los casos al operar con dos pistas.

3.2. GRASP + Hill Climbing (2 Pistas, Base Greedy Determinista)

La metaheurística GRASP (Greedy Randomized Adaptive Search Procedure) se adapta al escenario de dos pistas. En esta variante, la fase de construcción se basa en el algoritmo Greedy Determinista para dos pistas, y la fase de búsqueda local se realiza mediante Hill Climbing (HC) del tipo Alguna-Mejora, explorando movimientos tanto dentro de la misma pista como entre pistas.

3.2.1. GRASP con Greedy Determinista como base y HC Alguna-Mejora (2 Pistas)

Descripción del Algoritmo:

Fase de Construcción (2 Pistas): La solución inicial para el proceso GRASP se genera utilizando el algoritmo Greedy Determinista para dos pistas, tal como se describió en la Sección 3.1.1. Este algoritmo asigna cada avión a una de las dos pistas y a un tiempo de aterrizaje, priorizando el menor tiempo efectivo de aterrizaje y utilizando criterios lexicográficos para desempates. El resultado es una única solución determinista que sirve como punto de partida para la búsqueda local.

Fase de Búsqueda Local (Hill Climbing Alguna-Mejora para 2 Pistas): Una vez obtenida la solución del Greedy Determinista para dos pistas, se aplica un algoritmo de Hill Climbing (HC) para intentar mejorarla. Se mantiene la variante **Alguna-Mejora (First Improvement)** por su eficiencia.

El vecindario para Hill Climbing en el contexto de dos pistas se define mediante los siguientes operadores de movimiento, aplicados a cada avión en la solución actual:

- **Operador de Cambio de Tiempo (Misma Pista):** Para cada avión, se exploran nuevos

tiempos de aterrizaje en su pista asignada actualmente. Los tiempos candidatos incluyen:

1. El tiempo preferente del avión (P_k).
2. Tiempos alrededor del tiempo de aterrizaje actual del avión (t_{actual}), explorando deltas (ej. $\pm\delta, \pm2\delta, \dots$).
3. Los tiempos de aterrizaje más temprano (E_k) y más tardío (L_k) del avión.

Todos los tiempos candidatos deben estar dentro de la ventana $[E_k, L_k]$ del avión y respetar las separaciones con otros aviones en la misma pista.

- **Operador de Cambio de Pista:** Para cada avión, se evalúa la posibilidad de moverlo a la *otra* pista disponible. Al considerar mover un avión k de su pista original p_{orig} a la pista alternativa p_{alt} , se calcula el tiempo de aterrizaje más temprano factible en p_{alt} . Este cálculo considera el E_k del avión y los tiempos de separación requeridos con todos los aviones ya programados en p_{alt} . También se puede considerar el tiempo preferente P_k si resulta factible en la nueva pista.

Para cada avión, se prueban estos movimientos. Si un movimiento (ya sea cambio de tiempo en la misma pista o cambio a la otra pista con un tiempo ajustado) genera una solución vecina *factible* y con un *costo total menor*, esta nueva solución se acepta inmediatamente (estrategia Alguna-Mejora), y el proceso de exploración del vecindario se reinicia desde esta nueva solución mejorada.

Criterio de Parada para Hill Climbing (2 Pistas): La fase de Hill Climbing se detiene cuando se completa una exploración de todos los posibles movimientos para todos los aviones (según la definición del vecindario para dos pistas) sin encontrar ninguna mejora en el costo total de la solución, o si se alcanza un número máximo de iteraciones de HC predefinido.

Resultados y Análisis (2 Pistas, GRASP+HC Determinista): Se aplicó el procedimiento GRASP con construcción Greedy Determinista para dos pistas, seguido de Hill Climbing Alguna-Mejora, a los cuatro casos de prueba. Los resultados, que comparan el costo de la solución inicial del Greedy Determinista (2 Pistas) con el costo final obtenido después de la búsqueda local, se presentan en la Tabla 15.

Caso de Prueba	Costo Greedy Det. (2P)	Costo Final HC (2P)	Mejora Absoluta	Mejora Relativa (%)
Case 1 (15 aviones)	6690.00	390.00	6300.00	94.17 %
Case 2 (20 aviones)	9550.00	1350.00	8200.00	85.86 %
Case 3 (44 aviones)	7639.00	0.00	7639.00	100.00 %
Case 4 (100 aviones)	51775.43	12026.19	39749.24	76.77 %

Tabla 15: Resultados de GRASP + HC Alguna-Mejora (2 Pistas, Greedy Determinista)

- **Case 1:** El costo se redujo drásticamente de 6690.00 a 390.00, lo que representa una notable mejora del 94.17 %. La Figura 17 ilustra la convergencia de Hill Climbing, mostrando una

rápida disminución del costo. La capacidad de HC para reasignar aviones entre pistas y ajustar tiempos fue fundamental.

- **Case 2:** Se observó una mejora también muy sustancial, con una reducción del costo del 85.86 %, pasando de 9550.00 a 1350.00. El gráfico de convergencia (Figura 18) evidencia cómo Hill Climbing logró reconfigurar la solución inicial de manera efectiva.
- **Case 3:** Este caso presentó el resultado más impactante, donde Hill Climbing logró reducir el costo de 7639.00 a 0.00, una mejora del 100 %. Esto indica que la solución inicial del Greedy, aunque factible, estaba lejos de ser óptima, y HC pudo encontrar una configuración sin penalizaciones. La Figura 19 muestra esta convergencia hacia un costo cero. Es importante destacar que un costo de cero significa que todos los aviones aterrizaron exactamente en su tiempo preferente P_k y todas las restricciones se cumplieron.
- **Case 4:** Para la instancia más grande, con 100 aviones, Hill Climbing también demostró una gran capacidad de mejora, reduciendo el costo de 51775.43 a 12026.19 (una mejora del 76.77 %). El gráfico de convergencia (Figura 20) muestra un descenso sostenido, reflejando la complejidad de optimizar una mayor cantidad de asignaciones y tiempos.

En general, para el escenario de dos pistas, la fase de búsqueda local mediante Hill Climbing Alguna-Mejora fue consistentemente muy efectiva para refinar las soluciones generadas por el Greedy Determinista. Las mejoras obtenidas son mayores que las vistas en el escenario de una pista para los mismos casos. Esto sugiere que la flexibilidad adicional que ofrecen las dos pistas, combinada con la capacidad de HC para explorar movimientos de cambio de pista y reajuste de tiempos, permite escapar de óptimos locales de mayor costo y alcanzar soluciones de calidad superior.

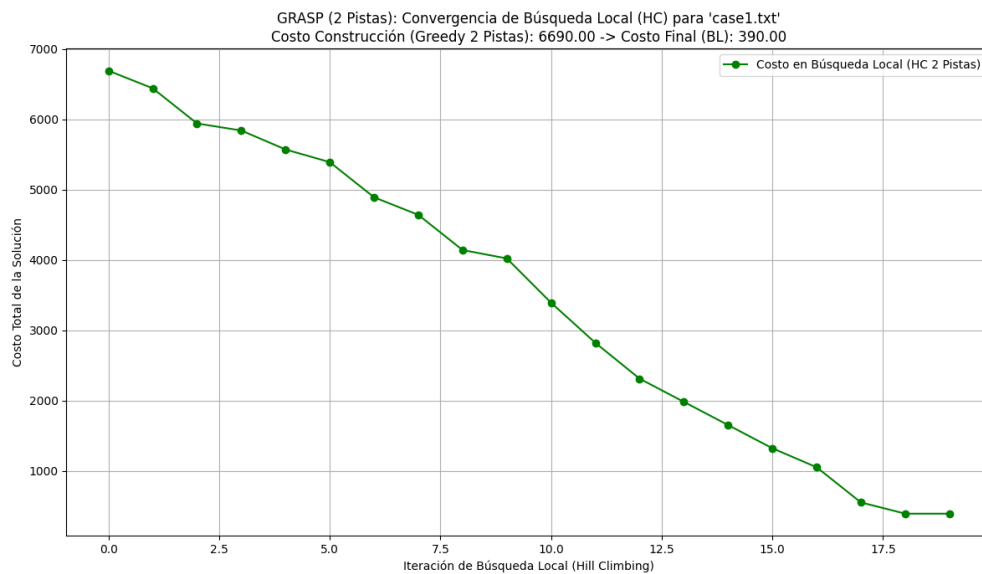


Figura 17: GRASP (2 Pistas): Convergencia de Búsqueda Local (HC Alguna-Mejora) para Case 1 (Greedy Determinista).

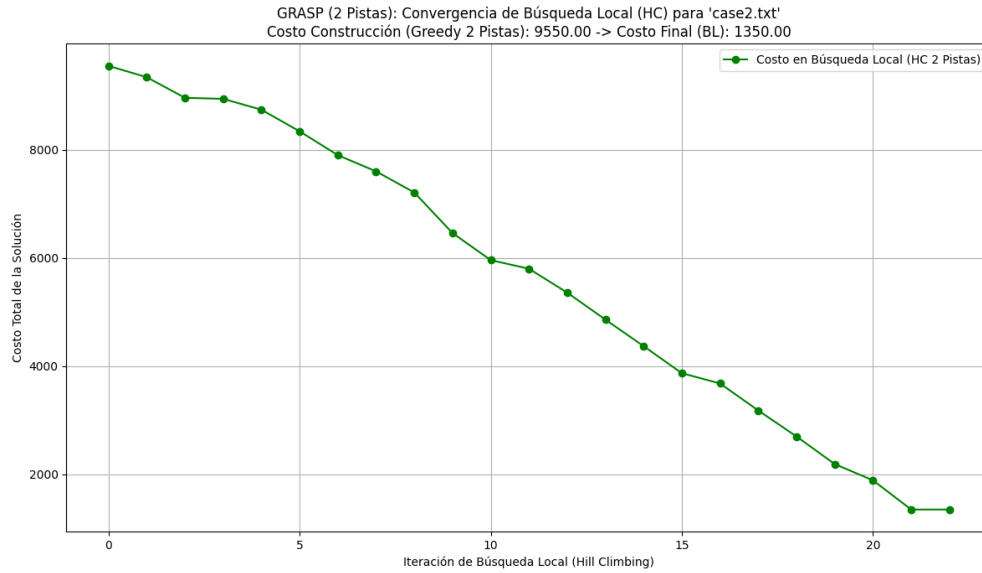


Figura 18: GRASP (2 Pistas): Convergencia de Búsqueda Local (HC Alguna-Mejora) para Case 2 (Greedy Determinista).

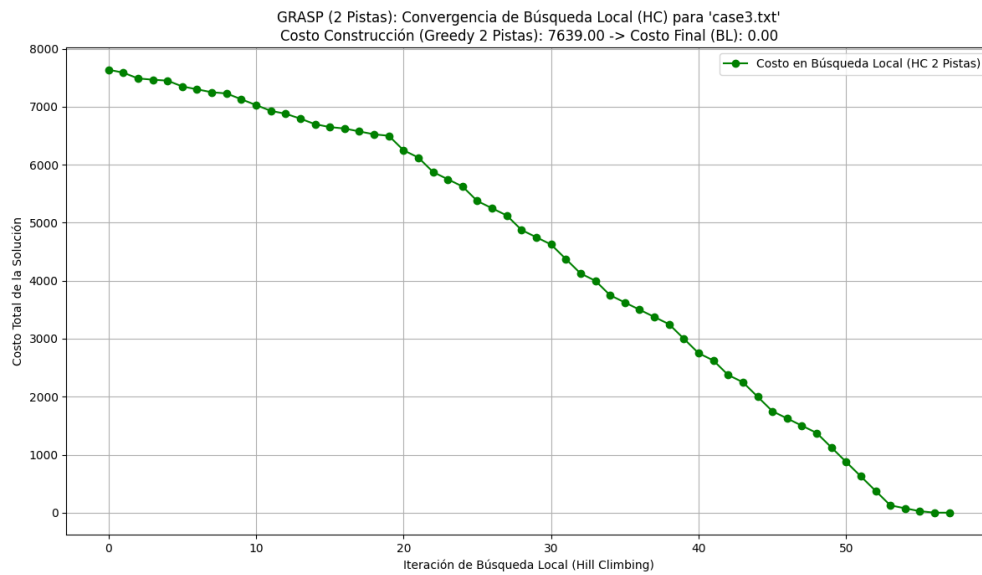


Figura 19: GRASP (2 Pistas): Convergencia de Búsqueda Local (HC Alguna-Mejora) para Case 3 (Greedy Determinista).

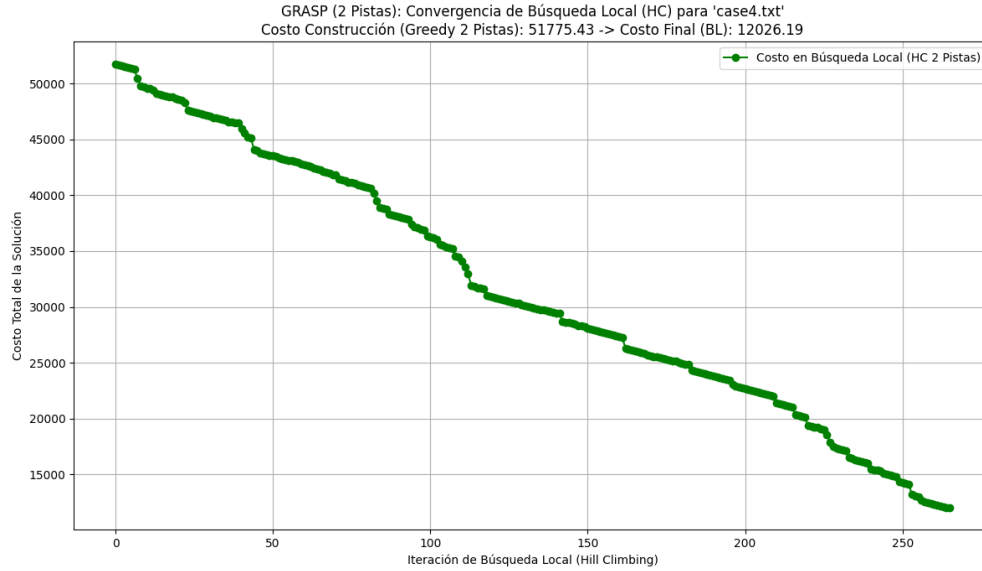


Figura 20: GRASP (2 Pistas): Convergencia de Búsqueda Local (HC Alguna-Mejora) para Case 4 (Greedy Determinista).

3.3. GRASP + Hill Climbing (2 Pistas, Base Greedy Estocástico con Componente de Restart ILS)

Para el escenario de dos pistas, esta implementación de GRASP (Greedy Randomized Adaptive Search Procedure) utiliza el Greedy Estocástico para dos pistas en su fase de construcción, generando así diversos puntos de partida. A cada solución construida se le aplica un Hill Climbing (HC) Alguna-Mejora adaptado para dos pistas. Para cumplir con el requisito de un componente de 'RESTART' y potenciar la exploración, se integra una Búsqueda Local Iterada (ILS) después del HC inicial. La ILS realiza perturbaciones en la solución actual seguidas de nuevas aplicaciones de HC, permitiendo escapar de óptimos locales. Este proceso completo (Construcción Estocástica + HC Inicial + ILS con sus reinicios internos) se repite para 10 semillas distintas del Greedy Estocástico, seleccionando al final la mejor solución global encontrada.

3.3.1. GRASP con Greedy Estocástico, HC Alguna-Mejora y Búsqueda Local Iterada con Reinicios (2 Pistas)

Descripción del Algoritmo:

El algoritmo GRASP para dos pistas sigue una estructura iterativa y multi-arranque:

1. **Múltiples Puntos de Partida (Arranques Globales GRASP):** El proceso GRASP completo se ejecuta 10 veces de forma independiente. Cada una de estas ejecuciones utiliza una semilla diferente para la fase de construcción.

2. **Fase de Construcción Aleatorizada (Greedy Estocástico para 2 Pistas) por Arranque:** Para cada una de las 10 ejecuciones GRASP, la solución inicial se construye utilizando el algoritmo Greedy Estocástico para dos pistas (descrito en la Sección 3.1.2). Se utilizan las 10 semillas predefinidas (42, 123, 7, 99, 500, 777, 2024, 1, 100, 314) para asegurar la diversidad en los puntos de partida. Este constructor estocástico ya considera la asignación a una de las dos pistas disponibles.
3. **Fase de Búsqueda Local Inicial (Hill Climbing Alguna-Mejora para 2 Pistas):** Tras la construcción de una solución inicial mediante el Greedy Estocástico, se aplica un algoritmo de Hill Climbing (HC) del tipo **Alguna-Mejora (First Improvement)**. Este HC es el mismo descrito en la Sección 3.2.1, capaz de explorar cambios de tiempo en la misma pista y cambios de avión a la pista alternativa. Esta fase refina la solución inicial hasta un óptimo local.
4. **Componente de Restart mediante Búsqueda Local Iterada (ILS para 2 Pistas):** A la solución obtenida tras el HC inicial, se le aplica un procedimiento de Búsqueda Local Iterada (ILS).
 - Se toma la mejor solución encontrada hasta el momento en la ejecución GRASP actual.
 - **Perturbación (2 Pistas):** Esta solución se perturba realizando un número específico ('num_switches_perturbacion') de intercambios de las asignaciones completas (tiempo y pista) entre pares de aviones seleccionados al azar. Se verifica que la solución perturbada siga siendo factible respecto a las dos pistas.
 - **Nueva Búsqueda Local (HC para 2 Pistas):** A la solución perturbada (si es factible) se le aplica nuevamente el mismo algoritmo Hill Climbing Alguna-Mejora para dos pistas.
 - Este ciclo de perturbación y HC se repite un número de veces definido por 'num_restarts_ils'.
 - Si alguna de estas nuevas búsquedas locales encuentra una solución mejor que la mejor conocida para la ejecución GRASP actual, esta última se actualiza.

El mejor resultado obtenido entre las 10 ejecuciones GRASP independientes constituye la solución final del algoritmo.

Resultados y Análisis (2 Pistas, GRASP+HC Estocástico con Restart ILS): Se realizaron 10 ejecuciones independientes del proceso GRASP descrito para el escenario de dos pistas. Dentro de cada ejecución, la Búsqueda Local Iterada (ILS) se configuró con 'num_restarts_ils = 5' y 'num_switches_perturbacion = 3'.

La Tabla 16 resume los costos de las soluciones iniciales (Greedy Estocástico para 2 pistas) y los costos finales obtenidos por el proceso completo GRASP+HC+ILS para cada una de las 10 semillas. También se indica el mejor resultado global encontrado para cada caso de prueba.

Métrica Case 1 (2P)	Semilla del Greedy Estocástico Inicial (2P)										Mejor Global
	42	123	7	99	500	777	2024	1	100	314	
Costo Inicial (Greedy Sto. 2P)	5720	5580	5480	5060	5680	5780	5210	4980	5700	5730	-
Costo Final (GRASP+ILS 2P)	270	360	420	630	360	270	270	600	870	330	270
Métrica Case 2 (2P)	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto. 2P)	8100	8370	7940	8370	8050	8310	8440	8090	8180	8070	-
Costo Final (GRASP+ILS 2P)	680	960	270	390	300	570	630	300	960	390	270
Métrica Case 3 (2P)	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto. 2P)	7465	7479	7440	7503	7460	7507	7479	7461	7487	7432	-
Costo Final (GRASP+ILS 2P)	0	0	0	0	0	0	0	0	0	0	0
Métrica Case 4 (2P)	42	123	7	99	500	777	2024	1	100	314	Mejor Global
Costo Inicial (Greedy Sto. 2P)	51317	51259	51019	50966	51032	51331	51328	51260	50895	51196	-
Costo Final (GRASP+ILS 2P)	7290	9160	6396	6896	8523	10987	7114	9320	8024	7533	6396

Tabla 16: Resultados de GRASP + HC Alguna-Mejora con Restart ILS (2 Pistas, Base Greedy Estocástico, 10 Ejecuciones GRASP)

- **Case 1 (2 Pistas):** El mejor costo global encontrado fue de 270.00, obtenido por múltiples semillas de construcción del Greedy Estocástico (42, 777, 2024) después del proceso GRASP+ILS. Este valor es significativamente inferior al costo inicial promedio del Greedy Estocástico (aprox. 5500) y también mejora el resultado del GRASP con base determinista para 2 pistas (390.00). Esto subraya cómo la diversificación de la construcción estocástica y la robustez de la ILS pueden hallar soluciones aún mejores. La Figura 21 muestra la convergencia.
- **Case 2 (2 Pistas):** Se alcanzó un excelente costo final de 270.00, partiendo de la solución Greedy Estocástica generada con la semilla 7 (costo inicial 7940.00). Este resultado es considerablemente mejor que el obtenido con la base determinista (1350.00), mostrando el beneficio de la aleatorización en la construcción y la capacidad de ILS para explotar diferentes regiones del espacio de búsqueda. Ver Figura 22.
- **Case 3 (2 Pistas):** De manera notable, para todas las 10 semillas, el proceso GRASP+HC+ILS logró reducir el costo a 0.00. Esto replica el hallazgo del GRASP con base determinista para este caso, indicando que, con dos pistas, es consistentemente posible encontrar una planificación sin penalizaciones para esta instancia, independientemente del punto de partida generado por el Greedy Estocástico (cuyos costos iniciales rondaban los 7400-7500). La Figura 23 ilustraría una rápida convergencia a cero para todas las semillas.
- **Case 4 (2 Pistas):** Para la instancia más grande, el mejor costo global fue de 6396.17 (aproximado), originado por la semilla Greedy 7 (costo inicial aprox. 51019). Este es el mejor resultado obtenido hasta ahora para este caso en dos pistas, superando marcadamente al GRASP de base determinista (12026.19). Esto resalta la efectividad de la combinación de construcción diversificada y búsqueda local intensificada con reinicios para problemas complejos con dos pistas. La Figura 24 muestra la variabilidad y la mejora.

El componente de ‘RESTART’ implementado a través de la ILS es importante, ya que perturba las soluciones y reinicia el HC desde nuevos puntos, por lo que el algoritmo evita estancarse pre-

maturamente en óptimos locales. Para el escenario de dos pistas, donde el espacio de soluciones es más amplio debido a la opción de asignación de pista, esta capacidad de exploración es aún más ventajosa. La combinación de múltiples arranques GRASP (10 semillas) con la ILS interna en cada uno, genera una búsqueda potente y diversificada, llevando a soluciones de muy alta calidad y confirmando la optimización total en el Case 3.

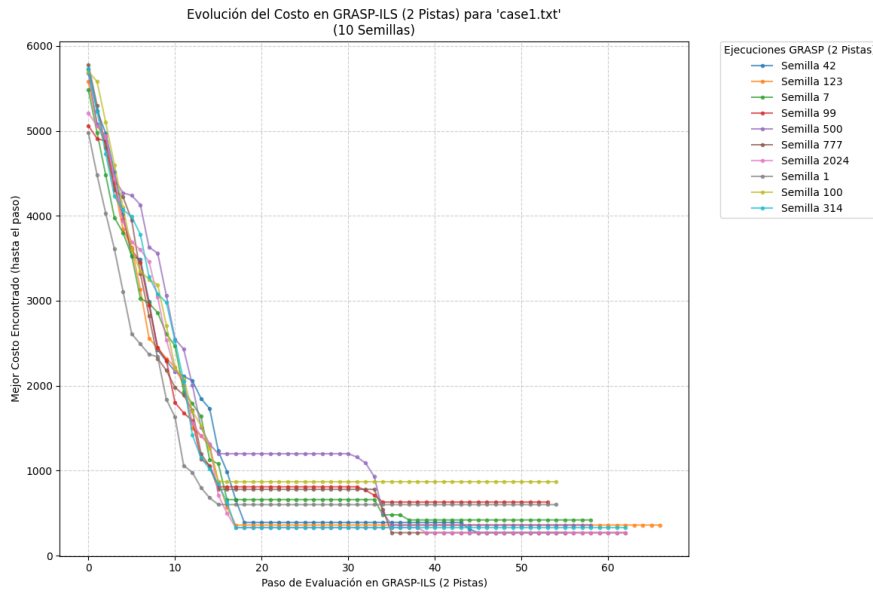


Figura 21: GRASP+ILS (2 Pistas) con Restart Interno: Evolución del Mejor Costo para Case 1 (10 Semillas Greedy).

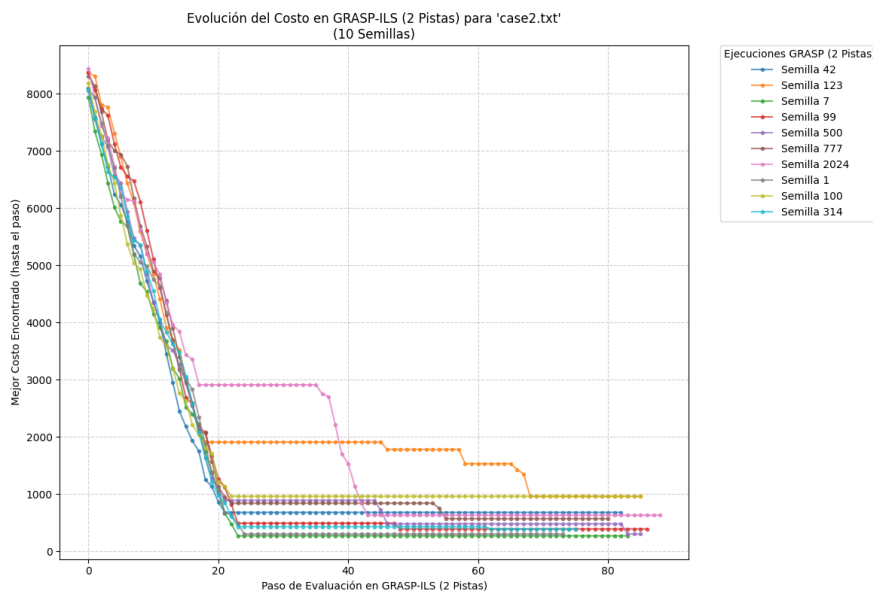


Figura 22: GRASP+ILS (2 Pistas) con Restart Interno: Evolución del Mejor Costo para Case 2 (10 Semillas Greedy).

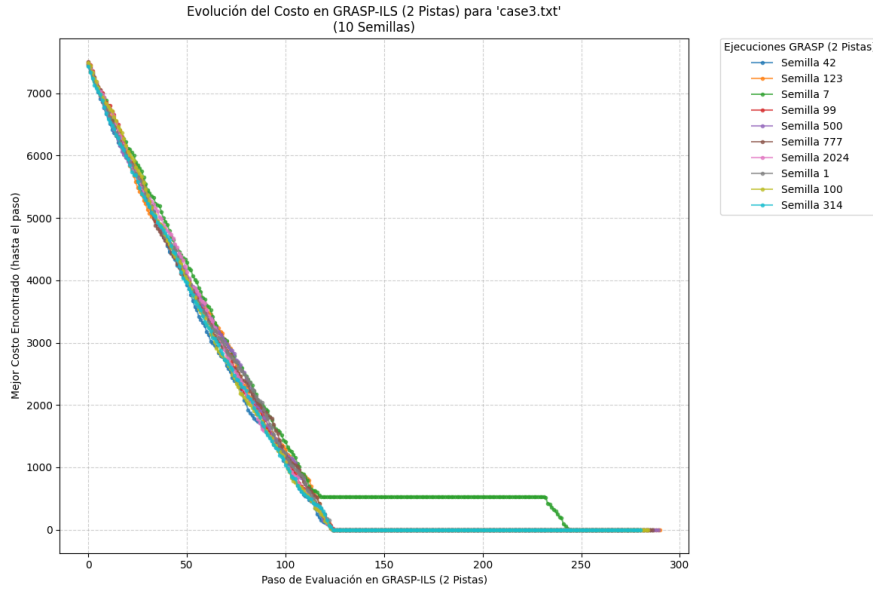


Figura 23: GRASP+ILS (2 Pistas) con Restart Interno: Evolución del Mejor Costo para Case 3 (10 Semillas Greedy).

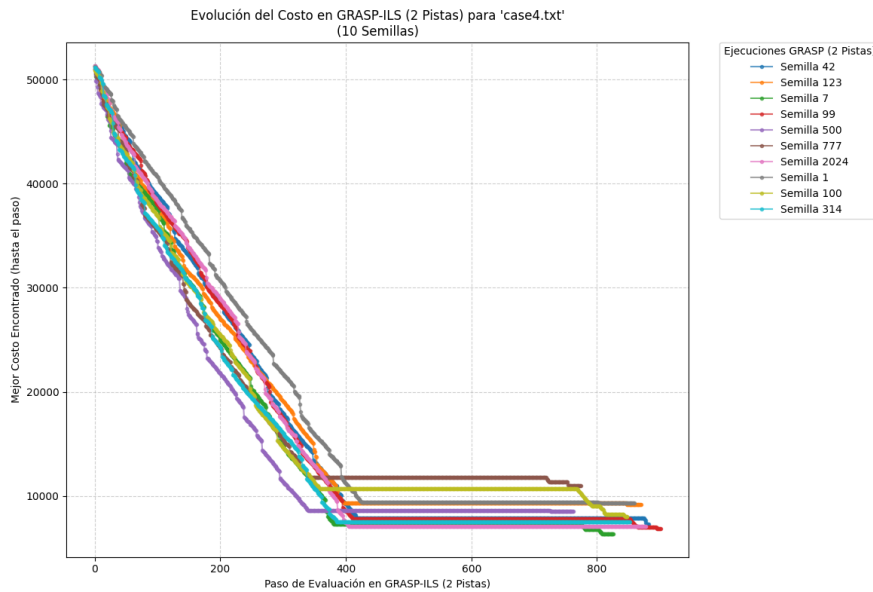


Figura 24: GRASP+ILS (2 Pistas) con Restart Interno: Evolución del Mejor Costo para Case 4 (10 Semillas Greedy).

3.4. Simulated Annealing (2 Pistas, Greedy Determinista)

El algoritmo de Simulated Annealing (SA) se aplica al escenario de dos pistas para explorar el espacio de soluciones y escapar de óptimos locales. De manera análoga al escenario de una pista, la

solución inicial para SA se obtiene del algoritmo **Greedy Determinista para dos pistas** (descrito en la Sección 3.1.1). Para mantener la consistencia y permitir un análisis controlado del impacto de los parámetros de SA, todas las ejecuciones de SA con diferentes configuraciones utilizan una **semilla de aleatoriedad global fija** (`SEMILLA_GLOBAL_REPLICACION = 42`) para el propio algoritmo SA.

3.4.1. Diseño del Algoritmo Simulated Annealing (Determinista, 2 Pistas)

Solución Inicial: La secuencia de aterrizaje y las asignaciones de tiempo y pista generadas por el algoritmo Greedy Determinista para dos pistas (ver Sección 3.1.1) sirven como punto de partida para el SA.

Mecanismos de Generación de Vecinos (2 Pistas): Los operadores para generar soluciones vecinas se adaptan para el entorno de dos pistas:

- **Cambio de Atributo Simple ('Single Attribute Change'):** Se selecciona un avión al azar. Con una probabilidad se modifica su tiempo de aterrizaje (dentro de su ventana $[E_k, L_k]$) en su pista actual, y con la probabilidad complementaria se cambia su asignación a la otra pista (manteniendo el tiempo o ajustándolo si es necesario para la factibilidad).
- **Intercambio de Atributos Completos ('Swap All Attributes'):** Se seleccionan dos aviones al azar y se intercambian sus tiempos de aterrizaje y sus pistas asignadas.
- **Reinserción de Avión ('Reinsert Plane'):** Se selecciona un avión al azar y se le reasigna un nuevo tiempo de aterrizaje aleatorio (dentro de su ventana $[E_k, L_k]$) y una nueva pista aleatoria (0 o 1).

La factibilidad de cada vecino generado (respetando ventanas de tiempo y separaciones en ambas pistas) se verifica antes de considerar su aceptación.

Esquema de Enfriamiento y Criterio de Aceptación: Se utiliza un esquema de enfriamiento geométrico ($T_{\text{nueva}} = T_{\text{actual}} \times \alpha$). Las soluciones que mejoran el costo actual se aceptan siempre. Las soluciones peores se aceptan con una probabilidad $e^{-\Delta C/T}$, donde ΔC es el incremento en el costo y T es la temperatura actual.

3.4.2. Experimentación y Resultados (2 Pistas, Greedy Determinista)

Se ejecutó el algoritmo SA utilizando la solución del Greedy Determinista para dos pistas como punto de partida. Para cada caso de prueba, se evaluaron las mismas 5 configuraciones de parámetros de SA utilizadas en el escenario de una pista, manteniendo la semilla global de SA en `SEMILLA_GLOBAL_REPLICACION = 42`. Estas configuraciones son (reutilizando las etiquetas definidas previamente):

- **Config1_Mod_ExploitFocus_From_C2**
- **Config2_Fast_LowIter_MedT**

- **Config3_VeryFast_VLowIter_LowT**
- **Config4_HighIter_BalancedOps**
- **Config5_Fast_RapidCool_MedIter**

La probabilidad de cada operador de vecindad se gestiona internamente según la configuración.

Los resultados obtenidos para cada caso de prueba y cada configuración se resumen en la Tabla 17. El costo inicial corresponde al obtenido por el Greedy Determinista para dos pistas.

Configuración SA	Case 1 (C. Ini: 6690.00)	Case 2 (C. Ini: 9550.00)	Case 3 (C. Ini: 7639.00)	Case 4 (C. Ini: 51775.43)
Config1_Mod_ExploitFocus_From_C2	270.00	160.00	108.00	2825.84
Config2_Fast_LowIter_MedT	630.00	400.00	217.00	6145.36
Config3_VeryFast_VLowIter_LowT	1130.00	1450.00	457.00	13306.72
Config4_HighIter_BalancedOps	210.00	60.00	1.00	826.69
Config5_Fast_RapidCool_MedIter	310.00	420.00	128.00	6706.29

Tabla 17: Resultados del Simulated Annealing (2 Pistas, Greedy Determinista, Semilla Global SA: 42)

A continuación, se presenta un análisis individual para cada caso de prueba, junto con los gráficos de evolución del costo actual encontrado por SA partiendo de la solución Greedy Determinista para dos pistas.

Análisis para Case 1 (2 Pistas):

- Costo Inicial (Greedy Det. 2P): 6690.00
- Mejor Costo SA (2P): 210.00 (obtenido con Config4_HighIter_BalancedOps)
- Observaciones: Todas las configuraciones de SA lograron mejoras muy sustanciales respecto a la solución Greedy inicial. La ‘Config4_HighIter_BalancedOps’ fue la más destacada, alcanzando el costo más bajo. Le siguen ‘Config1’ (270.00) y ‘Config5’ (310.00). La Figura 25 muestra la rápida convergencia de las configuraciones más efectivas. La disponibilidad de dos pistas permite a SA encontrar soluciones de muy bajo costo.

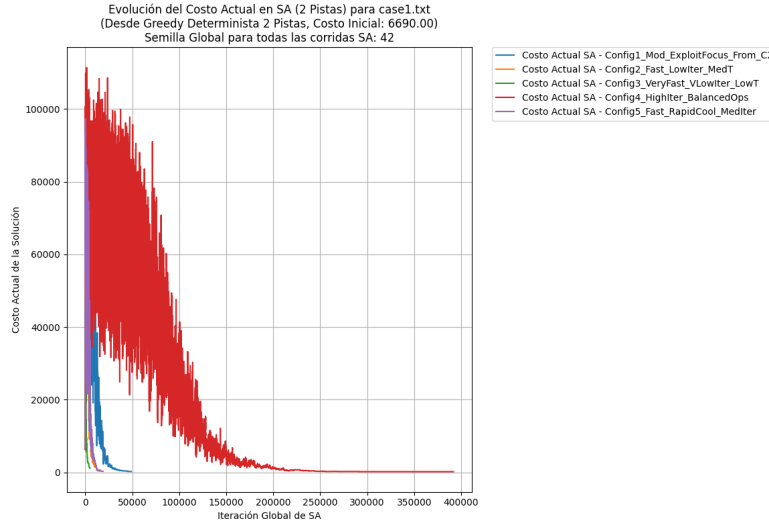


Figura 25: Evolución del costo actual en SA para Case 1 (2 Pistas, Greedy Det., Semilla SA 42).

Análisis para Case 2 (2 Pistas):

- Costo Inicial (Greedy Det. 2P): 9550.00
- Mejor Costo SA (2P): 60.00 (obtenido con Config4_HighIter_BalancedOps)
- Observaciones: SA demuestra una capacidad de mejora excepcional en este caso, reduciendo el costo en más del 99 %. Nuevamente, ‘Config4_HighIter_BalancedOps’ obtiene el mejor resultado. ‘Config1’ (160.00) también es muy competitiva. La Figura 26 ilustra cómo las configuraciones eficientes exploran el espacio y convergen a soluciones de costo notablemente bajo.

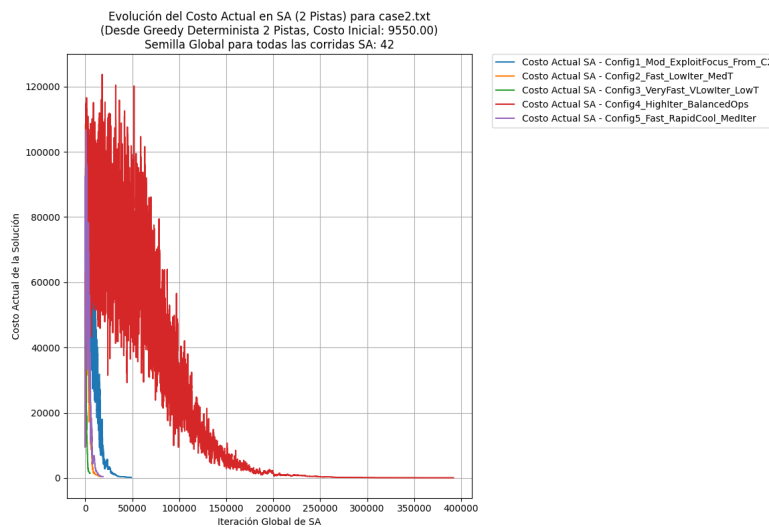


Figura 26: Evolución del costo actual en SA para Case 2 (2 Pistas, Greedy Det., Semilla SA 42).

Análisis para Case 3 (2 Pistas):

- Costo Inicial (Greedy Det. 2P): 7639.00
- Mejor Costo SA (2P): 1.00 (obtenido con Config4.HighIter.BalancedOps)
- Observaciones: Para el Case 3, SA logra un resultado casi perfecto. La ‘Config4.HighIter.BalancedOps’ reduce el costo a prácticamente cero (1.00), lo cual es una mejora drástica. Otras configuraciones como ‘Config1’ (108.00) y ‘Config5’ (128.00) también son muy efectivas, aunque no alcanzan el mínimo absoluto. La Figura 27 muestra la convergencia. Esto contrasta con el escenario de una pista donde SA no mejoraba el Greedy para este caso, resaltando el impacto de las dos pistas y la capacidad de SA para explotar esta flexibilidad.

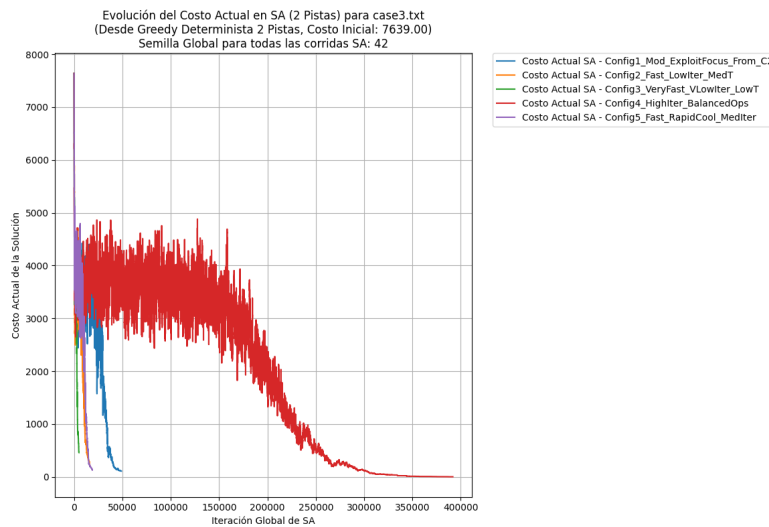


Figura 27: Evolución del costo actual en SA para Case 3 (2 Pistas, Greedy Det., Semilla SA 42).

Análisis para Case 4 (2 Pistas):

- Costo Inicial (Greedy Det. 2P): 51775.43
- Mejor Costo SA (2P): 826.69 (obtenido con Config4.HighIter.BalancedOps)
- Observaciones: En el caso más grande, SA también consigue una reducción de costo extremadamente significativa (más del 98 %). ‘Config4.HighIter.BalancedOps’ se consolida como la configuración más robusta. ‘Config1’ (2825.84) también ofrece una mejora considerable, aunque es superada. La Figura 28 muestra la evolución, donde ‘Config4’ desciende a una región de costo mucho menor que las demás.

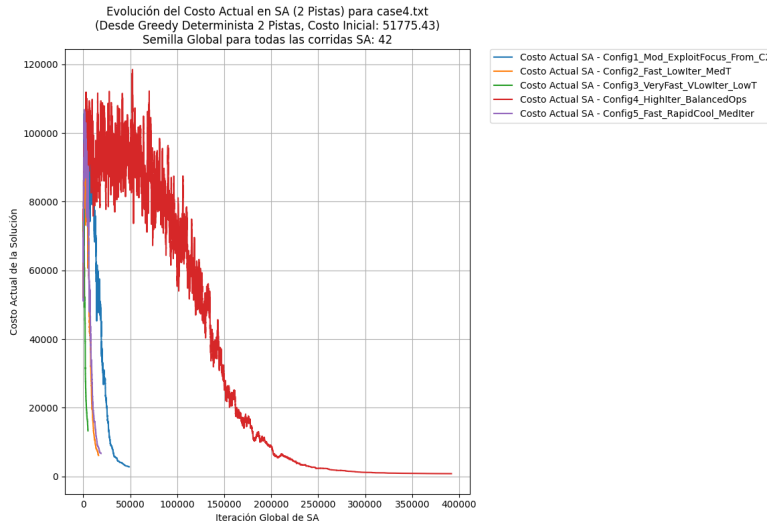


Figura 28: Evolución del costo actual en SA para Case 4 (2 Pistas, Greedy Det., Semilla SA 42).

Discusión General de los Parámetros: Para el escenario de dos pistas y partiendo de la solución Greedy Determinista, Simulated Annealing demuestra ser una metaheurística extraordinariamente efectiva.

- La configuración ‘Config4_HighIter_BalancedOps’ (temperatura inicial media-alta, enfriamiento lento, muchas iteraciones por temperatura y operadores balanceados) consistentemente produce los mejores resultados en los cuatro casos de prueba, logrando reducciones de costo drásticas. Esto sugiere que una exploración exhaustiva en cada nivel de temperatura es crucial cuando se dispone de la flexibilidad de dos pistas.
- ‘Config1_Mod_ExploitFocus_From_C2’ también muestra un rendimiento muy competitivo, siendo la segunda mejor en la mayoría de los casos. Su enfoque en la explotación, combinado con una temperatura inicial moderada, parece funcionar bien en el contexto de dos pistas.
- Las configuraciones con enfriamiento más rápido o menos iteraciones por temperatura (‘Config2’, ‘Config3’, ‘Config5’) tienden a ser superadas, aunque aún logran mejoras significativas sobre el Greedy inicial.
- Es notable la diferencia de rendimiento de SA en el Case 3 para dos pistas en comparación con una pista. Mientras que con una pista SA no mejoraba el resultado del Greedy, con dos pistas logra reducir el costo a casi cero. Esto evidencia cómo la estructura del problema cambia con la adición de recursos (pistas) y cómo SA puede capitalizar esta nueva flexibilidad.

La semilla global fija para las ejecuciones de SA (‘SEMILLA_GLOBAL_REPLICACION = 42’) fue crucial para aislar el efecto de los diferentes conjuntos de parámetros de SA sobre una misma solución inicial, permitiendo una comparación directa de su efectividad. Los resultados indican que SA, con una parametrización adecuada, es capaz de encontrar soluciones de muy alta calidad para el problema de secuenciación en dos pistas.

3.5. Simulated Annealing (2 Pistas, Greedy Estocástico)

En esta sección se evalúa el rendimiento de Simulated Annealing (SA) cuando se inicializa con un conjunto diverso de soluciones generadas por el método Greedy Estocástico para dos pistas (descrito en la Sección 3.1.2). Al proporcionar múltiples puntos de partida, se busca explorar si SA puede alcanzar regiones del espacio de soluciones de mejor calidad en comparación con una única inicialización determinista, especialmente en el complejo escenario de dos pistas.

3.5.1. Diseño del Algoritmo y Metodología Experimental (Estocástica, 2 Pistas)

Soluciones Iniciales Múltiples (2 Pistas): Se utilizan las 10 soluciones generadas por el algoritmo Greedy Estocástico para dos pistas, cada una obtenida con una de las 10 semillas predefinidas (42, 123, 7, 99, 500, 777, 2024, 1, 100, 314). Cada una de estas 10 soluciones, con sus respectivas asignaciones de aviones a tiempos y a una de las dos pistas, sirve como un punto de partida independiente para el proceso de SA.

Configuraciones de Simulated Annealing y Asignación de Semillas SA: Para cada una de las 10 soluciones iniciales del Greedy Estocástico (2 Pistas), se aplican las mismas 5 configuraciones de parámetros de SA detalladas en la sección anterior (ver Sección 3.4.2 para los nombres de las configuraciones, ej. ‘Config1_Mod_ExploitFocus_From_C2’, ‘Config4_HighIter_BalancedOps’, etc.).

Para asegurar la reproducibilidad de cada ejecución individual de SA y aislar el efecto de la configuración de SA sobre una partida estocástica específica, a cada combinación (Solución Inicial Greedy Estocástico $\#i$, Configuración SA $\#j$) se le asigna una semilla única y determinista para el propio algoritmo SA. Por ejemplo, la solución Greedy de la semilla 42, al ser procesada por ‘Config1’, usa la semilla SA $(1 \times 1000) + (1 \times 100) = 1100$; al ser procesada por ‘Config2’, usa la semilla SA 1200, y así sucesivamente. Para la solución Greedy de la semilla 123, las semillas SA para las 5 configuraciones serían 2100, 2200, \dots , 2500.

Mecanismos de SA (2 Pistas): La lógica interna del algoritmo SA (generación de vecinos adaptada a dos pistas, función de evaluación, esquema de enfriamiento, criterio de aceptación y condición de parada) es idéntica a la descrita en la Sección 3.4.1.

3.5.2. Experimentación y Resultados (2 Pistas, Greedy Estocástico)

Para cada uno de los 4 casos de prueba, se generaron 10 soluciones iniciales mediante el Greedy Estocástico para dos pistas. Cada una de estas 10 soluciones sirvió como punto de partida para 5 ejecuciones de Simulated Annealing, resultando en $10 \times 5 = 50$ ejecuciones de SA por caso de prueba.

La Tabla 18 presenta un resumen de los hallazgos. Para cada caso, se indica el rango de costos de las soluciones iniciales del Greedy Estocástico (2 Pistas) y el mejor costo final obtenido por SA después de procesar todas las combinaciones, junto con la semilla Greedy inicial y la configuración de SA que condujeron a dicho óptimo.

Métrica	Case 1 (2P)	Case 2 (2P)	Case 3 (2P)	Case 4 (2P)
Costos Iniciales (Greedy Sto. 2P)	4980.00 - 5780.00	7940.00 - 8440.00	7432.00 - 7507.00	50894.90 - 51331.42
Mejor Costo Inicial (Greedy Sto. 2P)	4980.00 (Semilla 1)	7940.00 (Semilla 7)	7432.00 (Semilla 314)	50894.90 (Semilla 100)
Mejor Costo Final SA (desde Greedy Sto. 2P)	210.00	60.00	0.00	779.02
Semilla Greedy Inicial para Mejor SA	Varias*	Varias**	100	314
Config. SA para Mejor SA	Config4	Config4	Config4	Config4
Semilla SA Específica para Mejor SA	(e.g., 1400, 2400, etc.)	(e.g., 1400, 2400, etc.)	9400	10400
Costo Inicial Greedy Específico del Mejor SA Global	(e.g., 5720.00 de Sem. 42)	(e.g., 8100.00 de Sem. 42)	7487.00	51195.96

Tabla 18: Resumen de Resultados: Simulated Annealing (2 Pistas, Greedy Estocástico)

A continuación, se muestra para cada caso de prueba un gráfico que ilustra la evolución del costo de SA para la combinación (solución inicial Greedy Estocástica específica y configuración SA) que produjo el mejor costo global para ese caso.

Análisis para Case 1 (2 Pistas, Estocástico):

- Costos Iniciales (Greedy Sto. 2P): Entre 4980.00 y 5780.00.
- Mejor Costo Final SA: 210.00.
- Detalles: Este costo óptimo de 210.00 fue alcanzado por la ‘Config4_HighIter_BalancedOps’ de SA partiendo de diversas soluciones iniciales del Greedy Estocástico (incluyendo la de semilla Greedy 42, costo inicial 5720.00, con semilla SA 1400; o la de semilla Greedy 123, costo inicial 5580.00, con semilla SA 2400, entre otras).
- Observaciones: El resultado iguala al mejor obtenido por SA partiendo del Greedy Determinista (210.00). Esto indica que, para esta instancia y con la robustez de ‘Config4’, la calidad del punto de partida (determinista vs. múltiples estocásticos) no marcó una diferencia en el óptimo final encontrado por SA. La Figura 29 muestra la convergencia para una de estas combinaciones (semilla Greedy 42).

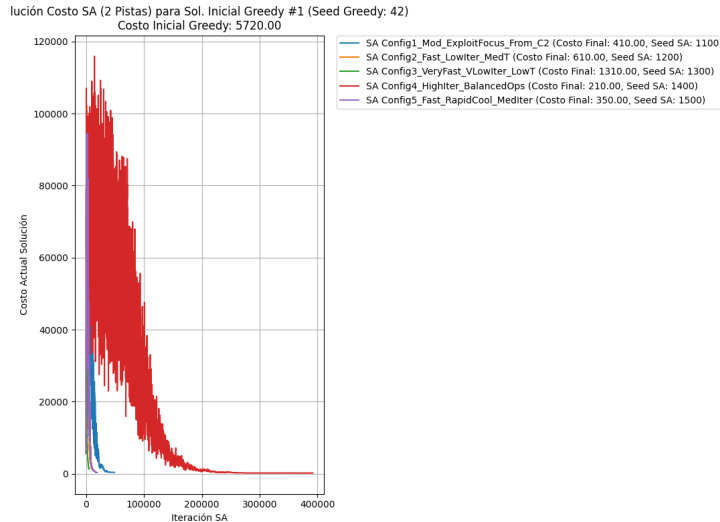


Figura 29: Evolución del costo SA para Case 1 (2 Pistas), partiendo de la Sol. Inicial Greedy Sto. (Semilla 42).

Análisis para Case 2 (2 Pistas, Estocástico):

- Costos Iniciales (Greedy Sto. 2P): Entre 7940.00 y 8440.00.
- Mejor Costo Final SA: 60.00.
- Detalles: El excepcional costo de 60.00 fue consistentemente alcanzado por la ‘Config4_HighIter_BalancedOps’ de SA, independientemente de cuál de las 10 soluciones iniciales del Greedy Estocástico se utilizara. Por ejemplo, partiendo de la solución Greedy (semilla 42, costo inicial 8100.00), SA con Config4 (semilla SA 1400) llegó a 60.00.
- Observaciones: Este resultado de 60.00 también iguala al mejor obtenido por SA con partida determinista. La consistencia de ‘Config4’ en alcanzar este valor desde múltiples puntos de partida estocásticos (todos con costos iniciales significativamente mayores) subraya su efectividad para este caso. La Figura 30 (usar una etiqueta diferente si es necesario) muestra la convergencia para la semilla Greedy 42.

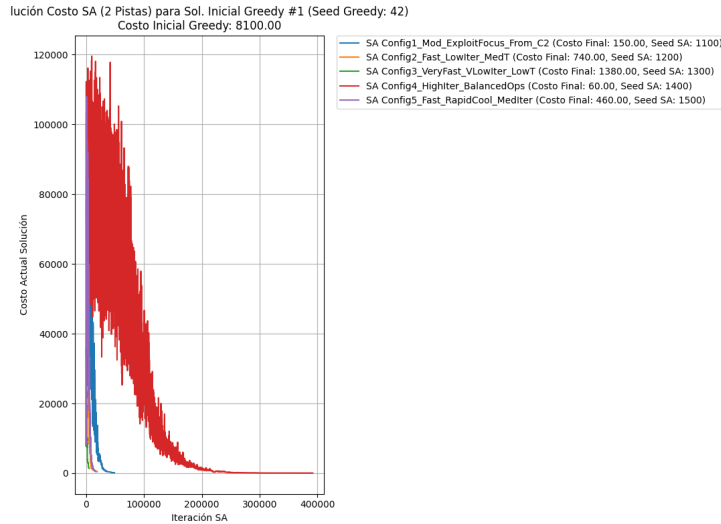


Figura 30: Evolución del costo SA para Case 2 (2 Pistas), partiendo de la Sol. Inicial Greedy Sto. (Semilla 42).

Análisis para Case 3 (2 Pistas, Estocástico):

- Costos Iniciales (Greedy Sto. 2P): Entre 7432.00 y 7507.00.
- Mejor Costo Final SA: 0.00.
- Detalles: El costo perfecto de 0.00 fue alcanzado por la 'Config4_HighIter_BalancedOps' de SA partiendo de varias soluciones iniciales del Greedy Estocástico, incluyendo la de semilla Greedy 100 (costo inicial 7487.00, con semilla SA 9400) y la de semilla Greedy 314 (costo inicial 7432.00, con semilla SA 10400).
- Observaciones: SA, con la configuración adecuada, logra encontrar la solución óptima de costo cero, superando ligeramente al SA con partida determinista que obtuvo un costo de 1.00. Esto demuestra que, aunque la mejora es marginal, una partida estocástica puede, en algunos casos, guiar a SA a un óptimo global aún mejor. La Figura 31 ilustra la convergencia para la semilla Greedy 100.

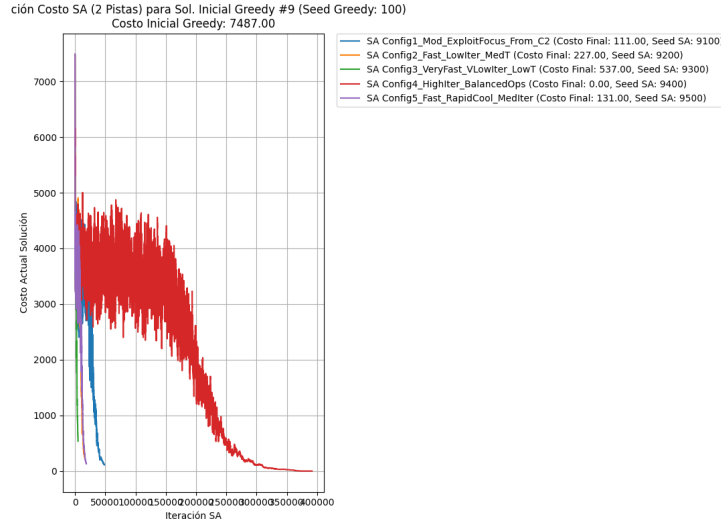


Figura 31: Evolución del costo SA para Case 3 (2 Pistas), partiendo de la Sol. Inicial Greedy Sto. (Semilla 100).

Análisis para Case 4 (2 Pistas, Estocástico):

- Costos Iniciales (Greedy Sto. 2P): Entre 50894.90 y 51331.42.
- Mejor Costo Final SA: 779.02.
- Detalles: La mejor solución de SA, con un costo de 779.02, se obtuvo al aplicar 'Config4_HighIter_BalancedOps' (con semilla SA 10400) a la solución inicial generada por el Greedy Estocástico con semilla 314 (costo inicial 51195.96).
- Observaciones: En este caso más complejo, utilizar una partida del Greedy Estocástico permitió a SA encontrar una solución notablemente mejor (779.02) que cuando se partió de la solución del Greedy Determinista (donde el mejor costo SA fue 826.69). Esto resalta la sinergia entre la diversificación del Greedy Estocástico y la capacidad de búsqueda intensiva de SA, especialmente cuando la solución determinista podría estar en una región menos prometedora para la instancia más grande. La Figura 32 muestra la evolución.

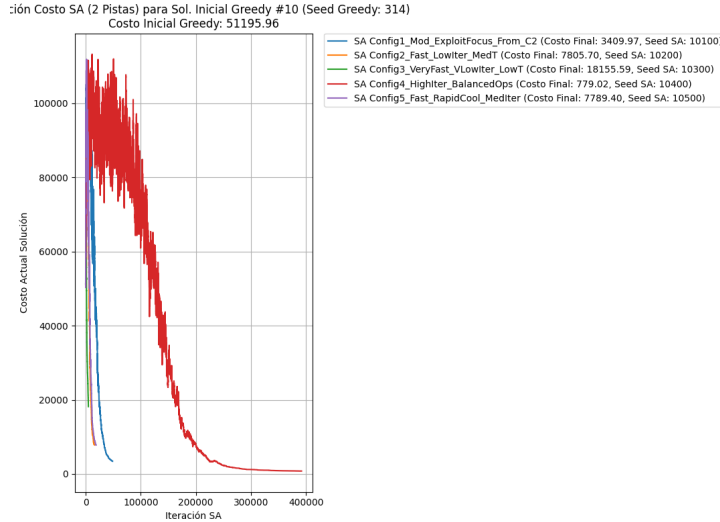


Figura 32: Evolución del costo SA para Case 4 (2 Pistas), partiendo de la Sol. Inicial Greedy Sto. (Semilla 314).

Discusión General de Parámetros (2 Pistas, Greedy Estocástico): La estrategia de inicializar Simulated Annealing con múltiples soluciones del Greedy Estocástico para dos pistas demuestra ser muy efectiva.

- La ‘Config4.HighIter_BalancedOps’ de SA se consolida como la más robusta y efectiva, produciendo consistentemente los mejores o casi los mejores resultados para cada una de las 10 partidas estocásticas en todos los casos de prueba, y alcanzando los mejores costos globales para los 4 casos.
- Para el Case 1 y Case 2, el mejor resultado de SA partiendo de soluciones estocásticas igualó al obtenido partiendo de la solución determinista, sugiriendo que ‘Config4’ es lo suficientemente potente para converger al mismo óptimo local de alta calidad desde diversas buenas regiones iniciales.
- Para el Case 3, una partida estocástica permitió a SA (con Config4) alcanzar un costo de 0.00, superando marginalmente el 1.00 obtenido desde la partida determinista.
- Para el Case 4, el más grande, la diversificación de las soluciones iniciales del Greedy Estocástico fue claramente beneficiosa, permitiendo a SA (con Config4) encontrar una solución significativamente mejor (779.02) que la obtenida desde la única partida determinista (826.69).

Estos resultados refuerzan la idea de que, para problemas complejos, combinar una fase de construcción estocástica para generar diversidad con una metaheurística de búsqueda local potente como SA es una estrategia muy prometedora. La elección de múltiples puntos de partida aumenta la probabilidad de que SA inicie su búsqueda en una cuenca de atracción que contenga un óptimo global de muy alta calidad, o incluso el óptimo global.

4. Análisis Comparativo y Conclusiones

El presente trabajo se centró en la exploración y evaluación de diversas estrategias algorítmicas para abordar el Problema de Secuenciación de Aterrizajes de Aeronaves (ALP), considerando escenarios con una y dos pistas. A través de la implementación de algoritmos Greedy (determinista y estocástico), GRASP con Hill Climbing (con y sin componente de reinicio ILS) y Simulated Annealing (con diferentes configuraciones y puntos de partida), se ha obtenido una visión integral de su desempeño y aplicabilidad. Esta sección ofrece un análisis comparativo de los resultados y extrae las conclusiones principales del estudio.

4.1. Análisis del Rendimiento de los Algoritmos

4.1.1. Algoritmos Greedy: Una Base Constructiva

Los algoritmos **Greedy Determinista** sirvieron como un punto de referencia inicial, proporcionando soluciones factibles de manera rápida. Su naturaleza, basada en la selección localmente óptima (generalmente, el aterrizaje más temprano posible), demostró ser sensible a las características de las instancias. En el escenario de **1 Pista**, el Caso 3 fue una anomalía, donde el Greedy Determinista obtuvo un costo (1550.00) que no fue consistentemente superado por variantes estocásticas simples ni algunas metaheurísticas, sugiriendo que para ciertas configuraciones de datos, una heurística sencilla puede alinearse bien con la estructura del óptimo. Sin embargo, en general, sus soluciones fueron subóptimas, especialmente en instancias más grandes (Caso 4). Al pasar a **2 Pistas**, el Greedy Determinista ofreció una mejora natural debido a la mayor flexibilidad, pero sus costos iniciales siguieron siendo considerablemente altos (e.g., 51775.43 para el Caso 4), evidenciando las limitaciones de una toma de decisiones puramente local sin una visión global.

El **Greedy Estocástico**, al introducir aleatoriedad en la selección de candidatos y, en el caso de 2 pistas, en la baraja de aviones y la adición de un pequeño margen de tiempo, demostró su valor. En el escenario de **1 Pista**, logró superar al determinista en algunas instancias (Caso 1 y Caso 4), aunque no consistentemente, y con una variabilidad dependiente de la semilla. La necesidad de múltiples intentos internos en algunas ejecuciones del Caso 3 (1 Pista) resaltó que la aleatorización simple no siempre garantiza una fácil navegación por el espacio de búsqueda. Para **2 Pistas**, el Greedy Estocástico fue más consistentemente superior a su contraparte determinista en todas las instancias, logrando mejores costos promedio y mínimos. Esto sugiere que la mayor dimensionalidad del problema con dos pistas se beneficia más de la exploración diversificada que ofrece la estocasticidad.

4.1.2. GRASP con Hill Climbing: El Poder de la Búsqueda Local

La incorporación de una fase de **Búsqueda Local (Hill Climbing Alguna-Mejora)** tras la construcción Greedy marcó una mejora sustancial en la calidad de las soluciones. Para **GRASP con base Greedy Determinista (1 Pista)**, Hill Climbing logró reducciones de costo muy significativas (hasta un 71.15 % en el Caso 2), excepto en el Caso 3, donde la solución Greedy inicial ya era un

óptimo local respecto al vecindario de HC. Esto subraya que HC es efectivo para refinar soluciones, pero su éxito depende de la calidad del punto de partida y la estructura del paisaje de búsqueda. En el escenario de **2 Pistas**, el impacto de HC sobre la solución Greedy Determinista fue aún más drástico, con mejoras de hasta el 100 % (Caso 3, alcanzando costo cero). La capacidad de HC para explorar movimientos de cambio de pista fue crucial.

El **GRASP con base Greedy Estocástico y componente de Restart ILS** demostró ser una de las estrategias más potentes. En **1 Pista**, esta variante superó al GRASP de base determinista en los casos más complejos (Caso 2 y Caso 4), obteniendo costos de 1680.00 y 20058.73 respectivamente. La combinación de múltiples puntos de partida (10 semillas Greedy) y la capacidad de la ILS para perturbar y reiniciar la búsqueda local desde nuevos puntos, facilitó la evasión de óptimos locales. No obstante, para el Caso 1, el GRASP determinista simple fue ligeramente mejor, y para el Caso 3, el Greedy Determinista original siguió siendo insuperable por esta metaheurística. En el escenario de **2 Pistas**, esta variante de GRASP brilló consistentemente. Logró costos notablemente bajos (e.g., 270.00 para Caso 1 y Caso 2, 0.00 para Caso 3, y 6396.17 para Caso 4). Estos resultados fueron, en general, los mejores o estuvieron entre los mejores obtenidos, demostrando que la sinergia entre una construcción estocástica diversificada y una búsqueda local iterada es altamente efectiva cuando se dispone de mayor flexibilidad operativa.

4.1.3. Simulated Annealing: Exploración Robusta del Espacio de Soluciones

Simulated Annealing (SA) se destacó por su capacidad para escapar de óptimos locales y encontrar soluciones de muy alta calidad, especialmente con una parametrización adecuada. Partiendo del **Greedy Determinista (1 Pista)**, SA (particularmente con ‘Config4.HighIter.BalancedOps’) logró mejoras sustanciales en los Casos 1, 2 y 4, alcanzando costos como 1510.00 (Caso 1) y 8236.25 (Caso 4). Sin embargo, al igual que HC, no pudo mejorar la solución del Greedy Determinista para el Caso 3 (1550.00), utilizando la misma semilla global de SA para la comparación de configuraciones. Cuando SA partió de soluciones del **Greedy Estocástico (1 Pista)**, se observaron mejoras adicionales en algunos casos. Para el Caso 4, se alcanzó un costo de 7797.66, superando el resultado de SA con partida determinista. Para el Caso 1, la mejora fue marginal (1500.00 vs 1510.00). Esto sugiere que la diversificación de los puntos de partida puede ser beneficiosa para SA. La ‘Config4’ siguió siendo la más robusta.

En el escenario de **2 Pistas**, SA demostró un rendimiento excepcional. Partiendo del **Greedy Determinista (2 Pistas)**, la ‘Config4’ de SA produjo reducciones de costo masivas, alcanzando 210.00 (Caso 1), 60.00 (Caso 2), un impresionante 1.00 (Caso 3), y 826.69 (Caso 4). Esto fue significativamente mejor que los resultados de SA en 1 pista, especialmente para el Caso 3. Al utilizar el **Greedy Estocástico (2 Pistas)**, SA (nuevamente con ‘Config4’) consolidó su superioridad. Logró igualar los excelentes resultados de 210.00 (Caso 1) y 60.00 (Caso 2). Notablemente, para el Caso 3, alcanzó un costo perfecto de 0.00, y para el Caso 4, mejoró aún más el resultado a 779.02. Esto subraya que, con dos pistas, la combinación de múltiples inicios estocásticos y un SA bien parametrizado es una estrategia ganadora para las instancias más complejas.

4.2. Impacto del Número de Pistas

La disponibilidad de una **segunda pista** tuvo un impacto fuerte en la calidad de las soluciones obtenibles. Para todos los algoritmos y casos de prueba, se observó una reducción general de costos, ya que los costos finales en el escenario de 2 pistas fueron drásticamente inferiores a los de 1 pista; esto es intuitivo, pues la capacidad adicional permite mayor flexibilidad para programar aterrizajes sin incurrir en grandes penalizaciones por desviación o conflictos de separación. Notablemente, en el Caso 3, que resultó problemático para algunas metaheurísticas en 1 pista, se logró la consecución de óptimos perfectos, alcanzándose consistentemente un costo de 0.00 o cercano a cero con las metaheurísticas más potentes (GRASP+ILS y SA) en el escenario de 2 pistas, lo que indica que, con recursos suficientes, el problema puede volverse considerablemente más fácil de optimizar. Además, se constató una mayor efectividad de las metaheurísticas, dado que algoritmos como Hill Climbing y Simulated Annealing pudieron explotar mucho mejor el espacio de soluciones con dos pistas, siendo los operadores de vecindario que implicaban cambiar un avión a otra pista clave para encontrar configuraciones superiores.

4.3. Influencia de los Parámetros y Estrategias de Inicialización

- **Parámetros de SA:** La configuración ‘Config4_HighIter_BalancedOps’ (temperatura inicial media/alta, enfriamiento lento, muchas iteraciones por temperatura y operadores balanceados) emergió consistentemente como la más robusta para SA, tanto en 1 como en 2 pistas. Esto resalta la importancia de permitir suficiente exploración en cada nivel de temperatura. Configuraciones con enfriamiento muy rápido o pocas iteraciones tendieron a converger prematuramente a soluciones subóptimas.
- **Semillas de Aleatoriedad:** El uso de múltiples semillas para los algoritmos estocásticos (Greedy Estocástico, GRASP con base estocástica) fue crucial para la diversificación. Los resultados mostraron variabilidad, indicando que una sola ejecución estocástica no garantiza encontrar la mejor solución posible del método. La estrategia de múltiples arranques es, por lo tanto, fundamental.
- **Determinista vs. Estocástica para Metaheurísticas:** Para **GRASP**, la variante con construcción estocástica y reinicios ILS fue generalmente superior a la que partía de un único Greedy Determinista, especialmente en los casos más complejos y en el escenario de 2 pistas. Para **Simulated Annealing**, iniciar desde múltiples soluciones del Greedy Estocástico ofreció ventajas, particularmente en las instancias más grandes (Caso 4) y en el escenario de 2 pistas, donde permitió alcanzar mejores soluciones finales que partiendo de la única solución del Greedy Determinista. No obstante, la calidad de la configuración de SA (‘Config4’) fue tan dominante que, en algunos casos más simples, el punto de partida específico (dentro de un rango de ‘buenas’ soluciones iniciales) no alteró drásticamente el óptimo final encontrado por SA.

- **Componente de Restart (ILS en GRASP):** La Búsqueda Local Iterada, con su mecanismo de perturbación y reinicio de la búsqueda local, fue vital para que GRASP escapara de óptimos locales y explorara más ampliamente el espacio de soluciones. Esto fue particularmente notorio en el escenario de 2 pistas.

4.4. Conclusión

Este estudio experimental sobre el Problema de Secuenciación de Aterrizajes de Aeronaves ha sido muy útil para entender la complejidad de este tipo de problemas de optimización y ver cómo diferentes algoritmos rinden de manera distinta. Quedó claro que, aunque los algoritmos constructivos simples como Greedy dan un primer resultado rápido y válido, su capacidad es limitada. La verdadera fuerza para enfrentar la complejidad del ALP se encuentra en las metaheurísticas. Durante este trabajo, vimos de forma repetida cómo metaheurísticas como GRASP —especialmente usando una base estocástica y reinicios con Búsqueda Local Iterada— y, de forma notable, Simulated Annealing —bien ajustado en sus parámetros y comenzando desde diversas soluciones iniciales— logran superar las limitaciones de quedarse atrapados en soluciones que parecen buenas localmente, y así pueden revisar el espacio de soluciones de una manera mucho más completa y efectiva.

Uno de los mejores aprendizajes es ver el gran cambio que producen los recursos disponibles, como el número de pistas. La mejora en la calidad de las soluciones al usar dos pistas en vez de una fue muy clara, incluso se llegó a soluciones de costo cero o casi cero en casos que antes parecían demasiado difíciles. Esto muestra que la propia estructura del problema y la dificultad para encontrar buenas soluciones están muy conectadas con la flexibilidad que se tenga para operar. Además, los experimentos confirmaron lo importante que es variar la búsqueda —usando diferentes semillas o puntos de inicio al azar— y tener formas sólidas para que los algoritmos no se estanquen en soluciones subóptimas. Esto se vio en cómo Simulated Annealing a veces acepta peores soluciones para seguir buscando y en cómo las alteraciones en la Búsqueda Local Iterada ayudan a encontrar nuevos caminos. Por lo que al final, el proceso de diseñar, programar y analizar estos algoritmos nos ha dado una visión clara sobre el equilibrio necesario entre buscar nuevas soluciones y mejorar las que ya se tienen, y cómo las elecciones que se hacen al diseñar un algoritmo pueden cambiar drásticamente la habilidad para encontrar respuestas eficientes a problemas reales. Además mencionar que el informe actual es extenso. Con el objetivo de mantener la concisión y no sobrecargar el documento, se ha decidido no incluir la totalidad de las imágenes y los datos detallados asociados a los gráficos de cada sección. Este material de apoyo se encuentra disponible para consulta en el repositorio del proyecto en GitHub: <https://github.com/renato357/Tarea-2-A-y-M-JB-RC>.

5. Referencias

1. Duarte Muñoz, A., Pantrigo Fernández, J. J., & Gallego Carrillo, M. (2007). *Metaheurísticas*. Dykinson.
2. Moreno Ramírez, J. (2018). Metaheurística GRASP para el problema Vertex Bisection Mini-

- mization. *Revista Cubana de Ciencias Informáticas*, 12(Extra 2), 30-41. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000500003
3. Moreno, F. (2016). *Cómo la naturaleza enseña a solucionar los problemas intratables: Algoritmos Genéticos y el Recocido Simulado* [Trabajo de grado, Universidad de los Andes]. Repositorio Funes. <https://funes.uniandes.edu.co/wp-content/uploads/tainacan-items/32454/1243242/Moreno2016Como.pdf>
 4. García Sabater, J. P. (n.d.). *Apuntes de Optimización Combinatoria (Modelado y Resolución de Problemas de Organización Industrial)* [Apuntes de curso]. Universidad Politécnica de Valencia. Recuperado de <https://personales.upv.es/jpgarcia/LinkedDocuments/ApuntesOptimizacionCombinatoria.pdf>
 5. García, [Inicial del nombre por confirmar]. (n.d.). *Optimización estática de la secuencia de aterrizajes en entornos con varias pistas* [Trabajo Fin de Máster, Universidad de Sevilla]. Repositorio Institucional de la Universidad de Sevilla. https://biblus.us.es/bibing/proyectos/abreproy/71393/descargar_fichero/TFM-1393-GARCIA.pdf
 6. Huguet Garcia, P. (2013). *Optimización Multi-Objetivo de la Secuencia de Aterrizaje de Aviones* [Trabajo de Fin de Grado, Universitat Autònoma de Barcelona]. Dipòsit Digital de Documents de la UAB. <https://ddd.uab.cat/pub/tfg/2013/113639/HuguetGarciaPatricia-TFGAa2012-13.pdf>