



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

UN MÉTODO DE BÚSQUEDA LOCAL PARA EL PROBLEMA DE ASIGNACIÓN DE
TRIPULACIONES UTILIZANDO ALGORITMOS DE GRAFOS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA,
MENCIÓN MATEMÁTICAS APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO

IGNACIO VICENTE RIEGO HUNT

PROFESOR GUÍA:
JORGE AMAYA ARRIAGADA

MIEMBROS DE LA COMISIÓN:
FRANCO BASSO SOTZ
FERNANDO ORDÓÑEZ PIZARRO
JAIME ORTEGA PALMA

SANTIAGO DE CHILE

2022

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS DE LA
INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
Y MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO
CIVIL MATEMÁTICO
POR: IGNACIO VICENTE RIEGO HUNT
FECHA: 2022
PROF. GUÍA: JORGE AMAYA ARRIAGADA

UN MÉTODO DE BÚSQUEDA LOCAL PARA EL PROBLEMA DE ASIGNACIÓN DE TRIPULACIONES UTILIZANDO ALGORITMOS DE GRAFOS.

Dado un conjunto de viajes entre distintas estaciones, con tiempos de inicio y fin determinados, se plantea el problema de cubrir todos estos viajes utilizando una cantidad mínima de tripulaciones. Estos deben respetar que cada tripulación puede participar solamente en un viaje a la vez, que debe descansar entre cada par de viajes y que debe comenzar los viajes en las estaciones apropiadas. Adicionalmente se impone la restricción de que una vez terminado el período estudiado, los mismos viajes deben volver a realizarse con el mismo número de tripulaciones, por lo que la planificación debe ser compatible con el próximo periodo de tiempo.

El problema se modela construyendo un grafo adecuado, donde se plantea un método de búsqueda local que utiliza los algoritmos de camino de peso máximo y de matching bipartito como subrutinas para encontrar soluciones cada vez mejores.

Por otro lado, se demuestran dos cotas inferiores a la mínima cantidad de tripulaciones necesarias para resolver una instancia del problema. Una de estas se obtiene gracias a una versión simplificada del problema que puede ser resuelta en tiempo polinomial, mientras que la otra se obtiene al analizar el número mínimo de tripulaciones que se necesita en cada instante de la planificación.

En la memoria de Vicente.

Agradecimientos

Quiero extender mis agradecimientos a todas las personas e instituciones que me acompañaron y me enseñaron a lo largo de mi carrera.

A mis papás y mi hermano, por aguantarme, aconsejarme y prestarme un oído siempre que lo necesité.

A toda mi enorme familia por su alegría e incondicionalidad.

A Raimundo, Borja, Martín y Vicente. No puedo poner en palabras todo lo que han significado para mi.

A Eugenio y Claudia, por permitirme estar ahí cuando importaba.

A todos los amigos que hice en la universidad, especialmente a Felipe, Javier y Camila, ustedes son quienes dan sentido a todo esto.

Al club chilien por ser mi familia postiza en el extranjero.

Au BDI, vous êtes vraiment le paradis.

À Florent et Steven, pour tout ce que nous avons appris ensemble et pour avoir fait confiance sur moi.

谢谢 club 朋友，我想你们，我希望我们能很快再见面。

淑慧，你给我灵感和幸福

A todos mis profesores a quienes admiro profundamente, por el conocimiento que comparten con nosotros y por empujarnos a ser todo lo que podemos ser.

Al departamento de relaciones internacionales por construir los puentes sobre los que yo caminé.

A todas las personas que construyeron el conocimiento que forma la base sobre la cual nos paramos.

Tabla de Contenido

1. Introducción	1
1.1. Trabajos relacionados	1
1.2. Organización	2
1.3. Resultados	3
2. El problema	4
2.1. Las restricciones locales	5
2.1.1. La compatibilidad entre viajes	6
2.1.2. Los descansos legales	9
2.1.3. El modelo de grafo	10
2.2. Otras restricciones	11
2.2.1. Turnos y asignaciones	12
2.2.2. Rotación de turnos	13
2.2.3. Situación inicial de las tripulaciones	15
2.2.4. Restricciones globales	16
2.3. La función objetivo	19
3. Modelo de optimización lineal	20
3.1. Las variables	20
3.2. Las restricciones	21
3.3. Los problemas lineales	23

4. Casos de estudio	25
4.1. Caso MxN	25
4.2. Caso rotativo	26
5. El Algoritmo de Búsqueda Local	27
5.1. Heurísticas para la factibilidad	27
5.2. Espacio de búsqueda de los descansos legales	34
5.3. Solución inicial	38
5.4. Perturbaciones horizontales	38
5.5. Perturbaciones verticales	39
6. Resultados experimentales	42
6.1. Las instancias de optimización	42
6.2. Las cantidad de tripulaciones	43
6.3. Las soluciones iniciales	44
6.4. Búsqueda local fase 1	50
6.5. Comparación con Gurobi	50
7. Conclusión	53
Bibliografía	55

Índice de Tablas

2.1. Viajes de ejemplo	4
2.2. Tripulaciones de ejemplo	5
2.3. Ejemplo de la función γ	9
2.4. Ejemplo de la función $\bar{\gamma}$	15
3.1. Viajes de Ejemplo	22
6.1. Resumen de las instancias estudiadas	42
6.2. Valores y tiempo de ejecución de la heurística de mejor turno	43
6.3. Valores y tiempo de ejecución de la heurística de mejor turno.	44
6.4. Cotas inferiores obtenidos por las distintas heurísticas aplicadas a \mathcal{R}_0	45
6.5. Tiempos de ejecución de las distintas heurísticas aplicadas a \mathcal{R}_0	46
6.6. Valores de S después de verificar factibilidad	47
6.7. Soluciones iniciales calculadas para cada instancia	47
6.8. Resultados de la búsqueda local inicial con 500 iteraciones	51
6.9. Resultados de la búsqueda local inicial con 2000 iteraciones	51
6.10. Resultados de 300 segundos de Gurobi comenzando desde una asignación vacía	52
6.11. Resultados de 300 segundos de Gurobi comenzando desde la solución inicial comparado con la búsqueda local fase 1 con 1500 iteraciones	52

Índice de Ilustraciones

2.1. Grafo de ejemplo	11
2.2. Ejemplo de asignación	13
2.3. Caso Repetitivo y Caso Rotativo	14
2.4. Ejemplo de asignación con tripulaciones.	16
2.5. Ejemplo de H	18
5.1. Ejemplo de instancia	28
5.2. Algoritmo para resolver el problema simplificado (PS)	33
5.3. Ejemplo visual de la demanda	35
6.1. Tiempo de ejecución según cantidad de viajes.	44
6.2. Algoritmo para determinar $S_{\text{asignación}}$	46
6.3. Demanda de las instancias 10x5 A y Rotativo A	48
6.4. Asignación de descansos legales \mathcal{R}_0 en la instancia 10x5 A	48
6.5. Asignación de descansos legales \mathcal{R}_0 en la instancia Rotativo A	49
6.6. Asignación de descansos aceptada.	49
6.7. Soluciones iniciales obtenidas	50

Capítulo 1

Introducción

En el contexto de la investigación de operaciones, el problema de asignar tareas a agentes que las pueden realizar surge muchas veces y con distintas variaciones. En aquellos casos en que las tareas son viajes entre puntos que deben ser realizados en momentos precisos, se habla de un problema de asignación de tripulaciones o *Crew Scheduling*.

Utilizando como punto de partida el caso de los conductores de trenes de carga en una empresa minera del norte de Chile, se plantea un problema de asignación de tripulaciones con restricciones adicionales. Por un lado, para obtener una planificación que sirva para el largo plazo, se exige que una vez terminado un periodo de planificación (semana, quincena, etc.), cada tripulación esté en posición para comenzar a trabajar en el periodo siguiente. Dado que se trata de un escenario real, las tripulaciones también tienen derecho a extensos periodos de descanso, los cuales deben ser realizados periódicamente. Además, se busca que el régimen de trabajo resultante sea equitativo en términos de carga de trabajo para las distintas tripulaciones.

Este problema se modela a través de un grafo dirigido cuyos vértices son viajes y descansos. Las tripulaciones son agentes que deben atravesar el grafo, donde un camino equivale a un turno de trabajo. Con este modelo se observa un parentesco con los problemas de ruteo de vehículos y del vendedor viajero múltiple. Sin embargo, las restricciones adicionales y el objetivo de tener cargas de trabajo similares hacen que los métodos de resolución de estos problemas clásicos no se adapten bien al caso estudiado.

1.1. Trabajos relacionados

Como se describe en [10], [8], los problemas de Crew Scheduling tienen una larga historia y utilizan distintos modelos dependiendo de cuáles aspectos se quieren enfatizar en el caso particular. Además de una diversidad de modelos, múltiples heurísticas de optimización se han utilizado para las distintas variantes del problema.

En [12] se emplea un algoritmo genético para un caso de asignación de tripulaciones en aerolíneas, donde se plantea el problema como una instancia de Set Partitioning, donde los

elementos son los viajes a realizar y los conjuntos son las posibles rutas que puede realizar una tripulación. En [15] se estudian tripulaciones de trenes en las que también se utiliza un modelo de Set Partitioning que luego se resuelve utilizando un método de Ant Colony Optimization, de forma similar a [17]. En [16] se considera un caso de asignación de tripulaciones de aerolíneas, en que las tripulaciones son heterogéneas y tienen preferencias sobre cuales viajes realizar. Este se resuelve de manera aproximada a través de un algoritmo de descomposición lagrangeana. En [5] se plantea un modelo para asignación de conductores de buses y se presenta un método de búsqueda aleatoria glotona adaptativa (GRASP por su sigla en inglés).

También se ha modelado el problema de Crew Scheduling como un problema de grafos: existen ejemplos como [6] donde se replantea el problema como una versión del vendedor viajero y sobre este se utiliza un algoritmo de colonia de hormigas. En [9] se utiliza un modelo de partición de grafos en instancias muy grandes para dividir los viajes en subproblemas que se resuelven independientemente. De forma similar, [14] reduce el problema a encontrar caminos sobre un grafo y emplea un algoritmo genético para construir soluciones.

El problema estudiado tiene sus orígenes en los modelos planteados en los artículos anteriores [2], [1]. En estos se plantean versiones particulares del modelo que será generalizado en esta tesis y se emplea un método de programación lineal para encontrar soluciones factibles, seguidas de una búsqueda local para mejorar dichas soluciones.

1.2. Organización

Se comienza en el capítulo 2 con una descripción detallada del modelo visto como un grafo, al igual que una formulación de programación lineal. Como parte de la descripción del modelo, se explicitan las propiedades matemáticas que debe cumplir una instancia para ser resuelta con estos métodos.

Una vez planteado el modelo, en el capítulo 4 se presentan los casos particulares que serán estudiados en la sección experimental. Ambos provienen de situaciones enfrentadas por la empresa minera y constan de instancias reales sobre las cuales se podrán evaluar los algoritmos.

En el capítulo 5 se introduce un algoritmo para la resolución aproximada del problema. La explicación va acompañada de un análisis de complejidad para cada una de sus componentes y, en los casos en que corresponde, demostraciones de su correctitud.

Finalmente, en el capítulo 6 se presentan los resultados experimentales del algoritmo sobre todas las instancias previamente descritas. Para cada fase del algoritmo se acompaña con ejemplos y datos sobre los tiempos de ejecución. También se comparan los resultados obtenidos con el solver comercial **Gurobi**.

1.3. Resultados

Los principales aportes de este trabajo consisten en la introducción de un modelo para el problema de asignación de tripulaciones que permite aprovechar propiedades de los grafos involucrados. Para este modelo, se prueban tres cotas inferiores para el número de tripulaciones necesarias para cubrir un conjunto de viajes. Finalmente se introduce un algoritmo de búsqueda local basado en lo anterior, acompañado de resultados experimentales que muestran su eficiencia.

Las tres cotas inferiores presentadas representan un mínimo para el número de tripulaciones necesarias para realizar un conjunto de viajes. De esta forma se logra disminuir el espacio de búsqueda, restringiéndolo solamente a los números de tripulaciones factibles. Estas cotas se pueden calcular en tiempo polinomial y son confirmadas por buenos resultados experimentales.

El algoritmo de búsqueda local aprovecha las características del problema para poder explorar el espacio de búsqueda sin salir del espacio factible. Esto es de especial interés dado que este problema tiene una alta cantidad de restricciones, lo que dificulta los procesos de búsqueda local. Este método está además respaldado por los resultados experimentales, en los que no solamente logra alcanzar valores óptimos o cercanos al óptimo, si no que además lo logra con gran rapidez.

Las ganancias en términos de interpretabilidad y velocidad de ejecución presentadas en este trabajo permiten encontrar en minutos soluciones a problemas que tomarían horas usando un solver comercial. De esta forma, permite tratar instancias que por su tamaño serían imposibles de resolver como problemas lineales.

Así, se transforma el proceso de planificación en un proceso dinámico en el que una firma puede generar planificaciones, evaluarlas y luego hacer ajustes, agilizando la toma de decisiones y permitiendo que se tomen en cuenta múltiples escenarios.

Capítulo 2

El problema

El problema a estudiar consiste en que, dados un conjunto de viajes que se deben realizar y un conjunto de tripulaciones encargadas de realizarlos, se debe encontrar una forma de asignar los viajes a las tripulaciones de tal manera que se cumplan las restricciones correspondientes y que todos los viajes sean realizados.

Sea un conjunto de viajes \mathcal{V}_0 , que transcurren en un intervalo de planificación $[0, t_{\text{máx}}]$ medido en minutos, pero que también podría estar medido en días u horas dependiendo del caso. Cada viaje $v \in \mathcal{V}_0$ consta de:

- El intervalo de tiempo en el que se realiza, $[t_v^i, t_v^f]$ y su duración $\Delta t_v = t_v^f - t_v^i$. Además se pide que $t_v^i \in [0, t_{\text{máx}}]$, es decir que el tiempo de fin puede estar fuera del intervalo de planificación $[0, t_{\text{máx}}]$ pero el tiempo de inicio debe estar dentro.
- El origen y destino geográfico I_v, F_v . Estos toman valores en el conjunto de estaciones \mathcal{J} .

Los viajes pueden implícitamente incluir múltiples paradas o procesos, por lo que un viaje puede comenzar y terminar en la misma estación y dos viajes entre el mismo par de estaciones pueden tener duraciones distintas. La información de los procesos que componen un viaje es irrelevante para el modelo.

A modo de ejemplo, en la tabla 2.1 se presenta un caso con tan sólo cuatro viajes en que $t_{\text{máx}} = 10080$ minutos, lo que corresponde a una semana. El conjunto de estaciones está dado por $\mathcal{J} = \{A, C\}$.

ID	I	F	t^i	t^f	Δt
AA0	A	A	300	3180	2880
AA1	A	A	6060	8940	2880
AC	A	C	2040	4920	2880
CA	C	A	7200	9880	2680

Tabla 2.1: Viajes de ejemplo

Además, se considera un conjunto de tripulaciones \mathcal{C} (por su nombre en inglés, Crew) las cuales pueden realizar estos viajes. Todas las tripulaciones pueden realizar cualquier viaje, pero están sujetas a distintas condiciones iniciales. Cada tripulación $c \in \mathcal{C}$ está definida por

- Su tiempo de inicio, es decir, en qué momento pueden comenzar a realizar viajes, denotado por $t_c \in [0, t_{\text{máx}}]$
- Su lugar de inicio, es decir, la estación en la que se encuentran al comenzar la planificación, denotado por F_c . Este toma valores en el conjunto de estaciones \mathcal{J} .

Sólo se consideran las condiciones de inicio de una tripulación, ya que estas corresponden a los trabajos o condiciones previas en que se encuentran antes de comenzar el período estudiado.

A modo de ejemplo, se presenta un caso con dos tripulaciones disponibles, con sus tiempos de inicio y orígenes correspondientes.

ID	t_c	F_c
TRA	0	A
TRC	1440	C

Tabla 2.2: Tripulaciones de ejemplo

En lo que sigue se detallarán las distintas restricciones que aplican al problema.

2.1. Las restricciones locales

En esta sección se tratarán aquellas restricciones que determinan cuáles viajes pueden ser realizados de forma consecutiva por una misma tripulación. Las restricciones que involucran viajes que no son consecutivos serán tratadas más adelante en la sección siguiente.

Algunas restricciones son consecuencias evidentes de la descripción del problema. A modo de ilustración, una tripulación no puede comenzar un viaje hasta haber terminado el viaje anterior. Por otro lado, distintas condiciones de trabajo pueden requerir condiciones adicionales, por ejemplo, que cada tripulación puede comenzar solamente un viaje cada día.

Para tomar en cuenta estos posibles cambios, la construcción del problema se realiza de forma modular, es decir que los distintos tipos de restricciones no se afectan entre sí.

En esta sección se presentan tres posibles restricciones locales que se resumen en la función γ . En las otras secciones del trabajo siempre se va a considerar la función γ como dada, sin entrar en los detalles de su definición. De esta manera, independientemente de los parámetros elegidos para las restricciones locales o del conjunto de restricciones que entren en consideración, esto no va a afectar la construcción de las otras restricciones.

2.1.1. La compatibilidad entre viajes

Por distintas razones puede haber pares de viajes que son incompatibles entre sí, es decir que no pueden ser realizados por la misma tripulación en un turno de trabajo. Un ejemplo claro son los viajes cuyos intervalos se intersectan, como es el caso de los viajes AA0 y AC en la tabla 2.1, al momento de comenzar AC en el minuto 2040, la tripulación aún está ocupada realizando AA0 que transcurre en el intervalo $[300, 3180]$.

Si bien pueden haber muchos tipos de restricciones entre viajes, en esta sección se tratarán todas aquellas restricciones que cumplen la propiedad de transitividad que se define a continuación. Para modelar una restricción i se define una función $\gamma_i : \mathcal{V}_0 \times \mathcal{V}_0 \rightarrow \{0, 1\}$ que asigna 1 a los pares de viajes compatibles según esta restricción y 0 a los que no lo son.

Definición 2.1 *Una restricción γ_i es transitiva en tiempo creciente si cumple*

$$\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{V}_0, \quad t_u^i \leq t_v^i \leq t_w^i, \quad \gamma_i(\mathbf{u}, \mathbf{v}) = 1 \wedge \gamma_i(\mathbf{v}, \mathbf{w}) = 1 \implies \gamma_i(\mathbf{u}, \mathbf{w}) = 1.$$

Muchas de las restricciones a considerar cumplen esta propiedad y serán detalladas en este capítulo. Vale la pena estudiar estas restricciones de forma independiente, ya que se prestan muy bien para escribir el problema utilizando grafos. Las restricciones de compatibilidad que no son transitivas en tiempo creciente serán tratadas en la sección 2.2.

La primera restricción a considerar es la compatibilidad temporal. Desde que una tripulación comienza un viaje, no puede comenzar uno nuevo hasta que termine el primero, tenga el tiempo de descansar una cantidad apropiada y tenga el tiempo de trasladarse hasta el lugar de inicio del viaje siguiente.

El tiempo de descanso entre viajes se describe con el parámetro $\delta \geq 0$, es decir que después de terminar un viaje se debe descansar al menos δ unidades de tiempo antes de comenzar otro. Por otro lado, una tripulación que termina un viaje en una estación $x \in \mathcal{J}$, no puede comenzar un viaje partiendo desde otra estación $y \in \mathcal{J}$ a menos de que tenga la oportunidad de transportarse entre las dos estaciones. Este tiempo de transporte no cuenta dentro del tiempo de descanso.

Se define entre cada par de estaciones $x, y \in \mathcal{J}$ la cantidad $\theta_{x,y} \geq 0$ que representa el tiempo necesario para que una tripulación se traslade desde la estación x hasta y , y que son consideradas datos del problema. Vale la pena notar que un viaje a realizar entre dos estaciones x e y puede durar más que la distancia $\theta_{x,y}$, y se debe a que un viaje puede incluir implícitamente más acciones que solamente el transporte entre las estaciones de inicio y fin. Sin embargo, un viaje nunca puede ser más corto que la distancia de traslado entre sus estaciones. También se asume que estas distancias cumplen la desigualdad triangular. A continuación se resumen estas propiedades de los $\theta_{x,y}$.

$$\begin{aligned} \forall x, y, z \in \mathcal{J}, \quad & \theta_{x,z} \leq \theta_{x,y} + \theta_{y,z} \\ \forall x \in \mathcal{J}, \quad & \theta_{x,x} = 0 \\ \forall \mathbf{u} \in \mathcal{V}_0, \quad & \theta_{I_u, F_u} \leq \Delta M_u. \end{aligned}$$

Juntando la restricción de descanso entre viajes relacionada a δ y el transporte entre estaciones relacionado a θ se define γ_0 según

$$\gamma_0(\mathbf{u}, \mathbf{v}) = 1 \iff \begin{cases} t_u^f + \delta + \theta_{F_u, I_v} \leq t_v^i & \text{si } t_u^i \leq t_v^i \\ \gamma_0(\mathbf{v}, \mathbf{u}) = 1 & \text{si } t_u^i > t_v^i. \end{cases}$$

Proposición 2.2 *La función γ_0 es transitiva en tiempo creciente.*

Demostración: En efecto, sean $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{V}_0$, con $t_u^i \leq t_v^i \leq t_w^i$, y tales que

$$\gamma_0(\mathbf{u}, \mathbf{v}) = \gamma_0(\mathbf{v}, \mathbf{w}) = 1.$$

Juntando las desigualdades provenientes de la definición de γ_0 se obtiene

$$t_u^f + \delta + \theta_{F_u, I_v} \leq t_v^i \quad \wedge \quad t_v^f + \delta + \theta_{F_v, I_w} \leq t_w^i.$$

Como además $t_v^i \leq t_v^f$, se puede reemplazar en la segunda desigualdad

$$t_u^f + \delta + \theta_{F_u, I_v} + \delta + \theta_{F_v, I_w} \leq t_w^i.$$

Luego, utilizando la desigualdad triangular se obtiene

$$t_u^f + \delta + \theta_{F_u, I_w} \leq t_u^f + 2\delta + \theta_{F_u, I_v} + \theta_{F_v, I_w} \leq t_w^i.$$

Luego se cumplen ambas condiciones para que $\gamma_0(\mathbf{u}, \mathbf{w}) = 1$, es decir que γ_0 es transitiva a tiempo creciente. ■

La restricción γ_0 está presente en todos los casos a estudiar, pero también existen otras restricciones transitivas a tiempo creciente que están presentes solamente en algunos de los casos.

Por ejemplo, la restricción de tiempo entre comienzos de viaje se da solamente en algunas de las instancias estudiadas. Esta restricción exige que entre el comienzo de un viaje y el comienzo del viaje siguiente exista al menos un periodo de largo δ_{inicio} . Por lo que podemos definir γ_{inicio} según

$$\gamma_{\text{inicio}}(\mathbf{u}, \mathbf{v}) = 1 \iff |t_v^i - t_u^i| \geq \delta_{\text{inicio}}.$$

Proposición 2.3 *Esta restricción también es transitiva a tiempo creciente.*

Demostración: Sean $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{V}_0$, con $t_u^i \leq t_v^i \leq t_w^i$, y tales que

$$\gamma_{\text{inicio}}(\mathbf{u}, \mathbf{v}) = \gamma_{\text{inicio}}(\mathbf{v}, \mathbf{w}) = 1.$$

Gracias al orden de los tiempos de inicio se pueden remover los valores absolutos y se tiene

$$t_v^i - t_u^i \geq \delta_{\text{inicio}} \quad \wedge \quad t_w^i - t_v^i \geq \delta_{\text{inicio}}.$$

Basta con sumar ambas desigualdades para obtener

$$t_w^i - t_u^i = |t_w^i - t_u^i| \geq 2\delta_{\text{inicio}} \geq \delta_{\text{inicio}}.$$

Con lo que se concluye que γ_{inicio} es transitiva a tiempo creciente. ■

Finalmente, la última de las restricciones transitivas a tiempo creciente a estudiar es la de los subintervalos exclusivos. Sea $\{a_i\}_{i \in \{1, \dots, N\}} \subseteq [0, t_{\text{máx}}]$ una partición del intervalo de planificación. La restricción de subintervalos exclusivos exige que cada tripulación sólo puede comenzar un viaje en cada intervalo $[a_i, a_{i+1})$. A modo de ejemplo, si el intervalo de planificación es de una semana, esta restricción puede pedir que cada tripulación comience como máximo un viaje cada día. Con esto se define $\gamma_{\text{intervalos}}$ según

$$\gamma_{\text{intervalos}}(\mathbf{u}, \mathbf{v}) = 0 \iff \exists i \in \{1, \dots, N-1\}, \quad t_u^i, t_v^i \in [a_i, a_{i+1}).$$

La restricción $\gamma_{\text{intervalos}}$ es transitiva a tiempo creciente.

Demostración: En efecto, sean

$$\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{V}_0, \text{ con } t_u^i \leq t_v^i \leq t_w^i,$$

tales que

$$\gamma_{\text{intervalos}}(\mathbf{u}, \mathbf{v}) = \gamma_{\text{intervalos}}(\mathbf{v}, \mathbf{w}) = 1.$$

Por contradicción, sea i tal que $t_u^i, t_w^i \in [a_i, a_{i+1})$. Como los viajes $\mathbf{u}, \mathbf{v}, \mathbf{w}$ están en orden creciente de inicio, entonces necesariamente se tiene que $t_u^i, t_v^i, t_w^i \in [a_i, a_{i+1})$, lo que contradice la hipótesis $\gamma_{\text{intervalos}}(\mathbf{u}, \mathbf{v}) = \gamma_{\text{intervalos}}(\mathbf{v}, \mathbf{w}) = 1$. ■

Dado que cada versión del problema puede requerir la consideración de un conjunto particular de restricciones, de manera general se considera que el conjunto de índices I contiene las restricciones que son relevantes al problema en cuestión y se define una función γ como la intersección de estas restricciones.

$$\gamma(\mathbf{v}, \mathbf{w}) = 1 \iff \forall i \in I, \quad \gamma_i(\mathbf{v}, \mathbf{w}) = 1$$

La función γ es evidentemente transitiva a tiempo creciente por ser la intersección de funciones que cumplen la propiedad.

Continuando con el ejemplo anterior, asumiendo los parámetros $\theta_{C,A} = \theta_{A,C} = 600$ y $\delta = 600$, y $I = \{0\}$, se obtiene el siguiente resultado para la función γ

ID	Origen	Destino	t^i	t^f
AA0	A	A	300	3180
AA1	A	A	6060	8940
AC	A	C	2040	4920
CA	C	A	7200	9880

→

γ	AA0	AA1	AC	CA
AA0	0	1	0	1
AA1	1	0	0	0
AC	0	0	0	1
CA	1	0	1	0

Tabla 2.3: Ejemplo de la función γ

Se puede notar que al terminar AA0 hay tiempo para que la tripulación se traslade de A hasta C para comenzar CA, pero no hay suficiente tiempo para que ocurra lo mismo entre AC y AA1.

2.1.2. Los descansos legales

En algunas variantes del problema una tripulación puede tener derecho a un descanso legal, que corresponde a un periodo regular en que el operario o tripulación está libre de compromiso de ejecución de tareas. Estos serán modelados como viajes ficticios que no tienen restricciones espaciales, es decir que las tripulaciones pueden descansar en cualquier lugar. Los descansos legales suelen ser periodos de larga duración con respecto a los viajes, por ejemplo tres días completos de descanso en turnos de una semana.

Para modelar estos descansos se define un conjunto de viajes ficticios \mathcal{V}' . Este conjunto contiene una copia del viaje ficticio por cada momento en el que se podría realizar un descanso (por ejemplo, si puede realizarse cualquier día comenzando a las 21 horas). Estos viajes ficticios comienzan y terminan en una estación ficticia de descanso h tal que $\forall d \in \mathcal{J}, \theta_{h,d} = \theta_{d,h} = 0$. Dado que se asume que los descansos son relativamente largos, si una tripulación termina un viaje en la estación x , realiza un descanso y luego debe comenzar un viaje en una estación y , se asume que la tripulación se transporta entre x e y durante la duración de su descanso.

Se define el conjunto de todos los viajes $\mathcal{V}_0 \cup \mathcal{V}' = \mathcal{V}$. Con $\mathcal{V}' = \emptyset$ en el caso en que no hay descansos legales.

Estos descansos legales pueden tener restricciones más allá de las restricciones presentes para los viajes reales. Por ejemplo puede haber un mínimo de un descanso por turno o exigirse al menos un descanso cada cierta cantidad de días. Este tipo de restricciones serán modeladas como restricciones globales que serán explicadas en la sección del mismo nombre.

La función γ se extiende para incluir a los descansos legales de la siguiente manera.

$$\forall u \in \mathcal{V}, v \in \mathcal{V}', \quad \gamma(\mathbf{u}, \mathbf{v}) = 1 \iff t_u^f \leq t_v^i \quad \vee \quad t_v^f \leq t_u^i.$$

Es decir que los descansos legales ignoran todas las restricciones, reduciendo la compatibilidad a que los intervalos no intersecten. Dado que los descansos legales suelen ser de un

largo considerable, se asume que su extensión cumple las siguientes características:

$$\begin{aligned} \forall v \in \mathcal{V}', \quad \Delta t_v &\geq \theta_{x,y} + \delta & \forall x, y \in \mathcal{J} \\ \Delta t_v &\geq a_{i+1} - a_i & \forall i \in \{1, \dots, N-1\} \\ \Delta t_v &\geq \delta_{\text{inicio}}. \end{aligned}$$

Asumiendo que los descansos legales cumplen estas características, la transitividad a tiempo creciente de γ se mantiene al incluirlos.

2.1.3. El modelo de grafo

Para modelar este problema se define el grafo dirigido $G = (V, E)$, y una función

$$i : V \rightarrow \mathcal{V}, \quad i(v) \rightarrow \mathbf{v}.$$

Se denotan las pre imágenes $V_0 = i^{-1}(\mathcal{V}_0)$, $V' = i^{-1}(\mathcal{V}')$, tal que la restricción de la función i al conjunto V_0 es biyectiva. Notar que para los descansos legales puede haber más de un vértice en V' que corresponde a la misma imagen en \mathcal{V}' . En lo que sigue, para un vértice $v \in V$ se denotará $i(v) = \mathbf{v}$, es decir que los viajes están en negrita y sus vértices correspondientes están en texto normal.

Se define el conjunto de aristas E de tal forma que existe una arista $(u, v) \in E$ si y solo si los viajes \mathbf{u}, \mathbf{v} son compatibles y \mathbf{u} comienza antes que \mathbf{v} :

$$E = \{(u, v) \mid \gamma(\mathbf{u}, \mathbf{v}) = 1 \quad \wedge \quad t_u^i < t_v^i\}.$$

Gracias a la transitividad a tiempo creciente que se impuso sobre γ se puede demostrar la siguiente propiedad de G

Proposición 2.4 *Asumiendo que la función γ es transitiva a tiempo creciente, todo camino $P \subseteq G$, $P = v_{1N}$ cumple que*

$$\forall j, k \in \{1, \dots, N\}, \quad j < k, \quad \gamma(\mathbf{v}_j, \mathbf{v}_k) = 1$$

Demostración: Sea $v_1 v_2 \dots v_N = P$ camino en G . Dado que γ es simétrica, la propiedad que se quiere demostrar es equivalente a

$$\forall p \in \{1, \dots, N\}, \forall j \in \{1, \dots, N-p\} \quad \gamma(\mathbf{v}_j, \mathbf{v}_{j+p}) = 1$$

Se procederá por inducción sobre p .

Caso base $p = 1$: Sea $j \in \{1, \dots, N - 1\}$. Los vértices v_j, v_{j+1} son adyacentes en G por definición, por lo que se tiene que $\gamma(\mathbf{v}_j, \mathbf{v}_{j+1}) = 1$.

Paso inductivo: Se asume como hipótesis inductiva que

$$\forall j \in \{1, \dots, N - p\}, \quad \gamma(\mathbf{v}_j, \mathbf{v}_{j+p}) = 1$$

Sea $j \in \{1, \dots, N - p - 1\}$, por hipótesis inductiva se sabe que $\gamma(v_j, v_{j+p}) = 1$ y por el caso base se observa que $\gamma(v_{j+p}, v_{j+p+1}) = 1$. Por la construcción de G , los tiempos de inicio de los viajes asociados a los vértices v_1, \dots, v_N son crecientes, es decir que $t_{v_j}^i \leq t_{v_{j+p}}^i \leq t_{v_{j+p+1}}^i$. Gracias a la transitividad a tiempo creciente de γ se tiene

$$t_{v_j}^i < t_{v_{j+p}}^i < t_{v_{j+p+1}}^i \wedge \gamma(v_j, v_{j+p}) = \gamma(v_{j+p}, v_{j+p+1}) = 1 \implies \gamma(v_j, v_{j+p+1}) = 1.$$

Con lo que se deduce

$$\forall j \in \{1, \dots, N - p - 1\}, \quad \gamma(v_j, v_{j+p+1}) = 1.$$

Esto concluye la demostración ■

En la figura 2.1 se presenta el grafo correspondiente a los 4 viajes de la tabla 2.1 construido utilizando la función γ .

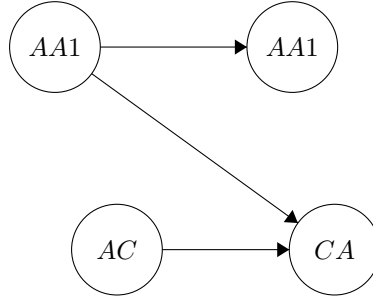


Figura 2.1: Grafo de ejemplo

2.2. Otras restricciones

Además de las restricciones locales descritas anteriormente, las soluciones que se desea encontrar deben cumplir con restricciones que van más allá de la interacción entre viajes inmediatamente contiguos.

Para describir estas restricciones se definen las estructuras de turno y asignación, que describen, respectivamente, los viajes realizados por una tripulación en un periodo de tiempo, y los turnos realizados por cada tripulación en el periodo de tiempo.

Utilizando estas definiciones se pueden introducir restricciones que gobiernan los descansos legales, por ejemplo imponiendo un número de descansos por turno o un número de descansos por periodo de tiempo trabajado. Estas estructuras también pueden utilizarse para considerar diferentes restricciones sobre los viajes, por ejemplo imponiendo que una tripulación no puede realizar 3 viajes nocturnos seguidos.

Si bien el modelo que se plantea permite incluir todo tipo de restricciones, para la sección experimental y algorítmica sólo se consideran restricciones de este tipo sobre los descansos, dejando el análisis de las restricciones no locales sobre viajes para trabajos futuros.

2.2.1. Turnos y asignaciones

Para poder definir restricciones no locales se definen los turnos y las asignaciones como las estructuras más generales que describen el problema. Los turnos de trabajo son series de viajes y descansos que pueden ser realizados por una tripulación a lo largo de un periodo de planificación. Por ejemplo si el periodo es de una semana, un turno de trabajo sería un itinerario de trabajo de una semana que puede ser realizado por una tripulación. Una asignación está formada por un conjunto de turnos y corresponde a los itinerarios de todas las tripulaciones en el periodo de planificación.

Todo turno de trabajo que podría ser realizado por una tripulación en un periodo de tiempo $[0, t_{\text{máx}}]$ puede ser representado por un camino $P \subseteq G$ ya que sus viajes son siempre compatibles entre sí. Por lo tanto se habla indistintamente de turnos y caminos en G .

El objetivo de este problema es encontrar una forma de realizar todos los viajes del conjunto \mathcal{V}_0 cumpliendo las restricciones y utilizando solamente la cantidad de tripulaciones posible S . La cantidad de tripulaciones no es necesariamente un parámetro del modelo, en el capítulo 5 se detalla cómo se elige este número. Asumiendo una cantidad de tripulaciones S se introduce el concepto de asignación.

Definición 2.5 *Una asignación es un conjunto de $\mathcal{P} = \{P_j\}_{j \in \{1, \dots, S\}}$ de caminos disjuntos en G . Se denota $V(\mathcal{P})$ el conjunto de vértices que pertenecen a algún camino en \mathcal{P} .*

Se dice que una asignación \mathcal{P} es completa si para todo viaje $v \in \mathcal{V}_0$, el vértice correspondiente v está en $V(\mathcal{P})$.

Continuando con el ejemplo, sobre el grafo G se define la asignación $\mathcal{P} = \{P_1, P_2\}$ de la siguiente forma

Hay distintos tipos de restricciones que pueden imponerse sobre las asignaciones, estas dependen del caso específico que se quiere modelar. A continuación se detallan los tipos de restricciones consideradas en este modelo.

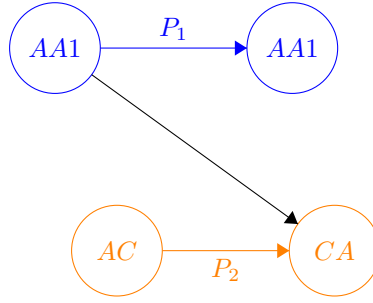


Figura 2.2: Ejemplo de asignación

2.2.2. Rotación de turnos

En algunas variantes del problema se impone que los mismos viajes tendrán que ser realizado por las mismas tripulaciones múltiples veces a lo largo del tiempo. Es decir que una vez que termina un periodo $[0, t_{\text{máx}}]$ con viajes \mathcal{V} comienza otro periodo igual con los mismos viajes.

Para enfrentar esta situación, los turnos se emparejan unidireccionalmente de tal forma que si s, s' están emparejados, una vez terminado un periodo de trabajo $[0, t_{\text{máx}}]$, la tripulación que acaba de realizar el turno s debe a continuación realizar el turno s' . Este emparejamiento de turnos se denota por $\mathcal{E} \subseteq \{1, \dots, S\}^2$.

Un posible régimen de emparejamiento que se estudiará es aquél en que cada turno está emparejado consigo mismo, es decir que cada tripulación repite el mismo turno una y otra vez (caso repetitivo).

Otro caso es aquel en que s se empareja con $s + 1 \pmod T$, es decir que cada tripulación eventualmente realiza cada viaje una vez en una rotación constante (caso rotativo). Ambos casos quedan ilustrados en la figura 2.3.

Si s está emparejado con s' , esto se denota por $\text{suc}(s) = s'$, y se define inductivamente $\text{suc}^{k+1}(s) = \text{suc}^k(\text{suc}(s))$, donde la notación suc indica *sucesor*.

Compatibilidad entre turnos

Considerando el caso en que los turnos están emparejados, se define la función $\bar{\gamma} : \mathcal{V}^2 \rightarrow \{0, 1\}$ la que, a diferencia de la función γ , considera la compatibilidad entre turnos. Es decir que $\bar{\gamma}(\mathbf{u}, \mathbf{v}) = 1$ si y sólo si es posible para una tripulación realizar \mathbf{u} en un periodo y realizar \mathbf{v} en el periodo siguiente. A modo de ejemplo, en el caso en que $[0, t_{\text{máx}}]$ es una semana, un viaje \mathbf{u} que termina en la noche del domingo no será compatible con un viaje \mathbf{v} que comienza el lunes en la madrugada siguiente, en ese caso $\bar{\gamma}(\mathbf{u}, \mathbf{v}) = 0$. Esto a pesar de que estos viajes si podrían ser compatibles si se realizan en la misma semana.

Para determinar $\bar{\gamma}(\mathbf{u}, \mathbf{v})$ se considera \mathbf{v}' , viaje con las mismas características que \mathbf{v} pero desplazado en el tiempo por $t_{\text{máx}}$

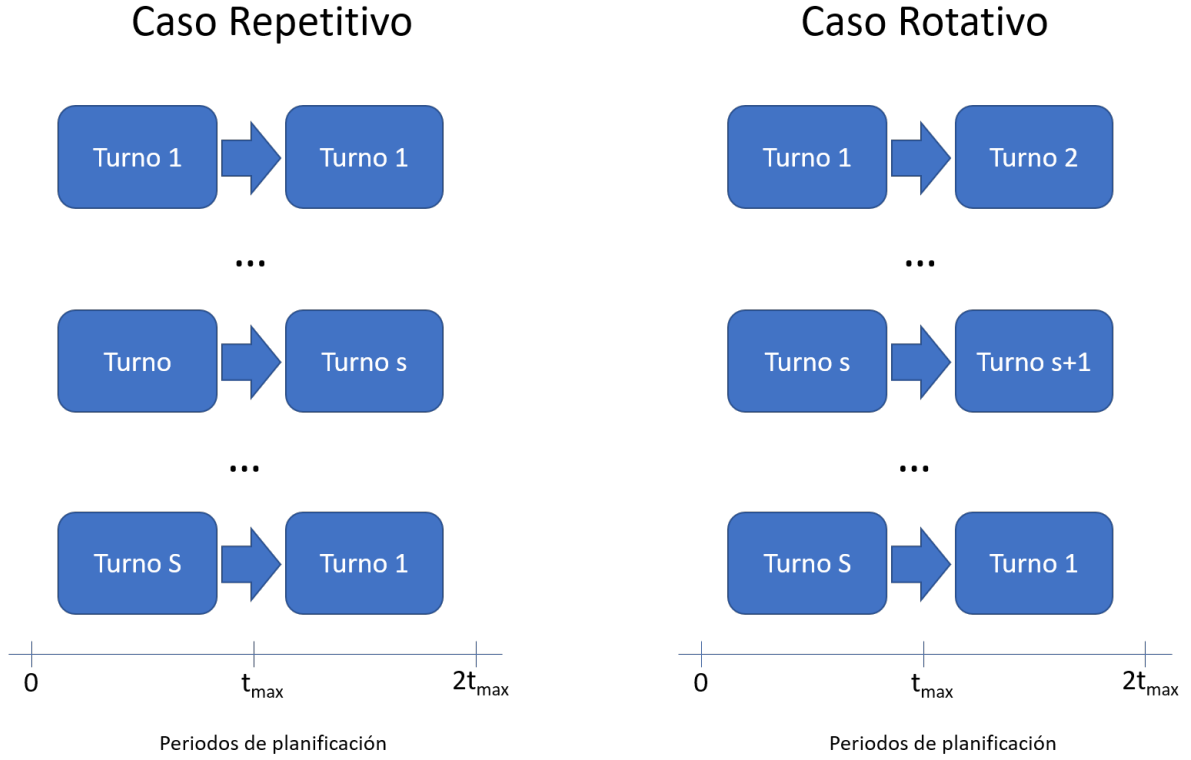


Figura 2.3: Caso Repetitivo y Caso Rotativo

$$t_{v'}^i = t_v^i + t_{\max}, \quad t_v^f = t_v^f + t_{\max}.$$

$$\bar{\gamma}(\mathbf{u}, \mathbf{v}) = \gamma(\mathbf{u}, \mathbf{v}').$$

En el caso de estar presente la restricción de subintervalos exclusivos, esta se ignora para efectos de $\bar{\gamma}$ ya que el intervalo en el que transcurre \mathbf{v}' está fuera de $[0, t_{\max}]$.

Utilizando la definición de $\bar{\gamma}$ se define un segundo conjunto de aristas F sobre los vértices V dado por

$$\forall u, v \in V, \quad (u, v) \in F \iff \bar{\gamma}(\mathbf{u}, \mathbf{v}) = 1.$$

Con esto se define la factibilidad entre turnos distintos.

Definición 2.6 Dada un conjunto de aristas F y dos caminos $P, P' \subseteq G$, se dice que el par ordenado (P, P') es compatible (respecto a $\bar{\gamma}$) si:

$$\forall u \in P, v \in P', \quad (u, v) \in F.$$

Además se dice que una asignación \mathcal{P} es compatible entre turnos si

$\forall s, s' \in \mathcal{E}, \quad \text{el par ordenado } (P_s, P_{s'}) \text{ es compatible.}$

Gracias a la transitividad a tiempo creciente, para dos caminos $P = u_1, \dots, u_k, \quad P' = v_1, \dots, v_l$ si se tiene que $(u_k, v_1) \in F$, entonces el par P, P' es compatible, es decir que basta con la compatibilidad de los extremos de los caminos.

Calculando la función $\bar{\gamma}$ con los viajes del ejemplo, utilizando nuevamente los mismos parámetros $\delta = 600, \theta_{A,C} = \theta_{A,C} = 600, t_{\text{máx}} = 10080$ se obtiene la tabla 2.4.

ID	Origen	Destino	t^i	t^f
AA0	A	A	300	3180
AA1	A	A	6060	8940
AC	A	C	2040	4920
CA	C	A	7200	9880

→

$\bar{\gamma}$	AA0	AA1	AC	CA
AA0	1	1	1	1
AA1	1	1	1	1
AC	1	1	1	1
CA	0	1	1	1

Tabla 2.4: Ejemplo de la función $\bar{\gamma}$

2.2.3. Situación inicial de las tripulaciones

Puede darse que las tripulaciones estén en distintos lugares al inicio de la planificación y/o que se liberen de sus trabajos anteriores a distintas horas.

En este caso se define la función $\eta : \mathcal{C} \times \mathcal{V} \rightarrow \{0, 1\}$ que a cada par tripulación-viaje asigna 1 si esta tripulación puede realizar el viaje en su primer turno y asigna 0 si no.

$$\forall \mathbf{c} \in \mathcal{C}, \forall \mathbf{u} \in \mathcal{V}, \quad \eta(\mathbf{c}, \mathbf{u}) = 1 \iff t_c + \theta_{F_c, I_v} \leq t_v^i$$

Es decir que la tripulación sólo puede realizar tareas que comienzan después de su momento de inicio y debe tener tiempo de transportarse a la estación correcta de ser necesario. Se define la compatibilidad entre una asignación y un conjunto de tripulaciones según:

Definición 2.7 Dada una tripulación \mathbf{c} y un camino $P = u_1, \dots, u_k \in G$ se dice que la tripulación \mathbf{c} puede realizar el turno P si $\eta(\mathbf{c}, \mathbf{u}_1) = 1$. Además se dice que una asignación \mathcal{P} es compatible con las tripulaciones \mathcal{C} en el orden $\mathbf{c}_1, \dots, \mathbf{c}_S$ si

$$\forall P_s \in \mathcal{P}, \quad \mathbf{c}_s \in \mathcal{C}, \quad \mathbf{c}_s \text{ puede realizar } P_s.$$

En el caso de que el número de tripulaciones sea mayor que el número de turnos, el orden $\mathbf{c}_1, \dots, \mathbf{c}_S$ puede definirse sobre un subconjunto de \mathcal{C} . Por otro lado, si el número de tripulaciones es inferior al número de turnos S , esto quiere decir que se necesitarían tripulaciones adicionales para realizar esta asignación, por lo que la asignación no es compatible con el conjunto de tripulaciones.

Incorporando las tripulaciones al ejemplo, si se considera la asignación $\{P_1, P_2\}$ y el orden de tripulaciones TRA, TRC , entonces la tripulación TRC tiene justo el tiempo necesario para comenzar su turno de trabajo en C , transportarse hasta A en tiempo $\theta_{C,A} = 600$ y comenzar P_2 como se ve en la figura 2.4

ID	Origen	Destino	t^i	t^f
AA0	A	A	300	3180
AA1	A	A	6060	8940
AC	A	C	2040	4920
CA	C	A	7200	9880

ID	t	Lugar de inicio
TRA	0	A
TRC	1440	C

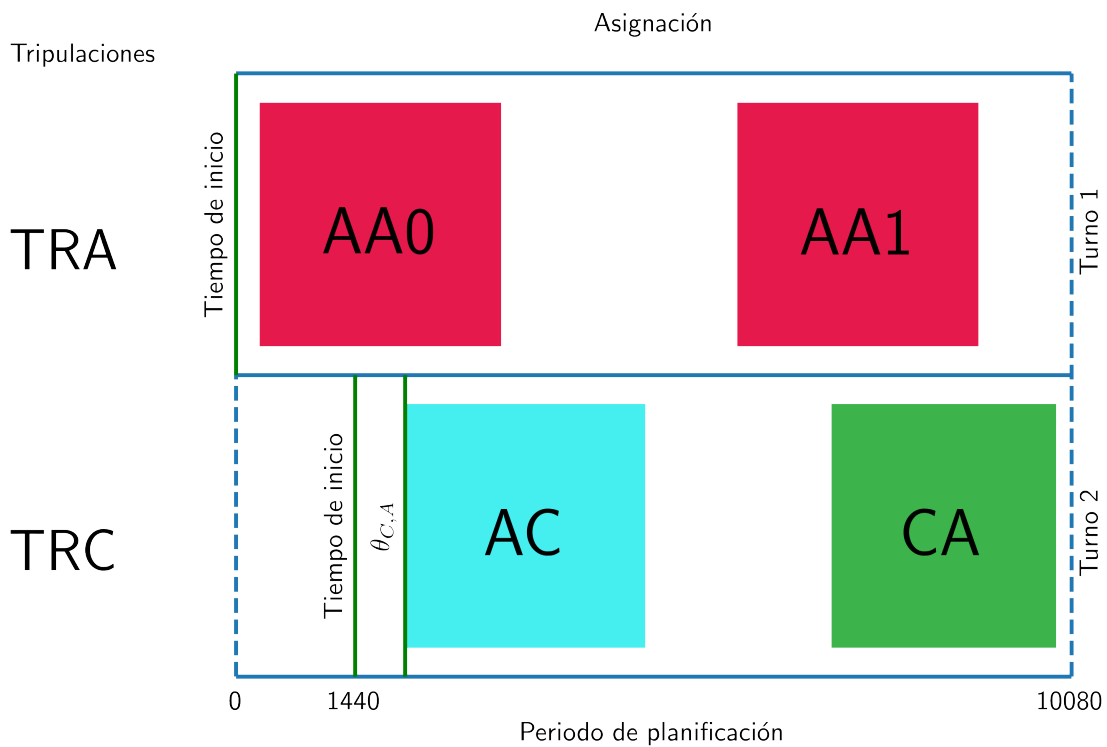


Figura 2.4: Ejemplo de asignación con tripulaciones.

2.2.4. Restricciones globales

Además de las restricciones más comunes descritas anteriormente, se pueden imponer restricciones que no caben en ninguna de las categorías previas y que pueden involucrar viajes que no son necesariamente sucesivos.

Estas restricciones caben en una de 2 categorías:

Restricciones de turno

Estas restricciones sólo consideran un turno de trabajo a la vez. Por ejemplo, estas pueden exigir que en cada turno de trabajo debe haber al menos un descanso legal. Estas están caracterizadas por una tupla de la forma (R, r_{inf}, r_{sup}) , formada por un conjunto de viajes restringidos $R \subseteq \mathcal{V}$ y las cotas $r_{inf}, r_{sup} \in \mathbb{N}$ que definen la restricción:

$$\forall s \in \mathcal{S}, \quad r_{inf} \leq \#(V(P_s) \cap R) \leq r_{sup}.$$

Dónde $\#$ representa la cantidad de elementos de un conjunto y $V(P_s)$ son los vértices visitados por el camino P_s . El ejemplo de tener un descanso legal por turno se escribiría como $(\mathcal{V}', 1, 1)$. El conjunto de restricciones de turno se denota por $\mathcal{G}_{\text{turno}}$.

Restricciones de intervalo

Estas restricciones representan un mínimo o máximo de eventos que pueden ocurrir dentro de un intervalo de tiempo desde el punto de vista de una tripulación. Ejemplos de estas restricciones son tener como mínimo un descanso legal en cada ventana de 7 días de trabajo o tener un máximo de 3 viajes nocturnos en un plazo 5 días.

Para modelar estas restricciones se considera la trayectoria que tendría una tripulación realizando los viajes de una asignación.

Si una tripulación comienza su trabajo en el turno s y una vez terminado este pasa al turno $suc(s) = s'$ y continúa de esta forma hacia el infinito, su trayectoria está dada por la concatenación de caminos

$$P_s^* = P_s P_{suc(s)} P_{suc^2(s)} \dots P_{suc^k(s)} \dots$$

Se define una función de trayectoria H , tal que $H(s, [h_i, h_f])$ considera una tripulación que comenzó en el turno s y entrega una lista ordenada con los viajes que esta tripulación realiza en el intervalo $[h_i, h_f]$ de su trabajo.

En la figura 2.5 se presenta un diagrama que ejemplifica el funcionamiento de la función H para el caso rotativo y el caso repetitivo, considerando un intervalo $[h_i, h_f]$ tal que $h_i \leq t_{\text{máx}} \leq h_f$.

Como se puede apreciar en el diagrama, la función $H(1, [t_i, t_f])$ entregaría $[AA1, AA0]$ en el caso repetitivo y $[AA1, AC]$ en el caso rotativo.

Cada una de estas restricciones se caracteriza por el conjunto de viajes restringidos $R \subseteq \mathcal{V}$, las cotas c_{inf}, c_{sup} y el intervalo sobre el que aplica la restricción $[h_i, h_f]$. Luego la tupla $(R, r_{inf}, r_{sup}, [t_i, t_f])$ define la restricción

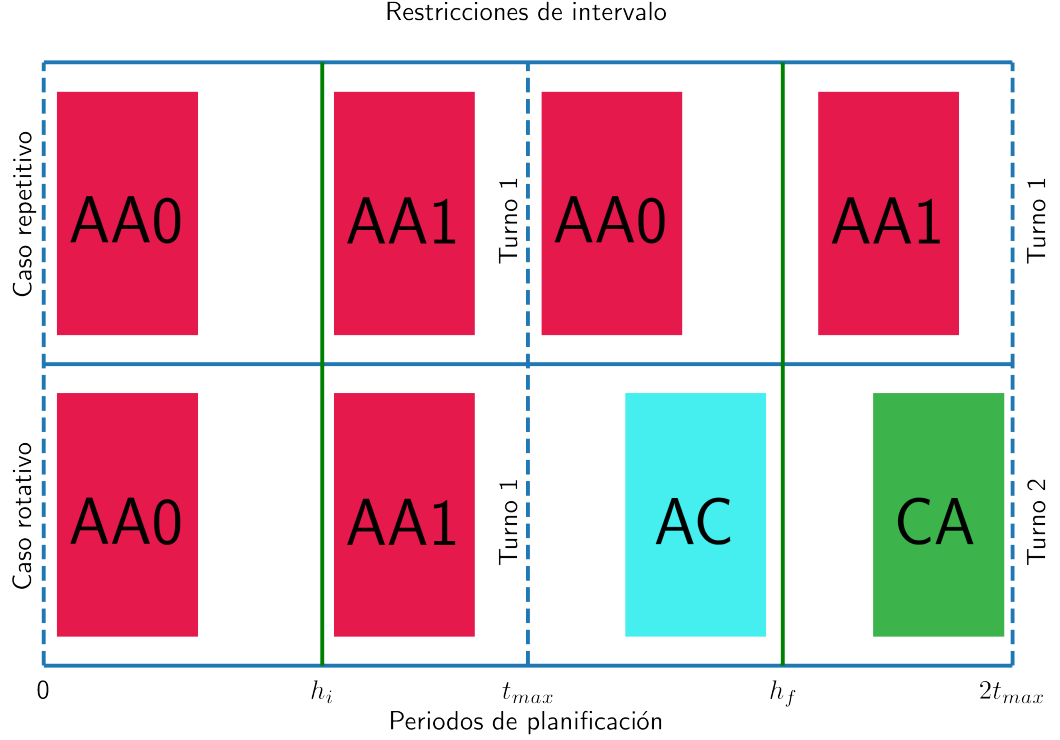


Figura 2.5: Ejemplo de H

$$\forall s \in \{1, \dots, S\}, \quad [v_1, \dots, v_k] = H(s, [h_i, h_f]), \quad r_{inf} \leq \#\{i \in \{1, \dots, k\} \mid v_i \in R\} \leq r_{sup}.$$

En otras palabras, para toda tripulación, en el intervalo $[h_i, h_f]$ de su tiempo de trabajo, deben encontrar un mínimo de r_{inf} y un máximo de r_{sup} viajes restringidos. Por ejemplo, para expresar que debe haber un descanso en cada ventana de 7 días se define la familia de restricciones

$$\{(\mathcal{V}', 1, +\infty, [i \text{ días}, i + 7 \text{ días}])\}_{i \in \{0, \dots, 6\}}.$$

Por otro lado, para definir que pueden haber máximo 3 viajes nocturnos en cada intervalo de 5 días, se define $\mathcal{N} \subseteq \mathcal{V}$ el conjunto de los viajes nocturnos y la familia de restricciones

$$\{(\mathcal{N}, 0, 3, [i \text{ días}, i + 5 \text{ días}])\}_{i \in \{0, \dots, 6\}}.$$

El conjunto de restricciones de intervalo se denota por $\mathcal{G}_{intervalo}$ y el conjunto de todas las restricciones globales se denota por $\mathcal{G} = \mathcal{G}_{turno} \cup \mathcal{G}_{intervalo}$. Se define la factibilidad global como sigue.

Definición 2.8 Dado el conjunto de restricciones globales $\mathcal{G} = \mathcal{G}_{turno} \cup \mathcal{G}_{intervalo}$, una asignación \mathcal{P} se dice globalmente factible si para cada restricción de turno $(R, r_{inf}, r_{sup}) \in \mathcal{G}_{turno}$ se cumple

$$\forall s \in \{1, \dots, S\}, \quad r_{inf} \leq \#|V(P_s) \cap R| \leq r_{sup}.$$

Y para cada restricción de intervalo $(R, r_{inf}, r_{sup}, [h_i, h_f])$ se cumple

$$\forall s \in \{1, \dots, S\}, \quad [v_1, \dots, v_k] = H(s, [h_i, h_f]), \quad r_{inf} \leq \#\{i \in \{1, \dots, k\} \mid v_i \in R\} \leq r_{sup}$$

Finalmente se define una asignación factible, es decir una asignación que podría ser una solución al problema.

Definición 2.9 Dado un problema definido por $\mathcal{V}, \mathcal{C}, \gamma, \bar{\gamma}, \eta, \mathcal{E}, \mathcal{G}$, una asignación \mathcal{P} , y un orden de las tripulaciones c_1, \dots, c_S . Se dice que esta asignación es factible con este orden si se cumplen las siguientes restricciones.

1. La asignación \mathcal{P} es completa.
2. La asignación \mathcal{P} es compatible entre turnos.
3. La asignación \mathcal{P} es compatible con las tripulaciones \mathcal{C} en el orden c_{1S} .
4. La asignación es globalmente factible respecto de \mathcal{G} .

2.3. La función objetivo

De forma general se consideran dos pasos en la resolución de este problema. El primero es el de encontrar una asignación completa y factible, el segundo es el de encontrar una asignación que minimice la diferencia de tiempo trabajado entre distintos los turnos, es decir, que asigne cargas de trabajo similares a cada tripulación. Para cada uno de estos pasos se necesitarán funciones objetivo distintas, ambas serán presentadas a continuación.

Para la primera parte, dada una asignación \mathcal{P} , se define la cantidad de viajes no ficticios asignados como $\#(V(P_s) \cap V_0)$, es decir la cantidad de viajes no ficticios que están cubiertos por algún camino en \mathcal{P} .

Por otro lado se define la carga de trabajo w_s de cada turno en términos de la duración de los viajes Δt_v

$$w_s = \sum_{\substack{v \in P_s \\ v \notin \mathcal{V}'}} \Delta t_v, \quad w_- = \min_{s \in \{1, \dots, S\}} w_s, \quad c_+ = \max_{s \in \{1, \dots, S\}} c_s.$$

Minimizar la desigualdad de carga de trabajo equivale a minimizar la diferencia de carga entre los turnos con más y menos carga.

$$\min [w_+ - w_-].$$

Capítulo 3

Modelo de optimización lineal

Una forma de resolver este problema consiste en plantearlo como un problema de programación lineal mixta, es decir, que incluye variables enteras y continuas. Estos modelos luego pueden optimizarse a través de sistemas comerciales para programación lineal, tales como Gurobi o Cplex. A continuación se presenta un modelo lineal para este problema.

Se considera un problema definido por el conjunto de viajes \mathcal{V} , el conjunto de tripulaciones \mathcal{C} , el grafo G , el conjunto de aristas F , el emparejamiento entre turnos \mathcal{E} y las restricciones η, \mathcal{G} . Se busca encontrar una asignación \mathcal{P} en G y un orden c_1, \dots, c_n tal que \mathcal{P} sea factible con este orden.

3.1. Las variables

Se definen dos conjuntos de variables binarias. Por un lado el conjunto $\{x_{s,v}\}_{s \in \{1, \dots, S, v \in V\}}$, tales que $x_{s,v} = 1$ si y sólo si el viaje v está en el camino s de la asignación. Por otro lado las variables $y_{c,s}$ tales que $y_{c,s} = 1$, si la tripulación c ocupa el lugar s en el orden c_1, \dots, c_S , es decir, que c realiza P_s .

Para asegurar que los viajes $P_s \in \mathcal{P}$ visiten cada viaje no ficticio una sola vez se impone la restricción

$$\sum_{s=1}^S x_{v,s} \leq 1, \quad \forall v \in V_0.$$

De manera similar, para asegurar que cada tripulación aparece solamente una vez en el orden, se impone

$$\sum_{s=1}^S y_{c,s} = 1, \quad \forall c \in \mathcal{C}.$$

Adicionalmente se agregan variables w_+, w_- que representan las mínimas y máximas cargas de trabajo. Dado que la carga total del camino s en \mathcal{P} es $\sum_{v \in V} x_{s,v} \cdot \Delta t_v$, entonces se debe imponer

$$w_- \leq \sum_{v \in V} x_{s,v} \cdot \Delta t_v \leq w_+, \quad \forall s \in \{1, \dots, S\}.$$

Evidentemente, como se desea equilibrar la carga entre las tripulaciones, la función objetivo consiste en minimizar w_+ y w_- .

3.2. Las restricciones

A continuación se traduce al lenguaje de la programación matemática las distintas restricciones descritas en el capítulo anterior.

Para asegurar que las variables $x_{s,v}$ efectivamente formen un camino P_s en G se utilizarán ciertas propiedades de G . En la Proposición 2.4 se demostró que para todo camino de la forma $P = v_1, \dots, v_N \in G$, se cumple

$$\forall j, k \in \{1, \dots, N\}, \quad j < k, \quad (v_j, v_k) \in E.$$

De forma recíproca, es evidente que si un conjunto $Q \subseteq V$ cumple

$$\forall u, v \in Q : \quad t_u^i \leq t_v^i \quad \implies \quad (u, v) \in E. \quad (3.1)$$

Entonces al ordenar los elementos de Q por tiempo de inicio, se obtiene un único camino $P_Q = v_1, \dots, v_l$. Este camino es único ya que si $u, v \in Q$ cumplen $t_u^i = t_v^i$, entonces $(u, v) \in E$ lo que es contradictorio ya que una tripulación no puede comenzar dos viajes al mismo tiempo.

Se concluye que hay una equivalencia entre conjuntos que cumplen a restricción (3.1) y los caminos en P . Se impone así la siguiente restricción sobre las variables $\{x_{s,v}\}_{s \in \{1, \dots, S, v \in V\}}$

$$x_{s,u} + x_{s,v} \leq 1, \quad t_u^i \leq t_v^i, \quad (u, v) \notin E, \quad \forall s \in \{1, \dots, S\}. \quad (3.2)$$

Sea una asignación de las variables $x_{s,v}$ que cumple esta restricción, entonces el conjunto $Q_s = \{v \mid x_{s,v} = 1\}$ cumple la condición (3.1). Luego Q_s está asociado a un único camino en G . Recíprocamente, todo camino P_s en G puede identificarse con un único conjunto Q_s que cumple la condición (3.1) tal que sus variables correspondientes cumplen (3.2).

A la hora de implementar este modelo estas se pueden agrupar para disminuir la cantidad de restricciones.

De manera similar a lo anterior se introducen las restricciones entre turnos, correspondientes a aristas $(u, v) \in F$. En este caso, dado un par de turnos consecutivos $P_s, P_{s'}$, no puede haber ningún par $u \in V(P_s), v \in V(P_{s'})$ tal que $(u, v) \notin F$. Entonces se deduce la restricción lineal.

$$x_{s,u} + x_{s',v} \leq 1, \quad (u, v) \notin F, \quad \forall s, s' \in \mathcal{E}.$$

De la misma forma se introducen las restricciones asociadas a las condiciones iniciales de las tripulaciones, exigiendo que las tripulaciones asignadas a cada turno puedan realizarlo:

$$y_{c,s} + x_{s,v} \leq 1, \quad \eta(c, v) = 0 \quad \forall c \in \mathcal{C}, v \in V.$$

Dado el conjunto de restricciones globales $\mathcal{G} = \mathcal{G}_{turno} \cup \mathcal{G}_{intervalo}$, se definen las restricciones globales de turno para cada tupla (R, r_{inf}, r_{sup}) como

$$r_{inf} \leq \sum_{v \in R} x_{s,v} \leq r_{sup}, \quad \forall s \in \{1, \dots, S\}.$$

Por otro lado, para las restricciones de intervalo, para cada tupla $((R, r_{inf}, r_{sup}, [h_i, h_f]))$ se computa el conjunto

$$K(h_i, h_f) = \{(k, v) \mid h_i \leq k \cdot t_{max} + t_v^i \leq h_f\}.$$

Es decir que $K(h_i, h_f)$ contiene los viajes que podrían quedar en el intervalo h_i, h_f de trabajo de una tripulación, acompañado de cuantos intervalos de planificación hacia el futuro son necesarios. A modo de ejemplo, considerando los viajes de la tabla 3.1, con $t_{max} = 10080$.

ID	Origen	Destino	t^i	t^f
AA0	A	A	300	3180
AA1	A	A	6060	8940
AC	A	C	2040	4920
CA	C	A	7200	9880

Tabla 3.1: Viajes de Ejemplo

Se calcula el conjunto $K(5000, 15000) = \{(0, AA1), (1, AA0), (1, AC)\}$. Con estos conjuntos calculados se puede expresar la restricción global por intervalos como

$$r_{inf} \leq \sum_{\substack{(k,v) \in K(h_i, h_f) \\ v \in R}} x_{suc^k(s),v} \leq r_{sup}, \quad \forall s \in \{1, \dots, S\}.$$

3.3. Los problemas lineales

Utilizando las restricciones y variables definidas anteriormente se definen dos problemas, denotados por Fase 1 y Fase 2.

El problema de Fase 1 consiste en encontrar una asignación factible, para esto se fija como función objetivo la cantidad total de viajes asignados y se dejan de lado las variables w_+ , w_- . Todas las otras restricciones se hacen efectivas. La solución del modelo de Fase 1 es utilizada como punto inicial del modelo de fase 2.

En el problema de Fase 1, a continuación, las variables w_+ y w_- no son consideradas, de manera que se intenta solamente maximizar los viajes que tienen efectivamente asignada una tripulación. Si el problema original es factible, el valor máximo alcanzable en la Fase 1 debe ser igual a la cardinalidad del conjunto V_0 , es decir el total de viajes. En caso de que el problema no sea factible, se puede proceder utilizando una cantidad reducida de los viajes o se puede volver a realizar la fase 1 aumentando la cantidad de tripulaciones. Cual de estos dos métodos se usa dependerá de las condiciones del caso particular. Se detallarán condiciones para que una solución sea factible en capítulos posteriores.

Fase 1

$$\text{máx} \quad \sum_{s=1}^S \sum_{v \in V_0} x_{s,v} \quad (3.4)$$

$$s.a. \quad \sum_{s=1}^S x_{v,s} \leq 1 \quad \forall v \in V_0 \quad (3.5)$$

$$\sum_{s=1}^S y_{c,s} = 1 \quad \forall c \in \mathcal{C} \quad (3.6)$$

$$x_{s,u} + x_{s,v} \leq 1 \quad t_u^i \leq t_v^i, \quad (u, v) \notin E, \quad \forall s \in \{1, \dots, S\} \quad (3.7)$$

$$x_{s,u} + x_{s',v} \leq 1 \quad \forall (u, v) \notin F, \quad \forall s, s' \in \mathcal{E} \quad (3.8)$$

$$y_{c,s} + x_{s,v} \leq 1 \quad \eta(c, v) = 0 \quad \forall c \in \mathcal{C}, v \in V \quad (3.9)$$

$$r_{inf} \leq \sum_{v \in R} x_{s,v} \leq r_{sup} \quad \forall s \in \{1, \dots, S\} \quad (3.10)$$

$$r_{inf} \leq \sum_{\substack{(k,v) \in K(h_i, h_f) \\ v \in R}} x_{suc^k(s),v} \leq r_{sup} \quad \forall s \in \{1, \dots, S\} \quad (3.11)$$

$$x_{s,v} \in \{0, 1\} \quad \forall s \in \{1, \dots, S\}, v \in V \quad (3.12)$$

$$y_{c,v} \in \{0, 1\} \quad \forall c \in \mathcal{C}, v \in V. \quad (3.13)$$

En este caso se dejan fuera las variables w_+ , w_- y la restricción 3.5 permite dejar viajes sin tripulación asignada.

El modelo de Fase 2 incorpora w_+, w_- y se define como función objetivo minimizar la diferencia entre ambas. Además se cambia la restricción 3.5 a una igualdad, de esta manera se exige que todos los viajes sean utilizados. Este modelo considera como punto inicial una solución de la Fase 1 ya que de otra forma sería muy costoso el cómputo de un punto inicial. En trabajos anteriores esta Fase 2 fue realizada con métodos de búsqueda local.

Fase 2

$$\underset{x,y}{\text{mín}} \quad w_+ - w_- \quad (3.15)$$

$$s.a. \quad \sum_{s=1}^S x_{v,s} = 1 \quad \forall v \in V_0 \quad (3.16)$$

$$\sum_{s=1}^S y_{c,s} = 1 \quad \forall c \in \mathcal{C} \quad (3.17)$$

$$x_{s,u} + x_{s,v} \leq 1 \quad t_u^i \leq t_v^i \quad (u, v) \notin E, \quad \forall s \in \{1, \dots, S\} \quad (3.18)$$

$$x_{s,u} + x_{s',v} \leq 1 \quad \forall (u, v) \notin F, \quad \forall s, s' \in \mathcal{E} \quad (3.19)$$

$$y_{c,s} + x_{s,v} \leq 1 \quad \eta(c, v) = 0, \quad \forall c \in \mathcal{C}, v \in V \quad (3.20)$$

$$r_{inf} \leq \sum_{v \in R} x_{s,v} \leq r_{sup} \quad \forall s \in \{1, \dots, S\} \quad (3.21)$$

$$r_{inf} \leq \sum_{\substack{(k,v) \in K(h_i, h_f) \\ v \in R}} x_{suc^k(s),v} \leq r_{sup} \quad \forall s \in \{1, \dots, S\} \quad (3.22)$$

$$w_- \leq \sum_{v \in V} x_{s,v} \cdot \Delta t_v \leq w_+ \quad \forall s \in \{1, \dots, S\} \quad (3.23)$$

$$x_{s,v} \in \{0, 1\} \quad \forall s \in \{1, \dots, S\}, v \in V \quad (3.24)$$

$$y_{c,v} \in \{0, 1\} \quad \forall c \in \mathcal{C}, v \in V \quad (3.25)$$

$$w_+, w_- \in \mathbb{R}. \quad (3.26)$$

Notar que en este caso se exige, a través de la restricción (3.16), que todos los viajes sean asignados a una y sólo una tripulación.

Capítulo 4

Casos de estudio

El trabajo de la sección experimental se enfoca en dos casos particulares del modelo planteado. Estos casos se eligieron debido a su aplicación a escenarios reales, en particular corresponden a trenes de carga que deben realizar viajes entre distintas minas y centros de procesamiento. Estos trenes son lentos ya que llevan grandes cantidades de carga, así que una tripulación puede viajar entre dos estaciones por otros medios en relativamente poco tiempo.

Algunas de las posibilidades introducidas en el modelo no aplican a los casos de estudio, como es el caso de las condiciones iniciales de las tripulaciones, y las restricciones globales entre viajes como la de los viajes nocturnos.

4.1. Caso MxN

Uno de los casos estudiados consiste en un régimen de trabajo de M días continuos intercalados con descansos continuos de N días. Este régimen de trabajo también trae consigo una serie de especificaciones sobre la estructura de los turnos, las cuales se detallarán a continuación.

- **Intervalo de planificación:** El intervalo de planificación $[0, t_{\text{máx}}]$ está determinado por $t_{\text{máx}} = M + N$ días, donde un día consta de 1440 minutos.
- **Descanso entre viajes:** El descanso entre dos viajes es de $\delta = 600$. Además existe un mínimo de tiempo entre tiempos el inicio de dos viajes consecutivos determinado por $\delta_{\text{inicio}} = 1440$.
- **Los descansos legales:** Además de los viajes de \mathcal{V}_0 , la tripulación debe realizar un descanso legal al menos una vez en cada turno. Estos tienen una duración de $9 \cdot 60 + 1440N$ minutos y pueden realizarse cualquier día, comenzando a las 21h y terminando N días después a las 6 am. Por lo que el conjunto \mathcal{V}' se define como

$$\mathcal{V}' = \{v_d \mid \forall d \in \{0, \dots, N + M - 1\}\}$$

$$t_{v_d}^i = 1440d + 21 \cdot 60, \quad t_{v_d}^f = 1440(N + d) + 6 \cdot 60.$$

A estos descansos legales se les asocia la restricción por turnos dada por la tupla $(V', 1, 1)$, es decir que debe haber un y sólo un descanso legal por turno.

- **Repetición de turnos:** Este caso cada tripulación debe realizar el mismo turno una y otra vez, por lo que $\mathcal{E} = \{(s, s), \quad \forall s \in \{1, \dots, S\}\}$.

4.2. Caso rotativo

Este es otro caso a estudiar, dónde el horizonte de planificación es de M días y las tripulaciones realizan turnos distintos en cada horizonte de manera rotativa. Esto quiere decir que la tripulación que realiza el turno s , a continuación realizará el turno $s + 1 \pmod S$.

- **Intervalo de planificación:** El intervalo de planificación $[0, t_{\text{máx}}]$ está dado por $t_{\text{máx}} = M$ días, un día es equivalente a 1440 minutos.
- **Descanso entre viajes:** En este caso el descanso entre viajes es de $\delta = 9 \cdot 60$. Por otro lado no hay restricción asociada a δ_{inicio} .
- **Descansos legales:** En este caso debe realizarse un descanso legal en cada ventana de D días. Estos duran $9 \cdot 60 + 1440N$ minutos, y comienzan a las $21h$ de un día para terminar N días después a las $6h$.

$$\mathcal{V}' = \{v_d \mid \forall d \in \{0, \dots, M - 1\}\}$$

$$t_{v_d}^i = 1440d + 21 \cdot 60, \quad t_{v_d}^f = 1440(N + d) + 6 \cdot 60.$$

A estos descansos se asocian restricciones globales por intervalos para asegurar que efectivamente haya un viaje en cada ventana de D días. Estas quedan expresadas por el siguiente conjunto.

$$\{(V', 1, +\infty, [d \text{ días}, d + N \text{ días}])\}_{d \in \{0, \dots, M-1\}}.$$

Dónde 1 día = 1440 minutos. Cada una de estas tuplas asegura que comenzando desde cualquier turno, una tripulación va a realizar al menos un descanso en toda ventana de D días pero no tiene máximo de descansos posibles.

- **Turnos rotativos:** Las tripulaciones cambian de turno a realizar de manera rotativa, por lo que \mathcal{E} queda definido de la siguiente forma:

$$\mathcal{E} = \{(s, s') \mid s + 1 \pmod S = s', \quad s \in \{1, \dots, S\}\}.$$

Capítulo 5

El Algoritmo de Búsqueda Local

En este capítulo se describirán todos los algoritmos utilizados acompañados de pruebas de su correctitud donde corresponda. El desempeño práctico de estos algoritmos será analizado en el capítulo siguiente.

El algoritmo de búsqueda local es el principal algoritmo planteado en este trabajo. Este busca encontrar rápidamente una solución factible para poder mejorarla sucesivamente a través de una serie de perturbaciones que serán descritas en este capítulo.

Sin embargo, para poder iniciar la búsqueda, primero se deben utilizar una serie de heurísticas de factibilidad para encontrar una cantidad de tripulaciones que con la que sea factible encontrar una asignación factible.

Una vez encontrada la cantidad de tripulaciones por medio de las heurísticas para la factibilidad, una solución inicial es construida con un método goloso. Finalmente, la búsqueda local se lleva a cabo utilizando dos tipos de perturbaciones llamadas verticales y horizontales.

Las perturbaciones verticales consisten en elegir una serie de viajes que son incompatibles entre si, es decir que cada uno debe ser realizado por tripulaciones distintas. Una vez seleccionados estos viajes, se intenta reordenarlos sin disminuir la cantidad total de viajes asignados. Se le denomina perturbación vertical debido a que al graficar una asignación como en la figura 5.1, un conjunto de viajes alineados verticalmente serán siempre incompatibles entre si ya que sus tiempos de ejecución se intersectan.

Por otro lado las perturbaciones horizontales consisten en elegir un turno y resolver el problema de encontrar el mejor turno posible utilizando los viajes del turno elegido junto con los viajes que aun no se asignan.

5.1. Heurísticas para la factibilidad

Hasta ahora no se ha tratado el problema de la factibilidad del problema dado un conjunto de tripulaciones \mathcal{C} y un conjunto de viajes \mathcal{V} . La primera pregunta importante es si este

conjunto de tripulaciones es suficiente para cubrir todos los viajes en \mathcal{V} . De no ser el caso, es importante estimar el mínimo número de tripulaciones necesarias para cubrir todos los viajes.

Con este fin se presenta una heurística que permite determinar una cota inferior para el número de tripulaciones necesarias para satisfacer la demanda, es decir, cubrir el conjunto de viajes.

Para cada viaje $v \in \mathcal{V}_0$ se busca encontrar el turno factible que contiene a v y que maximiza la cantidad de viajes reales que contiene. Para ilustrar el concepto se considera un ejemplo simple con un tiempo de descanso entre viajes de $\delta = 900$ en la figura 5.1.

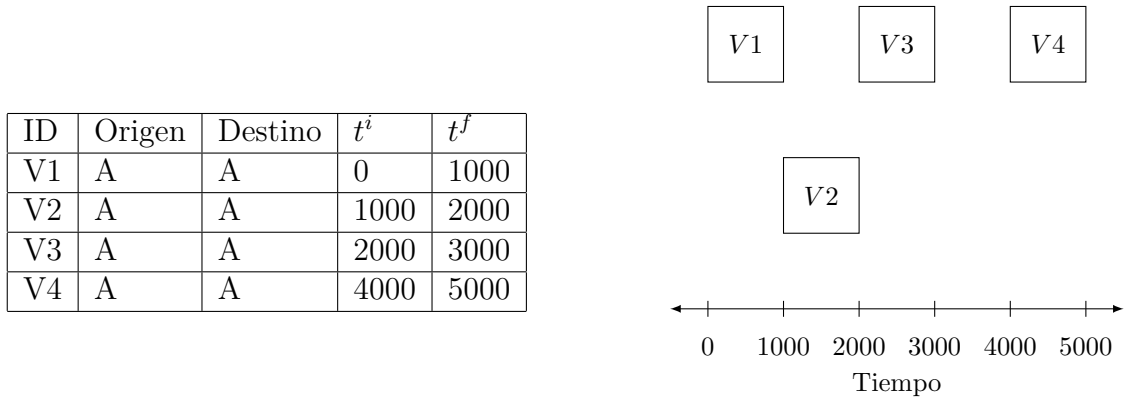


Figura 5.1: Ejemplo de instancia

En este caso es claro que el mejor turno que contiene a V1 es $\{V1, V3, V4\}$, mientras que el mejor turno que contiene a V2 es $\{V2, V4\}$. Formalmente, para cada viaje $v \in \mathcal{V}_0$ se define el mejor turno de su vértice correspondiente v como el turno factible más largo que pasa por v .

$$\text{mejor_turno}(v) = \text{máx}\{\#(V(P) \cap \mathcal{V}_0) \mid v \in V(P), \quad P \text{ es factible}\}.$$

Ahora, dado que una asignación completa cubre todos los viajes en \mathcal{V} , se quiere encontrar la menor cantidad de caminos necesarios para cubrir todo \mathcal{V}_0 . Por definición, un camino que contiene al vértice v tiene largo menor o igual a $\text{mejor_turno}(v)$. Utilizando esto se demostrará una cota inferior para el número de turnos necesarios para cubrir el conjunto \mathcal{V}_0 .

Para esto se define el problema simplificado (PS).

$$(PS) \quad \text{mín } N \quad (5.1)$$

$$s.a \quad \bigcup_{j=1}^N K_j = V_0 \quad (5.2)$$

$$\#|K_j| \leq \text{mín}_{v \in K_j} \text{mejor_turno}(v) \quad \forall j \in \{1, \dots, N\} \quad (5.3)$$

$$\#|K_j \cap K_l| = \emptyset \quad \forall j, l \in \{1, \dots, N\}, j \neq l. \quad (5.4)$$

En otras palabras N es la mínima cantidad de conjuntos necesarios para cubrir V_0 imponiendo la restricción de que cada conjunto no puede ser más grande que el mejor camino de cada uno de sus elementos. Es similar a cubrir \mathcal{V}_0 con caminos pero removiendo la exigencia de que los vértices deben estar conectados en el grafo G . Se quiere demostrar que una para una solución $\{K_j\}_{j \in \{1 \dots N\}}$ de este problema simplificado, el valor N corresponde efectivamente a una cota inferior para la cantidad de tripulaciones necesarias.

Proposición 5.1 *Sea $\mathcal{P} = \{P_j\}_{j \in \{1, \dots, S\}}$ una asignación completa y factible con S tripulaciones. Además sea $\{K_j\}_{j \in \{1 \dots N\}}$ una solución óptima al problema simplificado de valor N , entonces $N \leq S$.*

Demostración: Sea $\mathcal{P} = \{P_j\}_{j \in \{1, \dots, S\}}$ una asignación completa y factible con S tripulaciones. Sea N el valor de la solución a (PS). Por contradicción se asume que $N > S$.

Por definición de mejor_turno es evidente que

$$\forall P_j \in \mathcal{P}, \quad \#(P_j \cap V_0) \leq \text{mín}_{v \in P_j \cap V_0} \text{mejor_turno}(v).$$

Luego los conjuntos $K_j = P_j \cap V_0$ cumplen:

- $\#(K_j \cap V_0) \leq \text{mín}_{P_j \cap V_0} \text{mejor_turno}(v)$.
- Como la asignación \mathcal{P} es completa, también se cumple $\bigcup_{j=1}^N K_j = V_0$.
- Al ser una asignación factible, sus caminos son disjuntos.

Luego $\{K_j\}_{j \in \{1, \dots, S\}}$ es una solución factible del problema simplificado con $N > S$, lo que contradice el hecho de que N es el mínimo de (PS). ■

Desde un punto de vista computacional, primero se debe calcular mejor_turno para cada vértice y luego se debe resolver el problema simplificado.

Para calcular mejor_turno se utiliza el siguiente algoritmo.

- Se considera el grafo $G = (V_0 \cup V', E)$, $V = V_0 \cup V'$, como fue definido anteriormente con una copia de cada descanso legal en V' y se agrega una función de costo a los vértices w dada por

$$w(u) = \begin{cases} 1 & \text{si } u \in V_0 \\ 0 & \text{si } u \in V'. \end{cases}$$

- Para cada descanso legal $u \in V'$ se calculan $d_u, d'_u : V_0 \cup V' \rightarrow \mathbb{N}$ tal que d_u asocia el camino de máximo peso desde u hasta v y $d'_u(v)$ asocia el camino de máximo peso desde v hasta u . Se considera que el peso de un camino es la suma de los pesos de sus vértices.
- Dependiendo de la forma que tomen las restricciones de turno se consideran las configuraciones de descansos legales que pueden darse en una asignación factible. Estas se pueden codificar como una colección de conjuntos $\{R_j\}_{j \in \{1, \dots, N_{\text{conf}}\}}$, con $R_j \subseteq V'$. Para cada viaje $v \in V_0$ y cada configuración R_j compatible con v se ordenan los descansos junto con v de manera creciente en tiempo de inicio para obtener un orden de la forma $u_1 \dots u_k$ con $u_j = v$ para algún $j \in \{1 \dots k\}$ y $u_l \in V'$ si no. Luego el mejor camino de este vértice y esta configuración de descansos está dado por

$$\text{turno}(R_j, v) = d_{u_1}(u_2) + \dots + d_{u_{j-1}}(v) + d'_v(u_{j+1}).$$

$$\text{mejor_turno}(v) = \max_j \text{turno}(R_j, v).$$

Es decir que para cada configuración se considera el camino de mayor peso que pasa por v y se toma el máximo entre ellos.

Se utilizan una serie de optimizaciones para hacer más eficiente este algoritmo, sin embargo su complejidad está dada por la suma entre la construcción de los d_u y el cálculo de los mejor_turno . La construcción de los d_u requiere utilizar el algoritmo de Dijkstra de complejidad $O(|E| + |V| \log |V|)$ una vez por cada descanso legal, mientras que la construcción de mejor_turno requiere, para cada vértice, comparar una constante por cada una de las N_{conf} configuraciones. Se concluye que la complejidad de este algoritmo está dada por

$$O(\#\mathcal{V}' \cdot \#|E| + \#\mathcal{V}' \cdot \#\mathcal{V} \log \#\mathcal{V} + N_{\text{conf}} \cdot \#\mathcal{V}_0).$$

En los casos a estudiar, N_{conf} siempre es pequeño ya que se pueden utilizar turnos que contienen un solo descanso legal. Las complejidades presentadas para los distintos algoritmos provienen de [4].

A continuación se demostrará que el problema simplificado puede resolverse con un algoritmo glotón en tiempo lineal. Se define el problema simplificado restringido a $U \subseteq V_0$ (PSR) según:

$$(PSR) \tag{5.5}$$

$$\text{mín } N \tag{5.6}$$

$$\text{s.a } \bigcup_{j=1}^N K_j = U \tag{5.7}$$

$$\#|K_j| \leq \text{mín}_{v \in K_j} \text{mejor_turno}(v) \quad \forall j \in \{1, \dots, N\} \tag{5.8}$$

$$\#|K_j \cap K_l| = \emptyset \quad \forall j, l \in \{1, \dots, N\}, j \neq l. \tag{5.9}$$

Además se introduce la función ϕ que a cada subconjunto $U \subseteq V_0$ asocia la solución al problema simplificado restringido a U . Se demostrarán dos propiedades sobre el problema simplificado.

Proposición 5.2 Sean $P \subseteq Q \subseteq V_0$, luego $\phi(P) \leq \phi(Q)$.

Demostración: Sea $\{K_j\}$ solución al problema simplificado de tamaño $\phi(Q)$. Este puede convertirse en una solución del mismo tamaño para P definiendo los conjuntos $K'_j = K_j \cap P$. Estos forman una solución factible con el mismo valor que $\phi(Q)$. ■

Proposición 5.3 Sean $P, Q \subseteq V_0$ tales que $\#|P| = \#|Q|$ y tales que al ordenarlos de forma no decreciente en mejor_turno se cumple

$$\begin{aligned} P &= p_1 \dots p_k, & Q &= q_1 \dots q_k \\ \text{mejor_turno}(p_j) &\leq \text{mejor_turno}(q_j), & \forall j &\in \{1, \dots, \#|P|\}. \end{aligned}$$

Entonces $\phi(P) \geq \phi(Q)$

Demostración: Sea $\{K_j\}_{j \in \{1, \dots, N\}}$ con $N = \phi(P)$ una solución de tamaño $\phi(P)$ al problema simplificado restringido a P . Se construye $\{K'_j\}$ de la siguiente forma

$$p_l \in K_j \implies q_l \in K'_j, \quad \forall j \in \{1, \dots, N\}.$$

Luego $\{K'_j\}$ es solución al problema simplificado restringido a Q , en efecto, para cualquier $q_l \in Q$ existe un $p_l \in P$ y un K_j tal que $p_l \in K_j$, por lo que $q_l \in K'_j$. Utilizando esta propiedad para todos los q_l se obtiene

$$\bigcup_{j=1}^N K'_j = Q.$$

Además, por hipótesis se sabe que $\text{mejor_turno}(p_l) \leq \text{mejor_turno}(q_l)$ para todo $l \in \{1, \dots, \#|Q|\}$ entonces

$$\#|K'_j| = \#|K_j| \leq \min_{p \in K_j} \text{mejor_turno}(p) \leq \min_{q \in K'_j} \text{mejor_turno}(q), \quad \forall j \in \{1, \dots, N\}.$$

Finalmente, los conjuntos K'_j son disjuntos porque los K_j lo son. Se concluye que los $\{K'_j\}$ forman una solución factible del problema simplificado restringido a Q con $N = \phi(P)$, por lo que $\phi(Q) \leq \phi(P)$. ■

A continuación se utilizan las propiedades anteriores para construir de manera inductiva una colección $\{K_j\}_{j \in \{1, \dots, N\}}$ óptima para el problema simplificado.

Para un conjunto $U \subseteq V_0$ sea $u^* \in \underset{u \in U}{\text{argmín}} \text{mejor_turno}(u)$.

- Si la cantidad de elementos en U denotada por $\#|U|$ es menor o igual a $\text{mejor_turno}(u^*)$. Entonces $\phi(U) = 1$ ya que se cumple

$$\#|U| \leq \text{mejor_turno}(u^*) = \min_{v \in U} \text{mejor_turno}(v).$$

Y por lo tanto $\{U\}$ constituye una solución factible al problema simplificado restringido a U con $N = 1$.

- Por otro lado, si $\#|U| > \text{mejor_turno}(u^*)$, entonces en cualquier solución del problema simplificado debe haber un conjunto K_{u^*} que contenga a u^* , por lo que $\phi(U)$ se puede escribir como

$$\phi(U) = 1 + \min_{K_{u^*} \subseteq U} \phi(U \setminus K_{u^*}), \quad u^* \in K_{u^*}, \quad \#|K_{u^*}| \leq \text{mejor_turno}(u^*).$$

Se busca encontrar un K_{u^*} que minimice la expresión, para esto se utilizan las proposiciones 5.2 y 5.3.

Sea un K_{u^*} tal que $\#|K_{u^*}| < \text{mejor_turno}(u^*)$. Como $\#|U| > \text{mejor_turno}(u^*)$ entonces existe K'_{u^*} tal que

$$\#|K'_{u^*}| = \text{mejor_turno}(u^*), \quad K_{u^*} \subseteq K'_{u^*}.$$

Por lo tanto gracias a la proposición 5.2

$$\phi(U \setminus K'_{u^*}) \leq \phi(U \setminus K_{u^*}).$$

Por lo que se puede restringir la búsqueda del mínimo a los conjuntos con exactamente $\text{mejor_turno}(u^*)$ elementos

$$\phi(U) = 1 + \min_{K_{u^*} \subseteq U} \phi(U \setminus K_{u^*}), \quad u \in K_{u^*}, \quad \#|K_{u^*}| = \text{mejor_turno}(u^*).$$

Sea $U = u_1 u_2 \dots u_k$ ordenado de forma no decreciente en mejor_turno , con $u_1 = u^*$. Se define $K_{u^*} = \{u_l\}_{l \in \{1, \dots, \text{mejor_turno}(u^*)\}}$. Sea K'_u otro conjunto que contiene a u^* con $\text{mejor_turno}(u^*)$ elementos.

Para utilizar la propiedad 5.2 y concluir que K_{u^*} es el mínimo, sea K_{u^*} que cumple

$$u^* \in K'_{u^*}, \quad \#|K'_{u^*}| = \text{mejor_turno}(u^*).$$

Y sea $U \setminus K'_{u^*} = u_{f(1)} \dots u_{f(m)}$ dónde

$$m = k - \text{mejor_turno}(u^*), \quad f : \{1, \dots, m\} \rightarrow \{1, \dots, k\}.$$

Con f una función creciente. Es decir que se escribe el conjunto $U \setminus K'_{u^*}$ como un sub orden de $U = u_1 u_2 \dots u_k$. En esta misma notación se define $h(l) = l + \text{mejor_turno}(u^*)$ y se escribe $K_{u^*} = u_{h(1)} \dots u_{h(m)}$ por definición de K_{u^*} . Se deduce que

$$\forall l \in \{1 \dots m\}, \quad f(l) \leq h(l).$$

Y por lo tanto

$$\forall l \in \{1 \dots m\}, \quad \text{mejor_turno}(u_{f(l)}) \leq \text{mejor_turno}(u_{h(m)}).$$

Es decir que se cumplen las condiciones de 5.3 y

$$\phi(U \setminus K_{u^*}) \leq \phi(U \setminus K'_{u^*}).$$

Por lo que $\phi(U \setminus K_{u^*})$ es el mínimo buscado. Finalmente la expresión queda como

$$\phi(U) = 1 + \phi(U \setminus K_{u^*}).$$

De esta demostración se deduce el siguiente algoritmo:

```

N ← 0
U ← V0
k ← #U
while k > 0 do
  u* ← argmínv ∈ U mejor_turno(v)
  c ← mejor_turno(u*)
  if c ≥ #|U| then
    N ← N + 1
    U ← ∅
  else
    N ← N + 1
    u1 ... uk ← order(U)
    Ku* ← {u1, ..., uc}
    U ← U \ Ku*
  end if
k ← #U

```

Figura 5.2: Algoritmo para resolver el problema simplificado (PS)

La complejidad del algoritmo 5.2 es la misma que la complejidad de ordenar el conjunto V_0 respecto a `mejor_turno`, esto es porque los elementos que se retiran siempre serán los primeros elementos del orden, por lo que una vez calculado el orden para V_0 , este se mantiene para los pasos siguientes. Por otro lado, cada iteración del bucle remueve como mínimo un elemento de U y se procesa en tiempo constante asumiendo que el orden se mantiene. Como se trata de ordenar un conjunto acotado de números naturales, utilizando el algoritmo *Counting Sort* se obtiene una complejidad lineal sobre los elementos de V_0 .

5.2. Espacio de búsqueda de los descansos legales

Será conveniente para lo que sigue considerar separadamente los descansos legales del resto de los viajes.

El objetivo de esta sección es caracterizar el espacio de búsqueda de los descansos legales para entender cómo se puede elegir uno inicialmente y cómo navegar en él a lo largo del proceso de búsqueda local. Se define una asignación de descansos legales $\mathcal{R} = \{R_i\}_{i \in \{1, \dots, S\}}$ como una colección de caminos R_i sobre G que consisten exclusivamente de elementos en V' , es decir que constan sólo de vértices correspondientes a descansos legales.

A pesar de los distintos tipos de restricción global que pueden determinar los viajes, las propiedades que vienen a continuación aplican a ambos casos. En la primera parte se explorarán restricciones a los descansos legales que pueden obtenerse de manera general, en la segunda parte se planteará un algoritmo que dada una posible asignación de descansos, permite determinar si es factible construir una asignación completa a partir de ella o no.

Asumiendo una cantidad de turnos S y un conjunto de viajes \mathcal{V} se define la demanda por minuto según

$$\forall t \in [0, t_{\text{máx}}], \quad d_t = \#\{v \mid t_v^i \leq t \leq t_v^f \quad \vee \quad t_v^i \leq t + t_{\text{máx}} \leq t_v^f, \quad v \in \mathcal{V}_0\}.$$

Es decir que la demanda de cada minuto del intervalo $[0, t_{\text{máx}}]$ es la cantidad de viajes que ocupan ese minuto en su intervalo de operaciones, por lo que en una asignación completa al menos d_t tripulaciones están ocupadas en el minuto t con algún viaje. Es fácil utilizar esto para construir una primera restricción para las asignaciones de descansos legales \mathcal{R} . De manera análoga a la demanda, se define el consumo de tiempo r_t dado por

$$r_t = \#\{v \mid t_v^i \leq t \leq t_v^f, \quad v \in V(\mathcal{R})\} \leq S, \quad \forall t \in [0, t_{\text{máx}}].$$

De donde se deduce la restricción

$$d_t + r_t \leq S, \quad \forall t \in [0, t_{\text{máx}}].$$

Notar que cada descanso legal $v \in \mathcal{V}'$ puede tener muchos vértices correspondientes en \mathcal{R} . Para ilustrar el concepto de demanda de manera visual se presenta el siguiente ejemplo de caso rotativo de 135 viajes. Los viajes comparten colores si tienen la misma duración.

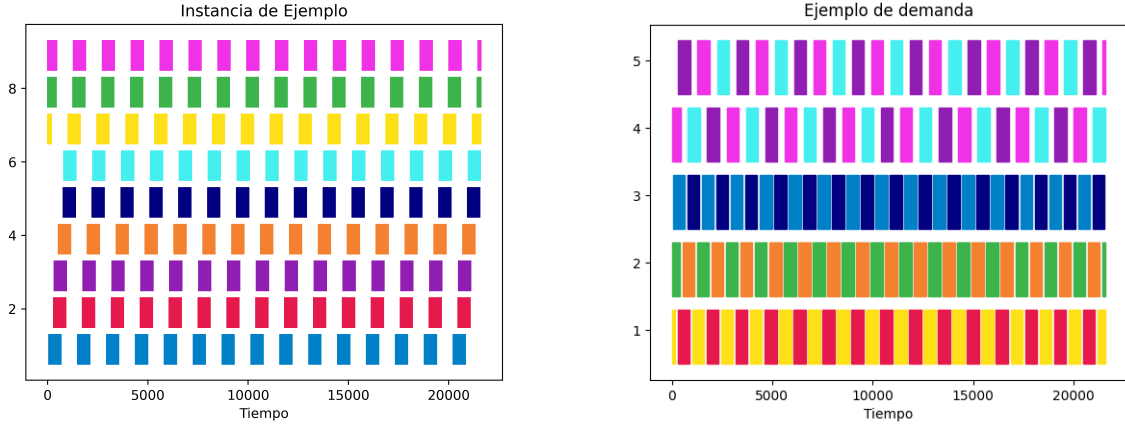


Figura 5.3: Ejemplo visual de la demanda

En el primer gráfico (izquierda) se observan los viajes ordenados para su fácil visualización, todos los viajes con los mismo colores tienen la misma duración y mismos lugares de inicio y fin, es decir son viajes equivalentes pero que se realizan en distintos momentos. Por otro lado, en el gráfico de la derecha se intenta colocar los viajes en el menor número de líneas posible. Se puede apreciar que se necesitan 5 un mínimo de líneas para colocar todos los viajes, a pesar de que puede haber minutos específicos en que se requieran menos, hay minutos en que 5 viajes están activos a la vez. Esto significa que $\max_{t \in [0, t_{\text{máx}}]} d_t = 5$, por lo que se deduce que no puede haber una asignación completa para estos viajes con menos de 5 tripulaciones.

En su forma actual esta restricción no entrega mucha información, pero es un buen punto de partida para obtener restricciones más estrictas. Una forma de hacer esto es tomar en consideración que después de cada viaje no necesariamente hay otro viaje que puede comenzar inmediatamente, entre dos viajes tiene que haber un mínimo descanso δ a menos de que uno de los dos sea un descanso legal. Para incorporar esta idea y mejorar la cota, se introduce el mínimo tiempo de espera de un viaje dado por

$$e_u = \min_{(u,v) \in E \cup F} \begin{cases} t_v^f - t_u^i & \text{si } (u, v) \in E \\ t_{\text{máx}} + t_v^f - t_u^i & \text{si } (u, v) \in F. \end{cases}$$

Es decir que para cada viaje, e_u es el mínimo tiempo que puede haber antes de que comience el viaje siguiente. Notar que, por la existencia de los descansos legales, e_u puede ser menor a δ . Se puede definir la demanda con tiempo de espera d'_t según

$$\forall t \in [0, t_{\text{máx}}], \quad d'_t = \#\{v \mid t_v^i \leq t \leq t_v^f + e_v \quad \vee \quad t_v^i \leq t + t_{\text{máx}} \leq t_v^f + e_v, \quad v \in \mathcal{V}_0\}.$$

Estas cumplen $d_t \leq d'_t$ y también se mantiene la restricción

$$d'_t + r_t \leq S, \quad \forall t \in [0, t_{\text{máx}}]. \quad (5.10)$$

Ya que para cada minuto en el intervalo $t_v^i, t_v^f + e_v$ una tripulación debe estar ocupada realizando el viaje v o esperando antes de poder realizar otro viaje. Volviendo a la figura 6.6, en el gráfico de la izquierda se observa que si se respetan los tiempos mínimos entre viajes hace falta un mínimo de 9 tripulaciones, lo que coincide con el resultado que se obtiene experimentalmente $\max_{t \in [0, t_{\text{máx}}]} d'_t = 9$.

Desde un punto de vista computacional, para calcular $\{d'_t\}_{t \in [0, t_{\text{máx}}]}$ basta con iterar sobre los pares $u, v \in V$ para obtener $\{e_u\}_{u \in V}$. Una vez calculados, basta con una iteración sobre el intervalo $[0, t_{\text{máx}}]$ para obtener $\{d'_t\}_{t \in [0, t_{\text{máx}}]}$. Es decir que la complejidad de calcular $\{d'_t\}_{t \in [0, t_{\text{máx}}]}$ es $O(|V|^2)$, considerando $t_{\text{máx}}$ como una constante.

Vale la pena notar que una vez calculado $\{d'_t\}_{t \in [0, t_{\text{máx}}]}$, evaluar si una asignación particular \mathcal{R} cumple con 5.10 se puede realizar de manera muy rápida, iterando por cada $v \in V(\mathcal{R})$ y calculando r_t , con una complejidad de $O(V(\mathcal{R})) = O(S)$.

Siguiendo la misma línea de pensamiento del mínimo tiempo de espera, se asigna una función de pesos a los conjuntos de aristas E y F dada por

$$w(u, v) = \begin{cases} t_v^f - t_u^i & \text{si } (u, v) \in E \\ t_v^f + t_{\text{máx}} - t_u^i & \text{si } (u, v) \in F. \end{cases}$$

Luego para cualquier asignación completa \mathcal{P} cuyos descansos legales corresponden a una asignación de descansos \mathcal{R} y con un emparejamiento de turnos \mathcal{E} , se tiene que

$$\forall (j, k) \in \mathcal{E}, \quad \forall P_j, P_k \in \mathcal{P}, \quad P_j = u_1 \dots u_n, \quad P_k = v_1 \dots v_m, \quad (u_n, v_1) \in F.$$

Se denotan por $E(\mathcal{P})$ y $F(\mathcal{P})$ los conjuntos de aristas de \mathcal{P} , vale la pena notar que cada vértice aparece solamente una vez como entrada y una vez como salida en $E(\mathcal{P}) \cup F(\mathcal{P})$ ya que los caminos en \mathcal{P} son disjuntos. Dado un vértice v en la asignación \mathcal{P} , y su arista correspondiente $(u, v) \in E(\mathcal{P}) \cup F(\mathcal{P})$ se define el tiempo de espera $e_u(\mathcal{P})$ según

$$e_u(\mathcal{P}) = w(u, v), \quad (u, v) \in E(\mathcal{P}) \cup F(\mathcal{P}).$$

Este está bien definido ya que cada vértice sólo tiene una arista de salida en $E(\mathcal{P}) \cup F(\mathcal{P})$.

Osea que $e_u(\mathcal{P})$ entrega el tiempo que ocurre entre que termina el viaje u y comienza el siguiente viaje en la asignación \mathcal{P} . Es claro que cada instante de una asignación, una tripulación puede estar realizando un viaje (real o ficticio) o estar en espera entre dos viajes, por lo que

$$\sum_{v \in V(\mathcal{P})} e_v(\mathcal{P}) + \Delta t_v = \#|\mathcal{P}| \cdot t_{\text{máx}}.$$

Es decir que la suma de la duración de los viajes más el tiempo entre viajes es igual a la cantidad de turnos multiplicados por la duración de aquellos turnos. Utilizando todo esto, se define el problema de asignación (PA), considerando $V = V(\mathcal{R}) \cup V_0$ y los E, F correspondientes.

$$(PA) \tag{5.11}$$

$$\text{mín} \quad \sum_{u,v \in E \cup F} x_{u,v} \cdot w(u,v) \tag{5.12}$$

$$s.a \quad \sum_{v \in V} x_{u,v} = 1 \quad \forall u \in V \tag{5.13}$$

$$\sum_{u \in V} x_{u,v} = 1 \quad \forall v \in V \tag{5.14}$$

$$x_{u,v} \in \{0, 1\} \quad (u,v) \in E \cup F \tag{5.15}$$

$$x_{u,v} = 0 \quad (u,v) \notin E \cup F. \tag{5.16}$$

Esto corresponde al problema de asignación que minimiza el tiempo total de espera entre viajes. Si $\{x_{u,v}\}_{u,v \in V}$ es una solución óptima, entonces es claro que para cualquier asignación completa \mathcal{P} con los mismos viajes y descansos se cumple

$$\sum_{u,v \in E \cup F} x_{u,v} \cdot w(u,v) \leq \sum_{v \in V(\mathcal{P})} e_v(\mathcal{P}).$$

Por lo que

$$\sum_{v \in V} \Delta t_v + \sum_{u,v \in E \cup F} x_{u,v} \cdot w(u,v) \leq \#\mathcal{P} \cdot t_{\text{máx}}.$$

Es decir que, dada una asignación de descansos $\mathcal{R} = \{R_i\}_{i \in \{1 \dots S\}}$, se puede resolver el problema de asignación para obtener una cota inferior para el número de tripulaciones que puede tener una asignación completa que utilice la asignación de descansos \mathcal{R} . Se denota esta cota por $S_{\text{asignación}}$. Luego si S es inferior a la cota obtenida con el problema de asignación, se dice que no es factible ya que una asignación completa no puede ser construida a partir de \mathcal{R} utilizando S tripulaciones.

El problema de asignación se puede resolver en tiempo polinomial, con una complejidad cúbica sobre el número de vértices, lo que en este caso corresponde a $O(\#|V|^3)$.

En el capítulo siguiente se verá que para distintos tipos de instancias las distintas cotas entregarán mejor o peores resultados, lo que justifica la existencia de más de una.

Para obtener una asignación de descansos que cumpla con las restricciones descritas se utiliza un método glotón que intenta asignar descansos comenzando en los distintos días hasta que no sea posible. Este método funciona bien para los casos estudiados, pero el desarrollo

de un método más robusto para encontrar asignaciones de descansos factibles podría ser una avenida de mejoramiento de este algoritmo.

5.3. Solución inicial

Para construir una solución inicial se asume una asignación de descansos $\mathcal{R} = \{R_j\}_{j \in \{1 \dots S\}}$. La construcción consiste en tomar un camino de descansos legales R_j y encontrar el camino de peso máximo en G que lo contenga, cada vez eliminando los vértices que fueron usados en caminos previos.

Formalmente, se define el conjunto de vértices sin usar que inicialmente está dado por $V^0 = V_0$ y el conjunto de índices sin usar inicialmente dado por $I^0 = \{1 \dots S\}$.

Cada paso $1 \leq n \leq S$ consta de las siguientes acciones

1. Se elige aleatoriamente $j \in I^n$.
2. Se busca el camino de mayor peso que contenga a R_j sobre el grafo inducido $G(V(R_j) \cup V^k)$. En otras palabras se resuelve el problema de solución inicial (PSI).

$$\begin{aligned}
 & (PSI) \\
 & \text{máx } \#|P| \\
 & \text{s.a } V(R_j) \subseteq V(P) \\
 & P \subseteq G(V(R_j) \cup V^k)
 \end{aligned}$$

3. Se define $P_j = P$, $V^{k+1} = V^k \setminus V(P)$, $I^{k+1} = I^k \setminus \{j\}$

Al cabo de S pasos estarán definidos los caminos P_j para cada $j \in \{1 \dots S\}$ y se define la asignación inicial $\mathcal{P}_{inicial} = \{P_j\}_{j \in \{1 \dots S\}}$. Notar que $\mathcal{P}_{inicial}$ es una asignación parcial que no necesariamente cubre a todos los viajes.

5.4. Perturbaciones horizontales

Las perturbaciones horizontales son una continuación del método para encontrar una solución inicial. Estas sirven para aumentar la cantidad de viajes en una asignación. Dada una asignación \mathcal{P} , una asignación horizontal consiste en elegir un turno al azar $P_s \in \mathcal{P}$ y restringir el grafo G a los vértices que no están asignados en \mathcal{P} , dados por $V \setminus V(\mathcal{P})$ junto con los vértices en $V(P_s)$. Se denotará este grafo restringido como G' .

Luego se resuelve el problema de encontrar el camino con mayor número de vértices en G' . Como el camino P_s es subgrafo de G' , el camino de peso máximo P^* inevitablemente va

a tener al menos tantos vértices como P_s . En los casos en que múltiples caminos tengan el largo máximo se elige uno al azar.

A diferencia del método utilizado para la solución inicial, en el que los descansos están fijos, en este caso se prueba con todas las combinaciones de descansos factibles para el turno s y tales que al ser integradas a la asignación de descansos total respeten las restricciones de demanda.

La complejidad de cada perturbación horizontal equivale a resolver el problema de camino de mayor peso en el grafo G' .

5.5. Perturbaciones verticales

Estas perturbaciones consisten en tomar un conjunto de viajes K que son incompatibles entre sí y reasignarlos manteniendo el resto de los viajes intactos.

Sean una asignación $\mathcal{P} = \{P_s\}_{s \in \{1 \dots S\}}$ y un conjunto de viajes incompatibles entre sí K . Se define \mathcal{P}' como la asignación que se obtiene al retirar todos los elementos de K . Dado que el grafo G cumple con la transitividad a tiempo creciente se tiene la certeza de que $\mathcal{P}' = \{P'_s\}_{1 \dots S}$ está bien definida. En efecto

$$\forall s \in \{1 \dots S\}, \quad P_s = p_1 \dots p_l \dots p_n, \quad p_l \in K \quad \implies \quad P'_s = p_1 \dots p_{l-1} p_{l+1} \dots p_n.$$

Y como los p están en orden creciente de tiempo de inicio

$$(p_{l-1}, p_l), (p_l, p_{l+1}) \in E \implies (p_{l-1}, p_{l+1}) \in E.$$

Se busca asignar los viajes en K a los turnos P'_s con tal de maximizar una función objetivo. Se dice que P'_s es compatible con $v \in K$ si existe una extensión del camino P'_s con $V(P''_s) = V(P'_s) \cup \{v\}$ y tal que $\{P'_s\} \cup \{P''_s\}$ es factible. En otras palabras, son compatibles si el vértice v se puede insertar en alguna parte del camino P'_s manteniendo la compatibilidad con el resto de la asignación.

El concepto de compatibilidad se resume en el conjunto

$$H = \{(s, v) \in \{1 \dots S\} \times K \mid v \text{ es compatible con } P'_s\}.$$

Utilizando H , se puede plantear el problema de asignación vertical (PAV) de la siguiente manera, con una función de peso a definir f .

$$(PAV) \tag{5.17}$$

$$\text{máx} \quad \sum_{(s,v) \in H} y_{s,v} \cdot f(s,v) \tag{5.18}$$

$$s.a \quad \sum_{v \in K} x_{s,v} = 1 \quad \forall s \in \{1 \dots S\} \tag{5.19}$$

$$\sum_{s=1}^S x_{s,v} = 1 \quad \forall v \in K \tag{5.20}$$

$$x_{s,v} = 1 \in \{0, 1\} \quad (s,v) \in H \tag{5.21}$$

$$x_{s,v} = 0 \quad (s,v) \notin H \tag{5.22}$$

En el caso de que \mathcal{P} es una asignación parcial y hay elementos en K que no están asignados en \mathcal{P} , el objetivo consiste en maximizar la cantidad de elementos asignados. Por lo que la función de peso puede tomarse como la constante 1, $f(s,v) = 1, \forall (s,v) \in H$.

Sin embargo, para asegurar que las perturbaciones verticales no repitan los mismos patrones cada vez y que la búsqueda no se detenga, se define $k = \#(K)$ y se agrega un término aleatorio a la función f dado por

$$f(s,v) = 1 + r_{s,v}, \forall (s,v) \in H, \quad 0 \leq r_{s,v} \leq \frac{1}{k}.$$

La restricción $r_{s,v} \leq \frac{1}{\#(k)}$ asegura que para cualquier valor de los $r_{s,v}$, la asignación de máximo peso también es aquella que asigna la mayor cantidad de viajes.

En efecto, cualquier asignación $\{y_{j,v}\}$ que asigna l viajes cumple

$$\sum_{(s,v) \in H} y_{s,v} \cdot f(s,v) \leq l.$$

Por otro lado, el valor máximo de cada arista está dado por $1 + \frac{1}{k}$, luego cualquier asignación $\{y'_{j,v}\}$ con $l' < l \leq k$ aristas cumple

$$\sum_{(s,v) \in H} y_{s,v} \cdot f(s,v) \leq l' \cdot \left(1 + \frac{1}{k}\right) < l.$$

Con lo que se verifica la propiedad. ■

Las perturbaciones verticales permiten iterar sobre el espacio de búsqueda rápidamente, y gracias a su componente aleatoria permiten el cambio gradual de la solución. En el caso de la fase 1, al realizar una perturbación vertical esta siempre va a entregar una nueva asignación con igual número o más viajes asignados que la asignación anterior. Para el caso de la fase

2 este ya no es el caso, ya que no es seguro que una asignación vertical va a disminuir la diferencia de carga entre turnos.

Es por esto que para la fase dos, después de cada perturbación vertical se evalúa $w_+ - w_-$ y se compara con el valor que tenía la asignación previa a la perturbación. Luego se elige si guardar o descartar el cambio de manera probabilista según la diferencia entre los valores de la función objetivo.

Capítulo 6

Resultados experimentales

En esta sección se analizan los resultados experimentales de cada una de las componentes del algoritmo propuesto, y se comparan con los resultados obtenidos con el solver comercial Gurobi.

Todas las medidas de tiempo de ejecución presentadas se obtuvieron con un procesador AMD Ryzen 7 3750H, programadas en `python` y medidas con la función `perf_counter` de la librería `time`. En algunos casos se presenta un promedio de múltiples mediciones para obtener mayor precisión, estos casos serán indicados según corresponda.

6.1. Las instancias de optimización

Para esta parte del trabajo se consideraron 12 instancias de optimización cuyas características se resumen en la siguiente tabla. Estas corresponden a los casos de estudio descritos en el capítulo 3.

Nombre	Número de viajes	Régimen	$t_{\text{máx}}$	Largo de los descansos legales
10x5 A	135	Repetitivo	15 días	129 horas
10x5 B	810	Repetitivo	15 días	129 horas
10x5 C	180	Repetitivo	15 días	129 horas
10x5 D	150	Repetitivo	15 días	129 horas
10x5 E	570	Repetitivo	15 días	129 horas
10x5 F	765	Repetitivo	15 días	129 horas
Rotativo A	63	Rotativo	7 días	33 horas
Rotativo B	378	Rotativo	7 días	33 horas
Rotativo C	68	Rotativo	7 días	33 horas
Rotativo D	70	Rotativo	7 días	33 horas
Rotativo E	238	Rotativo	7 días	33 horas
Rotativo F	357	Rotativo	7 días	33 horas

Tabla 6.1: Resumen de las instancias estudiadas

Vale la pena notar que, como regla general, las instancias de régimen rotativo tienen menos viajes que las de régimen repetitivo. Esto se debe a que el intervalo de planificación $[0, t_{\text{máx}}]$ es más corto, entonces una menor cantidad de viajes entrega una cantidad similar de tripulaciones.

6.2. Las cantidad de tripulaciones

Se comienza el análisis evaluando el comportamiento de las heurísticas presentadas en el capítulo anterior. Para cada instancia se calcula la heurística del mejor valor.

Nombre	Valor de la heurística (S_0)	Tiempo de ejecución [s]
10x5 A	15	0.05
10x5 B	89	1.83
10x5 C	20	0.09
10x5 D	17	0.06
10x5 E	63	0.95
10x5 F	84	1.73
Rotativo A	11	0.01
Rotativo B	63	0.27
Rotativo C	12	0.01
Rotativo D	12	0.01
Rotativo E	40	0.10
Rotativo F	60	0.25

Tabla 6.2: Valores y tiempo de ejecución de la heurística de mejor turno

Cada una de estas mediciones se obtuvo promediando 10 repeticiones del cálculo. Para ayudar con la interpretación de estos datos, en la figura 6.1 se pueden observar los tiempos de ejecución según la cantidad de viajes

Este gráfico es consistente con el cálculo de complejidad para el cálculo de esta heurística presentado en el capítulo anterior dado por

$$O(\#\mathcal{V}' \cdot \#|E| + \#\mathcal{V}' \cdot \#\mathcal{V} \log \#\mathcal{V} + N_{\text{conf}} \cdot \#\mathcal{V}_0).$$

Como $\#\mathcal{V}'$ es igual a 15 o 7 en los casos a estudiar, y N_{conf} es pequeño, la complejidad puede simplificarse a

$$O(\#|E| + \#\mathcal{V} \log \#\mathcal{V}).$$

Lo que corresponde a lo observado en el gráfico.

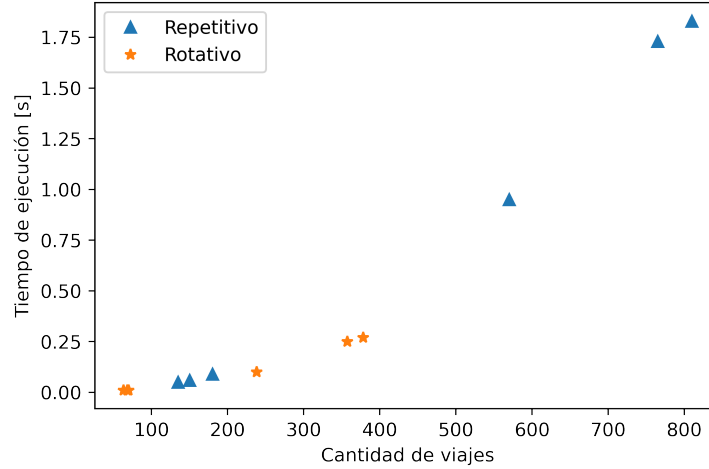


Figura 6.1: Tiempo de ejecución según cantidad de viajes.

6.3. Las soluciones iniciales

Para construir soluciones iniciales \mathcal{P}_0 primero se deben encontrar asignaciones de descansos factibles \mathcal{R}_0 . Utilizando los resultados de la heurística anterior se busca encontrar asignaciones de descanso con S_0 tripulaciones. Se obtienen los siguientes resultados:

Nombre	Valor de la heurística (S_0)	Tiempo de ejecución [s]
10x5 A	15	0.05
10x5 B	89	1.83
10x5 C	20	0.09
10x5 D	17	0.06
10x5 E	63	0.95
10x5 F	84	1.73
Rotativo A	11	0.01
Rotativo B	63	0.27
Rotativo C	12	0.01
Rotativo D	12	0.01
Rotativo E	40	0.10
Rotativo F	60	0.25

Tabla 6.3: Valores y tiempo de ejecución de la heurística de mejor turno.

Una vez obtenidos los valores S_0 para cada instancia, sabiendo que S_0 es una cota inferior al número de tripulaciones necesarias para una asignación completa, se generan asignaciones de descansos legales \mathcal{R}_0 para cada instancia y se utilizan las heurísticas correspondientes para determinar si los descansos en \mathcal{R}_0 podrían ser consistentes con una asignación completa. En la tabla 6.4 se presentan los valores que toman las distintas heurísticas, marcando los máximos de cada columna con el color rojo.

Instancia	S_0	$\max d'_t + r_t$	$S_{\text{asignación}}$
10x5 A	15	14	14
10x5 B	89	84	84
10x5 C	20	20	20
10x5 D	17	18	19
10x5 E	63	61	62
10x5 F	84	81	81
Rotativo A	11	12	12
Rotativo B	63	64	64
Rotativo C	12	11	12
Rotativo D	12	12	13
Rotativo E	40	45	45
Rotativo F	60	59	60

Tabla 6.4: Cotas inferiores obtenidos por las distintas heurísticas aplicadas a \mathcal{R}_0

Vale la pena recordar que para que \mathcal{R}_0 sea factible, deben cumplirse las restricciones

$$\max_{t \in [0, t_{\text{máx}}]} d'_t + r_t \leq S_0 \quad \wedge \quad S_{\text{asignación}} \leq S_0.$$

Es decir que en todas aquellas filas en que la columna S_0 no está marcada en rojo, la asignación \mathcal{R}_0 encontrada es infactible. En estos casos hay dos posibilidades

- El problema es infactible con S_0 tripulaciones, por lo que si se desea encontrar una asignación completa es necesario aumentar la cantidad de tripulaciones. Es posible que en ciertas aplicaciones del problema el número de tripulaciones esté fijo como parámetro y por lo tanto se acepten instancias completas.
- El problema es factible con S_0 tripulaciones, pero no para la asignación de descansos R_0 . En este caso hace falta encontrar un mejor \mathcal{R}_0 .

En este trabajo no se presenta una solución exacta para diferenciar entre ambos casos. A falta de una manera exacta se presenta una solución heurística que consiste en generar múltiples asignaciones de descansos legales \mathcal{R}_0 y verificar si alguna es factible. Para detallar mejor este método es importante considerar los tiempos de ejecución de cada heurística presentados en la tabla 6.5, que corresponden a los tiempos necesarios para calcular los resultados en la tabla 6.4 y siguen el mismo esquema de colores.

Instancia	S_0	Calculo d'_t	máx $d'_t + r_t$	$S_{\text{asignación}}$
10x5 A	0.05	0.02	0.00	0.04
10x5 B	1.78	0.55	0.05	1.38
10x5 C	0.09	0.03	0.00	0.07
10x5 D	0.07	0.02	0.00	0.04
10x5 E	0.93	0.22	0.03	0.66
10x5 F	1.84	0.44	0.04	1.11
Rotativo A	0.01	0.00	0.00	0.02
Rotativo B	0.27	0.11	0.02	0.33
Rotativo C	0.01	0.00	0.00	0.01
Rotativo D	0.01	0.00	0.00	0.01
Rotativo E	0.10	0.04	0.01	0.13
Rotativo F	0.28	0.09	0.02	0.27

Tabla 6.5: Tiempos de ejecución de las distintas heurísticas aplicadas a \mathcal{R}_0

Es claro que máx $d'_t + r_t$ es una restricción más débil que $S_{\text{asignación}}$ pero es mucho más rápida de calcular repetidamente. Esto se coincide con los cálculos de complejidad del capítulo anterior. Es por esto que se utiliza el siguiente algoritmo que evita calcular S_0 siempre que no sea necesario, probando con N asignaciones distintas:

```

 $S_0 \leftarrow \text{heurística\_mejor\_camino}(V)$ 
 $\{d'_t\} \leftarrow \text{demanda\_extendida}(V)$ 
 $k \leftarrow 0$ 
aceptar_ $\mathcal{R} \leftarrow \text{Falso}$ 
while  $k < N$  do
   $k \leftarrow k + 1$ 
   $\mathcal{R} \leftarrow \text{asignacion\_descansos}$ 
   $\{r_t\} \leftarrow \text{consumo}(\mathcal{R})$ 
  if máx  $d'_t + r_t \leq S_0$  then
     $S_{\text{asignación}} \leftarrow \text{asignacion\_heuristica}(\mathcal{R})$ 
    if  $S_{\text{asignación}} \leq S_0$  then
      aceptar_ $\mathcal{R} \leftarrow \text{Verdadero}$ 
      Terminar bucle

```

Figura 6.2: Algoritmo para determinar $S_{\text{asignación}}$

De esta forma sólo se calcula $S_{\text{asignación}}$ solamente en aquellos casos en que \mathcal{R} cumpla máx $d'_t + r_t \leq S_0$, de esta forma se evita tener que resolver el problema de asignación de manera innecesaria. En el caso de que no sea posible aceptar ningún \mathcal{R} , se debe aumentar el número de tripulaciones y volver a realizar el proceso. Con esto se obtienen los números de tripulaciones finales S para cada instancia presentados en la tabla 6.6.

Nombre	S_0	S
10x5 A	15	15
10x5 B	89	89
10x5 C	20	20
10x5 D	17	19
10x5 E	63	63
10x5 F	84	84
Rotativo A	11	12
Rotativo B	63	64
Rotativo C	12	12
Rotativo D	12	13
Rotativo E	40	46
Rotativo F	60	60

Tabla 6.6: Valores de S después de verificar factibilidad

Como parte del proceso de calcular S también se genera una asignación de descansos factible \mathcal{R} . Utilizando ambos elementos se puede calcular una solución inicial para cada instancia. En la tabla 6.7 se presentan los resultados de las soluciones iniciales con S tripulaciones y los tiempos de ejecución correspondientes solamente al calculo de la solución inicial con el método goloso descrito en el capítulo anterior, al igual que el tiempo total de ejecución hasta ahora.

Instancia	Viajes asignados/Total	T. de ejec. solución inicial [s]	T. total de ejec. [s]
10x5 A	135 / 135	0.016	0.65
10x5 B	802 / 810	3.25	10.46
10x5 C	179 / 180	0.06	0.37
10x5 D	143 / 150	0.02	1.75
10x5 E	557 / 570	1.23	4.60
10x5 F	757 / 765	2.69	9.18
Rotativo A	62 / 63	0.02	0.13
Rotativo B	367 / 378	0.77	2.82
Rotativo C	66 / 68	0.01	0.08
Rotativo D	66 / 70	0.01	0.31
Rotativo E	237 / 238	0.26	1.47
Rotativo F	347 / 357	0.75	1.96

Tabla 6.7: Soluciones iniciales calculadas para cada instancia

Se puede observar que en algunos casos como 10x5 A marcado en rojo, la asignación ya incluye todos los viajes, por lo que la solución inicial también concluye la fase 1. Para el resto de las instancias es claro que el número de viajes sin asignar es un pequeño porcentaje del total. Estas soluciones iniciales luego pasan a la búsqueda local para tratar de alcanzar el máximo número de viajes.

Para ilustrar de forma más clara los distintos pasos realizados, se seguirá en detalle el funcionamiento para las instancias 10x5 A y Rotativo A. Primero se construyen las asignaciones de descansos legales. En la figura 6.3 se muestra para cada una de estas instancias,

cómo se asignan los descansos legales representados por rectángulos horizontales, con líneas punteadas verticales para marcar los días, equivalentes a 1440 minutos.

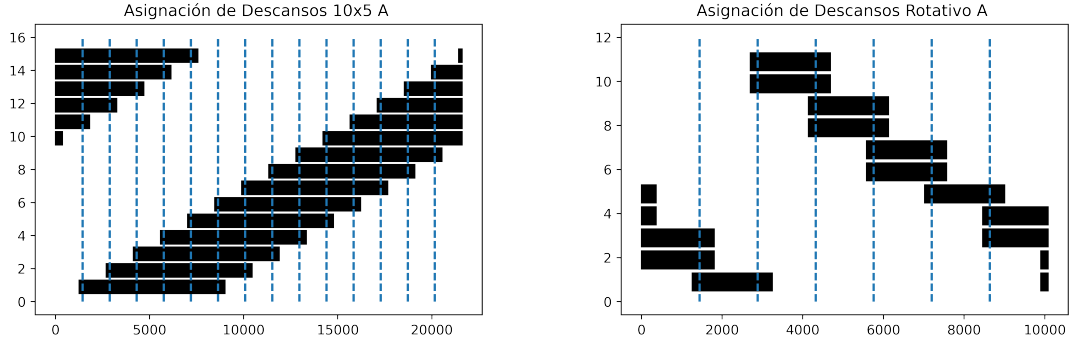


Figura 6.3: Demanda de las instancias 10x5 A y Rotativo A

En el caso de 10x5 A los descansos (representados por rectángulos en negro) tienen una duración de 129 horas y están bajo un régimen repetitivo, por lo que puede verse que los extremos de los descansos que sobrepasan $t_{\text{máx}}$ vuelven a aparecer en la misma línea por la izquierda. En esta instancia la heurística de mejor camino entregó un valor de $S_0 = 15$, al entregar esto al algoritmo de asignación de descansos se asignó exactamente un descanso a cada turno y de tal forma que en el turno s la tripulación comienza su descanso en el día s . Esto no tiene por qué darse así en el caso general.

En contraste, para el caso Rotativo A debe haber un descanso en cada intervalo de 7 días de trabajo y estos duran $33h$. En este caso $S_0 = 11$ pero se asignaron 12 descansos legales para cumplir con las restricciones.

Una vez determinadas las asignaciones de descansos, se debe evaluar su factibilidad con los métodos explicados en el capítulo anterior. Para esto se deben calcular los valores $\{d'_t\}_{t \in [0, t_{\text{máx}}]}$ y $\{r_t\}_{t \in [0, t_{\text{máx}}]}$ y resolver el problema de asignación. Para cada una de las instancias se presentan los viajes a asignar ordenados por color, recordando que dos viajes son del mismo color si tienen la misma duración y lugares de origen y destino. También se presentan los valores d'_t, r_t y $S_{\text{asignación}}$

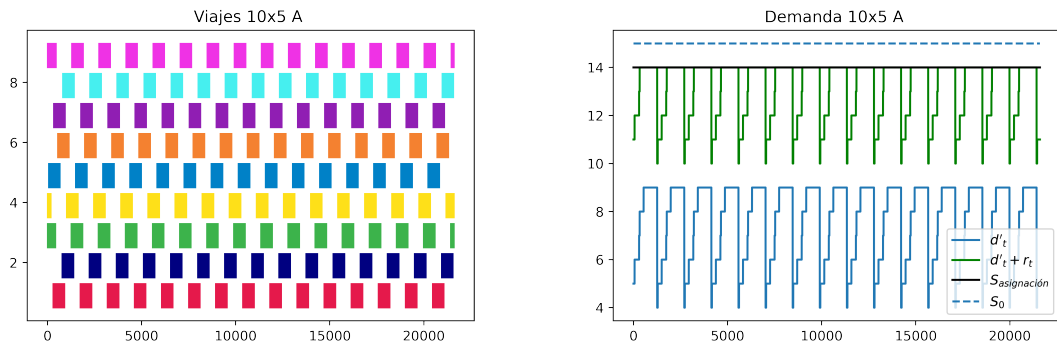


Figura 6.4: Asignación de descansos legales \mathcal{R}_0 en la instancia 10x5 A

Es claro en este caso que la restricciones obtenidas por la demanda y por el problema

de asignación son menores que S_0 . Esto significa que todas las heurísticas están apuntando a que esta asignación de descansos es factible en el sentido de que una asignación completa puede construirse a partir de ella.

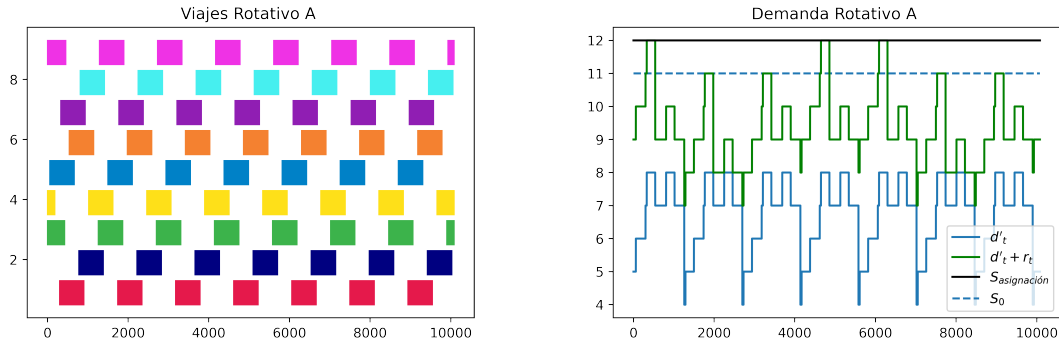


Figura 6.5: Asignación de descansos legales \mathcal{R}_0 en la instancia Rotativo A

Para la instancia Rotativo A el panorama es el opuesto. Tanto el problema de asignación como la restricción asociada a d'_t indican que esta asignación de descansos no es factible, es decir que a partir de esta asignación de descansos no es posible construir una asignación completa. Del gráfico de la demanda además se observa que no pareciera haber ninguna modificación fácil que se puede hacer a \mathcal{R}_0 para que se vuelva factible.

Por lo tanto se elige $N = 100$ y se generan 100 asignaciones de descansos \mathcal{R} para ser evaluadas siguiendo el algoritmo 6.2. En 0,38s se generan y evalúan 100 asignaciones, las cuales son todas descartadas por violar la restricción $d'_t + r_t$. Al aumentar S de 11 tripulaciones a 12, la primera asignación es automáticamente aceptada.

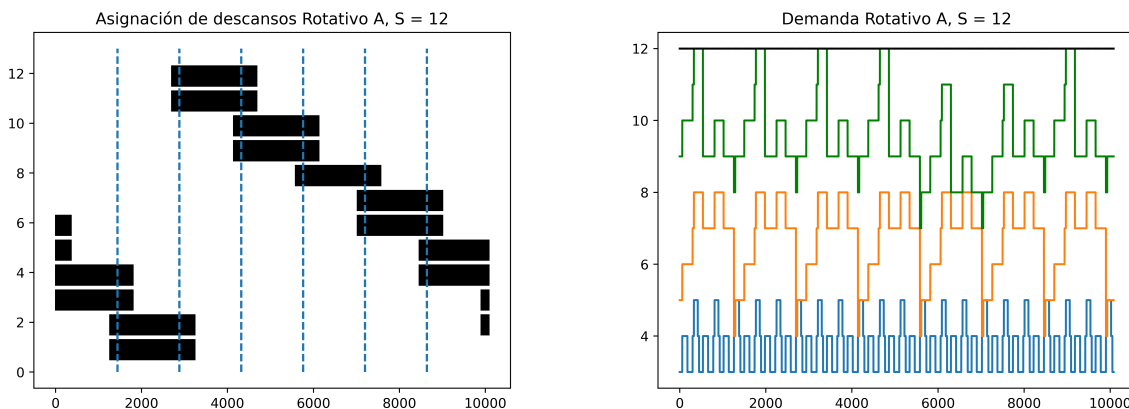


Figura 6.6: Asignación de descansos aceptada.

Ahora que se determinaron las cantidades de tripulaciones apropiadas y asignaciones de descansos factibles, se pueden calcular las soluciones iniciales correspondientes.

Para el caso de 10x5 A se trata de una solución inicial completa mientras que para Rotativo A queda un viaje fuera de 63.

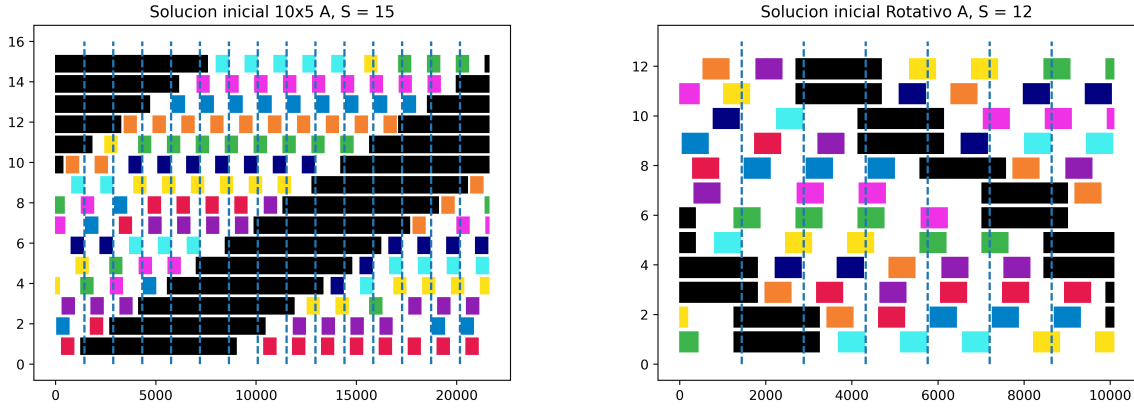


Figura 6.7: Soluciones iniciales obtenidas

6.4. Búsqueda local fase 1

Utilizando las soluciones iniciales como puntos de partida, para cada instancia se realiza una búsqueda local de fase 1 con el objetivo de encontrar una solución que maximice la cantidad de viajes asignados. Para analizar su efectividad, para cada instancia se realizan 500 iteraciones de la búsqueda local. En aquellos casos en que se encuentra una solución con el máximo número de viajes, el algoritmo se detiene automáticamente. En la tabla 6.8 se resumen los resultados.

Una iteración consiste en una serie de perturbaciones verticales y horizontales. Para realizar las perturbaciones verticales se identifican conjuntos de viajes incompatibles entre sí. Para encontrar conjuntos de este tipo fácilmente basta con tomar todos los viajes que comienzan dentro de un intervalo de largo $\delta + \min_{v \in V_0} \Delta t_v$, ya que necesariamente estos son incompatibles entre sí. Luego se separa el intervalo de planificación $[0, t_{\text{máx}}]$ en sub intervalos de largo $\delta + \min_{v \in V_0} \Delta t_v$ y se realiza una perturbación vertical para cada uno de ellos.

Además se elige de manera aleatoria un porcentaje de los turnos de la asignación y se realizan perturbaciones horizontales en cada uno de ellos, para el caso de los experimentos correspondientes a la tabla 6.8 se tomó el 50% de los turnos en cada iteración.

Luego con aquellas instancias que no alcanzaron el máximo de su fase 1 se realiza la búsqueda local con 2000 iteraciones.

6.5. Comparación con Gurobi

Para calcular el rendimiento de los algoritmos presentados con el solver **Gurobi** se realizan una serie de experimentos utilizando las mismas instancias anteriores. En cada caso se limita al solver a 300 segundos de procesamiento con el parámetro `time_limit`.

Instancia	Asignados / Total	T. de ejecución [s]	Iteraciones
10x5 A	135 / 135	0	0
10x5 B	808 / 810	125.66	500
10x5 C	180 / 180	0.46	1
10x5 D	148 / 150	8.03	500
10x5 E	565 / 570	63.89	500
10x5 F	764 / 765	112.07	500
Rotativo A	63 / 63	0.16	7
Rotativo B	376 / 378	37.14	500
Rotativo C	68 / 68	0.14	1
Rotativo D	68 / 70	2.64	500
Rotativo E	238 / 238	0.58	2
Rotativo F	357 / 357	19.04	286

Tabla 6.8: Resultados de la búsqueda local inicial con 500 iteraciones

Instancia	Asignados / Total	T. de ejecución [s]	Iteraciones
10x5 B	808 / 810	480	2000
10x5 D	148 / 150	8.03	500
10x5 E	567 / 570	253.433	2000
10x5 F	764 / 765	435.93	2000
Rotativo B	378 / 378	88.22	1142
Rotativo D	69 / 70	10.27	2000

Tabla 6.9: Resultados de la búsqueda local inicial con 2000 iteraciones

Primero se utiliza el solver **Gurobi** con el modelo lineal de Fase 1 descrito en el capítulo 3 sin condiciones iniciales. Estos resultados están tabulados en la tabla 6.10.

En la tabla 6.10 se puede observar que para aquellas instancias con menos de 200 viajes, **Gurobi** alcanza soluciones óptimas rápidamente, mientras que en las instancias con mayor número de viajes, las soluciones iniciales calculadas con los métodos glotones son considerablemente mejores y demoran considerablemente menos.

Para continuar la comparación, se utiliza el mismo programa de **Gurobi** con límite en 300 segundos, pero se utilizan las soluciones iniciales como puntos de partida. Estos resultados se comparan con los resultados de la búsqueda local de fase uno en la tabla 6.11.

En resumen, los métodos propuestos son similares en términos de velocidad para las instancias de menos de 200 viajes y para estas instancias son menos precisos (por ejemplo, no encuentra soluciones óptimas para 10x5 D ni para Rotativo D). Por otro lado, para las instancias de más de 200 viajes, el algoritmo de búsqueda local por sí solo logra encontrar asignaciones con casi todos los viajes asignados, mientras que incluso con más tiempo, **Gurobi** no obtiene resultados igualmente buenos. Sin embargo, como se observa en las tablas 6.8 y 6.9, la búsqueda local deja de aumentar su valor tan rápidamente pasado cierto punto, y no ofrece certeza de encontrar la solución eventualmente.

Instancia	Valor Gurobi	T. de E. Gurobi [s]	Valor Sol. Inicial	T. de E. Sol Inicial [s]
10x5 A	135 / 135	<1	135 / 135	0.65
10x5 B	732 / 810	300.00	802 / 810	10.46
10x5 C	180 / 180	50	179 / 180	0.37
10x5 D	150 / 150	80	143 / 150	1.75
10x5 E	528 / 570	300	557 / 570	4.60
10x5 F	764 / 765	300	757 / 765	9.18
Rotativo A	63 / 63	<1	62 / 63	0.13
Rotativo B	357 / 378	300	367 / 378	2.82
Rotativo C	68 / 68	<1	66 / 68	0.08
Rotativo D	70 / 70	9	66 / 70	0.31
Rotativo E	237 / 238	300	237 / 238	1.47
Rotativo F	335 / 357	300	347 / 357	1.96

Tabla 6.10: Resultados de 300 segundos de Gurobi comenzando desde una asignación vacía

Instancia	Valor Gurobi	T. de E. Gurobi [s]	Valor B.L. Fase 1	T. de E. Sol Inicial [s]
10x5 A	135 / 135	<1	135 / 135	0.65
10x5 B	805 / 810	300	808 / 810	480
10x5 C	180 / 180	0	180 / 180	0.46
10x5 D	150 / 150	156	148 / 150	8.03
10x5 E	560 / 570	300	567 / 570	253.433
10x5 F	760 / 765	300	764 / 765	435.93
Rotativo A	63 / 63	0	63 / 63	0.16
Rotativo B	368 / 378	300	378 / 378	88.22
Rotativo C	68 / 68	<1	66 / 68	0.08
Rotativo D	70 / 70	27	69 / 70	10.27
Rotativo E	238 / 238	44	238 / 238	0.58
Rotativo F	350 / 357	300	357 / 357	19.04

Tabla 6.11: Resultados de 300 segundos de Gurobi comenzando desde la solución inicial comparado con la búsqueda local fase 1 con 1500 iteraciones

La diferencia más notoria está en la tabla 6.10, dónde se observa que en casi todas las instancias de más de 200 viajes, 5 minutos de procesamiento de Gurobi no obtienen resultados tan buenos como la heurística de solución inicial glotona.

Capítulo 7

Conclusión

En este trabajo se presentó un modelo comprensivo y versátil del problema de Crew Scheduling. Adicionalmente se presentaron cotas inferiores para la cantidad de tripulaciones necesarias para cubrir un conjunto de viajes, las cuales permiten rápidamente verificar la factibilidad de un problema particular. Finalmente se presentó un algoritmo de búsqueda local basado en algoritmos de grafos que presenta buenos resultados experimentales.

Los métodos presentados para la búsqueda local, al igual que las heurísticas para la factibilidad están diseñados para ser versátiles, por lo que pueden fácilmente adaptarse para ser utilizadas en otros problemas similares.

Por otro lado, los resultados experimentales confirmaron el valor de este tipo de técnicas basadas en heurísticas, ya que cuando se piensa en aplicaciones industriales, la velocidad que entregan los algoritmos presentados puede ser de gran relevancia.

Una de las ventajas del aumento de velocidad es que este algoritmo ahora puede interactuar con otros sistemas. Por ejemplo, en este trabajo los viajes a realizar se tomaron como datos del problema, pero en una aplicación real, una firma debe decidir cómo y cuando realizar los viajes antes de poder llamar al software que realiza las asignaciones. Estas heurísticas rápidas permiten que los planificadores puedan probar distintas configuraciones de viajes, recibiendo información en tiempo real de parte del algoritmo sobre cuales configuraciones son factibles y cuántos trabajadores se necesitan para realizarlas.

Esta es una situación que se da frecuentemente a la hora de evaluar nuevos proyectos, dónde surge la necesidad de predecir los costos e iterar sobre múltiples posibles escenarios.

Bibliografía

- [1] J. Amaya, Molina E., N. Morales, and P. Uribe. Optimization model for rostering and crew assignment for train transportation. *Mining Goes Digital. Mueller et al. (Eds) © Taylor Francis Group, London, pp. 419-425*, 2019.
- [2] J. Amaya and P. Uribe. A model and computational tool for crew scheduling in train transportation of mine materials by using a local search strategy. *TOP-Journal of the Spanish Society of Statistics and Operations Research (26)383-402*, 2018.
- [3] O. Cheikhrouhou and I. Khoufi. A comprehensive survey on the multiple travelling salesman problem: Applications, approaches and taxonomy. *Computer Science Review 40:100369*, 2021.
- [4] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction to algorithms, third edition. 2009.
- [5] R. De Leone, P. Festa, and E. Marchitto. A bus driver scheduling problem: a new mathematical model and a grasp approximate solution. *Transportation Science 23(1):1-13.*, 2011.
- [6] G. Deng and W. Lin. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications 38(5):5787-5793*, 2011.
- [7] C. Ding, Y. Cheng, and M. He. Two-level genetic algorithm for clustered traveling salesman problem with application in large-scale tsps. *TSINGHUA SCIENCE AND TECHNOLOGY, 12(4):459-465*, 2007.
- [8] J. Heil, K. Hoffmann, and U. Buscher. Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research, 283(2):405-425*, 2020.
- [9] S. Jütte and U. Thonemann. A graph partitioning strategy for solving large-scale crew scheduling problems. *OR Spectrum, 37(1):137-170*, 2015.
- [10] A. Kasirzadeh, Saddoune M., and Soumis F. Airline crew scheduling: models, algorithms, and data sets. *The Association of European Operational Research Societies, 6(2):111-137*, 2015.
- [11] E. Khmeleva and A. A. Hopgood. Fuzzy-logic controlled genetic algorithm for the rail-freight crew-scheduling problem. *KI - Künstliche Intelligenz, 32(1):61-75*, 2018.

- [12] D. Levine. Alternating lagrangian decomposition for integrated airline crew scheduling problem. *Computers and Operations Research* 23(6):547-558, 1996.
- [13] H. Lourenço, O. Martin, and T. Stützle. 2010. iterated local search: Framework and applications. In M. GENDREAU and J. POVIN, editors, *Handbook of Metaheuristics, Springer US*, pages pp. 363–397.
- [14] H. T. Ozdemir and C. K. Mohan. Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences*, 133(3):165-173, 2001.
- [15] Z. Tian and Sont Q. Modeling and algorithms of the crew scheduling problem on high-speed railway lines. *Procedia - Social and Behavioral Sciences*, 96:1443-1452, 2013.
- [16] V. Zeighami, Saddoune M., and F. Soumis. Alternating lagrangian decomposition for integrated airline crew scheduling problem. *European Journal of Operational Research*, 287(1):211-224, 2020.
- [17] S. Zhou, Z. Zhan, Z. Chen, S. Kwong, and J. Zhang. A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):6784-6798, 2021.