Universidad Carlos III de Madrid

Ingeniería Informática + ADE 2023-24

Heurística y Optimización

Práctica 1

Curso 2024-25

Carlos Bravo Garrán - 100474964

Índice

1. Introducción

Este documento presenta la modelización y resolución de un problema de asignación de recursos y optimización de los beneficios y costes de una aerolínea. La práctica está dividida en tres partes: un modelo básico para la asignación de asientos y tarifas, un modelo avanzado enfocado en la programación de aterrizajes minimizando los costes de retraso, y finalmente, una fusión de ambos modelos en uno solo que contempla ambas optimizaciones simultáneamente.

En el desarrollo de cada modelo, se utilizaron herramientas como *Calc*, para una solución inicial en la primera parte, y *GLPK*, para la implementación avanzada y la combinación de ambos problemas. La memoria detalla el planteamiento de las variables y restricciones, así como las decisiones de diseño tomadas en cada modelo, finalizando con un análisis de resultados y conclusiones sobre el proceso y los resultados obtenidos.

2. Descripción de Modelos

En este apartado se detallan los modelos desarrollados para abordar el problema de optimización en distintas etapas. Comenzamos con un modelo básico en *Calc*, seguido por el modelo avanzado y un modelo fusionado que combina ambas optimizaciones, ambos en *GLPK*. Cada sección describe la modelización a mano y su implementación en las respectivas herramientas, incluyendo la definición de variables, restricciones, y las decisiones de diseño fundamentales en cada caso.

2.1. Modelo Básico en Calc

Para modelizar el problema de asignación de asientos y tarifas de forma manual, definí las variables y restricciones clave de forma que optimicen la asignación de billetes en función de los beneficios y capacidades del avión. A continuación, se detallan los elementos fundamentales de la modelización:

Variables:

x_{ij} : Representa el número de billetes vendidos para cada tipo de tarifa *i* en cada avión *j*.

Esta variable es entera y no negativa, ya que no se pueden vender billetes parciales y no se permiten ventas negativas.

Se puede desglosar para simplificar la función objetivo:

x_{i1} = número de billetes estándar en el avión i.

x_{i2} = número de billetes leisure plus.

x_{i3} = número de billetes business plus.

Parámetros

$Precio_j$: Precio de cada tipo de tarifa *j*, que permite calcular los ingresos asociados a los billetes vendidos para cada categoría de pasajeros.

$Peso_j$: Peso permitido del equipaje para cada tipo de tarifa *j*, que influye en la capacidad de peso total que puede transportar cada avión.

$Asientos_i$: Cantidad de asientos del avión i, que limita la cantidad de billetes que pueden ser vendidos por avión.

$Capacidad_i$: Capacidad máxima de peso permitida en cada avión i, lo cual restringe la suma de peso de equipajes por tarifa.

Función Objetivo

El objetivo del modelo es **maximizar los ingresos** obtenidos por la venta de billetes, lo cual se representa como:

$$Max\ z = \sum_{i=1}^{5} (19x_{i1} + 49x_{i2} + 69x_{i3})$$

Restricciones

1. Asientos por avión: La suma de billetes vendidos no puede exceder la capacidad de asientos de cada avión:

$$R_1 : x_{i1} + x_{i2} + x_{i3} \leq Asientos_i$$

2. Peso por avión: La suma del peso total de los equipajes no debe exceder la capacidad máxima de peso del avión:

$$R_2 : x_{i1} + 20x_{i2} + 40x_{i3} \leq PesoMáx_i$$

3. Restricción mínima para tarifa Leisure Plus: Se requiere que se vendan al menos 20 billetes de esta tarifa por avión:

$$R_3 : x_{i2} \geq 20$$

4. Restricción mínima para tarifa Business: Se exige que se vendan al menos 10 billetes de la tarifa por avión:

$$R_4 : x_{i3} \geq 10$$

5. Restricción de proporción de billetes de tarifa Estándar: Al menos el 60% de todos los billetes vendidos en cada avión deben corresponder a la tarifa 'Estándar':

$$R_5 : \sum_{i=1}^{5} x_{i1} \geq 0.6 \sum_{j=1}^{3} \sum_{i=1}^{5} x_{ij}$$

Con esta formalización, cada parámetro y variable tiene una función específica, permitiendo la organización de datos y la implementación del *Solver* de *Calc* para optimizar la asignación de billetes respetando las limitaciones de la aerolínea.

Para implementar el modelo en *Calc*, he organizado toda la información de forma estructurada y clara en una hoja de cálculo, permitiendo una visualización intuitiva y detallada de cada parámetro y variable, facilitando su comprensión y manipulación.

Posteriormente, configuré el *Solver* para maximizar los ingresos mediante la asignación óptima de billetes a cada tipo de tarifa. En el Solver, definí la función objetivo como la suma de los ingresos por la venta de billetes, estableciendo las restricciones de capacidad de asientos, peso, y los billetes mínimos por tarifa. El *Solver* ajustó automáticamente los valores de las variables, encontrando así la combinación más eficiente que maximiza los beneficios de la aerolínea dentro de las limitaciones establecidas.

Para afianzar el modelo y familiarizarme con la sintaxis de *GLPK*, implementé este mismo problema en dicho lenguaje. Esto implicó trasladar la estructura de variables, parámetros y restricciones del modelo inicial a *GLPK*, conservando la función objetivo de maximizar ingresos y manteniendo todas las restricciones planteadas. Una vez ejecutado, corroboré que los resultados obtenidos coincidían, verificando así la precisión de la implementación en ambos enfoques.

2.2. Modelo Avanzado en GLPK

Para modelar el problema de asignación de aterrizajes en pistas y minimizar los costes asociados a retrasos, definimos los elementos clave a través de variables, parámetros y restricciones:

Variables:

x_{ijk}: Variable binaria que indica si el avión *i* aterriza en la pista *k* en el slot de tiempo *j*. La cual siempre debe ser positiva.

{1 si el avión *i* aterriza en la pista *k* en el slot de tiempo *j*; 0 en caso contrario}

Parámetros:

$Coste_i$: Coste de retraso por minuto para cada avión *i*, utilizado en la función objetivo para cuantificar los costes de retraso.

$Llegada_i$ Hora de llegada programada para cada avión *i*, expresada en minutos desde medianoche.

$Hora\_Límite_i$ Hora límite de aterrizaje para cada avión *i*, a partir de la cual el avión ya no debería aterrizar.

$Hora\_Inicio_j$: Hora de inicio de cada slot *j*, expresada en minutos desde medianoche.

$Disponibilidad_{jk}$: Matriz que indica la disponibilidad de slots en cada pista *k* y tiempo *j*. 1 si está disponible y 0 si está ocupado.

Función Objetivo:

La **función objetivo** en este modelo tiene como objetivo minimizar el coste total de retraso de todos los aviones, calculado como:

$$Min\ z = \sum_{i=1}^{5} \sum_{k=1}^{4} \sum_{j=1}^{n} C_i \cdot (t\_inicio_j - h\_prog_i) \cdot x_{ijk}$$

- $coste_i$: es el coste de retraso por minuto del avión *i*.
- t_inicio_j: representa el tiempo de inicio del slot *j*.
- h_prog_i: es la hora de llegada programada del avión *i*.

Restricciones:

1. Asignación de slots: Cada avión debe tener asignado un único slot de aterrizaje, asegurando que todos puedan aterrizar en una pista disponible y dentro de un tiempo válido.

$$R_1: \sum_{k=1}^{4}\sum_{j=1}^{n} x_{ijk} = 1 \qquad \forall i$$

2. Disponibilidad: Un slot solo puede asignarse si está disponible en la pista correspondiente. Esto limita las opciones de aterrizaje según la disponibilidad.

$$R_2: x_{ijk} \leq disponibilidad_{jk} \qquad \forall ijk$$

3. Exclusividad de slots: Solo un avión puede ocupar un slot específico en una pista. Esto evita conflictos de aterrizaje en los mismos slots.

$$R_3: \sum_{i=1}^{5} x_{ijk} \leq 1 \qquad \forall j, k$$

4. Límites de tiempo: Solo se asignan slots a los aviones dentro de su rango de llegada programada y límite de aterrizaje.
   a. *Llegada programada*: Restringe los slots antes de la hora programada.
   b. *Límite máximo*: Impide aterrizajes después de la hora límite.

$$R_4: h\_prog_i \leq x_{ijk} \cdot t_j \leq h\_lim_i \qquad \forall i, k$$

5. No consecutividad: Evita asignar slots consecutivos en la misma pista para un avión.

$$R_5: x_{ijk} + x_{ik(j+1)} \leq 1 \qquad \forall j, k$$

Este modelo asegura la asignación óptima de aterrizajes, minimizando los costes de retraso al considerar tanto la disponibilidad de slots como las limitaciones de tiempo y recursos en cada pista.

Para la implementación en *GLPK* de este modelo de minimización de costes de retraso, organicé las variables de tiempo en minutos desde las 00:00 para cada slot de aterrizaje, esta disposición en minutos facilita la comparación directa con los parámetros de llegada y límites de aterrizaje de cada avión, y nos evita la utilización de números negativos en caso de tomar como hora inicial las 9:00. Por ejemplo: $8:55 \ = \ 535\, minutos$

También estructuré una matriz para la disponibilidad de modo que los slots disponibles para cada pista se representen como “1”, mientras que los ocupados se representan como “0”, la matriz permite filtrar rápidamente los slots libres para cada pista en momentos específicos.

Inicialmente, para modelar la restricción de tiempo de aterrizaje, empleé una variable auxiliar *M* suficientemente grande para restringir los valores de x_{ijk} de modo que fueran “1” únicamente si el slot t_inicio_j estaba dentro del intervalo permitido para cada avión.

Sin embargo, para mejorar la eficiencia, implementé la restricción de tiempo mediante un subconjunto adicional, el cual filtra solo los slots válidos dentro del intervalo de tiempo permitido para cada avión. Al restringir el modelo a usar únicamente este subconjunto en las asignaciones, evitamos la necesidad de utilizar *M* para la limitación de tiempo, lo que simplifica el cálculo y mejora la claridad y eficiencia del modelo. Así, cada avión se asigna exclusivamente a los slots válidos en su rango de tiempo, cumpliendo la restricción de tiempo evitando la complejidad y optimizando el proceso.

Al ejecutar este modelo en *GLPK*, obtuvimos un resultado óptimo, logrando la asignación de slots de aterrizaje minimizando los costes de retraso, mediante una solución precisa y eficiente.

2.3. Modelo Fusionado en GLPK

Para fusionar ambos problemas, simplemente he unido sus modelos, sin necesidad de realizar cambios en las variables ni en las restricciones originales de cada parte. Así, las variables de asignación de tarifas y asientos y las restricciones de capacidad de los aviones permanecen intactas, al igual que las variables y restricciones asociadas a los aterrizajes en slots y pistas.

Para diferenciar las variables de aterrizaje de las de asignación de tarifas, he cambiado la notación de x_{ijk} a y_{ijk} en el segundo modelo, manteniendo la estructura y funcionalidad de las restricciones originales. De este modo, cada conjunto de restricciones y parámetros sigue cumpliendo su función en el modelo fusionado sin conflicto alguno.

La fusión de ambos problemas se logra sumando las funciones objetivo de cada parte: la maximización del beneficio por venta de billetes y la minimización de los costes de retraso. Esto permite optimizar ambos aspectos simultáneamente, sin modificar la lógica interna de los modelos originales.

Variables

x_{ij}: Número de billetes de cada tipo de tarifa *j* asignados al avión *i*.

y_{isk}: Variable binaria de asignación que toma el valor 1 si el avión *i* aterriza en la pista *k* en el slot de tiempo *s*; 0 en caso contrario.

Parámetros

- **Tarifas y Restricciones de Asientos (Parte 1)**:

$Precio_j$: Precio por billete para cada tarifa *j*.

$Peso_j$: Peso asociado al tipo de tarifa *j*.

$Asientos_i$: Capacidad máxima de asientos para cada avión *i*.

$Capacidad_i$: Capacidad máxima de peso para cada avión *i*.

- **Costes y Tiempos de Aterrizaje (Parte 2)**:

$Coste_i$: Coste de retraso por minuto para cada avión *i*.

$Llegada_i$: Hora de llegada programada para cada avión *i*.

$Hora\_Limite_i$: Hora límite de aterrizaje para cada avión i.

$Hora\_Inicio_s$: Hora de inicio de cada slot *s*.

$Disponibilidad_{sk}$: Matriz binaria que indica si el slot *s* en la pista *k* está disponible (1) o no (0).

Función Objetivo

Maximizar el beneficio de las tarifas y minimizar el coste de retraso:

$$Max\ z = \sum_{i=1}^{5} (19x_{i1} + 49x_{i2} + 69x_{i3}) - \sum_{i=1}^{5} \sum_{k=1}^{4} \sum_{s=1}^{n} C_i \cdot (t\_inicio_j - h\_prog_i) \cdot x_{isk}$$

Con esta fusión, el modelo resultante incorpora tanto la maximización del beneficio por tarifas como la minimización de costes de retraso, sin cambios en la estructura original de cada submodelo.

Restricciones

- **Restricciones de Asientos y Peso**:

1. Capacidad de asientos: Para cada avión, el número total de billetes no debe superar la capacidad de asientos.

$$R_1 : x_{i1} + x_{i2} + x_{i3} \leq \ Asientos_i$$

2. Capacidad de peso: La suma de pesos de los billetes no debe superar la capacidad de peso para cada avión.

$$R_2 : x_{i1} + 20x_{i2} + 40x_{i3} \leq \ PesoMáx_i$$

3. Mínimo tarifa Leisure Plus: Se deben vender al menos 20 billetes de esta tarifa por avión:

$$R_3 : x_{i2} \geq 20$$

4. Mínimo tarifa Business: Se deben vender al menos 10 billetes de la tarifa por avión:

$$R_4 : x_{i3} \geq 10$$

5. Proporción de billetes Estándar: Al menos el 60% de todos los billetes vendidos en cada avión deben corresponder a la tarifa 'Estándar':

$$R_5 : \sum_{i=1}^{5} x_{i1} \geq 0.6 \sum_{j=1}^{3} \sum_{i=1}^{5} x_{ij}$$

- **Restricciones de Aterrizaje**:

6. Asignación de slots: Cada avión debe tener asignado un único slot de aterrizaje, asegurando que todos puedan aterrizar en una pista disponible y dentro de un tiempo válido.

$$R_1 : \sum_{k=1}^{4} \sum_{s=1}^{n} x_{isk} = 1 \qquad\qquad \forall i$$

7. Disponibilidad: Un slot solo puede asignarse si está disponible en la pista correspondiente. Esto limita las opciones de aterrizaje según la disponibilidad.

$$R_2:\ x_{isk} \leq disponibilidad_{sk} \qquad \forall ijk$$

8. Exclusividad de slots: Solo un avión puede ocupar un slot específico en una pista. Esto evita conflictos de aterrizaje en los mismos slots.

$$R_3:\ \sum_{i=1}^{5} x_{isk} \leq 1 \qquad \forall j, k$$

9. Límites de tiempo: Solo se asignan slots a los aviones dentro de su rango de llegada programada y límite de aterrizaje.

$$R_4:\ h\_prog_i \leq x_{isk} \cdot t_s \leq h\_lim_i \qquad \forall i, k$$

10. No consecutividad: Evita asignar slots consecutivos en la misma pista para un avión.

$$R_5:\ x_{isk} + x_{ik(s+1)} \leq 1 \qquad \forall j, k$$

Para implementar el modelo fusionado en *GLPK*, mantuve las variables y restricciones de ambos problemas originales, adaptando las estructuras y organizando los parámetros de forma que ambos aspectos se pudieran optimizar simultáneamente. La implementación en *GLPK* incluye la combinación de las variables de asignación de billetes y las de aterrizaje, así como la unión de ambas funciones objetivo. Para ello, utilicé una suma que agrega el beneficio de los billetes vendidos y resta el coste de los retrasos en el aterrizaje.

Similar al enfoque en el problema 2, utilicé el subconjunto de slots válidos para cada avión para gestionar los tiempos. De esta forma, filtra previamente los slots válidos según el intervalo entre la llegada programada y el límite de aterrizaje de cada avión, permitiendo que solo se consideren los slots viables para cada avión.

Además, como en el problema de aterrizaje, mantuve la matriz de disponibilidad en función de minutos, asignando cada slot de acuerdo con su inicio en el día. De esta forma, la disponibilidad y los tiempos se alinean, lo que facilita el cumplimiento de restricciones de espacio y tiempos para cada pista y slot.

Finalmente, al ejecutar el modelo en *GLPK*, obtuvimos una solución óptima que simultáneamente maximiza los beneficios y minimiza los costes de retraso, cumpliendo con las restricciones de cada uno de los problemas originales.

3. Análisis de Resultados

En el análisis de resultados de esta práctica, el modelo fusionado en *GLPK* ha permitido obtener una solución óptima que maximiza los beneficios derivados de la venta de billetes y minimiza los costes asociados a los retrasos en los aterrizajes. La solución cumple con todas las restricciones establecidas en el enunciado, incluyendo la capacidad máxima de peso y asientos por avión, los porcentajes mínimos de ciertos tipos de billetes, y los límites de disponibilidad de las pistas para cada slot de tiempo. La estructura en *GLPK* ha permitido mantener ambas funciones objetivo, generando un modelo que se adapta a las necesidades de una aerolínea en términos de ingresos y eficiencia en los aterrizajes.

Restricciones limitantes del problema: Las principales restricciones que limitan la solución incluyen las relativas a la capacidad de asientos y peso en cada avión, así como la disponibilidad de slots en cada pista y el cumplimiento de los tiempos de aterrizaje programados y límite de cada vuelo. Estas restricciones impiden que el modelo se optimice en uno u otro aspecto sin respetar los límites de cada categoría, creando un equilibrio en el que la asignación de billetes y la programación de aterrizajes deben coordinarse para no exceder ninguna de las capacidades o tiempos definidos.

Análisis de complejidad: El modelo presenta una complejidad significativa debido al número de variables y restricciones. En este caso, hemos definido variables para la asignación de billetes por tarifa y avión, así como para la asignación de slots de aterrizaje en función de cada avión y pista. Además, el número de restricciones también es considerable, dado que cada restricción controla un aspecto específico de capacidad o tiempo, generando un sistema complejo de condiciones interdependientes.

Modificando los parámetros o variando el número de aviones y pistas, la dificultad de resolución se incrementa considerablemente. Por ejemplo, al añadir más aviones o más pistas, el espacio de búsqueda se amplía, y el sistema necesita gestionar más combinaciones posibles de aterrizajes y asignaciones de billetes. Esto aumenta el número de restricciones y complica la obtención de una solución óptima, requiriendo más recursos computacionales para evaluar cada combinación.

Consideración de retrasos en las llegadas: En la segunda parte de la práctica, los retrasos en la llegada de vuelos podrían provocar la necesidad de replantear la asignación de slots en las pistas. Si consideramos un retraso de 20 minutos en el avión AV1, sería necesario actualizar el parámetro de "hora de llegada programada" para ese avión hasta las 9:30. Esto permitiría al modelo adaptarse al nuevo tiempo de llegada y minimizar más los costes, en este caso hasta 4000, bajando 500.

Ventajas y desventajas de LibreOffice y GLPK: *LibreOffice Calc* y *GLPK* presentan ventajas y desventajas según la fase del modelado. *Calc* es intuitivo y facilita la organización y visualización de datos para problemas sencillos, especialmente en la asignación inicial de tarifas y optimización básica de beneficios mediante el *Solver*.

Sin embargo, su capacidad es limitada para problemas con múltiples variables y restricciones complejas. Por otro lado, *GLPK*, aunque requiere mayor conocimiento en la estructura de programación y modelado, es capaz de gestionar problemas más grandes y sofisticados, optimizando múltiples objetivos y respetando restricciones de mayor complejidad. Su desventaja es la mayor curva de aprendizaje, lo cual lo hace menos accesible para usuarios que no estén familiarizados con la programación matemática. En conclusión, Calc es ideal para cálculos y optimizaciones básicas, mientras que *GLPK* es más adecuado para problemas avanzados y altamente estructurados como el de esta práctica.

4. Conclusiones

En conclusión, los modelos planteados en esta práctica me han permitido obtener una solución óptima al problema descrito por varios medios. Podemos comprobar que el enfoque es aplicable a otras situaciones en las que los parámetros varíen, siempre y cuando se mantenga un sentido lógico y realista en las restricciones y los datos. Durante el desarrollo del trabajo, el modelo ha sido perfeccionado a medida que analizaba los resultados que iba obteniendo y tratando de alcanzar la solución solicitada.

Una de las principales dificultades que he encontrado fue al modelar el segundo ejercicio, donde trabajé con una matriz de disponibilidad en el problema de programación lineal. Al no haber trabajado previamente con este tipo de estructuras en *GLPK*, inicialmente me resultó confuso. Sin embargo, con el manual proporcionado y un poco de investigación logré entender mejor cómo aplicar matrices en este contexto, lo que facilitó la resolución del modelo y me permitió implementar las restricciones de forma efectiva.

Para finalizar, esta práctica ha sido fundamental para afianzar mis conocimientos en modelado de problemas avanzados mediante programación lineal. Además, he tenido la oportunidad de explorar herramientas útiles como *Solver* y *GLPK*, lo que ha mejorado mi comprensión de las aplicaciones de optimización en la práctica y me ha dado una base sólida para resolver problemas complejos en el futuro.

Optimización de Problemas en Programación Lineal

Realizado por:
Carlos Bravo Garrán

Asignatura:
Heurística y Optimización

Profesor:
Cesar Alberto Escribano Otero

Institución:
Universidad Carlos III de Madrid

Fecha de entrega:
28/10/2024