



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Programación Lineal Entera aplicada al Problema de Reprogramación del Cronograma Ferroviario ante grandes interrupciones

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Agustín Mosteiro

Directores: Juan José Miranda-Bront, Federico Pousa
Buenos Aires, 2017

Resumen

Las operaciones del transporte ferroviario son influenciadas diariamente por eventos no planeados que afectan, por ejemplo, el material rodante o la infraestructura y pueden producir demoras e interrupciones en la red. Como consecuencia, hay un alto impacto en la calidad del servicio percibida por los pasajeros. Bajo estos escenarios, una de las claves es construir un nuevo cronograma para recuperarse de la interrupción e informar a los pasajeros lo más rápido posible, usualmente en unos pocos minutos. Debido a su complejidad, el problema general es usualmente dividido en tres etapas secuenciales, a saber *reprogramación de cronograma*, *reprogramación de material rodante* y *reprogramación de tripulación*. Uno de los principales desafíos involucra la factibilidad de la solución obtenida en una etapa respecto de las restricciones impuestas por la etapa próxima. Luego, el modelo intenta incluir, al menos de una manera relajada, algunas de las restricciones de las posteriores etapas con el fin de obtener una solución más robusta.

En los últimos años, ha habido una tendencia en la literatura a considerar modelos de PLEM como prototipos de sistemas automatizados de decisión para hacer frente a este tipo de situaciones. Esta investigación se basa principalmente en los trabajos de Louwerse y Huisman [9] y Veelenturf et al. [12] para el problema de reprogramación de cronograma. En ambos modelos, a pesar de las diferencias en las características particulares modeladas, el problema es formulado por medio de una red evento-actividad que luego se traduce a la formulación de PLEM, con el objetivo de obtener el nuevo cronograma. También consideran algunos aspectos generales del material rodante en la etapa del armado del cronograma. En ambos casos, los resultados muestran que se producen soluciones de buena calidad en tiempos de cómputo razonables. En particular, el trabajo de Veelenturf et al. [12] hace algunas extensiones al problema original y también tiene en cuenta el ida y vuelta desde el nuevo cronograma hacia el cronograma original, lo que es una cuestión clave para poder implementar la solución en la práctica.

En base a nuestro conocimiento, la mayor parte de la investigación relativa al problema de reprogramación de cronograma se dedica a incorporar nuevas características al modelo, mientras que pocos trabajos se enfocan en algoritmos exactos alternativos (ver Cacchiani et al. [3] para una revisión actualizada). Mejoras en los algoritmos permitirían considerar redes de mayor tamaño como así también ciertos escenarios en los cuales los tiempos de cómputo exceden los tiempos disponibles. Nuestra contribución apunta en esa dirección. Extendemos el trabajo de Veelenturf et al. [12], proponiendo una formulación PLEM alternativa basada en la llamada *Formulación por Índice de Tiempo* considerada en Dash et al. [4] para el Problema del Viajante de Comercio con Ventanas de Tiempo. Mostramos que la formulación propuesta tiene una relajación LP más ajustada. También presentamos resultados computacionales, incluyendo comparaciones respecto a los tiempos de cómputo y a la calidad de las relajaciones LP. Finalmente discutimos futuras direcciones de investigación.

Palabras claves: Gestión de interrupciones, Reprogramación de cronograma ferroviario, Programación Lineal Entera, Formulación por Índice de Tiempo.

Abstract

Normal railway operations are often influenced by unexpected situations that affect, for example, the rolling stock and the infrastructure and may cause delays and disruptions in the network. As a consequence, there is a heavy impact on the quality of service perceived by the passengers. Under these scenarios, one of the key issues is to construct a new *disposition* timetable to recover from the disruption and to inform the passengers as fast as possible, usually within a few minutes. Due to its complexity, the overall problem is usually divided into three phases which are solved sequentially, namely *timetable rescheduling*, *rolling stock rescheduling* and *crew rescheduling*. One of the major issues involves the feasibility of the solution obtained in one phase regarding the constraints imposed by the following one. Therefore, the models try to include, at least in a relaxed fashion, some of the constraints involved in successive phases in order to provide a more robust solution.

There has been a trend in the literature in the last few years to consider MILP as prototypes for automated decision systems to tackle this kind of situations. This research builds mainly upon the works by Louwerse and Huisman [9] and Veelenturf et al. [12] for the timetable rescheduling problem. In both models, despite the differences regarding the particular characteristics modeled, the problem is formulated by means of an event-activity network which is later translated into an MILP formulation, aiming to obtain the disposition timetable. They also consider some general aspects of the rolling stock rescheduling in the timetabling phase. In both cases, the results show that the approach produces good quality solutions in reasonable computing times. In particular, the work by Veelenturf et al. [12] makes some extensions to the original problem and also takes into account the back and forth from the disposition timetable to the original timetable, which is a key issue in order to implement the solution in practice.

To the best of our knowledge, most of the research regarding timetable rescheduling is devoted to incorporate new modeling characteristics, but there is limited research focusing on alternative exact algorithms for the problem (see Cacchiani et al. [3] for an updated review). Improvements in the algorithms could allow to consider larger railway networks as well as to consider certain scenarios where computing times exceed the available time. Our contribution aims in that direction. We extend the work by Veelenturf et al. [12], proposing an alternative MILP formulation based on the so-called *Time-Index Formulation* considered in Dash et al. [4] for the Traveling Salesman Problem with Time Windows. We show that the proposed formulation has a tighter LP relaxation. We further present computational results, including a comparison regarding the overall computing times and quality of the LP relaxations, and discuss future research directions.

Keywords: Disruption management, Railway timetable rescheduling, Integer programming, Time-Index Formulation.

A mi vieja por estar siempre para todo. A mi viejo por encender la chispa de la curiosidad en muchas cosas, en particular la computación.

A mi abuela por su cariño incondicional, a mi abuelo que siempre fue un ejemplo a seguir con su historia en exactas, a ambos por la gran ayuda hospedándome durante mi primer año en la facultad. A mi hermana Vicky, en especial por los años viviendo juntos en Amenabar. A toda mi familia en general.

A mis directores. A Juanjo por la infinita ayuda durante todo el trabajo, disponible en absolutamente todo momento para despejar cualquier consulta. Ni hablar de los charlas y debates súper interesantes durante todo este tiempo. A Fede, por la gran ayuda en el CBC, por ser fuente de consulta durante la carrera, por incentivar-me a ser ayudante y obviamente por todo el laburo como director. Muchas gracias a ambos!

A la cátedra de Métodos Numéricos, por la increíble experiencia de la docencia durante tres años. En especial a Juanjo y a Fede, que además de su destacado rol de directores, fueron excelentes JTPs.

A todos mis docentes y al DC en general, por la gran formación que me brindaron.

A los jurados, por analizar en detalle nuestro trabajo y contribuir a mejorarlo.

A Fede, Agus, De, Garyu, Tute, Maxi y Chou. Por todas las cosas vividas en este camino, ni hablar de las incontables épicas en la cajita del 2, vamo Fondo!

A Lucho, Fede, Mati e Igal, por todo lo que compartimos haciendo prácticamente toda la carrera juntos.

A los pibes, Tebi, Jony, Tincho, Juve, Matu, Seba, Cochi, Dug, Camisa, Maken, Facu, Fede, Pablito. Que equipito.. gracias por tantos años de una amistad increíble.

A Facu, Tito, Juli, Pitu y todos mis amigos del colegio Hotton por tantas historias.

Al Garyu y a Cochi, por los momentos vividos junto al rojo, todas las canchas, alegrías, tristezas y las interminables anécdotas.

A Lucho, Mati, Agus, Tobi, Sherman, Juan, Facu, Ro, Tincho y Maxi por la gran compañía de todos los días junto a Eryx.

A Flo, mi compañera de siempre, mi cómplice en todo, mi amor.

Índice general

| | |
|---|----|
| 1. Introducción al problema de Gestión de Interrupciones en transporte ferroviario | 1 |
| 2. Introducción a la Programación Lineal Entera | 6 |
| 2.1. Definición | 6 |
| 2.2. Formulaciones | 7 |
| 2.3. Algoritmos para Problemas de Programación Lineal Entera | 9 |
| 2.3.1. Algoritmos branch-and-bound | 10 |
| 2.3.2. Algoritmos de planos de corte | 13 |
| 2.3.3. Algoritmos branch-and-cut | 15 |
| 3. Red Evento-Actividad y Formulación PLE del problema | 18 |
| 3.1. Descripción general del problema | 18 |
| 3.1.1. Programación original y restricciones generales | 19 |
| 3.1.2. Recursos disponibles para el funcionamiento del servicio | 19 |
| 3.1.3. Bloqueo de vías | 20 |
| 3.2. Red Evento-Actividad | 21 |
| 3.2.1. Eventos | 22 |
| 3.2.2. Actividades | 23 |
| 3.3. Formulación PLE Base | 25 |
| 3.4. Restricciones de recursos | 27 |
| 3.4.1. Vías de tramos | 29 |
| 3.4.2. Vías dentro de una estación | 35 |
| 3.4.3. Material rodante | 38 |
| 3.4.4. Relación entre las vías de una estación y el manejo del material rodante | 41 |
| 3.4.5. Aclaraciones sobre los conjuntos $A^-()$ y $A^+()$ | 44 |
| 3.5. Bloqueo de una sección de vías | 45 |
| 3.5.1. Trenes sobre el área de bloqueo | 46 |
| 3.5.2. Armado del conjunto de eventos E | 47 |
| 3.5.3. Factibilidad post \lim_E^+ del material rodante | 49 |
| 3.5.4. Consideraciones para los trenes andando en \lim_E^- | 50 |
| 3.5.5. Disponibilidad de vías de tramo en la sección del bloqueo | 53 |
| 3.6. Formulación Orig | 57 |
| 4. Formulación alternativa | 58 |
| 4.1. Nuevos conjuntos de variables | 59 |
| 4.1.1. Binarización de variables de tiempo de eventos | 60 |
| 4.1.2. Reescritura de variables de actividad por índice de tiempo | 62 |
| 4.1.3. Variables de liberación de recurso | 63 |
| 4.2. Formulación TIF | 65 |
| 4.3. Relajación lineal TIF vs Orig | 66 |
| 4.4. Equivalencia del conjunto de soluciones | 73 |

| | |
|--|----|
| 5. Experimentación y Resultados | 75 |
| 5.1. Caso de estudio: Línea Mitre | 75 |
| 5.2. Parámetros | 77 |
| 5.3. Escenarios de bloqueo | 78 |
| 5.4. Preprocesamiento | 79 |
| 5.5. Configuración Algoritmo <i>Branch-and-Cut</i> | 81 |
| 5.5.1. Algoritmo de resolución nodo raíz y Selección de nodo | 81 |
| 5.5.2. Branching por variables | 82 |
| 5.6. Errores Numéricos | 86 |
| 5.7. Comparación Orig vs TIF | 87 |
| 6. Conclusiones y trabajo futuro | 96 |

1. Introducción al problema de Gestión de Interrupciones en transporte ferroviario

Las operaciones del transporte ferroviario son influenciadas diariamente por eventos no planeados que pueden producir desde pequeñas demoras hasta grandes interrupciones. En consecuencia, se tiene un impacto directo en la calidad del servicio. Bajo este contexto, uno de los principales objetivos es recuperarse de estas situaciones lo antes posible.

La investigación en el area de *Gestión de Interrupciones* es desafiante desde un punto de vista científico, pero al mismo tiempo prometedora desde un punto de vista práctico. En efecto, incrementar la porción de mercado del transporte público, en especial del transporte ferroviario, es considerada una de las soluciones principales a los problemas de movilidad en grandes ciudades. Además, el ferrocarril es considerado un modo ecológico de transporte. Luego, incrementar el uso del servicio de la red ferroviaria es una cuestión de alta prioridad para muchos gobiernos.

El problema principal es que muchos potenciales pasajeros son desalentados a utilizar este tipo de transporte por la percepción del nivel de servicio ante incidentes. La aplicación de modelos y algoritmos de recuperación para atacar las demoras e interrupciones, implementados mediante *Sistemas de soporte a decisiones* (DSS, por sus siglas en inglés), es clave para elevar la calidad del servicio percibida por los pasajeros.

En el trabajo de Cacchiani et al. [3] se presenta un resumen completo del estado del arte en la aplicación de técnicas de optimización e investigación operativa a problemas de recuperación. En dicho estudio, se realiza una distinción importante entre los eventos inesperados que pueden afectar a una red ferroviaria. Por un lado, las demoras son perturbaciones relativamente menores del sistema ferroviario que pueden ser manejadas modificando únicamente el cronograma, sin necesidad de reprogramar las tareas de material rodante ni de las tripulaciones. Por otro lado, las interrupciones son incidentes relativamente largos, que requieren una reprogramación tanto del cronograma como del programa de material rodante y tripulación.

En consecuencia, las demoras e interrupciones son atacadas con enfoques diferentes, dando lugar al problema de *Gestión de Demoras* y *Gestión de Interrupciones* respectivamente. En nuestro trabajo nos enfocamos en el segundo problema, bajo el contexto de grandes interrupciones.

Los causales de las interrupciones pueden ser muchos, entre ellos podemos mencionar desperfectos en el material rodante o en la señalización, accidentes u obras (planificadas) de mantenimiento. La gestión de interrupciones se ocupa de recuperarse de dichas situaciones y no de prevenirlas.

Como consecuencia de las interrupciones, la capacidad de la red se ve afectada (dis-

minuída) mientras dure la interrupción. El cronograma original puede volverse imposible de cumplir, siendo necesario no solo retrasar algunos servicios, sino también cancelar otros.

Para tener una primera intuición de la magnitud del problema, en la Figura 1.1 se muestran algunas estadísticas del año 2016 de la red ferroviaria de pasajeros del area Metropolitana de Buenos Aires¹. Como se puede ver, aproximadamente el 6% del las operaciones del 2016 se vio afectada por cancelaciones, llegando en algunos meses a estar cerca del 10%.

| RED FERROVIARIA DE PASAJEROS DEL AREA METROPOLITANA DE BUENOS AIRES | | | | | | | | |
|---|--------------------|-----------|-------------------|--------------------|------------|----------|-----------|-----------|
| Cumplimiento de la Programación de Trenes 2016 | | | | | | | | |
| mes 2016 | Regularidad Global | | Cump. de Programa | Cantidad de trenes | | | | |
| | Absoluta | Relativa | | Programados | Cancelados | Corridos | Puntuales | Atrasados |
| | Punt/Prog | Punt/Corr | Corr/Prog | | | | | |
| ene | 77.56% | 84.98% | 91.26% | 55.843 | 4.879 | 50.964 | 43.311 | 7.653 |
| feb | 77.66% | 83.89% | 92.57% | 50.463 | 3.750 | 46.713 | 39.188 | 7.525 |
| mar | 81.05% | 86.04% | 94.20% | 59.480 | 3.450 | 56.030 | 48.207 | 7.823 |
| abr | 76.31% | 82.08% | 92.98% | 58.932 | 4.139 | 54.793 | 44.972 | 9.821 |
| may | 77.32% | 82.93% | 93.24% | 60.292 | 4.075 | 56.217 | 46.620 | 9.597 |
| jun | 77.69% | 83.43% | 93.12% | 58.326 | 4.013 | 54.313 | 45.314 | 8.999 |
| jul | 79.07% | 84.64% | 93.41% | 59.563 | 3.923 | 55.640 | 47.094 | 8.546 |
| ago | 80.75% | 85.02% | 94.98% | 60.845 | 3.057 | 57.788 | 49.131 | 8.657 |
| sep | 79.96% | 83.96% | 95.24% | 59.222 | 2.819 | 56.403 | 47.354 | 9.049 |
| oct | 81.77% | 85.76% | 95.35% | 59.592 | 2.774 | 56.818 | 48.726 | 8.092 |
| nov | 83.85% | 86.95% | 96.44% | 58.469 | 2.081 | 56.388 | 49.028 | 7.360 |
| dic | 80.19% | 85.68% | 93.60% | 58.879 | 3.769 | 55.110 | 47.218 | 7.892 |
| Promedio | 79.46% | 84.63% | 93.90% | 58.326 | 3.561 | 54.765 | 46.347 | 8.418 |
| Total | --- | --- | --- | 699.906 | 42.729 | 657.177 | 556.163 | 101.014 |

Fig. 1.1: Estadísticas del año 2016 de la red ferroviaria de pasajeros del area Metropolitana de Buenos Aires. En color rojo se resaltan los datos relativos a cancelaciones.

Para nuestro trabajo consideramos interrupciones relacionadas con bloqueos de una o más vías entre estaciones por un período de tiempo determinado (e.g. dos horas). En efecto, interrupciones de este tipo son muy difíciles de manejar por los operadores ferroviarios y los administradores de la infraestructura, debido a que causan numerosos cambios en el sistema y se deben tomar decisiones rápidamente. En particular, algunas de las decisiones principales a tomar son:

- Confección de un nuevo cronograma: qué servicios matener, cuáles retrasar (y cuánto), y cuáles cancelar.
- Circulación de material rodante: cómo re-assignar el material rodante en base al nuevo cronograma.
- Reasignación de tripulaciones: cómo re-assignar el personal, cumpliendo las restricciones de seguridad y convenios laborales.

Más aún, como vimos en la Figura 1.1, estas interrupciones ocurren regularmente. En base a esto, el apoyo de una herramienta automatizada para resolver el problema es alta-

¹ Estadísticas oobtenidas de la CNRT (Comisión Nacional de Regulación del Transporte), <https://www.cnrt.gob.ar/estadisticas-del-transporte-ferroviario>

mente deseable.

Actualmente, estos problemas son resueltos en la práctica mediante la implementación de un plan de contingencia apropiado, seleccionado a partir de un conjunto de planes disponibles. Uno de los principales problemas de esta solución es que el conjunto de casos predefinido no puede capturar todos los posibles escenarios que pueden surgir. Los planes de contingencia aplicados tienen un impacto directo en la calidad y confiabilidad del servicio.

Además, otra problemática de esta solución es que las decisiones son tomadas en base a la experiencia del despachante, con poca o nula utilización de herramientas de IT que asistan a la toma de decisiones. En este contexto, en caso de que cambie la persona cumpliendo esta tarea, la solución ante un mismo escenario seguramente también se verá modificada. En los países más avanzados, la tendencia es a incorporar DSS como herramienta de asistencia.

Debido a la complejidad del problema de recuperación, en general su resolución en tiempo real en redes de mediano o gran tamaño es computacionalmente inviable. Debido a esto, el problema es decompuesto en subproblemas más pequeños, cada uno atacando etapas específicas del proceso. Típicamente en la literatura, el problema general se subdivide en los problemas de *Reprogramación de cronograma*, *Reprogramación del material rodante* y *Reprogramación de la tripulación*. En la Figura 1.2 podemos ver un diagrama que explica como estas distintas etapas se combinan para resolver el problema completo. El diagrama es incluido a partir del trabajo de Besinovic et al. [2], donde los autores desarrollan un *framework* en el cual comunican distintos modelos conocidos para resolver cada etapa.

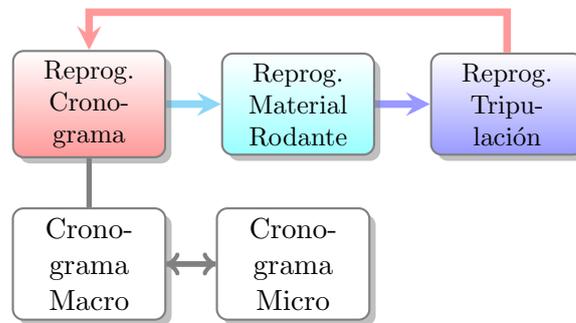


Fig. 1.2: Diagrama del problema de Gestión de Interrupciones, subdividido en etapas. Basado en el trabajo de Besinovic et al. [2].

En cuanto a la Figura 1.2, más allá de los detalles particulares de la implementación, es interesante destacar que uno de los desafíos principales es la factibilidad. Cuando se obtiene una solución para una etapa, la misma es utilizada como entrada para el próximo problema, el cual podría resultar infactible. En dicho caso, la solución de la etapa anterior tendrá que ser alterada para poder obtener una solución factible en la etapa siguiente. Dicho esquema iterativo se repite hasta encontrar una solución factible para el problema completo.

Notar como en el diagrama el problema de Reprogramación de cronograma es a su vez subdividido a nivel macroscópico y microscópico. Los abordajes a nivel macroscópico

modelan la infraestructura con un alto nivel de abstracción y usualmente se utilizan para resolver problemas ante grandes interrupciones. En contraste, los modelos microscópicos consideran la infraestructura ferroviaria con un alto nivel de detalle. Al hacerlo, la propagación de los retrasos se puede modelar con alta precisión. En general, estos modelos se aplican para resolver pequeñas alteraciones. Por ejemplo, en Besinovic et al. [2], se utilizan implementaciones de ambos modelos de manera conjunta para resolver de manera más precisa el problema de Reprogramación de cronograma ante una gran interrupción.

Finalmente, en Dollevoet et al. [5] se puede encontrar una versión más actualizada de una implementación de un framework para el problema completo, en el cual también se incluye un completo apartado de experimentos computacionales. En el mismo se trabaja sobre un caso de estudio de una red holandesa, donde se obtienen resultados interesantes para su aplicación en la práctica.

En nuestro trabajo, nos enfocamos en el problema de Reprogramación de cronograma a nivel macroscópico. Nos basamos principalmente en los trabajos relacionados de Louwerse y Huisman [9] y Veelenturf et al. [12]. Ambos estudios modelan el estado de la red y la capacidad de sus recursos mediante una *Red Evento-Actividad*. Es una estructura típicamente usada en problemas de este tipo. Por ejemplo, en Schachtebeck y Schöbel [11] es utilizada para resolver el problema de Gestión de demoras.

Los dos trabajos mencionados en los cuales nos basamos formulan el problema como un problema de *Programación Lineal Entera*, construido a partir de la red evento-actividad mencionada. Además, dichos estudios consideran, al menos parcialmente, algunas restricciones de la etapa de asignación del material rodante, con el objetivo de facilitar la factibilidad en la próxima etapa.

Una de las diferencias principales entre el trabajo de Veelenturf et al. [12] y Louwerse y Huisman [9] es que este último no considera la transición desde el cronograma original hacia el nuevo cronograma generado por el modelo, ni tampoco el traspaso de este al original nuevamente. Se asume además que la red se encuentra vacía al momento en que ocurre el bloqueo de vías. Tener en cuenta estos detalles es clave para la aplicación de estas herramientas en casos reales. Por dicho motivo, nuestro estudio se basa particularmente en el trabajo de Veelenturf et al. [12]. Vale mencionar que este modelo es el utilizado tanto en Besinovic et al. [2] como en Dollevoet et al. [5] para resolver el problema de Reprogramación de cronograma a nivel macroscópico.

Como se mencionó anteriormente, el desafío es por un lado construir un nuevo cronograma que minimice el impacto en la calidad del servicio. Por otro lado, se poseen solo algunos pocos minutos para decidir el nuevo cronograma, con el objetivo de obtener una solución en tiempo real. Mejoras en los modelos y algoritmos permitirían considerar redes de mayor tamaño como así también escenarios en donde los tiempos de cómputo exceden los tiempos deseados.

El objetivo de nuestro trabajo apunta en esta dirección. Para lograrlo, proponemos una formulación de PLE alternativa basada en la llamada *Formulación por Índice de Tiempo* considerada en Dash et al. [4] para el *Problema del Viajante de Comercio con Ventanas de Tiempo*. Por las características de la formulación utilizada en Veelenturf et al. [12] y por las ventajas del nuevo modelo propuesto, demostramos que nuestra formulación alternativa posee una relajación lineal más ajustada, aportando un resultado teórico que

relaciona ambos modelos. Asimismo, realizamos experimentos preliminares sobre un caso de estudio de una red del área Metropolitana de Buenos Aires. En dicha experimentación comparamos la calidad de la relajación lineal de ambas formulaciones como así también sus tiempos de cómputo. Observamos como los modelos se presentan factibles para su aplicación en la práctica. En particular nuestra formulación es capaz de resolver una mayor cantidad de instancias y disminuir los tiempos de cómputo sobre un subconjunto de ellas.

Nuestro trabajo se organiza de la siguiente manera. En el Capítulo 2 se incluye una introducción a la Programación Lineal Entera. En el Capítulo 3 se describe la red evento-actividad mencionada y la formulación de PLE de Veelenturf et al. [12]. Vale destacar que por la complejidad del problema, en dicho capítulo explicamos como modelar distintos aspectos del modelo en detalle, lo cual creemos es un valioso aporte dado que no encontramos este nivel de detalle en la literatura. Luego, en el Capítulo 4 presentamos nuestra formulación alternativa y un estudio teórico comparando ambas formulaciones. En el Capítulo 5 se incluyen los experimentos computacionales realizados. Finalmente, en el Capítulo 6 incluimos las conclusiones y el trabajo a futuro.

2. Introducción a la Programación Lineal Entera

2.1. Definición

La Programación Lineal es una técnica para resolver problemas de optimización. Mediante una formulación de Programación Lineal se pueden modelar situaciones donde se debe minimizar (o maximizar) una función lineal, denominada función objetivo, sujeta a un conjunto de restricciones, también lineales. Todo problema de Programación Lineal (PL) es resoluble de manera eficiente [8], pero existen muchos problemas que no se pueden modelar con esta técnica a menos que se restrinja el dominio de las variables.

La Programación Lineal Entera (PLE) está íntimamente relacionada con la Programación Lineal, pero tiene la diferencia de que todas las variables están restringidas a tomar valores enteros.

La Programación Lineal Entera Mixta (PEM) es una generalización de las dos anteriores, en donde algunas o todas las variables están restringidas a valores enteros.

La Programación Lineal Entera Mixta no solo puede resolver los mismos problemas que resuelve la Programación Lineal, por definición, sino que también puede modelar problemas donde las variables son enteras. Esto por un lado se traduce en problemas donde no se pueden tener valores reales, por ejemplo en los problemas de producción de unidades indivisibles; pero también se traduce en modelos mucho más potentes, ya que la presencia de variables enteras o binarias permiten la modelización de restricciones lógicas como conjunciones, disyunciones y consecuencias. El caso más emblemático es el problema SAT, que puede ser resuelto mediante Programación Lineal Entera Mixta y no mediante Programación Lineal.

Esta diferencia entre los problemas que se pueden modelar, hace que la Programación Lineal Entera Mixta sea NP-Difícil en general. Por lo que no existe ningún algoritmo polinomial que resuelva cualquier problema de Programación Lineal Entera Mixta a menos que $P = NP$. Una demostración de que la Programación Lineal Entera Mixta es NP-Difícil se obtiene justamente mostrando que SAT se puede reducir a un problema de Programación Lineal Entera Mixta.

La Programación Lineal Entera Mixta puede ser utilizada para resolver variados problemas de optimización, entre los que se encuentran problemas de ruteo de vehículos, problemas de telecomunicaciones, problemas de planificación de producción, problemas de asignación de tareas y turnos, entre otros.

En las próximas secciones se detallan algunos conceptos básicos de la Programación Lineal Entera Mixta y se presentan los algoritmos generales de resolución conocidos. Para una ampliación de los temas aquí introducidos ver [13].

2.2. Formulaciones

Un problema de PEM puede ser formulado de la siguiente manera:

$$\begin{aligned}
 & \text{Minimizar} && \sum_{j \in I} c_j x_j + \sum_{j \in C} c_j x_j \\
 & \text{sujeto a} && \\
 & && \sum_{j \in I} a_{ij} x_j + \sum_{j \in C} a_{ij} x_j \leq b_i \quad i = 1, \dots, m \\
 & && x_j \in \mathbb{Z}_+ \quad \forall j \in I \\
 & && x_j \in \mathbb{R}_+ \quad \forall j \in C
 \end{aligned}$$

donde I es el conjunto de variables enteras, C es el conjunto de variables continuas y m es la cantidad de restricciones.

Cada formulación PEM tiene asociado un poliedro $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$ con $A \in \mathbb{R}^{m \times n}$ y $b \in \mathbb{R}^m$ y el conjunto de soluciones factibles $S = P \cap \{x \in \mathbb{R}^n : x_j \in \mathbb{Z} \forall j \in I\}$. A P se lo denomina relajación lineal de S .

Cada desigualdad presentada en las restricciones divide el espacio donde viven las soluciones en dos subespacios. El subespacio que será tenido en cuenta para definir los puntos factibles dependerá de la orientación de la desigualdad que se tenga que cumplir.

A continuación se presenta el poliedro asociado a una formulación con solo dos variables, y ambas restringidas a ser enteras.

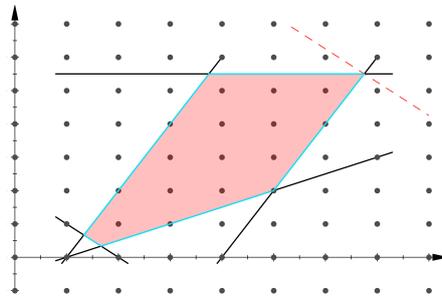


Fig. 2.1: Conjunto factible y poliedro asociado

Las líneas que definen el poliedro son las desigualdades presentes en la formulación, y el área encerrada por estas líneas resulta en el poliedro asociado gracias a la orientación de las desigualdades. Si bien este poliedro contiene muchos puntos, el conjunto de soluciones factibles S son los puntos enteros dentro de P .

La línea que se observa punteada corresponde a la función objetivo. Al ser esta una función lineal, en el espacio queda representada por un hiperplano, en este caso una línea, que solo tiene definida su pendiente ya que no tiene término independiente como las desigualdades de las restricciones. Es decir, la función objetivo es un hiperplano con pendiente definida que toma diferentes valores según donde esté ubicada. Lo buscado es que la función objetivo pase por el punto del conjunto factible de soluciones que mayor valor le haga

tomar. En el caso de la figura la función objetivo se encuentra optimizada para todos los puntos del poliedro P , y la mejor solución se encuentra en la esquina superior derecha.

En los problemas de Programación Lineal las soluciones factibles son todos los puntos del poliedro ya que no está la restricción de integralidad. Esto hace que en un poliedro acotado, como el mostrado en la figura, se pueda demostrar que la mejor solución está en uno de los puntos extremos o esquinas del mismo. Por lo que en un problema de Programación Lineal, se pueden recorrer las esquinas hasta encontrar la que tenga menor valor objetivo, siempre haciendo referencia a una minimización.

Si llamamos $conv(S)$ a la cápsula convexa de S (menor poliedro que contiene a S), entonces PEM es equivalente a resolver $Min\ cx : x \in conv(S)$. Si $P = conv(S)$, el problema PEM puede ser resuelto en forma eficiente por cualquier algoritmo de Programación Lineal.

Esto sucede porque un algoritmo para resolver Programación Lineal puede encontrar el mínimo en todo el poliedro P y, por lo dicho anteriormente, este mínimo será alguna de las esquinas del poliedro. Como en la cápsula convexa las esquinas coinciden con puntos enteros, entonces cuando se minimice el problema sobre todo P , resultará que el mínimo es un punto entero, por lo que este punto es mejor, en términos de la función objetivo, que cualquier otro punto de P y en particular que cualquier otro punto de S , ya que $S \subseteq P$.

Si se conociese la descripción de $conv(S)$ mediante un número polinomial (en la cantidad de variables) de desigualdades lineales, se podría resolver el problema como uno de Programación Lineal, lo cual es computacionalmente fácil. Es más, aún en el caso que esta caracterización no fuese polinomial, bajo ciertas circunstancias el problema podría ser resuelto en tiempo polinomial. Desafortunadamente, para la mayoría de los problemas no se ha podido obtener la descripción completa de la cápsula convexa y, en general, el número de restricciones lineales que la caracterizan es exponencial.

Para un mismo problema existen muchas formas de formularlo, a veces con las mismas variables existen otras restricciones que también modelan el problema, pero que devienen lógicamente en poliedros diferentes. Otras veces un problema se puede formular con dos conjuntos de variables diferentes, lo que hace que probablemente los poliedros asociados ni siquiera pertenezcan a la misma dimensión.

En la siguiente figura se pueden ver los poliedros asociados a dos formulaciones diferentes, siempre manteniéndose en dos dimensiones para poder visualizarlo correctamente.

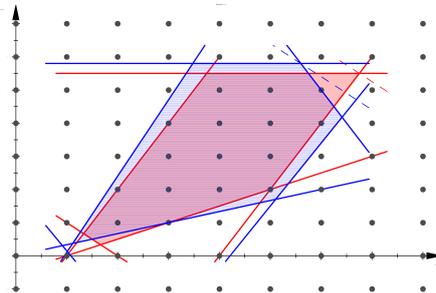


Fig. 2.2: Dos formulaciones y sus poliedros asociados

Por un lado se ve el poliedro rojo opaco, y por el otro el poliedro azul a rayas. Lo que se puede ver es que el conjunto de puntos enteros S es el mismo, pero el poliedro asociado P es diferente.

Se dice que una formulación $F1$ es *más ajustada* que otra, $F2$, cuando el poliedro asociado a $F1$ está incluido en el poliedro asociado a $F2$. En este caso $F1$ logra capturar todos los puntos enteros posibles y contiene menos puntos fraccionarios obsoletos que $F2$.

Esta noción conforma una relación de orden que no es total. En la figura anterior se ve que hay sectores coloreados de rojo, pero no de azul, y viceversa. Esto indica que ninguna de las dos formulaciones es más ajustada que la otra.

Bajo esta definición, la cápsula convexa de un conjunto de soluciones factibles representa la formulación más ajustada que se puede conseguir. En la siguiente figura se muestra la primera formulación visualizada, junto con la cápsula convexa de los puntos de S . En rojo opaco se encuentra la primera formulación, y con la línea gruesa azul se remarca la cápsula convexa.

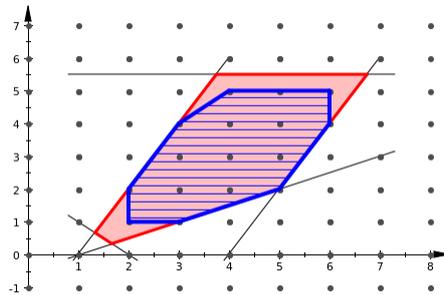


Fig. 2.3: Formulación y Cápsula convexa

El objetivo para un problema de Programación Lineal Entera es diseñar las formulaciones más ajustadas posibles. Esta tarea en general resulta difícil, dado que la ayuda gráfica solo sirve para los problemas en dos o tres dimensiones, cuando muchos de los problemas resueltos suelen tener cientos de miles de variables.

2.3. Algoritmos para Problemas de Programación Lineal Entera

El procedimiento más simple para resolver un problema de Programación Lineal Entera pura es enumerar todas las posibilidades. Sin embargo, debido a la explosión combinatoria esta técnica solo resulta aplicable a instancias sumamente pequeñas. En esta sección se describen los algoritmos más usados en la práctica.

Los algoritmos más utilizados se encuadran dentro de algunos de estos esquemas básicos:

- Enumeración inteligente: algoritmos *Branch-and-Bound*.
- Caracterización de $conv(S)$ o ajuste de la relajación lineal: algoritmos de planos de corte.
- Una combinación de las dos técnicas anteriores: algoritmos *Branch-and-Cut*.

A continuación se describen los puntos más sobresalientes de cada uno.

2.3.1. Algoritmos branch-and-bound

Ya se mencionó que la enumeración de las soluciones factibles en busca de la solución óptima no es un procedimiento aconsejable para usar en la práctica. Para mejorar esta técnica básica muchas veces es posible eliminar algunas posibilidades mediante argumentos de dominancia o factibilidad. Es decir, argumentos que permiten afirmar que el óptimo no pertenece a un determinado subconjunto de soluciones sin la necesidad de enumerarlo.

Dentro de esta línea, en los años 60 fueron propuestos los algoritmos *Branch-and-Bound*, donde el *branching* se refiere a la parte enumerativa y el *bounding* al proceso de poda de posibles soluciones.

Estos algoritmos están asociados al concepto *divide y conquista*: si resulta difícil buscar el óptimo en un conjunto S , entonces es mejor buscar en partes de S y luego quedarse con la mejor solución.

Este esquema puede ser representado mediante un árbol cuya raíz corresponde al problema original y sus ramas resultan de la división en partes del espacio de búsqueda. A cada nodo del árbol le corresponde un subproblema que consiste en buscar el óptimo en una parte del espacio de soluciones. Los argumentos de dominancia y factibilidad son los que permitirán descartar ramas del árbol en el proceso de búsqueda.

Una forma de llevar a cabo la poda, *bounding*, es calcular en los nodos del árbol cotas inferiores del óptimo del problema restringido a esa parte del espacio de soluciones. Si la cota es peor que la mejor solución obtenida hasta el momento, no es necesario explorar dicha parte. El cálculo de estas cotas debe lograr un equilibrio entre calidad y esfuerzo en obtenerla. Una cota débil hará que se explore innecesariamente ramas del árbol, pero un procedimiento que brinde buenas cotas a un costo alto puede no justificarse.

Para obtener cotas inferiores, una posibilidad es relajar el problema de forma de obtener una relajación *fácil* de resolver. La idea es reemplazar un PEM difícil por un problema de optimización más simple cuyo valor óptimo sea menor o igual al óptimo del problema original. Las cotas obtenidas a partir del problema relajado son comúnmente llamadas *cotas duales*, mientras que las soluciones exploradas que cumplen las restricciones de nuestro problema original son llamadas *cotas primales*. Obviamente, es deseable obtener relajaciones *ajustadas*, es decir, que la diferencia relativa (*gap*) entre el valor óptimo de la relajación y el valor óptimo del PEM sea chica. Hay dos posibilidades obvias para que el problema relajado tenga esta característica. Se puede agrandar el conjunto de soluciones factibles sobre el cual se optimiza o reemplazar la función objetivo por otra que tenga igual o menor óptimo. Dentro de la primera posibilidad se encuentra la relajación lineal y en la segunda se enmarca la relajación lagrangeana. Las relajaciones no solo son útiles para obtener cotas duales, algunas veces permiten probar optimalidad.

La relajación lineal consiste en borrar del PEM la imposición de ser entera sobre las variables que la tengan, es lo que en los gráficos mostrados anteriormente se enunciaba como el poliedro P asociado a S . Es la relajación más natural y una de las más utilizadas.

Bajo este esquema, tendremos tres escenarios en los cuales podremos podar el árbol de subproblemas. Una es la comentada anteriormente, donde si el óptimo de la relajación en el nodo es peor que la mejor solución entera conocida, podemos descartar dicha parte. Llamamos a esta poda *bound* por dominancia. Por otro lado, si al calcular la relajación de un nodo, nos encontramos con que el problema es infactible, también podemos aplicar una poda, dado que si no hay solución para el problema relajado, tampoco lo habrá para

el problema original. Llamamos a esta poda *bound* por infactibilidad. Finalmente, en caso de que la solución de la relajación arroje una solución que cumpla con las restricciones de integralidad, tampoco será necesario dividir la región del nodo actual en subproblemas, dado que ya se obtuvo la solución óptima para nuestro problema original en dicha región. Llamamos a esta poda *bound* por integralidad.

En cuanto al cálculo de la relajación de nuestro problema, tanto en el nodo raíz como en los nodos restantes de nuestro árbol, existen distintos algoritmos para calcularla. Uno de los más utilizados es el algoritmo *Simplex*, dado que tiene la particularidad de ser muy eficiente en el contexto de *Branch-and-Bound*. Esto se debe a que al ir recorriendo cada nodo, *Simplex* puede aprovechar los resultados en los nodos anteriores y reutilizarlos, evitando así resolver las relajaciones en cada nodo desde el inicio.

Otro algoritmo muy conocido para resolver problemas de programación lineal donde las variables son continuas es el algoritmo *Barrier*. Este es un algoritmo de la clase denominada de punto fijo, y suele ser particularmente eficiente en problemas de gran tamaño.

Problemas que son numéricamente difíciles para un algoritmo pueden ser más fáciles para el otro y viceversa. Si bien el algoritmo de *Barrier* no posee la propiedad mencionada que posee *Simplex*, si debe considerarse para obtener el óptimo de la relajación en el nodo raíz. En muchos problemas de PLE, el cálculo de la relajación inicial conlleva un tiempo considerable del tiempo total necesario para resolver el problema.

Hay dos factores decisivos en la implementación de un algoritmo de este tipo que se deben determinar: las reglas de *branching* y el esquema de selección del próximo nodo a explorar. No hay una combinación de estos factores que resulte mejor para todos los problemas. Es necesario utilizar criterios basados en una combinación de teoría, sentido común y experimentación.

El proceso de *branching* consiste en dividir la región factible anterior en dos o más regiones factibles más pequeñas. Cada nueva región da origen a un nuevo subproblema o nodo hijo, originado por la adición de una nueva restricción al problema del nodo padre. Un requerimiento esencial es que cada solución entera factible del nodo padre pertenezca a, al menos, uno de los hijos. Estos nuevos subproblemas son agregados a la lista de nodos activos, es decir, aún no explorados. La regla de *branching* más simple consiste en considerar alguna variable entera que tiene valor fraccionario, d , en la solución actual. Se parte al problema en dos hijos, imponiendo en uno de ellos como cota superior de este variable el valor $\lfloor d \rfloor$ y en el otro como cota inferior $\lceil d \rceil$. Este procedimiento es aplicado recursivamente a cada nodo del árbol.

En la siguiente figura se puede ver como la formulación original es partida en dos subproblemas, tomando la variable x y separando cuando vale menor o igual a 3, y cuando vale mayor o igual a 4. Se puede ver que esta separación propuesta cumple con el requisito de que toda solución entera factible se encuentra en alguno de los dos subproblemas nuevos.

Es importante notar que la idea de esta simple regla de *branching* tiene por objetivo también que el punto fraccionario óptimo actual no se encuentre en ninguno de los dos subproblemas. Tomando la imagen anterior, si el óptimo de la relajación lineal hubiese estado en la esquina superior izquierda del poliedro donde la variable x vale entre 3 y 4, el *branching* propuesto lograría dejar de tener en cuenta este punto.

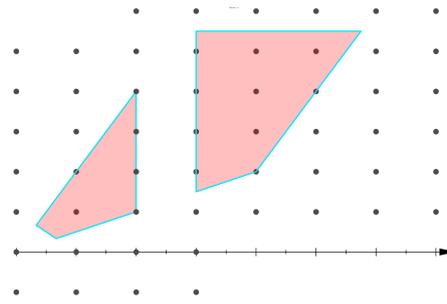


Fig. 2.4: Regla de Branching

La próxima decisión que se debe tomar es la selección del siguiente nodo a explorar de la lista de nodos activos. En la práctica hay varios argumentos contradictorios que pueden ser utilizados. Como solo es posible podar significativamente el árbol de enumeración si se cuenta con buenas cotas primales, entonces se debería descender lo más pronto posible en el árbol para encontrar rápidamente soluciones factibles. Esto sugiere el uso de una estrategia de *búsqueda en profundidad*. Otra estrategia sugiere elegir el nodo activo con mejor cota dual, buscando mejorar el valor actual de la misma. Esta estrategia es llamada *mejor cota primero*.

El esquema básico del algoritmo es el siguiente. Llámese PEM el problema entero mixto que se quiere resolver, \mathcal{N} al conjunto de subproblemas o nodos del árbol de enumeración activos. Para cada nodo k , $PL(k)$ representa la relajación lineal del PEM asociado a este nodo y Z^k el valor óptimo de $PL(k)$. En Z^* se almacena el valor de la mejor solución obtenida.

1. **Inicialización:**

$$\mathcal{N} = \{PEM\} \quad Z^* = \infty$$

2. **Elección de próximo nodo:**

Si $\mathcal{N} = \{\}$ el algoritmo termina. Si $Z^* \neq \infty$ entonces es óptimo. Si no, PEM es no factible

Si $\mathcal{N} \neq \{\}$, elegir y borrar un nodo k de \mathcal{N}

3. **Evaluación:**

Resolver $PL(k)$.

- a) Si no es factible, ir a **Elección**. (*Bound* por infactibilidad)
- b) Si $Z^k > Z^*$, ir a **Elección**. (*Bound* por dominancia)
- c) Si la solución óptima cumple las condiciones de integralidad, actualizar $Z^* = \min\{Z^*, Z^k\}$ e ir a **Elección**. (*Bound* por integralidad)

4. **División:** (*Branch*) Particionar la región factible de $PL(k)$ en dos o más regiones, agregando un nuevo nodo a \mathcal{N} por cada nueva región. Ir a **Elección**.

2.3.2. Algoritmos de planos de corte

Los algoritmos de planos de corte fueron originalmente propuestos por Gomory en los 60's [6] como un método general para resolver problemas de programación lineal entera.

Un algoritmo básico de planos de corte en un primer paso relaja las condiciones de integralidad sobre las variables y resuelve el programa lineal resultante. Si el programa lineal es infactible, el entero también lo es. Si la solución óptima del programa lineal cumple las condiciones de integralidad, se ha encontrado un óptimo del problema. En caso contrario, se busca identificar desigualdades lineales (*problema de separación*) que estén violadas por la solución fraccionaria del programa lineal y sean válidas para los puntos enteros factibles. Es decir, desigualdades que *separen* el óptimo fraccionario de $conv(S)$.

Siguiendo con el ejemplo gráfico que se presentó anteriormente, se puede visualizar la idea de este método. En la siguiente figura se muestra la formulación original con su óptimo en la relajación lineal, y una desigualdad válida que fortalece al modelo y deja afuera al anterior óptimo ya que era un punto fraccionario que no se quiere tener en cuenta.

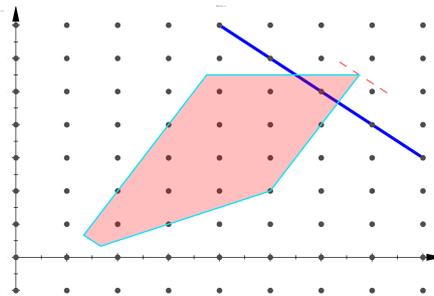


Fig. 2.5: Plano de corte

Se observa como la línea gruesa azul corta la solución fraccionaria óptima que existía hasta el momento, pero no corta ningún punto entero factible. De esta manera, si se vuelve a resolver la relajación lineal, pero ahora con el modelo fortalecido, dará un óptimo diferente.

El algoritmo continúa hasta que:

- una solución óptima entera es encontrada, en cuyo caso el problema es resuelto con éxito
- el programa lineal es infactible, lo que significa que el problema entero es infactible
- no se pudo identificar alguna desigualdad lineal que separe al óptimo de la relajación actual, ya sea porque no se conoce la descripción completa de la cápsula convexa o porque los algoritmos de separación no son exactos.

El éxito del algoritmo depende en gran medida de la posibilidad y la eficiencia de encontrar desigualdades violadas (*planos de corte*) que puedan ser agregadas a la formulación para separar las soluciones fraccionarias.

Los planos de corte pueden ser generados bajo dos enfoques:

- *Con herramientas generales aplicables a cualquier problema de programación lineal entera*

El algoritmo original propuesto por Gomory utiliza como planos de corte desigualdades derivadas de la solución óptima obtenida por *Simplex* para el problema lineal. Estos planos de corte reciben el nombre de *cortes de Gomory*. Aunque fue demostrado que este algoritmo, bajo ciertas condiciones, termina en un número finito de pasos, en la práctica su convergencia parece ser muy lenta. Por otro lado, la implementación computacional es numéricamente inestable, aunque en la actualidad han sido fortalecidos lográndose buenas implementaciones.

Posteriormente, se han desarrollado algoritmos que utilizan una variedad de cortes aplicables a cualquier *PEM*, como por ejemplo los cortes *disyuntivos*, *clique*, *cover*, etc. Si bien estos algoritmos tienen propiedades teóricas de mucho interés, su éxito en la práctica es discutible. Cualquiera de las técnicas mencionadas tienen la ventaja de poder ser utilizadas para cualquier problema de Programación Lineal Entera, independientemente de su estructura. Si bien esto es una propiedad deseable en un algoritmo, no siempre brinda la herramienta más adecuada para casos particulares. Un estudio más específico del problema ayuda a obtener mejores procedimientos. Este es el sentido del próximo enfoque.

- *Explotando la estructura particular del problema.*

En los 70's, resurgió el interés por los algoritmos de planos de corte debido al desarrollo de la teoría poliedral. Mediante el estudio de combinatoria poliedral, la intención es reemplazar el conjunto de restricciones de un programa de programación entera mixta por la descripción de la cápsula convexa del conjunto de soluciones factibles. Las desigualdades lineales necesarias para describir a $\text{conv}(S)$ se denominan *facet*s. Si se conoce de forma completa esta descripción, el problema entero puede ser resuelto como un problema de programación lineal. De esta manera, explotando la estructura particular de cada problema, los planos de corte resultarán más efectivos a la hora de cortar soluciones.

Desafortunadamente, no es fácil tener esta descripción y los problemas pertenecientes a la clase NP-Difícil tienen una cantidad exponencial de facets, a menos que $P = NP$. Aún en ese caso, la cantidad de facets podría ser exponencial, pero el problema de separación de las mismas sería polinomial. Alternativamente, es posible utilizar cualquier desigualdad válida para el conjunto de soluciones factibles como planos de corte, pero, en general, la eficiencia del algoritmo depende de la *fortaleza* de estas desigualdades, siendo las facets los *mejores* cortes posibles.

Con fines algorítmicos, el estudio poliedral debe estar acompañado de algoritmos de separación eficientes. En este sentido, hay un resultado muy importante debido a Grötschel, Lovász y Schrijver [8] que relaciona la complejidad del problema de separación con la complejidad del problema de optimización. Se establece que el problema de optimización $\max\{cx : x \in \text{conv}(S)\}$ puede resolverse polinomialmente si y solo si el problema de separación ($x \in \text{conv}(S)$ ó encontrar una desigualdad válida violada) es polinomial. Es decir que si el problema que se está tratando no es polinomial, existe al menos alguna familia de facets que no puede separarse en tiempo polinomial. Esto de alguna manera implica el grado de dificultad de encontrar

la descripción de todas las facetas de la cápsula convexa y del desarrollo de algoritmos de separación.

En forma general, para desarrollar un algoritmo de planos de corte, primero se busca una descripción parcial de la cápsula convexa del conjunto de las soluciones factibles enteras o desigualdades válidas *fuertes* para este conjunto. Luego es necesario el diseño de rutinas de separación para las familias de desigualdades encontradas. Estas rutinas toman como entrada una solución y retornan restricciones de estas familias violadas por este punto, si es que existe alguna. El problema de separación, en algunos casos, puede ser NP-difícil o tener complejidad alta, lo que lleva en la práctica a utilizar algoritmos heurísticos, o sea, que es posible que no sean capaces de encontrar una desigualdad violada aunque exista. La estrategia que se utilice para decidir la búsqueda en la diferentes familias es clave para la performance del algoritmo.

El esquema básico de un algoritmo de planos de corte es el siguiente. Llámese PEM al problema entero mixto que se quiere resolver, $PL(P)$ a la relajación lineal del problema P y x^P la solución óptima de esta relajación.

1. **Inicialización:**

$P = PEM$

2. **Evaluación:**

Resolver $PL(P)$

- a) Si es no factible, entonces PEM es no factible y el algoritmo termina.
 - b) Si x^P cumple las condiciones de integralidad, x^P es la solución óptima de PEM y el algoritmo termina.
 - c) **Separación:** Caso contrario, resolver el problema de separación para x^P .
 - Si se encuentran cortes, agregarlos a P e ir a **Evaluación**.
 - Caso contrario, retornar el funcional evaluado en x^P como una cota inferior de PEM .
-

El algoritmo de planos de corte puede no resolver el problema de forma óptima, ya sea por no encontrar desigualdades válidas violadas o porque el tiempo consumido excede el tiempo disponible. Sin embargo, puede ser utilizado para generar buenas cotas inferiores del valor óptimo del problema. Además, muchas veces a partir de la solución óptima de la relajación actual es posible encontrar buenas soluciones enteras mediante una heurística, brindando una cota superior del valor óptimo.

2.3.3. Algoritmos branch-and-cut

En muchas instancias, los dos algoritmos descritos arriba fallan en la resolución del problema. A comienzos de los 80's se comenzó a aplicar una metodología mixta que conjuga las dos ideas dando origen a los llamados algoritmos *Branch-and-Cut*. De esta manera

se lograron resolver exitosamente instancias de tamaño considerable de una gran cantidad de problemas de programación lineal entera, como por ejemplo el *Problema de Viajante de Comercio* [10] o el *Problema de Corte Máximo* [7].

Uno de los factores que influye en el fracaso de los algoritmos *Branch-and-Bound* es la baja calidad de las cotas obtenidas mediante las relajaciones lineales. Esto significa que resulta crucial poder ajustar las formulaciones, por ejemplo con planos de corte.

Un algoritmo *Branch-and-Cut* es un *Branch-and-Bound* en el cual se generan planos de corte a través del árbol de enumeración. El objetivo de esto es reducir significativamente el número de nodos del árbol mejorando la formulación de los subproblemas. En un *Branch-and-Cut*, la enumeración y los planos de corte se benefician mutuamente. Generalmente, la cota producida en cada nodo del árbol de enumeración es mejor que en un *Branch-and-Bound*, debido a las nuevas desigualdades agregadas a la formulación del correspondiente subproblema. Por otro lado, el proceso de *branching* perturba la solución fraccionaria ayudando a los algoritmos de separación.

Estos algoritmos no solo son capaces de producir la solución óptima, también pueden brindar soluciones aproximadas al óptimo con certificado de calidad en tiempos de cómputo moderado. Es decir, aún sin conocer el valor real del óptimo, se puede acotar que tan lejos se está de este.

En la implementación de un algoritmo *Branch-and-Cut* hay que tener en cuenta las estrategias propias de un algoritmo *Branch-and-Bound* sumadas a las de un algoritmo de planos de corte. Además, se agregan nuevas decisiones como ¿cuándo buscar planos de cortes?, ¿cuántos cortes agregar?, etc.

El esquema de un algoritmo *Branch-and-Cut* es el siguiente.

1. **Inicialización:**

$$\mathcal{N} = \{\text{PEM}\} \quad Z^* = \infty$$

2. **Elección de próximo nodo:**

Si $\mathcal{N} = \{\}$ Z^* es óptimo y el algoritmo termina

Si no, elegir y borrar un nodo k de \mathcal{N}

3. **Evaluación:**

Resolver $PL(k)$.

a) Si es no factible, ir a **Elección**.

b) Si $Z^k > Z^*$, ir a **Elección**.

c) Si la solución óptima cumple las condiciones de integralidad, poner $Z^* = \min\{Z^*, Z^k\}$ e ir a **Elección**.

4. **División vs Separación:**

Decidir si se buscarán planos de corte:

- **SI:** Ir a **Separación**
- **NO:** Ir a **División**

5. **División:** Particionar la región factible de $PL(k)$ en dos o más regiones, agregando un nuevo nodo a \mathcal{N} por cada nueva región. Ir a **Elección**.
 6. **Separación:** Resolver el problema de separación para la solución fraccionaria de $PL(k)$.
 - Si son encontrados cortes, agregarlos a la formulación e ir a **Evaluación**.
 - Si no se encuentran, ir a **División**.
-

3. Red Evento-Actividad y Formulación PLE del problema

En el presente capítulo se incluye una descripción completa del problema de Reprogramación de cronograma y el modelo de PLE en el cual nos basamos para nuestro estudio. Dicho modelo fue construido a partir del trabajo de Veulenturf et al. [12]. Sobre su trabajo, hemos añadido una explicación detallada de las consideraciones necesarias para abordar distintos escenarios, en particular, sobre las adaptaciones realizadas en el modelo para contemplar correctamente el bloqueo de una sección de vías. Por otro lado, dado que nos interesa estudiar un modelo alternativo para el problema tal como veremos en las siguientes secciones, hemos omitido algunas extensiones que los autores introducen dado que no son estrictamente necesarias para el funcionamiento normal del servicio de trenes, y corresponden a mejoras que pueden ser agregadas fácilmente en un futuro.

Dada la complejidad del modelo, una parte importante de nuestro trabajo consistió en desgarnar los detalles del mismo para poder entenderlos en profundidad, permitiéndonos luego estudiar posibles mejoras. A continuación se presenta una explicación completa con un enfoque incremental, buscando simplificar el entendimiento. En la Sección 3.1 introducimos los distintos conceptos que intervienen en el problema. En la Sección 3.2 se explica el armado de la *red evento-actividad*, que consiste en un grafo que modela las restricciones operativas del problema y a partir del cual se genera posteriormente el modelo de PLE. La Sección 3.3 explica el modelo de PLE base. A continuación, en la Sección 3.4, se detalla el traspaso de recursos entre trenes. Para esto, deben tenerse en cuenta numerosas cuestiones para modelarlo correctamente, tanto del armado de la *red evento-actividad* como del problema de PLE. En la Sección 3.5 se explican las extensiones requeridas para incluir el bloqueo de vías dentro del modelo. Por último, en la Sección 3.6 se presenta el modelo de PLE completo.

3.1. Descripción general del problema

Tal como se introdujo brevemente en el Capítulo 1, nuestro problema surge en el contexto de un servicio de trenes y una red asociada en la cual, a partir de una contingencia, se produce un **bloqueo** de algún tramo de vías de la red. Dicho bloqueo vuelve infactible la **programación** original del servicio de trenes. El objetivo del problema es recalcular el cronograma en tiempo real, minimizando el impacto en la calidad del servicio para los pasajeros. Para cumplir con este objetivo, el modelo debe decidir qué trenes son **cancelados** (total y/o parcialmente) y cuáles son **retrasados**, conformando un nuevo programa que será utilizado hasta reestablecer el servicio normal.

Para lograr esto, debe tenerse en cuenta toda la información disponible, tanto del programa original como del estado actual de la red sobre la que operan los trenes y el bloqueo de vías.

3.1.1. Programación original y restricciones generales

El programa original contiene la información del recorrido de cada uno de los trenes. Llamamos T al conjunto de todos los trenes. Para cada tren $t \in T$, el programa original indica el **horario de salida** y el **horario de llegada** en cada una de las estaciones de su recorrido. Llamamos S al conjunto de todas las estaciones de la red. Es importante mencionar que estos horarios originales deben ser considerados en el nuevo cronograma, dado que un tren no puede ser reprogramado en un tiempo anterior a su horario previo.

En cuanto a los **tiempos mínimos** de circulación entre dos estaciones y de espera dentro de una estación, los mismos podrían variar respecto del programa original. Por ejemplo, en caso de una sección afectada por el bloqueo, los tiempos de viaje en el nuevo cronograma pueden ser mayores a los tiempos de viaje originales en dicha sección. Por otro lado, en otras secciones se podrían permitir tiempos de viaje o espera menores a los originales, siempre y cuando no se viole ninguna restricción de seguridad.

Otro dato importante a tener en cuenta es el **tiempo máximo de retraso** que un tren puede tener en cada salida o llegada de una estación determinada. Este parámetro es de suma importancia ya que en caso de permitir un retraso máximo bajo, se limitan las opciones a considerar, forzando a que se produzcan muchas cancelaciones. Por otro lado, si los horarios originales pueden ser retrasados por mucho tiempo, esto implicará que existan muchas opciones a considerar, lo cual podría tener un impacto muy grande en los tiempos de cómputo. En las siguientes secciones profundizaremos sobre las implicaciones de la elección de este valor, dado que es un dato importante a tener en cuenta también a nivel del modelado.

A la hora de recalcular el programa original, debe establecerse una relación entre los retrasos y las cancelaciones. Un tren cancelado debería tener un impacto negativo mucho mayor que una llegada o salida retrasada, sin embargo, podría ser preferible tener unas pocas cancelaciones que eviten una gran cantidad de retrasos. Por dicho motivo, se incluye en el modelo una **penalización por tren cancelado** y una **penalización por unidad de retraso** que modelan esta relación.

3.1.2. Recursos disponibles para el funcionamiento del servicio

Para modelar correctamente el funcionamiento del servicio de trenes, es necesario comprender cómo está conformada la red sobre la que operan, cuáles son los **recursos** disponibles y cómo los trenes utilizan dichos recursos para circular. Tal como comentamos en el Capítulo 1, en nuestro problema consideramos la red a nivel macroscópico.

A grandes rasgos, se puede pensar a la red como un conjunto de **estaciones** interconectadas por **tramos de vías** (también llamados secciones de vías). Cada estación tiene una **capacidad** determinada, que no es más que la cantidad de vías dentro de la misma. De la misma manera, la capacidad de un tramo de vías estará dada por la cantidad de vías entre las estaciones que conecta. Es importante diferenciar entre las **vías dentro de una estación** de las **vías de un tramo**, dado que son distintos recursos, cada uno con su manejo particular. Mientras las vías de una estación solo pueden ser ocupadas por un único tren en un tiempo determinado, las vías entre estaciones pueden ser ocupadas por varios trenes al mismo tiempo, siempre y cuando los mismos circulen en la misma dirección. El modelo asume que al ingresar a una estación, un tren puede optar por tomar cualquiera de las vías libres en la misma, independientemente de la vía del tramo por la

que ingresa a la estación. Análogamente, se asume que desde una estación determinada, un tren puede tomar cualquier vía del tramo hacia el que parte.

Otro recurso que un tren necesita para circular es el **material rodante**. Cada tren utiliza una formación (conjunto de unidades) de material rodante durante todo su recorrido. Al finalizar el mismo, puede suceder que el material rodante que utilizó sea traspasado a otro tren cuyo recorrido comienza en dicha estación, o que las unidades sean guardadas en un **patio de maniobras**, si es que la estación posee uno. Luego, además de poseer una capacidad de vías, las estaciones poseen una **capacidad de material rodante** al inicio de cada día. Cuando un tren comienza su recorrido en una estación, puede tomar el recurso de material rodante de ese inventario inicial. Vale mencionar que se asume que el patio de maniobras posee una capacidad infinita para guardar unidades de material rodante. Otra cuestión importante a mencionar es que para nuestro problema, las formaciones de material rodante no se pueden particionar y recombinarse. En esta etapa se incluyen en el modelo ciertos aspectos del manejo del material rodante para simplificar la obtención de una solución factible en la próxima etapa del problema general de Gestión de grandes interrupciones, es decir, en la Reprogramación de material rodante, tal como vimos en el Capítulo 1. En dicha etapa, se consideran en detalle distintos aspectos del manejo de las formaciones, tales como el particionamiento y la recombinación de unidades.

Es importante modelar correctamente el uso secuencial de los recursos mencionados, asegurando que se respeten los **tiempos de seguridad** obligatorios a la hora del traspaso. En particular, los tiempos de seguridad que deben respetarse son:

- **Vías de una estación:** luego de que un tren libera la vía de una estación al partir hacia otra, debe respetarse un tiempo mínimo hasta que otro tren la pueda ocupar.
- **Vías de un tramo:** si una vía de un tramo es liberada por un tren al ingresar a una estación, otro tren que quiera tomar la misma vía en dirección opuesta, deberá esperar el tiempo de seguridad obligatorio. Por otro lado, también debe respetarse un tiempo de separación para dos trenes operando en una misma vía en la misma dirección.
- **Material rodante:** cuando un tren termina su servicio, puede traspasar su material rodante hacia otro tren, dicho traspaso debe respetar un tiempo mínimo. Aún más, en caso de que el material rodante sea guardado en un patio de maniobras para luego ser utilizado por otro tren, otro tiempo de seguridad es requerido.

La elección de los tiempos de seguridad puede tener un impacto fuerte en la calidad de la solución obtenida. Los mismos deben respetar ciertos estándares que aseguren no poner en riesgo la seguridad de los pasajeros, pero debe tenerse en cuenta que al tener la capacidad de la red disminuida por un bloqueo, estos tiempos limitarán las opciones disponibles a la hora de conformar un nuevo cronograma.

3.1.3. Bloqueo de vías

El bloqueo de una sección de vías vuelve infactible la programación original, siendo este el motivo que da origen a nuestro problema. El bloqueo ocurre en un **tramo** determinado, bloqueando una **cantidad** de vías durante un período de cierta **duración**.

El tramo de vías afectado queda determinado por el par de estaciones que conecta. En nuestro trabajo asumiremos que los bloqueos afectan una única sección para facilitar la experimentación. Sin embargo, el modelo es fácilmente adaptable para el caso de múltiples bloqueos en distintos tramos. En cuanto a los bloqueos de vías de estaciones, si bien no son considerados, pueden ser tratados de una manera similar a los bloqueos de vías de tramos.

En cuanto a la cantidad de vías bloqueadas, el bloqueo puede afectar solo algunas vías (bloqueo parcial) o el total de la capacidad del tramo (bloqueo total). Es importante notar que en caso de un bloqueo total, durante el transcurso del mismo ningún tren podrá circular por la sección afectada.

El último dato clave es la duración del bloqueo. Para esto, el modelo asume que no solo el **tiempo de inicio del bloqueo** es conocido sino también el **tiempo de fin del bloqueo**. Vale mencionar que en la práctica muchas veces se desconoce la duración del bloqueo, en estos casos, se trabaja preventivamente con una estimación del tiempo de fin que se puede ir actualizando a medida que se obtengan nuevos datos. Adicionalmente, se debe determinar un instante de tiempo posterior al bloqueo en el cual los trenes vuelven a funcionar de acuerdo a la programación original. Lo llamamos **tiempo límite de transición** dado que modela el instante en que los trenes dejan de seguir el nuevo cronograma y vuelven a estar programados bajo el cronograma original. Luego, el modelo debe asegurar que a partir de ese momento todos los trenes puedan circular sin ser retrasados ni cancelados. De esta manera, la elección de dicho instante de tiempo no es menor, dado que afectará las posibilidades a considerar para el armado de la nueva programación. La Figura 3.1 muestra los instantes mencionados en una línea de tiempo.

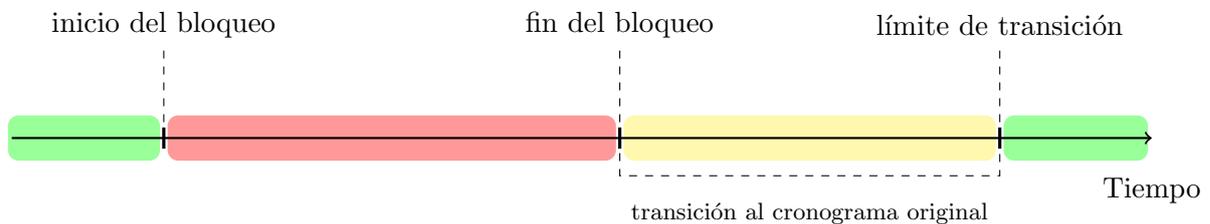


Fig. 3.1: Bloqueo y sus instantes de tiempo

Los trenes que estén programados para circular por la sección del bloqueo durante el transcurso del mismo deben tratarse con especial atención. Tanto el manejo de estos casos como el de varios otros detalles que deben tomarse para contemplar correctamente el bloqueo pueden verse en la Sección 3.5.

3.2. Red Evento-Actividad

Como se mencionó en la sección anterior, el objetivo de nuestro problema es recalculuar la programación original, la cual se vuelve infactible por el bloqueo de vías. Para esto, se debe decidir qué trenes son cancelados (total y/o parcialmente) y cuáles son retrasados. Cada tren tiene originalmente programadas una llegada y salida de cada estación de su recorrido en un horario determinado. Representamos a cada una de estas llegadas y salidas

como **eventos** vinculados con cada tren. Luego, el modelo debe decidir un nuevo horario para cada uno de los eventos de cada tren o si debe cancelar los mismos.

Los sucesivos eventos de un mismo tren están relacionados en base al orden temporal que tienen en el cronograma original. Asimismo, eventos de distintos trenes también se relacionan para traspasarse recursos. Decimos que dos eventos se unen por medio de una **actividad** cuando hay una relación entre los mismos.

Para modelar nuestro problema de reprogramación de cronograma, nos basamos en una *red evento-actividad*, conformada a partir de los eventos y actividades mencionados. Formalmente, representamos a la red como un grafo dirigido $N = (E, A)$, donde E es el conjunto de vértices (eventos) y A el conjunto de ejes (actividades).

El modelo de PLE se define en base a la red evento-actividad mencionada. En las siguientes secciones desarrollamos paralelamente tanto el detalle del armado de la red evento-actividad como del modelo PLE.

3.2.1. Eventos

El conjunto de eventos $E = E_{tren} \cup E_{inv}$ está conformado por un conjunto E_{tren} de **eventos de tren** y un conjunto E_{inv} de **eventos de inventario**.

El conjunto E_{tren} no es más que el conjunto de llegadas y salidas. Es decir, cada evento $e \in E_{tren}$ corresponde a un arribo o salida de una determinada estación. Denominamos t_e al tren asociado al evento e . El horario original en el cual el evento e está programado es notado como q_e . Como mencionamos en la sección anterior, el horario de cada llegada o salida tiene un límite máximo de retraso en el cual puede ser reprogramado, llamamos d_e a dicho límite. Finalmente, llamamos x_e al nuevo horario en el cual el evento e es programado. Luego, nuestro modelo debe determinar los valores de x_e para cada $e \in E_{tren}$ tales que cumplan $q_e \leq x_e \leq q_e + d_e$.

Vale destacar que se asume que el horizonte de tiempo se encuentra discretizado, donde por ejemplo cada unidad puede representar minutos o decena de minutos. En nuestro caso, tanto para los ejemplos utilizados durante nuestro estudio teórico, como para los experimentos realizados, utilizamos como unidad de tiempo el minuto.

Como se mencionó anteriormente, para que los trenes puedan circular necesitan utilizar un determinado conjunto de recursos cuya capacidad disponible depende de la red sobre la que se opera. Los eventos $e \in E_{inv}$ representan este inventario de recursos disponibles. Para cada tipo de recurso, se define un conjunto de eventos que definen estas capacidades. Luego, los eventos de inventario del recurso de tramo de vías, determinan la cantidad de vías disponibles en cada tramo. Similarmente, para las vías dentro de estaciones, los eventos determinan la cantidad de vías en cada estación. Por último, para el material rodante, se definen eventos de inventario que definen la cantidad de unidades de material rodante disponibles al inicio del día en una determinada estación. Llamamos i_e a la cantidad de recursos que deja disponible el evento e . Por ejemplo, si el evento e representa las vías disponibles en la estación s , i_e será la cantidad de vías de s .

A diferencia de los eventos en E_{tren} , los eventos de inventario no tienen ni horario original ni reprogramado (i.e. q_e, x_e). Su uso queda más claro al comprender como la red evento-actividad modela el traspaso de recursos, con lo cual el detalle de cómo se definen estos eventos se puede ver en la Sección 3.4, donde se explica el traspaso de cada recurso.

3.2.2. Actividades

Anteriormente mencionamos que los eventos se relacionan entre sí mediante dos tipos de actividades. Por un lado, los eventos de un mismo tren están relacionados de manera ordenada en base a su cronograma original, llamamos a estas relaciones **actividades de tren** y las notamos A_{tren} . Por otro lado, eventos de distintos trenes se relacionan en caso de que puedan traspasarse recursos entre sí, llamamos a estas relaciones **actividades de recursos** y las notamos A .

Para modelar el traspaso de recursos, decimos que una actividad $a = (e, f) \in A$, definida desde el evento e hacia el evento f ($e, f \in E$) denota el hecho de que el evento e le traspasa un recurso a f , o lo que es lo mismo, f utiliza un recurso que le deja e . Es importante notar que esta relación fuerza un orden temporal (i.e. e ocurre antes de f si le traspasa el recurso). Otra cuestión a destacar es el hecho de que los mismos eventos e y f pueden estar unidos por más de una actividad dado que hay más de un tipo de recurso, pero a lo sumo puede haber una única actividad del mismo recurso entre dos eventos. Luego, entre dos eventos e y f , puede haber a lo sumo 3 actividades: una para el traspaso del recurso de vías de tramos, otra para las vías de estación y otra para el material rodante.

Como se explicó al introducir el problema, al momento de traspasarse recursos entre trenes, deben respetarse ciertos tiempos de seguridad. A partir de esto, cada actividad $a = (e, f)$ tiene asociado un **tiempo mínimo** L_a el cual debe respetarse para que el recurso que utilizó el evento e esté disponible para ser usado por f .

Para un tipo de recurso dado, un evento e tiene actividades definidas hacia distintos eventos, pero solo puede traspasar su recurso a uno de ellos. Similarmente, un evento f tiene posibilidades de tomar su recurso necesario de distintos eventos, pero solo puede tomarlo de un evento particular. En base a esto, las actividades determinan los posibles órdenes en que cada recurso se puede ir traspasando secuencialmente entre los eventos. Nuestro modelo deberá decidir cómo elegir las actividades buscando la mejor reprogramación posible.

Al igual que como comentamos sobre los eventos de inventario, el entendimiento de cómo se definen las actividades para cada tipo de recurso es complejo. El detalle del mismo se incluye en la sección 3.4 donde se explica el armado del conjunto de actividades para cada tipo de recurso y su relación con el modelo de PLE.

Mencionamos que los eventos de un mismo tren se vinculan mediante otro tipo de actividades. Las actividades de tren son más sencillas en el sentido que el modelo no debe decidir si los eventos que vincula se relacionan o no, ya que eso ya está determinado de antemano por el cronograma original. Además, entre dos eventos de un mismo tren solo puede haber una única actividad de tren, la cual existe si los eventos son contiguos en la programación original. Sin embargo, lo que si deben respetar los eventos de un mismo tren en la nueva programación son los tiempos mínimos de viaje entre dos estaciones y de espera dentro de una estación. Luego, las actividades $a = (e, f) \in A_{\text{tren}}$ tienen asociado un L_a que representa esta duración mínima.

En la Figura 3.2 se presenta una red de ejemplo que utilizaremos como base en las siguientes secciones para explicar cómo se construyen los conjuntos de eventos y actividades y la interpretación del modelo de PLE. Esta red simplificada cuenta con 3 estaciones

conectadas por tramos de 2 vías. Las estaciones en las cabeceras cuentan con un patio de maniobras donde se guardan las unidades de material rodante.

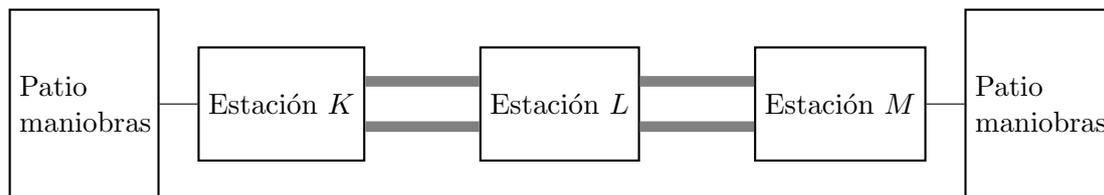


Fig. 3.2: Red de ejemplo

Junto con la red 3.2, en la Tabla 3.1 se incluye un programa de ejemplo del servicio de trenes. El servicio cuenta con cuatro trenes, dos en cada dirección, con su horario originalmente programado para cada arribo y salida. Vale aclarar que tanto la red como el cronograma de ejemplo son tomados del trabajo de Veelenturf et al. [12].

| Servicio | Salida K | Arribo L | Salida L | Arribo M |
|----------|------------|------------|------------|------------|
| Tren 1 | 7:00 | 7:20 | 7:23 | 7:30 |
| Tren 2 | 7:55 | 8:15 | 8:18 | 8:25 |
| Servicio | Salida M | Arribo L | Salida L | Arribo K |
| Tren 3 | 7:16 | 7:23 | 7:26 | 7:46 |
| Tren 4 | 7:45 | 7:52 | 7:55 | 8:19 |

Tab. 3.1: Cronograma de ejemplo

A partir del cronograma de ejemplo introducido, la Figura 3.3 muestra el conjunto de eventos de tren y actividades de tren que se construyen para dicho programa. Junto a cada actividad se incluye su respectivo tiempo mínimo de viaje o espera, i.e. L_a . Notar que en este ejemplo, los valores de L_a corresponden a los mismos tiempos de viaje entre estaciones y de espera en cada estación que en el cronograma original.

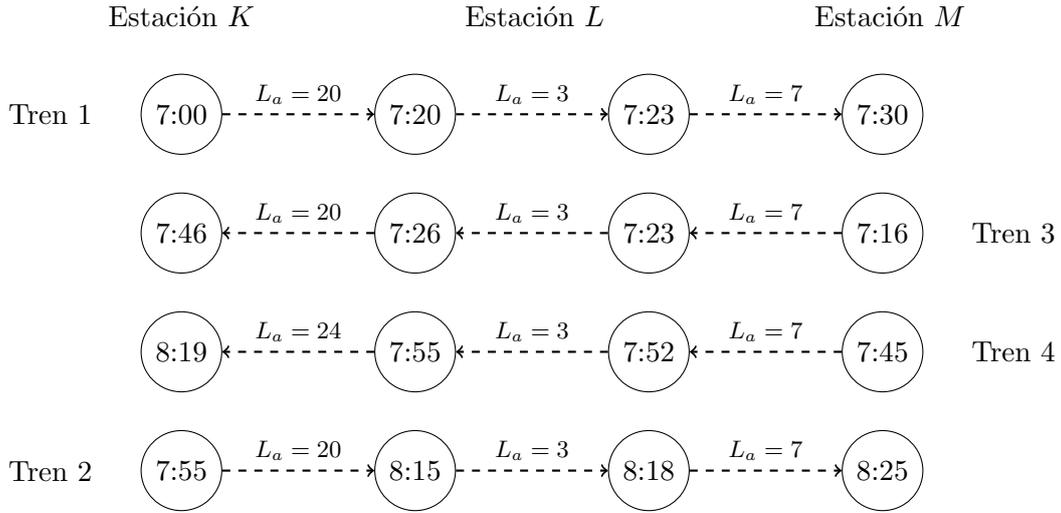


Fig. 3.3: Eventos y actividades de tren para el cronograma 3.1. Los eventos se organizan de manera horizontal según la estación y de manera vertical según el comienzo de cada servicio de tren. Los valores en cada evento son sus horarios originales (i.e. q_e).

3.3. Formulación PLE Base

A continuación se presenta la primera parte del modelo de PLE. La misma describe la función objetivo a minimizar y las restricciones básicas que deben cumplir los nuevos tiempos reprogramados para los eventos de cada tren. En la próxima sección se agregan las restricciones necesarias para modelar el traspaso de los recursos. Junto con el modelo, se agrega la Tabla 3.2 que resume la notación de los conceptos introducidos en las secciones previas. A la notación introducida en las secciones previas, la Tabla 3.2 agrega y_t como la variable binaria que determina si el tren $t \in T$ es cancelado, y λ_t y μ_e como las penalizaciones por cancelación y retraso respectivamente.

La función (3.1) modela nuestro objetivo de encontrar una reprogramación buscando minimizar el impacto en la calidad de servicio. Para esto, por un lado se suma una penalización por cada tren cancelado y por otro lado se suma una penalización por cada unidad de retraso de un evento en la nueva programación. Luego, si un tren es reprogramado en el mismo horario que lo estaba originalmente, su impacto será nulo. El modelo debe buscar la mínima cantidad de cancelaciones y retrasos dentro de las posibilidades que brinda la red al estar limitada por el bloqueo.

$$\text{Minimizar} \quad \sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{tren}} \mu_e (x_e - q_e) \quad (3.1)$$

$$\text{sujeto a} \quad x_e - q_e \geq 0 \quad \forall e \in E_{tren} \quad (3.2)$$

$$x_e - q_e \leq (1 - y_{t_e}) d_e \quad \forall e \in E_{tren} \quad (3.3)$$

$$x_f - x_e \geq L_a \quad \forall a = (e, f) \in A_{tren} \quad (3.4)$$

$$y_t \in \{0, 1\}, x_e \in \mathbb{N} \quad \forall t \in T, \forall e \in E_{tren} \quad (3.5)$$

| | |
|-------------|--|
| T | conjunto de trenes |
| y_t | variable de decisión de cancelación del tren $t \in T$ (1 = cancelado) |
| λ_t | penalización por cancelación del tren $t \in T$ |
| E_{tren} | eventos de tren |
| t_e | tren correspondiente al evento $e \in E_{tren}$ |
| q_e | horario original del evento $e \in E_{tren}$ |
| x_e | variable que determina el nuevo horario del evento $e \in E_{tren}$ |
| d_e | retraso máximo permitido para el evento $e \in E_{tren}$ |
| μ_e | penalización por unidad de retraso del evento $e \in E_{tren}$ |
| A_{tren} | actividades de tren |
| L_a | duración mínima de la actividad $a \in A_{tren}$ |

Tab. 3.2: Notación formulación PLE base

Las restricciones (3.2) aseguran que los eventos no puedan ser reprogramados antes de su horario original. Asimismo, las restricciones (3.3) aseguran que los eventos respeten el límite máximo de retraso en caso de que sus respectivos trenes no sean cancelados. Vale notar que en caso de que un tren sea cancelado, el modelo fuerza a que el nuevo horario de sus eventos sea igual al horario original. Esto último es importante para que no se pese la penalización por retraso de sus eventos sino solamente por la penalización por la cancelación del tren. Si bien una solución óptima forzaría esto mismo sin necesidad de incluir la restricción explícitamente, al hacerlo estamos limitando el espacio de posibles soluciones.

Las desigualdades (3.4) fuerzan a que se respeten los tiempos mínimos de viaje entre estaciones y de espera en una estación. En caso de que se utilicen como dichos valores las duraciones del cronograma original, si una actividad $a = (e, f)$ conecta los eventos e y f y el evento e es retrasado en el nuevo programa (i.e. $x_e > q_e$), el evento f también deberá retrasarse para que el tren pueda cumplir con la nueva programación.

Finalmente, las desigualdades (3.5) modelan las variables utilizadas como binarias y enteras según corresponda.

3.4. Restricciones de recursos

Para modelar correctamente el problema, es necesario incorporar las restricciones que describen el traspaso de recursos entre los eventos. Para esto, primero dividimos las actividades de recursos asociadas con cada evento $e \in E$ en dos: **actividades de entrada** y **actividades de salida**.

Como describe su nombre, las actividades de entrada de un evento e , notadas $A^-(e)$, corresponden a todas las actividades definidas desde otros eventos hacia el evento e . Es decir, $A^-(e) = \{a = (f, e) \in A \mid f \in E\}$. Una actividad de entrada $a = (f, e)$ de e significa que el evento e puede tomar el recurso utilizado por f al menos L_a minutos luego de que f ocurra.

Análogamente, las actividades de salida de un evento e , notadas $A^+(e)$, corresponden a todas las actividades definidas desde el evento e hacia otros eventos. Es decir, $A^+(e) = \{a = (e, f) \in A \mid f \in E\}$. Una actividad de salida $a = (e, f)$ de e significa que el evento e puede traspasar el recurso utilizado a f para que lo utilice al menos L_a minutos luego de que e ocurra.

Las actividades de entrada y salida de un evento representan las distintas posibilidades que tiene dicho evento para tomar y traspasar los recursos que se necesitan para circular. El modelo debe decidir cómo los eventos se traspasan recursos entre sí buscando que cada evento obtenga sus recursos necesarios a tiempo y evitar de esa manera demoras o cancelaciones no deseadas. Para esto, en el modelo de PLE, cada actividad $a = (e, f)$ tiene asociada una variable de decisión z_a que determina si la actividad a es utilizada (i.e. e traspasa el recurso a f) o no.

Para cada evento e , se define un subconjunto de sus actividades por cada tipo de recurso. Luego, se definen *i*) los subconjuntos $A_{\text{tr}}^-(e) \subset A^-(e)$ y $A_{\text{tr}}^+(e) \subset A^+(e)$ como las actividades de entrada y salida de e correspondientes al recurso de vías de tramos, *ii*) los subconjuntos $A_{\text{est}}^-(e) \subset A^-(e)$ y $A_{\text{est}}^+(e) \subset A^+(e)$ como las actividades de entrada y salida de e correspondientes al recurso de vías de estaciones, y *iii*) los subconjuntos $A_{\text{mr}}^-(e) \subset A^-(e)$ y $A_{\text{mr}}^+(e) \subset A^+(e)$ como las actividades de entrada y salida de e correspondientes al recurso de material rodante. Los conjuntos de entrada y salida quedan redefinidos entonces como la unión de estos subconjuntos: $A^-(e) = \{A_{\text{tr}}^-(e), A_{\text{est}}^-(e), A_{\text{mr}}^-(e)\}$, $A^+(e) = \{A_{\text{tr}}^+(e), A_{\text{est}}^+(e), A_{\text{mr}}^+(e)\}$. A su vez, definimos los subconjuntos $A_{\text{tr}} \subseteq A$, $A_{\text{est}} \subseteq A$, $A_{\text{mr}} \subseteq A$ correspondientes a cada tipo de recurso. En las siguientes subsecciones se describe cómo se define cada uno de ellos para una red y un cronograma dados.

A continuación, se presentan las restricciones que se agregan al modelo de PLE para modelar el traspaso de recursos. Se incluye también la Tabla 3.3 que resume la nueva notación utilizada.

$$\sum_{a \in C} z_a + y_{t_e} = 1 \quad \forall e \in E_{tren}, C \in A^-(e), C \neq \emptyset \quad (3.6)$$

$$\sum_{a \in C} z_a + y_{t_e} \leq 1 \quad \forall e \in E_{tren}, C \in A^+(e) \quad (3.7)$$

$$\sum_{a \in C} z_a \leq i_e \quad \forall e \in E_{inv}, C \in A^+(e) \quad (3.8)$$

$$x_f - x_e + M_a(1 - z_a) \geq L_a \quad \forall a = (e, f) \in A, e, f \in E_{tren} \quad (3.9)$$

$$z_a \in \{0, 1\} \quad \forall a \in A \quad (3.10)$$

| | |
|-----------|---|
| E_{inv} | eventos de inventario |
| i_e | cantidad de recursos disponibles por el evento de inventario $e \in E_{inv}$ |
| A | actividades de recursos |
| $A^-(e)$ | actividades de entrada del evento $e \in E$ |
| $A^+(e)$ | actividades de salida del evento $e \in E$ |
| C | subconjunto de actividades de entrada o salida de un recurso determinado |
| z_a | variable de decisión de selección de la actividad $a \in A$ ($1 =$ seleccionada) |
| L_a | tiempo de seguridad de la actividad $a \in A$ |

Tab. 3.3: Notación restricciones de recursos

La interpretación de las restricciones se puede analizar en detalle al conocer las definiciones de los conjuntos de entrada y salida para cada tipo de recurso. Sin embargo, más allá de su interpretación particular, las mismas modelan el traspaso de los recursos de manera genérica.

Las igualdades (3.6) modelan cómo los eventos toman sus recursos necesarios para circular. Un evento debe tomar una unidad de cada recurso a través de una actividad o, en caso de no poder hacerlo, su tren debe ser cancelado. Es importante notar que el conjunto de igualdades fuerza que los eventos tomen un único recurso de cada tipo y no más. Otro detalle a tener en cuenta es que, en caso de que un evento no tenga actividades de entrada para un tipo de recurso (porque el evento no necesita tomar recursos de ese tipo), la igualdad no se define, dado que forzaría a cancelar el tren asociado al evento. Esto último queda más claro al estudiar cómo se definen los conjuntos de entrada para cada recurso, como veremos en las siguientes secciones.

Las desigualdades (3.7) modelan cómo los eventos traspasan recursos. Para cada tipo de recurso, un evento puede traspasar el recurso utilizado a lo sumo a otro evento. Al tratarse de una desigualdad en este caso, podría pasar que un evento no traspase sus recursos utilizados, modelando así el caso del último evento que utiliza un recurso determinado. También vale notar como en caso de que el tren del evento sea cancelado, no puede traspasar ningún recurso, esto tiene sentido dado que al ser cancelado, el evento tampoco pudo haber tomado ningún recurso por las igualdades (3.6).

Las desigualdades (3.8) modelan el primer uso de cada unidad de recurso a través de los eventos de inventario. Como se explicó en secciones anteriores, los eventos de inventario modelan la capacidad de cada uno de los recursos disponibles en la red. Las actividades de salida desde estos eventos determinan cuales son los posibles eventos de tren que pueden tomar estos recursos por primera vez para luego ir traspasándolos (como marcan las restricciones (3.6) y (3.7)) hacia otros eventos. Un evento de inventario $e \in E_{inv}$ posee una capacidad i_e de recursos disponibles, las desigualdades (3.8) fuerzan a que los eventos de inventario no den más recursos de los que tienen disponibles. Es importante notar que los eventos de inventario no tienen actividades de entrada (i.e. $A^-(e) = \emptyset, \forall e \in E_{inv}$), su función es la de brindar los recursos disponibles de la red a los primeros eventos que los necesiten. Más aún, como los eventos de inventario corresponden a un tipo de recurso determinado, solo poseen actividades de salida para ese tipo de recurso, para los otros tipos de recurso restantes sus conjuntos de salida son vacíos.

Las desigualdades (3.9) aseguran que se cumplan los tiempos de seguridad de los traspasos de recursos. Notar también que se está forzando un orden temporal entre los eventos, es decir, si e le traspasa el recurso a f , entonces e es programado antes que f en el nuevo cronograma. Las desigualdades son similares a las (3.4) que modelan los tiempos mínimos de viaje y de espera, la diferencia es que deben tener en cuenta el hecho de que la actividad $a = (e, f)$ puede no ser utilizada (el evento f puede tomar el recurso de otro evento distinto de e). Para estos casos, al tener $z_a = 0$, un valor de M suficientemente grande hace que la desigualdad no imponga condiciones. Dicho valor depende de la actividad a , y por ese motivo notamos a la cota M_a . Se puede ver fácilmente que tomando $M_a = L_a + (q_e + d_e) - q_f$ es suficiente para que la desigualdad sea válida.

A continuación presentamos las definiciones de los conjuntos $A^-(e)$ y $A^+(e)$ para cada tipo de recurso, junto con su respectiva interpretación de las desigualdades (3.6) - (3.9). Adicionalmente se agregan al modelo, en caso de ser necesario, restricciones particulares a cada tipo de recurso.

3.4.1. Vías de tramos

Un tramo entre dos estaciones tiene una capacidad determinada de vías que pueden ser usadas en ambas direcciones. Múltiples trenes pueden circular en una vía al mismo tiempo si circulan en la misma dirección, separados por un tiempo de seguridad.

Los eventos y actividades del recurso de vías de tramos están relacionados con un tramo (k, l) particular, es decir, una sección de vías entre las estaciones k y l . Definimos $E_{sal}^k \subseteq E_{tren}$ como el subconjunto de los eventos de tren que corresponden a salidas desde la estación k . Más aún, como subconjunto de estos eventos definimos $E_{sal}^{kl} \subseteq E_{sal}^k$ como el conjunto de salidas desde la estación k correspondientes a trenes cuyo próximo evento es un arribo en la estación l . De manera análoga definimos $E_{arr}^l \subseteq E_{tren}$ como los arribos a la estación l y $E_{arr}^{kl} \subseteq E_{arr}^l$ como los arribos a la estación l con salida previa desde la estación k . Finalmente, $e_{tr}^{kl} \in E_{inv}$ representa el evento de inventario del tramo de vías entre las estaciones k y l . Luego, $i_{e_{tr}^{kl}}$ representa la cantidad de vías disponibles en el tramo (k, l) .

A continuación definimos para cada evento e los subconjuntos $A_{tr}^-(e)$ y $A_{tr}^+(e)$ relacionados con la estación k . Para la estación l se construyen de manera análoga. Estas actividades representan el uso secuencial de las vías del tramo (k, l) .

- Para una salida $e \in E_{sal}^{kl}$ desde la estación k hacia el tramo (k, l) :
 - El conjunto $A_{tr}^-(e)$ está formado por actividades $a = (f, e)$ desde i otras salidas en la misma dirección: $f \in E_{sal}^{kl} \setminus \{e\}$, ii) arribos en la dirección contraria: $f \in E_{arr}^{lk}$, iii) el evento de inventario e_{tr}^{kl} .
 - El conjunto $A_{tr}^+(e)$ está formado por actividades $a = (e, f)$ hacia i otras salidas en la misma dirección: $f \in E_{sal}^{kl} \setminus \{e\}$.
- Para un arribo $e \in E_{arr}^{kl}$ a la estación l desde el tramo (k, l) :
 - El conjunto $A_{tr}^-(e)$ es vacío, dado que un arribo en una estación no necesita tomar un recurso de vías de un tramo para ocurrir (el tren ya está en la vía).
 - El conjunto $A_{tr}^+(e)$ está formado por actividades $a = (e, f)$ hacia i otros arribos en la misma dirección: $f \in E_{arr}^{kl} \setminus \{e\}$, ii) salidas en la dirección contraria: $f \in E_{sal}^{lk}$.
- Para el evento de inventario del tramo (k, l) , el conjunto $A_{tr}^+(e_{tr}^{kl})$ está formado por actividades $a = (e_{tr}^{kl}, e)$ hacia salidas de ese tramo en ambas direcciones, i.e. $e \in E_{sal}^{kl} \cup E_{sal}^{lk}$.

La Figura 3.4 muestra el grafo de eventos y actividades del recurso de vías de tramos para el tramo (K, L) de la red de ejemplo 3.2 y el cronograma asociado 3.1. Para facilitar el entendimiento, se incluyen también las actividades de tren en líneas punteadas. Además, si bien por la definición de los conjuntos el modelo permite que un evento traspase su recurso a un evento programado en un tiempo anterior, dichas actividades no son incluidas en la figura. Esto es así dado que estas actividades no podrían ser seleccionadas debido a las desigualdades (3.9), como mencionamos anteriormente. Para el evento de inventario, se incluye entre paréntesis su respectiva capacidad, en este caso, al tratarse del tramo (K, L) de la red ejemplo, la capacidad es de dos vías.

En base a las definiciones de los conjuntos $A_{tr}^-(\cdot)$ y $A_{tr}^+(\cdot)$, podemos ahora interpretar las desigualdades de recursos en particular para el recurso de vías de tramo.

Las igualdades (3.6) implican que un tren puede partir desde una estación hacia una vía v de un tramo siempre y cuando: i) salga a continuación de otro tren que tomó la vía v en la misma dirección, ii) salga después de que otro tren arribó por la vía v en dirección opuesta o iii) tome la vía v por primera vez (i.e. tome el recurso del evento de inventario).

Las desigualdades (3.7) implican que si un tren parte desde una estación hacia un tramo tomando una vía, lo que puede ocurrir luego es que otro tren puede salir tomando esa misma vía o que dicha vía no sea utilizada más. Asimismo, si se produce un arribo desde una vía v de un tramo hacia una estación, luego de dicho arribo puede ocurrir: i) otro arribo desde v en la misma dirección o ii) una salida hacia la vía v en la dirección opuesta o iii) el arribo es el último evento en utilizar la vía v del tramo.

Las desigualdades (3.8) implican que a lo sumo $i_{e_{tr}^{kl}}$ salidas hacia el tramo (k, l) tomen una vía del mismo por primera vez. Por ejemplo, para nuestra red de ejemplo el tramo (K, L) posee dos vías. Luego, a lo sumo dos salidas hacia dicho tramo podrán tomar una vía del mismo por primera vez.

Por último, las desigualdades (3.9) aseguran que se respeten los tiempos de seguridad entre dos salidas consecutivas hacia una misma vía de un tramo, entre dos arribos consecutivos desde una misma vía y entre un arribo y posterior salida hacia la misma vía en dirección contraria.

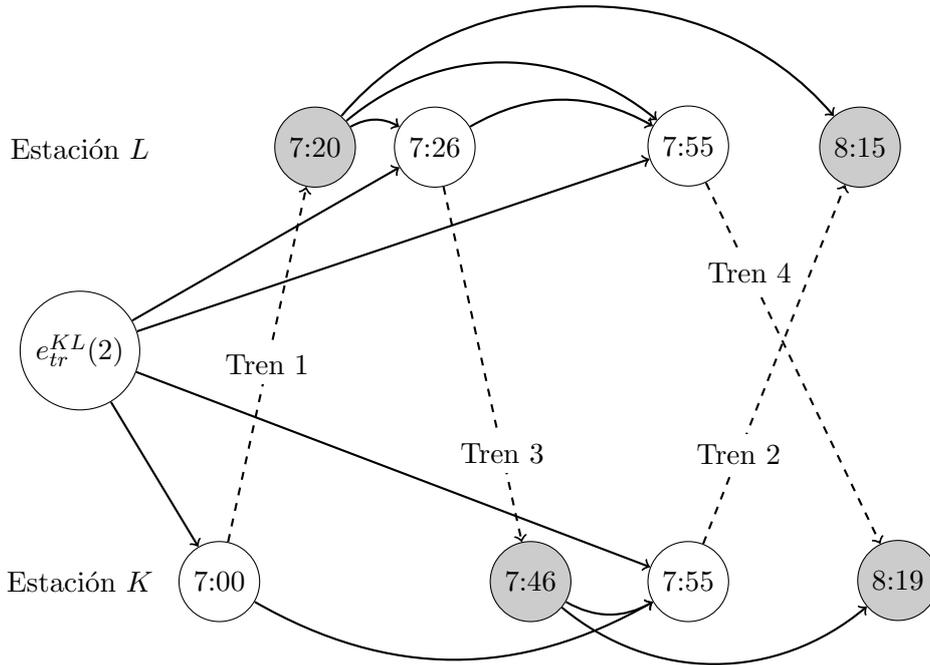


Fig. 3.4: Eventos y actividades relacionadas con el recurso de vías de tramo para el tramo (K, L) del cronograma ejemplo 3.1. Los eventos se organizan horizontalmente según su horario programado y verticalmente según su estación correspondiente. Los eventos correspondientes a salidas son blancos y los arribos grises. Las actividades del recurso del vías de tramo se dibujan con un eje normal mientras que las actividades de tren se dibujan con una línea punteada. El evento de inventario del tramo de vías se incluye a la izquierda y entre paréntesis figura su capacidad.

El conjunto de figuras 3.5 muestra las distintas soluciones posibles de traspaso de recursos para el conjunto de actividades del tramo (K, L) mostrado en la Figura 3.4. A la hora de seleccionar las actividades, el modelo debe elegir nuevos horarios para cada evento y decidir si se deben cancelar trenes o no. Dado que en esta sección solo nos interesa estudiar el traspaso del recurso de vías de un tramo, haremos una simplificación y asumiremos que los tiempos de cada evento son fijos (i.e. $x_e = q_e$).

Lo importante a notar de las distintas opciones que se muestran en la figura, es el concepto de que seguir un camino de actividades seleccionadas implica seguir el uso secuencial de un mismo recurso, en este caso una vía de un tramo, desde el evento de inventario hasta el último evento que lo utilice.

En particular para el recurso de vías de tramo, cuando una salida hacia un tramo toma una vía, obtiene un recurso que termina liberando posteriormente su evento contiguo, i.e., el arribo en la estación del otro lado del tramo. Como se detalló en la definición del conjunto de actividades, si e es un arribo, el conjunto $A_{tr}^-(e)$ es vacío. Si bien esto es así, existe un *traspaso implícito* del recurso entre su salida previa y el arribo en cuestión. Por este motivo, en la figura de selección de actividades marcamos también las actividades de tren entre las salidas y sus posteriores arribos para representar estos traspasos implícitos que permitan comprender mejor cómo las vías de un tramo se traspasan secuencialmente.

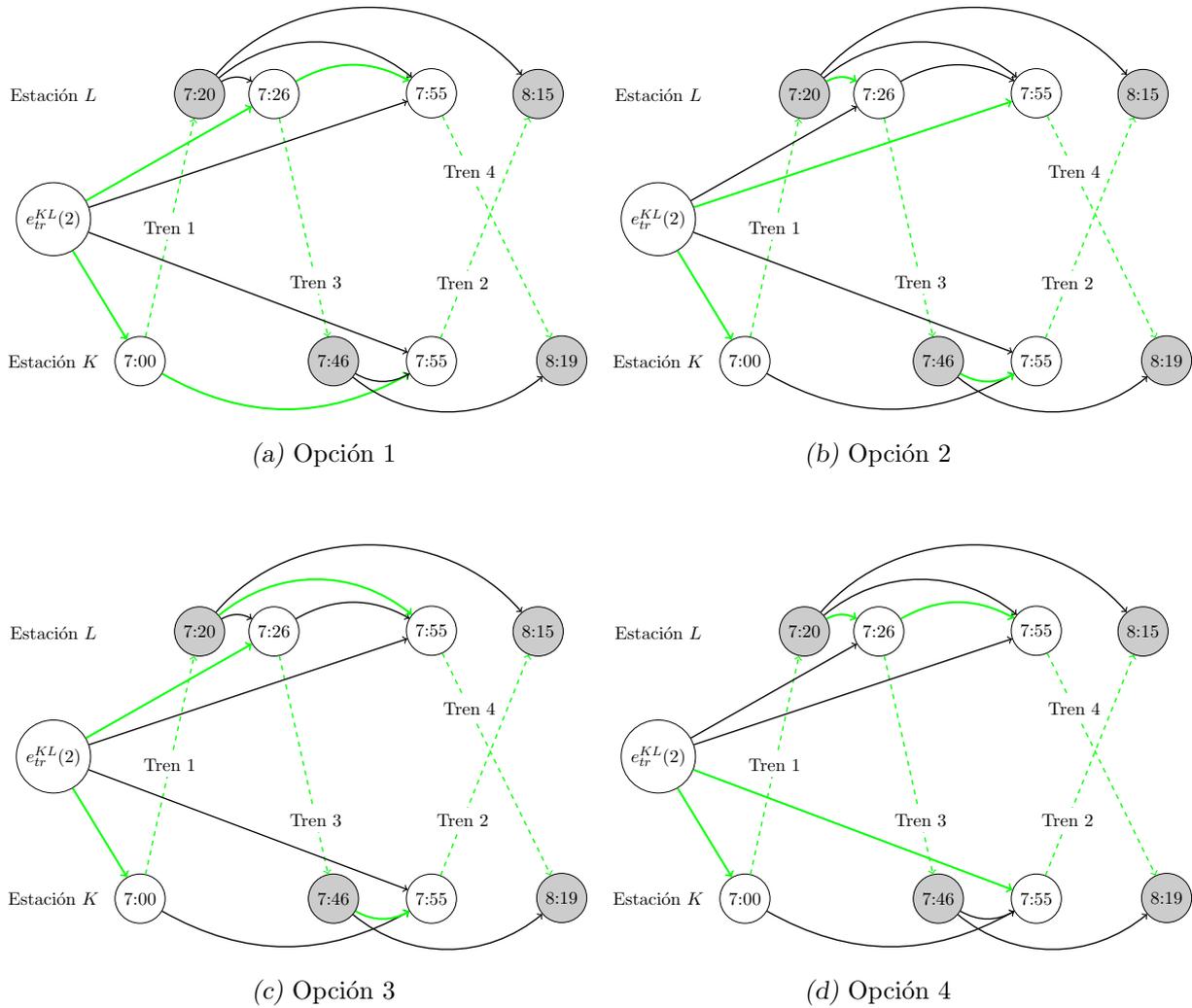


Fig. 3.5: Opciones disponibles de selección de actividades del recurso de vías de tramo para las actividades de la Figura 3.4. En cada opción, las actividades seleccionadas se muestran en color verde.

En cuanto a este ejemplo en particular, podemos ver como en la Figura 3.5a una de las vías del tramo es utilizada por los trenes que van desde la estación K hacia la L (trenes 1 y 2) mientras que la otra vía es utilizada por los trenes que van desde L hacia K (trenes 3 y 4). Otra cuestión interesante a notar es cómo la salida del tren 1 siempre toma el recurso de inventario. Esto se debe a que como su salida es la primera programada (7:00hs) y no puede recibir una vía del tramo de ningún otro evento, siempre deberá tomarlo del inventario.

Relación entre desigualdades de recursos

Una vez explicado en detalle el funcionamiento de uno de los tres tipos de recursos, incluimos un pequeño ejemplo para mostrar cómo las distintas desigualdades de recursos se relacionan entre sí para garantizar el correcto funcionamiento del modelo.

La Figura 3.6 muestra únicamente los trenes 1 y 3 para la red de ejemplo 3.2. Además,

para el ejemplo supondremos que entre las estaciones K y L no existe ninguna vía. Si bien en un caso real esto podría suceder solo en el contexto de un bloqueo total, el motivo de esta elección es que, dado que no hay vías en el tramo, el modelo debe forzar a que ambos trenes sean cancelados. Teniendo esto en mente, veamos cómo cada una de las desigualdades cumple un rol importante a la hora de modelar correctamente esta situación:

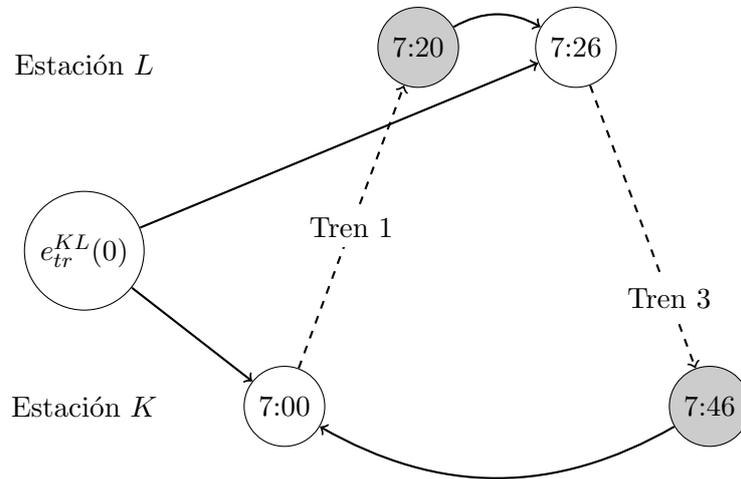


Fig. 3.6: Ejemplo de eventos y actividades del recurso de vías de tramo para interpretar la relación entre desigualdades de recursos. Ninguno de los dos trenes puede tomar los recursos necesarios y ambos son cancelados.

- Las actividades desde el evento de inventario no son permitidas por las desigualdades (3.8). De esta manera, ninguna de las dos salidas puede alimentarse de inventario, que es lo que pretendemos dado que su capacidad es 0.
- El arribo del tren 3 no puede traspasar el recurso a la salida del tren 1 por las desigualdades (3.9), dado que la salida del tren 1 ocurre 46 minutos antes que el arribo del tren 3 (y asumiendo que el máximo retraso permitido es menor a 46 minutos). Estas desigualdades que no son posibles por cuestiones de los tiempos de los eventos son las que directamente no fueron incluidas en la Figura 3.4 para simplificar su entendimiento.
- En base a lo anterior, como la salida del tren 1 no puede tomar ningún recurso, el tren 1 queda cancelado por las desigualdades (3.6). Sin embargo, con las restricciones mencionadas hasta el momento podríamos alimentar a la salida del tren 3 con el arribo del tren 1, a pesar de que este último haya sido cancelado, y aún así estaríamos cumpliendo todas las restricciones, permitiendo al tren 3 circular.
- Finalmente, incluyendo en el modelo las desigualdades (3.7), como el tren 1 fue cancelado, no puede traspasarle recursos a ningún otro tren, eliminando la posibilidad de que la salida del tren 3 tome el recurso y forzando a que ambos sean cancelados.

Adelantamiento sobre una misma vía

Tal como mencionamos anteriormente, múltiples trenes pueden circular en la misma vía al mismo tiempo, siempre y cuando circulen en la misma dirección y separados por un tiempo de seguridad mínimo. Sin embargo, con las actividades introducidas hasta el momento, el modelo no previene que un tren se adelante sobre otro que esté por delante en la misma vía, lo cual obviamente no debe permitirse. Por ejemplo, dados dos trenes t_1 y t_2 que parten desde la estación k hacia la estación l , si primero parte t_1 y luego t_2 (respetando el tiempo de seguridad entre ambas partidas), podría suceder que t_2 tenga un tiempo de viaje (i.e. L_a) menor que t_1 para el tramo (k, l) y de esa manera se produzca el adelantamiento.

Para prevenir estos casos, se introduce el concepto de *pares de actividades de vías*. Dada una sección de vías (k, l) y eventos $e, f \in E_{sal}^k$ y $e', f' \in E_{arr}^l$ tales que $t_e = t_{e'}$ y $t_f = t_{f'}$, sea $a = (e, f)$ la actividad representando las salidas consecutivas desde la estación k hacia una vía de la sección (k, l) y sea $a' = (e', f')$ la actividad representando los arribos consecutivos de los dos trenes a la estación l . Entonces el par (a, a') es un *par de actividades de vías*. Se define B_{tr} como el conjunto de todos los pares de actividades de vías. Para asegurar que no existan adelantamientos, se agrega al modelo el conjunto de desigualdades:

$$z_a = z_{a'} \quad \forall (a, a') \in B_{tr} \quad (3.11)$$

De esta manera, si la actividad a es seleccionada, también deberá serlo a' , por lo que se evitará el adelantamiento de t_f sobre t_e .

Es interesante notar que además de evitar el adelantamiento mencionado, también se está evitando que dos trenes puedan circular por la misma vía al mismo tiempo pero en sentido contrario. La Figura 3.7 muestra esta situación. La misma describe un ejemplo de un tramo (K, L) con una única vía en el cual transitan dos trenes en una dirección y otro en sentido contrario.

Como se muestra con el arco en rojo, el arribo del tren 1 le podría dejar el recurso de vía a la salida del tren 3. Esto implicaría que estén circulando sobre la misma vía y al mismo tiempo tanto el tren 2 como el tren 3, lo que no debería permitirse.

Sin embargo, el conjunto de desigualdades (3.11) no solo está forzando a que el arribo del tren 2 (llamémosle f') suceda después del arribo del tren 1 (llamémosle f), sino que también dice que f debe dejarle el recurso a f' (el arco verde de la Figura 3.7). De esta manera la salida del tren 3 no puede tomar el recurso de f , y se evita el problema mencionado.

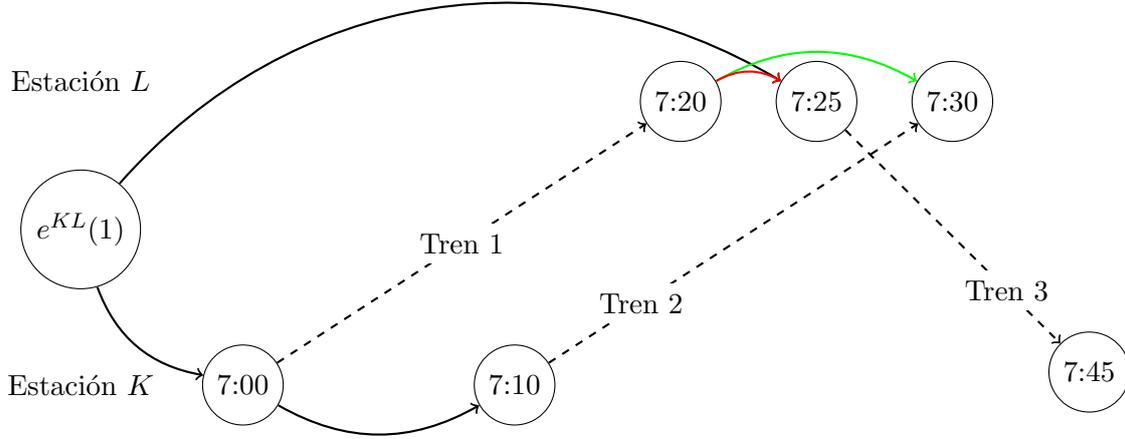


Fig. 3.7: Ejemplo de múltiples trenes en una misma vía. Los eventos se organizan horizontalmente según su horario programado y verticalmente según su estación.

3.4.2. Vías dentro de una estación

Cada estación tiene una capacidad determinada de vías. Al llegar a una estación, un tren necesita ser asignado a una vía determinada, la cual libera una vez que sale hacia la próxima estación, o si termina su recorrido y es guardado en el patio de maniobras. A diferencia de las vías correspondientes a tramos, no puede haber dos trenes al mismo tiempo en una vía de una estación.

Los eventos y actividades del recurso de vías de estación están asociados a una estación k particular. Para definir los conjuntos de actividades, utilizamos los conjuntos E_{sal}^k y E_{arr}^k definidos en la sección 3.4.1 previa. Además, definimos $e_{est}^k \in E_{inv}$ como el evento de inventario asociado a la estación k . Luego, $i_{e_{est}^k}$ representa la cantidad de vías disponibles en la estación k .

A continuación definimos para cada evento e los subconjuntos $A_{est}^-(e)$ y $A_{est}^+(e)$ relacionados con la estación k . Estas actividades representan el uso secuencial de las vías de la estación k .

- Para un arribo $e \in E_{arr}^k$ en la estación k :
 - El conjunto $A_{est}^-(e)$ está formado por actividades $a = (f, e)$ desde i) salidas desde la misma estación: $f \in E_{sal}^k, ii)$ el evento de inventario e_{est}^k .
 - El conjunto $A_{est}^+(e)$ es vacío, dado que es claro que el próximo evento en utilizar la vía de la estación tomada por el arribo es su posterior salida. El traspaso se modela de manera implícita.
- Para una salida $e \in E_{sal}^k$ desde la estación k :
 - El conjunto $A_{est}^-(e)$ es vacío por el mismo motivo mencionado arriba.
 - El conjunto $A_{est}^+(e)$ está formado por actividades $a = (e, f)$ hacia $i)$ otros arribos en la misma estación: $f \in E_{arr}^k$.

- Para el evento de inventario de la estación k , el conjunto $A_{est}^+(e_{est}^k)$ está formado por actividades $a = (e_{est}^k, e)$ hacia arribos en la estación: $e \in E_{arr}^k$.

La Figura 3.8 muestra el grafo de eventos y actividades del recurso de vías de estación para la estación L de la red de ejemplo 3.2 y el cronograma asociado 3.1. Para facilitar el entendimiento, se incluyen también las actividades de tren en líneas punteadas. Al igual que en la Figura 3.4, si bien por la definición de los conjuntos el modelo permite que un evento traspase el recurso a un evento programado en un tiempo anterior, dichas actividades no son incluidas dado que no podrían ser seleccionadas por las desigualdades (3.9). Para el evento de inventario, se incluye entre paréntesis su respectiva capacidad, en este caso, al tratarse de la estación L de la red ejemplo, la capacidad es de dos vías.

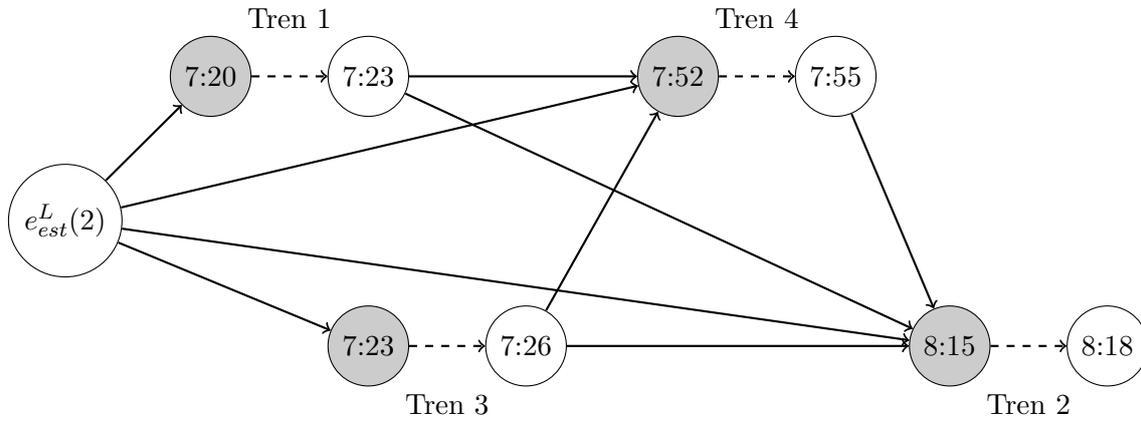


Fig. 3.8: Eventos y actividades relacionadas con el recurso de vías de una estación del cronograma ejemplo 3.1. Los eventos y actividades se organizan siguiendo la misma idea que la figura 3.4.

En base a las definiciones de los conjuntos $A_{est}^-(\cdot)$ y $A_{est}^+(\cdot)$, podemos ahora interpretar las desigualdades de recursos en particular para el recurso de vías de estación.

Las igualdades (3.6) implican que un tren puede arribar en una estación y tomar una vía v de la misma siempre y cuando: *i*) arribe a continuación de otro tren que liberó la vía v al salir de la estación o *ii*) tome la vía v por primera vez (i.e. tome el recurso del evento de inventario).

Las desigualdades (3.7) implican que si un tren parte desde una estación liberando una vía v , lo que puede ocurrir luego es que otro tren arribe a la estación y tome la vía v , o que dicha vía no sea utilizada más.

Las desigualdades (3.8) implican que la cantidad de arribos que toman una vía de una estación por primera vez no supere la cantidad total de vías que posee dicha estación. Por ejemplo, para nuestra red de ejemplo la estación L posee dos vías. Luego, a lo sumo dos arribos a L podrán tomar una vía por primera vez.

Por último, las desigualdades (3.9) aseguran que se respeten los tiempos de seguridad entre una salida de una estación que libere una vía v y un posterior arribo a la misma vía.

El conjunto de figuras 3.9 muestra las distintas soluciones posibles de traspaso de recursos para el conjunto de actividades de la estación L mostrado en la Figura 3.8. Al

igual que en la sección anterior, dado que aquí solo nos interesa estudiar las posibilidades de traspaso de recursos, haremos una simplificación y asumiremos que los tiempos de cada evento son fijos (i.e. $x_e = q_e$).

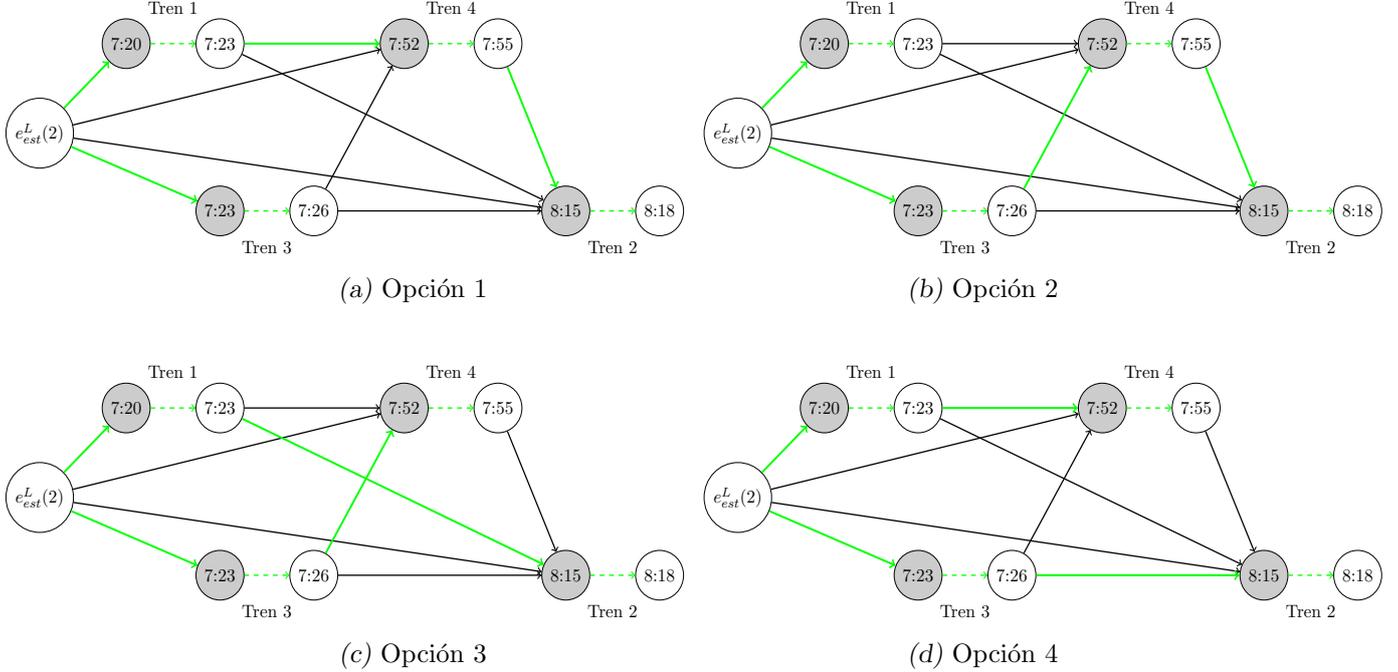


Fig. 3.9: Opciones disponibles de selección de actividades del recurso de vías de estación para las actividades de la Figura 3.8. En cada opción, las actividades seleccionadas se muestran en color verde.

Es importante recordar que seguir un camino de actividades seleccionadas implica seguir el uso secuencial de un mismo recurso, en este caso una vía de una estación, desde el evento de inventario hasta el último evento que lo utilice. Por ejemplo en el primer caso, en la Figura 3.9a vemos como los trenes 1, 4 y 2 son programados uno detrás del otro por la misma vía de L , mientras que el tren 3 utiliza la otra vía. Otra cuestión interesante a notar es que dados los horarios de los trenes del cronograma de ejemplo, y al estar asumiendo que los mismos son fijos, el modelo se ve forzado a que siempre los trenes 1 y 3 sean los que toman las vías desde inventario. Las distintas posibilidades radican en cómo esos trenes traspasan dichas vías a los próximos servicios programados a pasar por la estación L .

Manejos especiales primeras salidas y últimos arribos

Para terminar de modelar correctamente el recurso de las vías de una estación, deben tenerse en cuenta algunas consideraciones especiales para ciertos eventos.

Definimos $E_{com}^k \subseteq E_{sal}^k$ como las salidas desde la estación k que corresponden a la primera salida de un tren, es decir, salidas cuyos trenes comienzan su recorrido en k . Como se describió en la sección previa, las salidas de una estación obtienen de manera implícita el recurso a partir de su arribo previo. Para las salidas que corresponden a los comienzos de los servicios, no hay ningún arribo previo que tome ese recurso. Luego, para cada evento $e \in E_{com}^k$, construimos $A_{est}^-(e)$ como si se tratase de un arribo en vez de una

salida. De esta manera, para que el tren pueda salir por primera vez desde la estación k , necesitará tomar una vía de la misma (de inventario o de una salida anterior), y esto nos permite modelar correctamente este escenario. Notar como además de extender $A_{\text{est}}^-(e)$ para $e \in E_{\text{com}}^k$ con actividades desde eventos de inventario y desde otras salidas, también agregamos estas actividades a los conjuntos de salida correspondientes, i.e., $A_{\text{est}}^+(e_{\text{est}}^k)$ y $A_{\text{est}}^+(e)$ para $e \in E_{\text{sal}}^k$.

De manera similar, definimos $E_{\text{fin}}^k \subseteq E_{\text{arr}}^k$ como los arribos en la estación k que corresponden al último arribo de un tren, es decir, arribos cuyos trenes terminan su recorrido en la estación k . En la sección previa explicamos cómo se asume que los arribos siempre traspasan de manera implícita el recurso a su próxima salida. Para los arribos que corresponden al final de un servicio, no hay ninguna salida posterior, pero necesitamos que la vía de la estación pueda ser liberada. Luego, para cada evento $e \in E_{\text{fin}}^k$, construimos $A_{\text{est}}^+(e)$ como si se tratase de una salida en vez de un arribo. De esta manera, cuando un tren termina su recorrido en una estación determinada, puede liberar el recurso hacia otros próximos arribos. Nuevamente, también se extiende $A_{\text{est}}^-(e)$ para $e \in E_{\text{arr}}^k$ de manera de incluir estas actividades desde eventos en E_{fin}^k .

Por último, debemos tener en cuenta el caso donde tenemos ambas situaciones en conjunto. Si en una estación k tenemos un evento $e \in E_{\text{com}}^k$ y un evento $f \in E_{\text{fin}}^k$, estos eventos deben estar conectados: un tren que termina su recorrido en la estación k le puede traspasar la vía a un tren que empieza su recorrido en k . Luego, dada $a = (f, e)$, $a \in A_{\text{est}}^-(e)$ y $a \in A_{\text{est}}^+(f)$. Vale mencionar que para estos casos que relacionan trenes que terminan con otros que comienzan, otro factor importante que entra en juego es el recurso de material rodante. Una vez explicado su manejo en la próxima sección, veremos cómo manejar correctamente la interacción entre ambos recursos.

3.4.3. Material rodante

A continuación introducimos el último de los recursos involucrados en la red evento-actividad.

Cada tren necesita tener asignado durante su recorrido una formación (conjunto de unidades) de material rodante para poder circular. Un tren que comienza su recorrido en una estación determinada, puede tomar el recurso necesario de otro tren previo que terminó su servicio en la estación, o caso contrario tomar el material rodante del patio de maniobras, en caso de que haya formaciones disponibles. Cada estación con un patio de maniobras tiene asociada una capacidad de formaciones de material rodante disponibles al inicio del día.

Para definir los conjuntos de actividades, utilizamos los conjuntos E_{com}^k y E_{fin}^k definidos en la Sección 3.4.2 previa. Además, definimos $e_{\text{mr}}^k \in E_{\text{inv}}$ como el evento de inventario de material rodante asociado a la estación k . Luego, $i_{e_{\text{mr}}^k}$ representa la cantidad de formaciones de material rodante disponibles en el patio de maniobras de la estación k .

A continuación definimos para cada evento e los subconjuntos $A_{\text{mr}}^-(e)$ y $A_{\text{mr}}^+(e)$ relacionados con la estación k . Estas actividades representan tanto la liberación de formaciones de material rodante desde el patio de maniobras de la estación k , como así también el traspaso de material rodante proveniente de otras estaciones que se puede dar en k .

- Para un comienzo de un tren $e \in E_{com}^k$ desde la estación k :
 - El conjunto $A_{mr}^-(e)$ está formado por actividades $a = (f, e)$ desde i) arribos correspondientes a trenes que terminan en la estación: $f \in E_{fin}^k$, ii) el evento de inventario e_{mr}^k .
 - El conjunto $A_{mr}^+(e)$ es vacío, dado que el próximo evento en utilizar el material rodante es el arribo del tren t_e en la próxima estación. El traspaso se modela de manera implícita.
- Para un arribo final de un tren $e \in E_{fin}^k$ a la estación k :
 - El conjunto $A_{mr}^-(e)$ es vacío por el mismo motivo mencionado arriba.
 - El conjunto $A_{mr}^+(e)$ está formado por actividades $a = (e, f)$ hacia salidas correspondientes a trenes que comienzan en la estación: $f \in E_{com}^k$.
- Para el evento de inventario relacionado a la estación k , el conjunto $A_{mr}^+(e_{mr}^k)$ está formado por actividades $a = (e_{mr}^k, e)$ hacia comienzos desde la estación k : $e \in E_{com}^k$.

Es interesante notar como a diferencia de los otros dos tipos de recursos, para todos los eventos que no representan ni comienzos ni finales de trenes, no se definen actividades de material rodante. Esto se corresponde con el hecho de que no permitimos que un tren cambie sus unidades de material rodante durante su recorrido.

La Figura 3.10 muestra el grafo de eventos y actividades del recurso de material rodante para la red de ejemplo 3.2 y el cronograma asociado 3.1. Para facilitar el entendimiento, se incluyen también las actividades de tren en líneas punteadas. Al igual que para los recursos anteriores, si bien por la definición de los conjuntos $A_{mr}^-(\cdot)$ y $A_{mr}^+(\cdot)$ el modelo permite que un evento traspase el recurso a otro evento programado en un tiempo anterior, dichas actividades no son incluidas dado que no podrían ser seleccionadas por las desigualdades (3.9). Para los eventos de inventario, se incluye entre paréntesis su respectiva capacidad. En este caso, los mismos representan las unidades disponibles en los patios de maniobras de las estaciones K y M , por lo que en este ejemplo tenemos 1 y 2 unidades respectivamente.

En base a las definiciones de los conjuntos $A_{mr}^-(\cdot)$ y $A_{mr}^+(\cdot)$, podemos ahora interpretar las desigualdades de recursos en particular para el recurso de material rodante.

Las igualdades (3.6) implican que un tren puede comenzar su recorrido en una estación tomando una formación de material rodante m siempre y cuando: i) salga a continuación de otro tren anterior que le dejó disponible el material rodante m al terminar en la estación o ii) tome el material rodante m desde el patio de maniobras de la estación (i.e. tome el recurso del evento de inventario).

Las desigualdades (3.7) implican que si un tren termina su recorrido en una estación liberando el material rodante m , lo que puede ocurrir luego es que otro tren que comience en dicha estación tome el recurso m , o que el material rodante no sea utilizado más.

Las desigualdades (3.8) implican que a lo sumo $i_{e_{mr}^k}$ trenes pueden comenzar su recorrido en la estación k tomando material rodante disponible en el patio de maniobras de k . Por ejemplo, para nuestra red de ejemplo la estación K posee un único conjunto de unidades de material rodante disponible de inventario. Luego, a lo sumo un tren podrá comenzar desde la estación K tomando el material rodante desde el patio de maniobras.

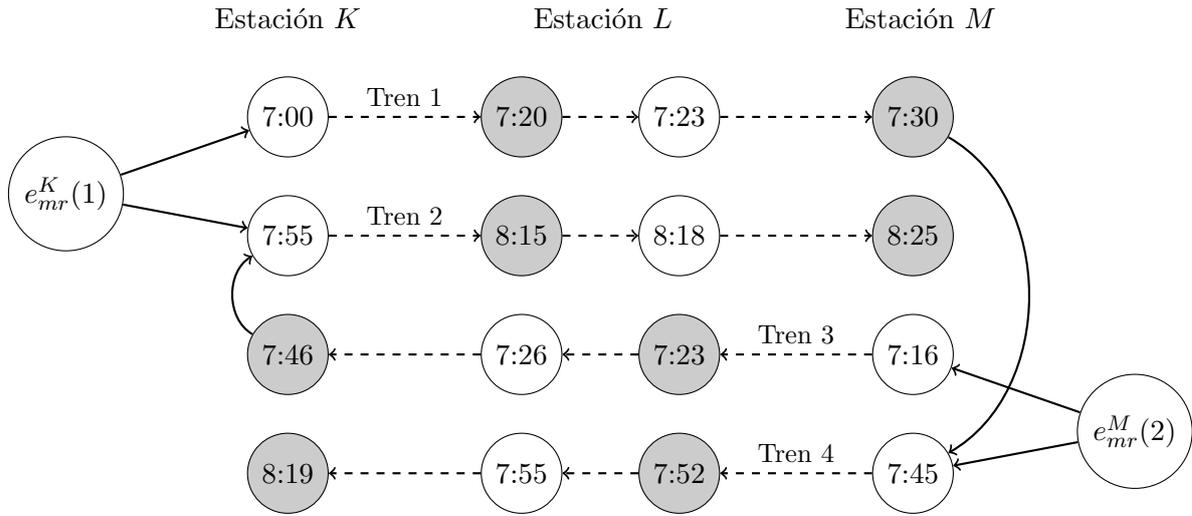


Fig. 3.10: Eventos y actividades relacionadas con el recurso de material rodante del cronograma ejemplo 3.1. Los eventos se organizan horizontalmente según su ubicación geográfica.

Por último, las desigualdades (3.9) aseguran que se respeten los tiempos de seguridad entre la finalización de un servicio que libere el material rodante m y un posterior servicio que comience en la misma estación utilizando m .

El conjunto de figuras 3.11 muestra las distintas soluciones posibles de traspaso de recursos para el conjunto de actividades de la Figura 3.10. Al igual que en las secciones anteriores, dado que aquí solo nos interesa estudiar las posibilidades de traspaso del material rodante, haremos una simplificación y asumiremos que los tiempos de cada evento son fijos (i.e. $x_e = q_e$).

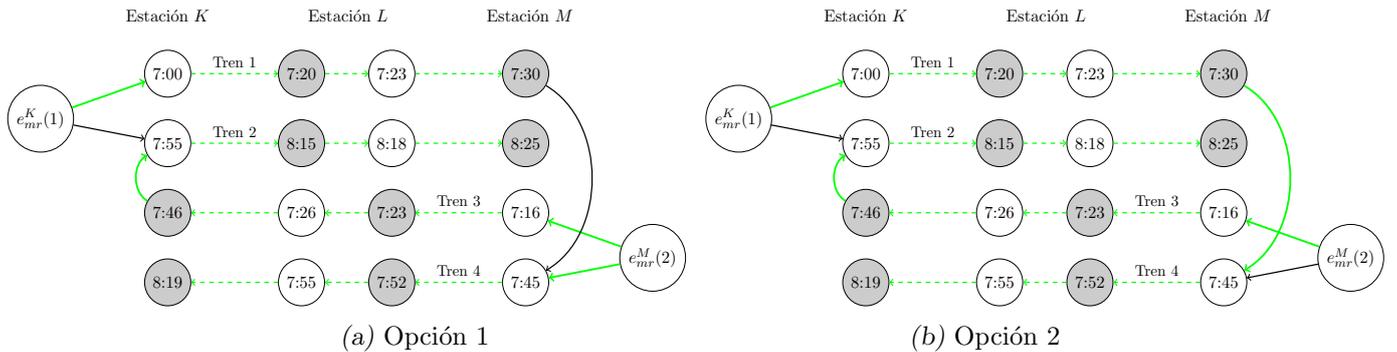


Fig. 3.11: Opciones disponibles de selección de actividades del recurso de material rodante para las actividades de la Figura 3.10. En cada opción, las actividades seleccionadas se muestran en color verde.

En este caso, tenemos dos eventos de inventario, uno para el patio de maniobras de la estación K y otro para el de M . Siguiendo un camino de actividades seleccionadas desde cada evento de inventario nos permite ver el uso secuencial de las unidades de material rodante de cada patio de maniobras. Por ejemplo en el primer caso, en la Figura 3.11a

vemos como el tren 1 toma el material rodante desde el patio de maniobras de K y lo utiliza para circular hasta la estación M . Asimismo, los trenes 3 y 4 toman el material rodante desde el patio de maniobras de M para poder comenzar su recorrido. Es interesante notar como el tren 2 utiliza el material rodante que termina dejando el tren 3 en K una vez que termina su recorrido.

La Figura 3.11b nos muestra la otra posibilidad de selección de las actividades del ejemplo. En este caso el tren 4 toma el material rodante que dejó el tren 1 en M . De esta manera, quedará una unidad de material rodante disponible en el patio de maniobras de M para posterior uso por parte de otro tren.

3.4.4. Relación entre las vías de una estación y el manejo del material rodante

Los recursos de vías de estación y de material rodante están estrechamente relacionados. Recordemos cómo se definen los conjuntos $A_{\text{est}}^+(e)$ para los arribos que corresponden a finales de recorridos en una determinada estación k . Estos pueden traspasar la vía de la estación que utilizaron tanto a posteriores arribos a k como a comienzos desde k . Si un fin de recorrido e le traspasa la vía v de k a un arribo posterior, significa que el tren t_e no está ocupando más la vía v , con lo cual su material rodante tuvo que ser guardado en el patio de maniobras de la estación.

Dicha situación da un buen indicio de la relación entre ambos recursos. A continuación discutiremos algunas consideraciones a tener en cuenta para manejar correctamente su interacción.

Traspaso de material rodante y el patio de maniobras

Detallando la idea anterior, supongamos una estación k , un evento $e \in E_{\text{fin}}^k$ y otro evento $f \in E_{\text{com}}^k$. Por la definición de $A_{\text{est}}^+(\cdot)$ en la Sección 3.4.2, sabemos que $A_{\text{est}}^+(e)$ estará compuesto por arribos a la estación k como así también comienzos desde la misma, con lo cual $f \in A_{\text{est}}^+(e)$. Para estos eventos, el recurso de material rodante funciona de manera similar. En la Sección 3.4.3 vimos como los trenes que terminan su recorrido en una estación k le pueden dejar el material rodante a los trenes que comienzan en k , con lo cual también sucede que $f \in A_{\text{mr}}^+(e)$.

Sin embargo, no está claro si el material rodante que e le traspasa a f debe ser guardado en el patio de maniobras de la estación antes de hacerse el traspaso, o si e se lo puede traspasar directamente sin pasar por el patio de maniobras, ocupando temporalmente una vía de la estación.

Para comprender cómo se pueden presentar ambas situaciones, se incluye a continuación la Figura 3.12, que muestra los dos posibles escenarios para los eventos mencionados.

Como podemos ver en la figura, en el caso 3.12b, el evento e , luego de llegar a la estación k le deja el recurso de vía a un arribo posterior a k , notado en la figura como e' . Por este motivo, en ese instante, para que el tren $t_{e'}$ pueda ocupar la vía de la estación, el material rodante de t_e debe guardarse en el patio de maniobras. Si el próximo evento en tomar dicho material rodante es f , significará entonces que durante el traspaso, las unidades de material rodante pasaron por el patio de maniobras. Dicha situación se refleja mediante la actividad de material rodante entre e y f a través del patio. Finalmente para este caso, como f no toma el recurso de vía de e , toma el mismo desde inventario (también

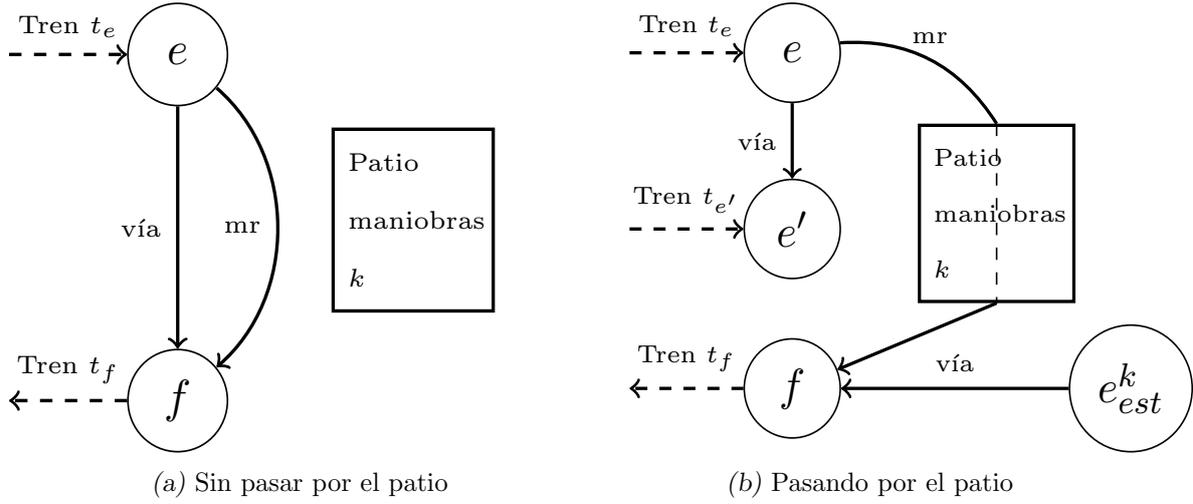


Fig. 3.12: Opciones de traspaso de material rodante desde el evento e hacia f . En el caso 3.12a el traspaso es directo, sin pasar por el patio de maniobras. En el caso 3.12b el traspaso es a través del patio. Las líneas punteadas desde y hacia los eventos indican si sus respectivos trenes corresponden a arribos o salidas de la estación k .

podría tomarlo de otro evento, por ejemplo e' , en caso de que su tren sea guardado en el patio de maniobras). Por otro lado, en el caso 3.12a el evento e le trasporta el recurso de la vía a f . Luego, también podría pasarle el recurso de material rodante de manera directa, sin necesidad de pasar por el patio, tal como muestra la figura.

Vemos entonces que para tener un modelo más robusto, debemos ser capaces de contemplar ambas situaciones, introduciendo la posibilidad de un traspaso de material rodante directo. Sin embargo, los tiempos de seguridad de dichas actividades son muy distintos. En el caso de tener un ida y vuelta hacia el patio de maniobras, el tiempo de traspaso es en general mucho mayor que si el material rodante simplemente espera en la plataforma para ser utilizado por un nuevo tren. El modelo, tal como fue introducido en la Sección 3.4.3, incluye una única actividad $a = (e, f)$ de traspaso de material rodante entre un evento $e \in E_{fin}^k$ y otro $f \in E_{com}^k$. Esta actividad representa el traspaso a través del patio. Luego, para el mismo par de eventos agregamos al modelo una nueva actividad $\tilde{a} = (e, f)$ para modelar el traspaso directo, con lo cual tendremos que $L_{\tilde{a}} < L_a$.

Sea $A_{mr}^{\overline{pm}} \subseteq A_{mr}$ el conjunto de actividades de material rodante correspondientes al traspaso desde un final de tren a un comienzo **sin** pasar por el patio de maniobras. Este conjunto de actividades, representa el caso de la figura de ejemplo 3.12a. Es importante notar que en estos casos en que el traspaso es directo, también se fuerza una restricción sobre el traspaso de la vía: si $e \in E_{fin}^k$ le trasporta el material rodante de manera directa a $f \in E_{com}^k$ mediante la actividad $a = (e, f) \in A_{mr}^{\overline{pm}}$, entonces necesariamente deberá pasarle también el recurso de vía, i.e. la actividad $a' = (e, f) \in A_{est}$ debe ser utilizada.

Dados dos eventos $e \in E_{fin}^k$ y $f \in E_{com}^k$ tales que $a = (e, f) \in A_{mr}^{\overline{pm}}$, sea $a' = (e, f) \in A_{est}$ la actividad que relaciona los mismos eventos para el recurso de vías de estación, entonces el par (a, a') es un *par de actividades de estación y material rodante*. Llamaremos B_{mr}^{est} al conjunto de todos estos pares. En base a este conjunto, se agrega al modelo la

siguiente desigualdad que asegura que se respete la situación mencionada para todos los casos:

$$z_a \leq z_{a'} \quad \forall (a, a') \in B_{mr}^{est} \quad (3.12)$$

Tomamos como contexto las dos situaciones de la Figura 3.12 entre los eventos $e \in E_{fin}^k$ y $f \in E_{com}^k$ para comprender cómo la nueva desigualdad modela correctamente lo que buscamos. En el caso 3.12a, como e traspasa el material rodante de manera directa a f , esto fuerza el traspaso del recurso de vía de estación. Luego, ningún otro tren podrá pasar por la vía durante el tiempo en que el tren t_e termina su recorrido y el tren t_f comienza el suyo. Por otro lado, en el caso 3.12b, como el evento e le traspasa el recurso de vía a un arribo, esto significa que la actividad $a' = (e, f) \in A_{est}$ no se utiliza. De esta manera, la actividad $a = (e, f) \in A_{mr}^{pm}$ no puede ser utilizada y el traspaso directo no es posible.

Estaciones sin patio de maniobras

Al introducir la relación entre los recursos de vías de estación y material rodante, mencionamos como disparador el hecho de que si un final de tren libera su vía de estación hacia un arribo posterior, su material rodante debe ser guardado en el patio de maniobras para liberar la vía. Esto da lugar a otra situación no contemplada: ¿qué sucede si la estación en cuestión no posee un patio de maniobras?

De acuerdo a la definición del conjunto de actividades A_{est} de la Sección 3.4.2, independientemente de la estación, todo fin de tren puede dejarle el recurso de vía a un arribo posterior a la misma. Luego, el modelo asume que todas las estaciones de la red tienen un patio de maniobras. Sin embargo, en una red de trenes típica lo que sucede es que muchas de las estaciones que la conforman no poseen un patio, sino que simplemente se comportan como estaciones de carga y descarga de pasajeros, pero donde no es posible guardar trenes.

Recordemos que S es el conjunto de estaciones de la red. Supongamos que $s \in S$ no posee patio de maniobras. Para asegurarnos que modelamos correctamente este tipo de estaciones, debemos hacer las dos siguientes adaptaciones:

- Como dijimos, todo tren que termine en s no debe poder liberar la vía de s que utilizó hacia otro arribo posterior, ya que no hay patio de maniobras para guardar su material rodante. Luego, para todo $e \in E_{fin}^s$, $A_{est}^+(e)$ estará compuesto únicamente por eventos correspondientes a comienzos de trenes desde la estación s , i.e., $A_{est}^+(e) = \{a = (e, f) | f \in E_{com}^s\}$.
- Para el traspaso de material rodante en la estación s , solo se definen las nuevas actividades A_{mr}^{pm} introducidas en la sección anterior, es decir, solo se permite el traspaso directo y no es posible el traspaso a través del patio de maniobras. Además, como ahora para la estación s poseemos una única actividad de material rodante entre un fin y un comienzo, necesitamos para esos casos cambiar también la desigualdad (3.12) por una igualdad. De esta manera, si un final $e \in E_{fin}^s$ le traspasa el recurso de vía a un comienzo $f \in E_{com}^s$, necesariamente le pasará también el material rodante, de manera directa. Luego, lo que está sucediendo para esta estación es que el traspaso de vía y el de material rodante se comportan como si fueran una misma actividad, i.e., o suceden ambas o ninguna.

Por último, es importante notar las posibles implicancias de las adaptaciones mencionadas. Si un tren termina su recorrido en la estación s ocupando la vía v de la misma, dicha vía estará ocupada hasta el momento en que otro tren comience su recorrido en s y la tome. En ciertos casos, escenarios como este podrían causar cancelaciones de servicios, producidas debido a estaciones sin patio de maniobras y con su capacidad de vías llenas, a las que no se puede arribar. Si bien es cierto que generalmente en los cronogramas de trenes todos los servicios terminan sus recorridos en estaciones con patio de maniobras, veremos como al introducir el bloqueo de una sección de vías en la Sección 3.5, la problemática mencionada puede surgir en otro contexto.

3.4.5. Aclaraciones sobre los conjuntos $A^-()$ y $A^+()$

Para culminar la sección donde detallamos cómo se modela el traspaso de cada tipo de recurso, hacemos unas aclaraciones finales.

Al introducir la red evento-actividad N , definimos a la misma como un grafo dirigido formado por eventos (nodos) relacionados por medio de actividades (ejes). Más aún, si tomamos solo el subconjunto de actividades y eventos de inventario relacionadas a un recurso r particular, podemos pensar a los subgrafos asociados con cada recurso como grafos separados, donde cada uno modela cómo se traspasan las unidades del recurso r entre todos los eventos. Los 3 subgrafos están relacionados, dado que comparten los mismos eventos de tren y actividades de tren (i.e. $e \in E_{tren}, a \in A_{tren}$). Estos eventos y actividades que comparten imponen restricciones en común que se deben cumplir para todos los recursos, con lo cual nuestro modelo buscará el mejor traspaso de recursos en su conjunto.

A lo largo de las distintas subsecciones, vimos cómo se definen para cada evento e los conjuntos $A_r^-(e)$ y $A_r^+(e)$, para cada tipo de recurso r . Es importante notar que existen ciertas diferencias entre la interpretación de esos conjuntos con los ejes de entrada y salida de un grafo tradicional. En particular, agregar una actividad $a = (e, f)$ al grafo N no implica que la misma deba ser agregada tanto al conjunto $A_r^+(e)$ como a $A_r^-(f)$.

La Figura 3.13 muestra un pequeño ejemplo donde se presenta este escenario. La misma muestra el conjunto de eventos y actividades del recurso de vías de tramo para las estaciones K y L de la red de ejemplo 3.2, junto con los trenes 1 y 2 del cronograma de ejemplo 3.1.

La actividad marcada en azul en la figura es la que se encuentra en la situación mencionada. Llamemos $a = (e, f)$ a dicha actividad, i.e., e representa el arribo del tren 1 y f el arribo del tren 2.

Tal como se describe en la definición de los conjuntos A_{tr} en la Sección 3.4.1, los arribos a una estación k pueden traspasarle el recurso de vía a un arribo posterior a la estación. En base a esto, se define que $a \in A_{tr}^+(e)$. Mirándolo desde el punto de vista de un grafo, podemos pensar que lo que hacemos es agregar al mismo el eje a , que une los vertices e y f . Si tratáramos al conjunto $A_{tr}^-(e)$ de la misma manera, también debería suceder que agreguemos a al conjunto $A_{tr}^-(f)$. Luego, como ahora $A_{tr}^-(f) \neq \emptyset$, quedaría definida la igualdad (3.6) relativa a la variable z_a :

$$z_a + y_{t_f} = 1$$

Para comprender la problemática de esto, supongamos que el tren 1 debe ser cancelado, por el motivo que fuere. Si nuestro modelo fuese correcto, esto no debería tener ningún

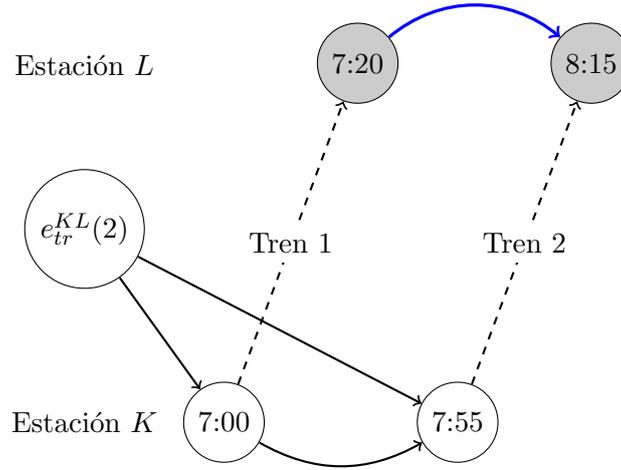


Fig. 3.13: Eventos y actividades relacionadas con el recurso de vías de tramo para el tramo (K, L) para los trenes 1 y 2 del cronograma ejemplo 3.1.

impacto sobre el funcionamiento del tren 2. Sin embargo, la variable z_a también aparece en las desigualdades (3.7):

$$z_a + y_{t_e} \leq 1$$

Como el tren 1 debe ser cancelado (i.e. la variable y_{t_e} toma valor 1), por las desigualdades (3.7) tenemos que $z_a = 0$. En base a esto, por las desigualdades (3.6) tenemos que $y_{t_f} = 1$, por lo que se termina forzando a que el tren 2 también sea cancelado.

El origen del problema viene del hecho de agregar la actividad a al conjunto $A_{tr}^-(f)$. En la Sección 3.4.1 se menciona como dado un evento e correspondiente a un arribo, el conjunto $A_{tr}^-(e)$ es vacío. En nuestro ejemplo, tomando $A_{tr}^-(f) = \emptyset$, se resuelve el problema mencionado.

En base al ejemplo visto, vale aclarar que esta situación particular puede ocurrir únicamente cuando un conjunto de actividades se fuerza a ser vacío, como en el caso de los arribos para las vías de tramos. De hecho, por cómo se definen los conjuntos A_{est} y A_{mr} , este es el único caso donde esta situación se presenta. Para cualquier otro caso, siempre que estemos agregando una actividad $a = (e, f)$ relativa a un recurso r a la red evento-actividad, esto implicará agregarla tanto al conjunto de salida de e como al conjunto de entrada de f , es decir, $a \in A_r^+(e)$ y $a \in A_r^-(f)$. También debe notarse como en estos casos, la variable z_a de nuestra formulación de PLE, al representar a la misma actividad que pertenece a ambos conjuntos, se verá involucrada tanto en las desigualdades (3.6) como en (3.7) o (3.8) según corresponda, tal como sucede en el ejemplo 3.13.

3.5. Bloqueo de una sección de vías

En la siguiente sección explicaremos en detalle las extensiones que deben realizarse sobre el modelo para contemplar correctamente el bloqueo de vías dentro de nuestro problema.

Recordemos, tal como introdujimos en la Sección 3.1.3, que el bloqueo afecta un tramo de vías entre dos estaciones. El mismo puede afectar solo algunas de las vías del tramo, en cuyo caso se tratará de un bloqueo parcial, o afectar todas las vías del tramo, produciendo un bloqueo total. En la misma sección introductoria comentamos que existen tres instantes de tiempo que le dan entidad al bloqueo. Por un lado tenemos el tiempo de inicio, al cual llamaremos en nuestro modelo τ_1 . Al tiempo de fin del bloqueo, que determina cuando las vías afectadas vuelven a estar disponibles, lo notaremos como τ_2 . Finalmente, el tiempo límite de transición desde el nuevo cronograma generado hacia el cronograma original lo notaremos τ_3 .

3.5.1. Trenes sobre el área de bloqueo

Supongamos que se produce un bloqueo parcial del tramo de vías (k, l) por el cual el tren $t \in T$ tiene su recorrido programado, en dirección desde k hacia l . Luego, si la salida desde la estación k está programada **durante** el bloqueo (i.e. entre τ_1 y τ_2), los eventos que estaban asociados con el tren t son particionados y asociados con tres nuevos trenes parciales, llamados α_t , β_t y γ_t . Llamamos $T^{blq} \subseteq T$ al conjunto de trenes en esta situación, es decir, $t \in T^{blq}$ si su recorrido pasa por el área del bloqueo y su salida hacia el tramo del bloqueo está programada durante el transcurso del mismo.

Continuando con el ejemplo, llamemos e^{com} y e^{fin} a los eventos correspondientes al comienzo y fin de recorrido del tren t respectivamente. e_k^{arr} y e_l^{arr} denotan los arribos a las estaciones k y l respectivamente, y de manera análoga e_k^{sal} y e_l^{sal} modelan las salidas desde k y l . Podemos ver cómo estarán contruidos los nuevos trenes parciales a partir de los eventos del tren t . El tren α_t contiene los eventos desde e^{com} hasta e_k^{arr} . El tren β_t contiene únicamente a los eventos e_k^{sal} y e_l^{arr} . Mientras que el tren γ_t posee los eventos desde e_l^{sal} hasta e^{fin} .

Se agregan al modelo las desigualdades (3.13) y (3.14). Las mismas aseguran que si el tren β_t es operado (la parte del tren original t sobre el área afectada) entonces tanto α_t como γ_t son también operados. Esto es equivalente a operar el tren t completo del cronograma original. Sin embargo, si β_t es cancelado, luego α_t y γ_t pueden, de manera independiente, circular o ser cancelados.

$$y_{\beta_t} \geq y_{\alpha_t} \quad \forall t \in T^{blq} \quad (3.13)$$

$$y_{\beta_t} \geq y_{\gamma_t} \quad \forall t \in T^{blq} \quad (3.14)$$

Las actividades de recursos relacionadas con los eventos de los trenes parciales son exactamente las mismas que para los eventos del tren original t . Sin embargo, para asegurarnos que α_t , β_t y γ_t usen el mismo material rodante en caso de que ninguno sea cancelado, agregamos una actividad de material rodante desde el evento e_k^{arr} del tren α_t hacia el evento e_k^{sal} del tren β_t y otra desde el evento e_l^{arr} de β_t hacia el evento e_l^{sal} del tren γ_t , ambas con tiempos de seguridad nulo. Además, estas son las únicas actividades de material rodante definidas para el tren β_t .

Por último, para el evento e_l^{arr} , la desigualdad (3.7) se vuelve una igualdad, i.e., si β_t no se cancela, necesariamente le traspasa el material rodante a γ_t .

Con estos pequeños ajustes, aseguramos que si ninguno de los trenes parciales se cancela, todos usen el mismo material rodante. Por otro lado, en caso de que β_t sea cancelado, el material rodante usado por α_t y γ_t puede ser utilizado por otros trenes desde las estaciones k y l respectivamente.

3.5.2. Armado del conjunto de eventos E

Analicemos el comportamiento de los eventos que se encuentran por fuera de τ_1 y τ_3 , es decir, eventos previos a τ_1 y eventos posteriores a τ_3 .

Por un lado, el horario de todos los eventos que ocurren previos a τ_1 es fijo, y es el horario programado originalmente, ya que hasta ese punto el bloqueo aún no comenzó y sabemos que la programación original es correcta. Además, los trenes asociados a estos eventos no pueden ser cancelados, ya que se encuentran andando previo al inicio del bloqueo.

Por otro lado, lo que se busca resolver mediante el modelo de PLE no es solo minimizar el impacto en la calidad del servicio producido por el bloqueo, sino también asegurar que en algún momento posterior al final del boqueo, los trenes vuelvan a circular de acuerdo al cronograma original. El instante de tiempo que define dicho momento es τ_3 , luego, cualquier evento programado en un tiempo posterior a τ_3 no puede ser retrasado. Asimismo, cualquier tren que comience su recorrido en un tiempo posterior a τ_3 no puede ser cancelado.

Debido a estas observaciones, no es necesario que el conjunto de eventos E contenga los eventos de todos los trenes programados en el cronograma original. Es suficiente con incluir únicamente los eventos programados después de τ_1 y antes de τ_3 , dado que para el resto se utilizan los horarios originales. En base a esto, es importante notar que no solo el tamaño de la red tendrá un gran impacto en la dificultad del problema, también la duración del bloqueo será otro factor clave a considerar.

Se deben tener algunas consideraciones especiales en los límites de E para por un lado modelar correctamente la disponibilidad de los recursos en τ_1 y por otro asegurar que los trenes puedan retomar el cronograma original a partir de τ_3 .

Manejo especial eventos previos a τ_1

Si solo considerásemos en E eventos e con $q_e \geq \tau_1$ estaríamos asumiendo que todos los recursos se encuentran libres y disponibles en el instante τ_1 . Esto se debe a que las actividades definidas desde los eventos de inventario no tienen tiempo de seguridad. Sin embargo, lo que sucede en realidad es que los eventos que suceden previo a τ_1 utilizan recursos de la red para operar, y luego traspasan dichos recursos a los eventos posteriores a τ_1 .

Supongamos que tenemos un evento e programado en el instante $q_e = \tau_1 - k$, $k > 0$ y un evento f programado en el instante $q_f = \tau_1$, es decir $e \notin E$, $f \in E$. Dado un recurso cualquiera r , el evento e puede traspasarle el recurso r a f mediante la actividad $a = (e, f)$. Dicha actividad tiene un determinado L_a . De esta manera, si la actividad a es utilizada, por (3.9) tendremos que:

$$x_f \geq x_e + L_a = q_e + L_a = \tau_1 - k + L_a$$

En el caso de que $k < L_a$, tendremos entonces que $x_f > \tau_1 = q_f$. Aquí vemos cómo sería incorrecto incluir en E solo a los eventos programados a partir de τ_1 , dado que al asumir que todos los recursos están disponibles en el instante τ_1 no modelaríamos correctamente la realidad.

Siguiendo la idea anterior, definimos $L_a^{max} = \max(\{L_a : a \in A\})$. Tomemos ahora el mismo evento e pero con $k = L_a^{max}$. Si la actividad a es utilizada, se fuerza a que:

$$x_f \geq x_e + L_a = q_e + L_a = \tau_1 - L_a^{max} + L_a$$

Como $\tau_1 - L_a^{max} + L_a \leq \tau_1$, el uso de a fuerza a lo sumo que $x_f \geq \tau_1$. Es decir, asumir que el evento e le traspasa el recurso a f o asumir que el recurso está disponible desde inventario en el instante τ_1 es lo mismo. De esta manera, podemos concluir que si incluimos en E todos los eventos programados en un tiempo $t \geq \tau_1 - L_a^{max}$ estamos modelando correctamente el traspaso de los recursos. Llamaremos a este límite inferior $lim_E^- = \tau_1 - L_a^{max}$.

Tal como dijimos anteriormente, los eventos anteriores a τ_1 no pueden ser retrasados, ni sus trenes asociados cancelados. Luego, para estos eventos adicionales incluidos en E definimos

$$d_e = 0, y_{t_e} = 0 \quad \forall e \in E, q_e < \tau_1$$

De esta manera, si un evento toma un recurso de un evento previo a τ_1 , como el horario de este último es fijo, el recurso estará disponible en el instante correcto y no en τ_1 como si se obtuviera de inventario.

Vale aclarar que los eventos adicionales incluidos con $q_e < \tau_1$ requieren recursos para operar como cualquier evento normal, caso contrario se dispondrían de más recursos que el inventario total dado que dichos eventos luego liberarían recursos que nunca consumieron. Como los eventos adicionales mencionados toman sus recursos necesarios de inventario, se modela también el hecho de que estos trenes previos cuentan originalmente con recursos de la red asignados.

Manejo especial eventos posteriores a τ_3

Como se comentó anteriormente, debemos asegurar que los eventos programados en un tiempo posterior a τ_3 no sean retrasados ni cancelados. En particular, si nos limitamos a incluir en E solo eventos e con $q_e \leq \tau_3$, no podemos asegurar que luego de τ_3 no tengamos que retrasar algún evento.

Para ver esto, haciendo un análisis similar al de los eventos anteriores a τ_1 , supongamos que tenemos un evento f con $q_f = \tau_3 + k, k > 0$. Supongamos también que el evento e con $q_e = \tau_3$ es el último evento en utilizar el recurso r . Luego $e \in E, f \notin E$. Si e le traspasa el recurso r a f mediante la actividad $a = (e, f)$, por las desigualdades (3.9) debe cumplirse que:

$$x_f \geq x_e + L_a \geq q_e + L_a = \tau_3 + L_a$$

En el caso de que $k < L_a$, significa que $q_f = \tau_3 + k < \tau_3 + L_a$, pero entonces:

$$q_f < \tau_3 + L_a \leq x_f$$

Vemos entonces que el evento f estaría obligado a retrasar su horario original, lo cual no debe permitirse dado que f está programado después de τ_3 . Luego, el evento f debe obtener su recurso necesario desde otro evento.

De hecho, a diferencia del análisis anterior, en este caso el evento e no está programado por fuera del límite considerado (no es mayor a τ_3) y por lo tanto puede ser retrasado hasta $q_e + d_e$, imponiendo en dicho caso como nuevo horario de f a $x_f = \tau_3 + d_e + L_a$. Como la idea es que no haya retrasos después de τ_3 , para los casos donde $q_e < \tau_3$, $q_e + d_e > \tau_3$ se redefine el máximo retraso permitido como

$$d_e = \tau_3 - q_e \quad \forall e \in E, q_e < \tau_3, q_e + d_e > \tau_3$$

En base a este ajuste, y siguiendo la misma deducción que para los eventos previos a τ_1 , se puede ver fácilmente que si incluimos en E todos los eventos programados en un tiempo $t \leq \tau_3 + L_a^{max}$, podemos asegurar que todos los eventos posteriores a τ_3 no serán retrasados. No es necesario ver más adelante en el tiempo ya que si sabemos que todos los eventos entre τ_3 y $\tau_3 + L_a^{max}$ no se retrasan, esto significará que todos los eventos posteriores tendrán sus recursos necesarios y no será necesario reprogramarlos. Llamaremos al límite superior definido $lim_E^+ = \tau_3 + L_a^{max}$.

En realidad, se requiere una atención especial para el recurso de material rodante, ya que debemos asegurar una cantidad determinada trasladada en cada estación, el detalle de dicho manejo se puede ver en la Sección 3.5.3.

Finalmente, al igual que para los eventos previos a τ_1 , para asegurar que los eventos posteriores a τ_3 no se puedan retrasar, y que sus trenes asociados no se puedan cancelar, definimos

$$d_e = 0, y_{t_e} = 0 \quad \forall e \in E, q_e > \tau_3$$

3.5.3. Factibilidad post lim_E^+ del material rodante

A diferencia de los recursos de vías de tramo y de vías de estación, el material rodante es un recurso que no es fijo de una determinada estación o conjunto de las mismas, sino que se mueve por las distintas estaciones. Como se mencionó previamente, necesitamos garantizar que los eventos programados posteriormente a τ_3 (y por consiguiente a lim_E^+) puedan andar sin retrasos y sin ser cancelados. Para asegurar esto, es suficiente que en lim_E^+ se cuente en cada estación con al menos la misma cantidad de material rodante que se cuenta en caso de que no haya bloqueo. De ese modo, sabemos que los trenes pueden operar sin problemas dado que la programación original es correcta.

Conocer la cantidad de material rodante que debería haber en cada estación en el instante lim_E^+ en caso de que no haya bloqueo es fácil. En base al programa original y a la cantidad de formaciones disponibles al inicio del día en cada estación, simplemente se simula la corrida de los servicios, sabiendo que cada tren que termina en una estación deja un recurso de material rodante en la misma, y contrariamente cada tren que comienza en una estación toma un recurso de la misma. Vale aclarar que un cálculo similar se hace para establecer la cantidad de material rodante disponible al inicio del bloqueo, que determina la cantidad de recurso de inventario disponible para cada evento e_{mr}^k .

Para asegurarnos que dichas cantidades sean respetadas, se agrega al modelo por cada estación, un evento de inventario de destino de material rodante. Para una estación k dada, llamaremos a este evento $e_{mr_d}^k$ y a la cantidad de recurso necesario $i_{e_{mr_d}^k}$. Luego, cada tren que termine en una estación k determinada, puede dejar su recurso de material rodante para que sea utilizado posteriormente a lim_E^+ . Para esto, incluimos actividades acordadas hacia el evento de inventario de destino:

$$A_{mr}^-(e_{mr_d}^k) = \{a = (e, e_{mr_d}^k) : e \in E_{fin}^k\} \quad \forall k \in S$$

Es importante notar que el uso de estas actividades es opcional. Cada tren que termine su recorrido en una estación puede dejar el recurso para uso posterior a lim_E^+ o bien traspasar el recurso a otro tren. También debemos aclarar que para el armado de $A_{mr}^-(e_{mr_d}^k)$, además de las actividades desde los eventos de fin de tren, se incluyen actividades desde arribos cuyas posteriores salidas no se encuentren en E . Como no se incluyen los siguientes eventos de dichos trenes (son trenes contenidos parcialmente en el modelo), su material rodante queda en la estación del último arribo considerado.

Una vez agregados estos eventos y actividades, el modelo debe forzar que las cantidades necesarias de material rodante en el instante lim_E^+ se respeten. Para lograr esto agregamos el conjunto de desigualdades (3.15):

$$\sum_{a \in A_{mr}^-(e_{mr_d}^k)} z_a + (i_{e_{mr}^k} - \sum_{a \in A_{mr}^+(e_{mr}^k)} z_a) \geq i_{e_{mr_d}^k} \quad \forall k \in S \quad (3.15)$$

De esta manera, para una estación k dada, si contamos la cantidad de material rodante que hay al inicio del bloqueo, y a esto le restamos el material rodante usado desde k y le sumamos el dejado en k , debemos quedarnos con al menos la misma cantidad que hubiese en caso de no tener el bloqueo.

Es importante notar que en caso de cancelar un tren que esté programado para dejar material rodante en la estación k , al no contar con el recurso para uso posterior a lim_E^+ , es posible que esto fuerce a que se tenga que cancelar otro tren que esté programado con comienzo en k , para contar con dicho recurso disponible.

Por último, dado que los eventos que se encuentran entre τ_3 y lim_E^+ no pueden ser retrasados, sabemos que todas las actividades que alimenten a $e_{mr_d}^k$ liberarán el recurso de material rodante a lo sumo en lim_E^+ . Por este motivo, agregando las desigualdades (3.15) podemos asegurar que en el instante lim_E^+ la cantidad de material rodante necesaria estará disponible.

3.5.4. Consideraciones para los trenes andando en lim_E^-

Algunos trenes del cronograma original se encuentran andando en el instante lim_E^- . Al limitar la cantidad de eventos a incluir en E , sucede entonces que para dichos trenes solo una parte de sus eventos están incluidos. El traspaso de recursos explicado en las secciones anteriores no considera en detalle estos casos, para los cuales se deben tener en cuenta algunas consideraciones para modelarlos.

Arribos sin salida previa incluida: conflicto vías de tramo

Sea (k, l) la sección de vías entre las estaciones k y l , con una capacidad de v vías. Llamemos e_{tr}^{kl} al evento de inventario relacionado, tenemos que $i_{e_{tr}^{kl}} = v$. Supongamos un

tren t para el cual el primero de sus eventos incluidos en E es un arribo en la estación l desde la estación k , llamémosle arr_t^{kl} . Luego, su salida previa desde la estación k no está incluida, llamamos a dicha salida sal_t^{kl} .

Como se detalla en la Sección 3.4.1, el traspaso del recurso de vía de tramo es implícito entre una salida y su posterior arribo. De esta manera, el arribo arr_t^{kl} no necesita un recurso de dicho tipo para ocurrir, dado que $A_{tr}^-(arr_t^{kl}) = \emptyset$. Sin embargo, sabemos también por 3.4.1 que en $A_{tr}^+(\cdot)$ de un arribo se incluyen arribos en el mismo sentido y salidas en sentido contrario. Luego, arr_t^{kl} no toma recurso de vía de tramo, pero si libera. Para el caso de la sección (k, l) tendremos entonces $v + 1$ recursos disponibles (v de inventario más 1 del arribo) cuando en realidad solo deberíamos tener v .

Una alternativa para solucionar este problema es conectar el evento de inventario e_{tr}^{kl} con el arribo arr_t^{kl} , es decir $A_{tr}^-(arr_t^{kl}) = \{e_{tr}^{kl}, arr_t^{kl}\}$. Al estar e_{tr}^{kl} incluido en $A_{tr}^-(arr_t^{kl})$, el evento arr_t^{kl} debe tomar obligatoriamente el recurso de inventario, y así estaríamos simulando el hecho de que el evento arr_t^{kl} ya tiene el recurso tomado al momento lim_E^- .

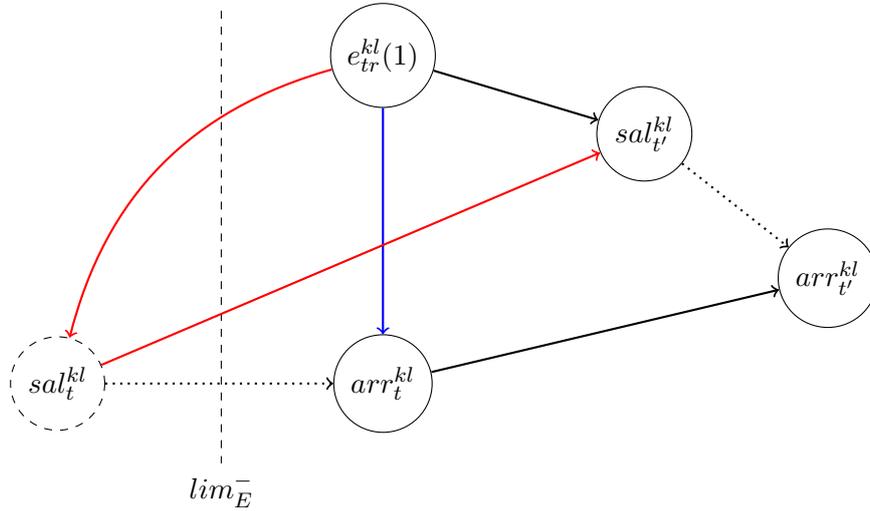


Fig. 3.14: Arribo sin salida previa incluida. Los eventos de tren se organizan horizontalmente de acuerdo a su horario original. Las aristas rojas muestran cómo sería el traspaso de recursos en caso de que el evento sal_t^{kl} estuviera incluido en E . Las aristas punteadas representan el traspaso implícito del recurso de vía de tramo desde una salida a su posterior arribo. La actividad especial agregada se muestra en azul.

Para analizar la problemática de esta solución, vemos un caso simple con $v = 1$. La Figura 3.14 muestra el traspaso del recurso de vías de tramo para este caso, donde además del tren t se agrega un tren t' que circula en la misma dirección y para el cual tanto la salida como el arribo entre las estaciones k y l están incluidos en E . Como el tren t se encuentra andando, el evento arr_t^{kl} toma obligatoriamente el recurso de inventario. Además, arr_t^{kl} no se conecta con $sal_{t'}^{kl}$, porque tal como se describe en la Sección 3.4.1, los arribos no se conectan con salidas en el mismo sentido. Luego, $sal_{t'}^{kl}$ no puede obtener el recurso necesario, recurso que tomaría de sal_t^{kl} si este último estuviera incluido (ilustrado a partir del eje rojo en la Figura 3.14).

Una opción para solucionar este problema sería no solo modificar $A_{tr}^-(arr_t^{kl})$, sino

también $A_{tr}^+(arr_t^{kl})$, definiendo en particular $A_{tr}^+(arr_t^{kl}) = A_{tr}^+(sal_t^{kl})$. En nuestro caso implicaría agregar la actividad marcada en rojo desde sal_t^{kl} hacia $sal_{t'}^{kl}$ también desde arr_t^{kl} . Más allá de la complejidad de considerar ese escenario, se mantienen otros problemas. Por ejemplo, la salida del tren t' podría estar programada antes que el arribo del tren t , y retrasarla podría ser infactible. Esta problemática surge por el hecho de que para el caso de vías de tramos, un mismo recurso puede ser utilizado al mismo tiempo por más de un tren.

Para resolver el problema ilustrado para este recurso, evitamos directamente esta situación. Para todos los arribos incluidos en E cuya salida previa no se encuentra incluida, lo que hacemos es incluir dicha salida previa en E . Evitamos así el problema para el recurso de vías de tramo, ya que las salidas siempre obtienen dicho recurso de manera explícita a través de una actividad.

Salidas sin arribo previo incluido: conflicto vías de estación

Un problema similar al anterior ocurre para el recurso de vías de estación en el caso de tener un tren para el cual una de sus salidas está incluida en E pero su arribo previo no. Como se detalla en la Sección 3.4.2, el traspaso del recurso es implícito entre un arribo y su posterior salida.

A diferencia que para el recurso de vías de tramos, aquí si podemos solucionar el problema agregando algunas actividades al subgrafo de vías de estación.

Sea k una estación con v cantidad de vías. Sea e_{est}^k el evento de inventario relacionado, i.e. $i_{e_{est}^k} = v$. Supongamos un tren t para el cual su primer evento incluido en E es una salida de la estación k , llamémosle sal_t^k . Como se define en 3.4.2, $A_{est}^-(sal_t^k) = \emptyset$. Similarmente a lo detallado para vías de tramos, si incluimos sal_t^k en E sin tener ninguna consideración especial, como sal_t^k libera un recurso, $v + 1$ recursos estarán disponibles cuando en realidad deberían ser v .

Veamos por qué en este caso incluir la actividad $a = (e_{est}^k, sal_t^k)$ en $A_{est}^-(sal_t^k)$ si resuelve el problema.

La Figura 3.15 muestra el traspaso del recurso de vías de estación para el caso mencionado. Además de los eventos del tren t , se incluyen otros arribos a la estación k .

Como t se encuentra andando, el evento sal_t^k toma obligatoriamente el recurso de inventario. Luego, para el resto de los eventos incluidos en E , quedan $v - 1$ recursos disponibles. Esto es exactamente lo mismo que hubiese sucedido en caso de que el evento arr_t^k estuviera incluido, así es como finalmente queda resuelto el problema. En la Figura 3.15 por simplicidad no se muestran otras salidas, ya que como se explicó, las mismas toman su recurso implícitamente de su arribo anterior. En caso de que hubieran otras salidas cuyo arribo previo no estuviera incluido, se procede de la misma manera, agregando una actividad desde el evento de inventario.

Es importante notar que la problemática del recurso de vías de tramo no se presenta en este caso dado que para las vías dentro de una estación, un mismo recurso no puede ser utilizado por más de un tren al mismo tiempo.

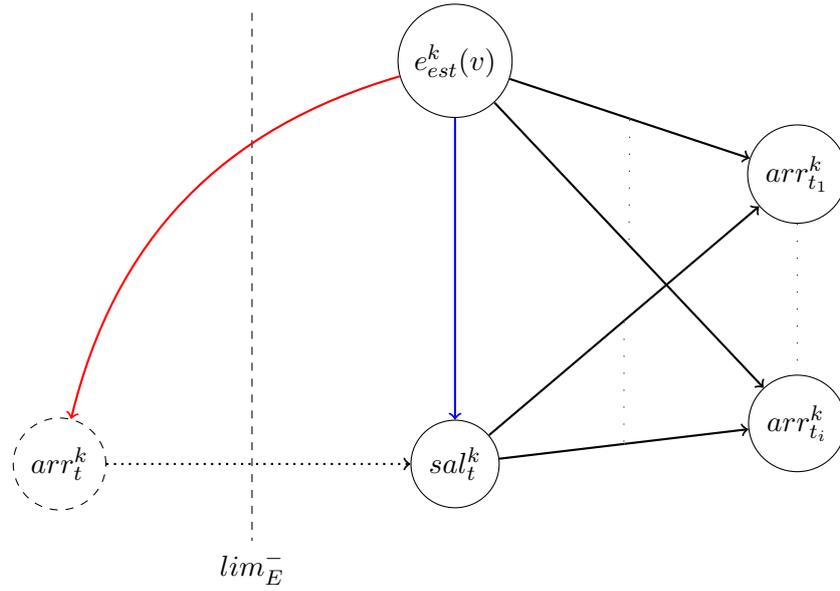


Fig. 3.15: Salida sin arribo previo incluido. Los eventos de tren se organizan horizontalmente de acuerdo a su horario original. La arista roja representa cómo sería el traspaso del recurso en caso de que arr_t^k estuviera incluido en E . Además la arista punteada representa el traspaso implícito de vía de estación de un arribo a su salida posterior. La actividad especial agregada se muestra en azul. Al no tener incluido el arribo arr_t^k , lo que se hace es simular el traspaso del recurso a su salida posterior sal_t^k .

3.5.5. Disponibilidad de vías de tramo en la sección del bloqueo

El manejo del recurso de vías de tramo debe tener especial atención entre las estaciones del bloqueo. Como se detalla en 3.5.2, en E no sólo se incluyen eventos programados durante el horario de bloqueo sino también por fuera del mismo. Por este motivo, hay que modelar el hecho de que las vías bloqueadas solo se encuentran disponibles antes y después del bloqueo.

Sea (k, l) la sección de vías entre las estaciones k y l donde ocurre el bloqueo. Supongamos que en dicha sección hay originalmente v vías entre las estaciones, y que el bloqueo produce que b vías queden inutilizadas entre τ_1 y τ_2 , i.e. $b \leq v$. Para el evento de inventario de vías de tramo no definimos la capacidad completa sino solo las vías que siempre están libres, es decir $i_{e_{tr}^{kl}} = v - b$. Adicionalmente, se incluye un evento de inventario especial, que habilita el uso de las b vías restantes una vez finalizado el bloqueo. Llamaremos a este evento $e_{tr}^{\tau_2+}$. Luego, $i_{e_{tr}^{\tau_2+}} = b$. Notar que este evento es único en toda la red N , con lo cual podemos referirnos al mismo de manera general, sin mencionar las estaciones.

El conjunto $A_{tr}^+(e_{tr}^{kl})$ se define normalmente, ya que las vías no bloqueadas pueden ser usadas en cualquier momento. En cambio, el evento $e_{tr}^{\tau_2+}$ solo tendrá actividades de salida hacia eventos que puedan programarse después de τ_2 , ya que dichas vías se liberan una vez terminado el bloqueo. Notar que en caso de bloqueo completo ($b = v$) solo se podrán utilizar las vías de la sección una vez terminado el bloqueo.

Como se detalla en 3.4.1, para una sección de vías (k, l) , e_{tr}^{kl} tiene actividades de salida

hacia las salidas de la sección (k, l) . $A_{tr}^+(e_{tr}^{\tau_2+})$ se define entonces de la misma manera, con la diferencia de que dichas salidas deben poder ser programadas después de τ_2 . Luego $A_{tr}^+(e_{tr}^{\tau_2+}) = \{e \in E_{sal}^{kl} \cup E_{sal}^{lk} : q_e + d_e \geq \tau_2\}$.

En el caso de que una salida originalmente programada antes de τ_2 obtenga su recurso a través de $e_{tr}^{\tau_2+}$, se debe forzar a que la salida sea programada posteriormente a τ_2 . Por dicho motivo se define el conjunto de desigualdades (3.16):

$$x_e \geq \tau_2 z_a \quad \forall a = (e_{tr}^{\tau_2+}, e) \in A_{tr}^+(e_{tr}^{\tau_2+}) \quad (3.16)$$

Manejo especial para eventos previos a τ_1

Al modelar la capacidad del recurso de vías de tramo con un evento de inventario con $v - b$ vías, que pueden ser utilizadas a partir de τ_1 y un evento de inventario de b vías, que libera las restantes a partir de τ_2 , no estamos considerando los eventos incluidos en E con $q_e < \tau_1$. Para dichos eventos, como todavía el bloqueo no comenzó, todas las v vías deben estar disponibles.

A diferencia de la solución implementada para los eventos posteriores a τ_2 , donde agregando un evento de inventario se liberan nuevas vías a partir de ese instante, aquí debería considerarse primero que el total de v vías esté disponible y luego a partir de τ_1 restringirlas a $v - b$. El problema que surge es que el modelo es fácilmente adaptable para el caso de agregar recursos a partir de un determinado instante, pero no lo es para quitar los mismos.

Sin embargo, la ventaja que se tiene para los eventos donde todas las v vías están disponibles, es que al estar programados previo al inicio del bloqueo, sabemos que pueden traspasarse los recursos sin ningún retraso, ya que la programación original es correcta. Aprovechando esto, una opción es no incluir en el modelo el traspaso del recurso de vías de tramo para dichos eventos. Esto se logra simplemente no incluyendo dichos eventos en el subgrafo asociado a este recurso.

Siendo más específicos, sea $E_{tren}^{kl} = E_{sal}^{kl} \cup E_{sal}^{lk} \cup E_{arr}^{kl} \cup E_{arr}^{lk}$ el conjunto de todos los eventos de tren involucrados en las estaciones del bloqueo, tendremos que:

$$A_{tr}^-(e) = A_{tr}^+(e) = \emptyset \quad \forall e \in E_{tren}^{kl}, q_e < \tau_1$$

El problema de este enfoque inicial es que al no incluir ninguno de los eventos previos al comienzo del bloqueo en el grafo de vías de tramo, $A_{tr}^+(e_{tr}^{kl})$ quedaría definido solo hacia salidas posteriores a τ_1 , y las $v - b$ vías estarían disponibles en el instante en que comienza el bloqueo. Es una situación similar a la planteada al estudiar el límite inferior de E , por la cual terminamos utilizando $\lim_{\bar{E}}$ en lugar de τ_1 .

Dependiendo de la programación original, esto podría modelar incorrectamente la disponibilidad de dichos recursos. Para modelarlo correctamente, se deben tener en cuenta algunos casos particulares de eventos programados cerca de τ_1 . Los mismos se detallan en la siguiente sección.

Consideraciones para eventos programados en el inicio del bloqueo

Supongamos que el bloqueo ocurre en la sección (k, l) de v vías y que un tren t tiene una salida desde la estación k (notada sal) programada antes del inicio del bloqueo, pero cuyo arribo posterior a l (notado arr) está programado después del inicio del mismo. Para este caso, no está claro si el tren t es afectado por el bloqueo o no. Ese nivel de detalle no es capturado por el modelo, por lo que asumimos que el tren sigue por su ruta original como si no fuera alcanzado por el bloqueo. Vale aclarar que en Veelenturf et al. [12], los autores hacen esta misma suposición.

En la sección anterior se detalla que $A_{tr}^-(sal) = A_{tr}^+(sal) = \emptyset$. Como sal no toma ningún recurso, una opción sería tampoco incluir a arr en el grafo de vías de tramo. El problema de esto es que arr podría estar programado en un tiempo muy posterior a τ_1 (el tramo entre sal y arr es muy largo), y si ambos eventos no se incluyen en el grafo, el recurso que utilizan se liberaría antes de lo previsto ya que estaría disponible en el instante τ_1 desde inventario. Luego, debemos definir de una manera diferente cómo dichos eventos toman y liberan recursos de vías de un tramo.

Supongamos que se tienen n trenes programados de la manera detallada arriba, es decir, si sal_i es la salida de la estación k del tren i y arr_i es el arribo programado a l del tren i , tendremos que $q_{sal_i} < \tau_1 < q_{arr_i}$. Ordenamos los n trenes según su horario de salida, i.e. $q_{sal_1} \leq q_{sal_2} \leq \dots \leq q_{sal_n}$. Vale aclarar que por simplicidad estamos asumiendo que los n trenes circulan en la misma dirección (salida desde k hacia l). El análisis es análogo en caso de tener trenes en ambas direcciones.

Como la sección posee v vías, en caso el caso de que $n > v$, asumimos que los primeros $n - v$ trenes traspasan sus recursos sin problemas hacia los eventos subsiguientes. Luego, las primeras $n - v$ salidas no quedan incluidas en el grafo de vías de tramo y en consecuencia, tampoco se incluyen los primeros $n - v$ arribos correspondientes a dichas salidas.

Sean $sal_1 \dots sal_v$ las v salidas restantes también ordenadas por tiempo (i.e. $q_{sal_1} \leq q_{sal_2} \leq \dots \leq q_{sal_v}$). Para dichas salidas, no está claro si las mismas toman vías bloqueadas o no, dado que el modelo tampoco captura ese nivel de detalle. Idealmente, esta información debería estar disponible como parte de los datos obtenidos del estado de la red.

Lo que haremos nosotros es asumir el peor caso, el cual estará dado por el caso en que la menor cantidad de vías quedan disponibles en el instante τ_1 . Para esto, asumimos que las últimas $v - b$ salidas circulan por las vías que no quedan bloqueadas. De esta manera, los recursos que utilizan quedarán liberados una vez que sus arribos correspondientes lleguen a destino, cuyo tiempo, como se explicó anteriormente, podría ser bastante posterior al instante τ_1 . Para las primeras b salidas restantes, asumimos que las mismas toman vías a punto de quedar bloqueadas, con lo cual quedarán liberadas una vez finalizado el bloqueo.

Luego, alimentamos a las últimas $v - b$ salidas del evento de inventario e_{tr}^{kl} e incluimos a sus correspondientes arribos normalmente al grafo de vías de tramo. Por otro lado, como asumimos que las primeras b salidas toman vías bloqueadas, sus arribos deberían liberar recursos únicamente para eventos posteriores a τ_2 . Como el evento de inventario $e_{tr}^{\tau_2+}$ libera recursos para eventos posteriores a τ_2 , dejamos que dichos recursos se liberen siempre mediante dicho evento, por lo que tampoco incluiremos las primeras b salidas ni sus respectivos arribos al grafo de vías de tramo.

La Figura 3.16 ilustra el manejo del recurso para eventos de este tipo. Para simplificar el entendimiento, mostramos un ejemplo particular con 5 trenes en esta situación particular

($n = 5$) y una sección con 4 vías ($v = 4$) de las cuales 2 quedan bloqueadas ($b = 2$).

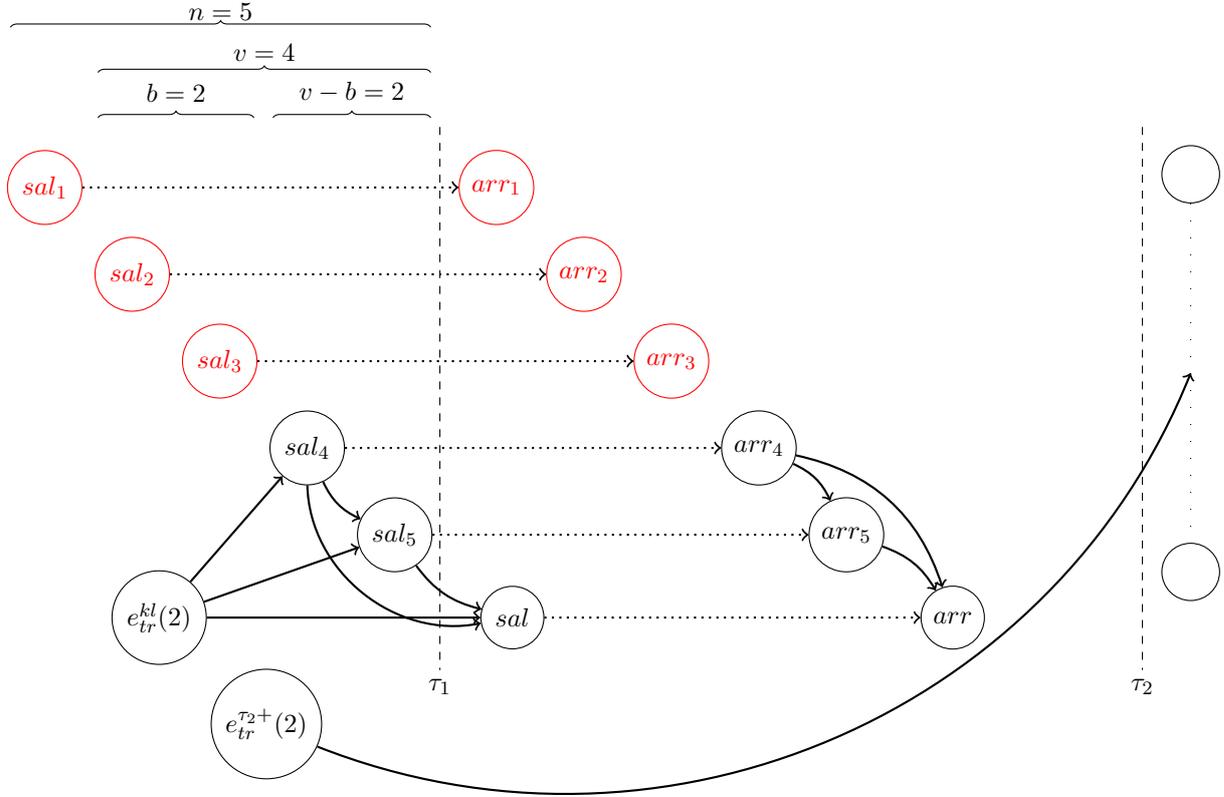


Fig. 3.16: Los eventos de tren se organizan horizontalmente de acuerdo a su horario original, de menor a mayor. Los eventos que quedan por fuera del grafo se muestran en rojo. Las actividades punteadas representan el traspaso implícito del recurso de vías de tramo.

Como se puede ver en la Figura 3.16, sal_1 y arr_1 quedan afuera del grafo dado que no están incluidos en los últimos v trenes. De manera similar, $sal_2, sal_3, arr_2, arr_3$ quedan afuera del grafo dado que son los primeros b trenes de los v restantes. Por último, $sal_4, sal_5, arr_4, arr_5$ son incluidos normalmente al ser los últimos $v - b$ trenes.

Los eventos sal y arr ejemplifican eventos normales que están programados posteriormente a τ_1 . Las vías bloqueadas que asumimos toman los eventos sal_2, sal_3 quedan liberadas a partir de τ_2 mediante el evento de inventario $e_{tr}^{\tau_2^+}$.

Es importante notar que las últimas $v - b$ salidas no pueden ser retrasadas ni canceladas por estar programadas antes de τ_1 . En el caso de que haya menos de $v - b$ trenes que cumplan estas condiciones (i.e. salida previa a τ_1 y arribo posterior) se procede de la misma manera, quedando solo dichos trenes alimentados del evento de inventario e_{tr}^{kl} .

3.6. Formulación Orig

A continuación se presenta la formulación de PLE completa. A partir de este momento, nos referiremos a la misma como *Orig*, dado que representa la formulación original en la cual basamos nuestro trabajo.

$$\text{Min } \sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{tren}} \mu_e (x_e - q_e) \quad (3.1)$$

$$\text{s.a. } x_e - q_e \geq 0 \quad \forall e \in E_{tren} \quad (3.2)$$

$$x_e - q_e \leq (1 - y_{t_e}) d_e \quad \forall e \in E_{tren} \quad (3.3)$$

$$x_f - x_e \geq L_a \quad \forall a = (e, f) \in A_{tren} \quad (3.4)$$

$$y_t \in \{0, 1\}, x_e \in \mathbb{N} \quad \forall t \in T, \forall e \in E_{tren} \quad (3.5)$$

$$\sum_{a \in C} z_a + y_{t_e} = 1 \quad \forall e \in E_{tren}, C \in A^-(e), C \neq \emptyset \quad (3.6)$$

$$\sum_{a \in C} z_a + y_{t_e} \leq 1 \quad \forall e \in E_{tren}, C \in A^+(e) \quad (3.7)$$

$$\sum_{a \in C} z_a \leq i_e \quad \forall e \in E_{inv}, C \in A^+(e) \quad (3.8)$$

$$x_f - x_e + M_a(1 - z_a) \geq L_a \quad \forall a = (e, f) \in A, e, f \in E_{tren} \quad (3.9)$$

$$z_a \in \{0, 1\} \quad \forall a \in A \quad (3.10)$$

$$z_a = z_{a'} \quad \forall (a, a') \in B_{tr} \quad (3.11)$$

$$z_a \leq z_{a'} \quad \forall (a, a') \in B_{mr}^{est} \quad (3.12)$$

$$y_{\beta_t} \geq y_{\alpha_t} \quad \forall t \in T^{blq} \quad (3.13)$$

$$y_{\beta_t} \geq y_{\gamma_t} \quad \forall t \in T^{blq} \quad (3.14)$$

$$\sum_{a \in A_{mr}^-(e_{mr_d}^k)} z_a + (i_{e_{mr}^k} - \sum_{a \in A_{mr}^+(e_{mr_d}^k)} z_a) \geq i_{e_{mr_d}^k} \quad \forall k \in S \quad (3.15)$$

$$x_e \geq \tau_2 z_a \quad \forall a = (e_{tr}^{\tau_2^+}, e) \in A_{tr}^+(e_{tr}^{\tau_2^+}) \quad (3.16)$$

4. Formulación alternativa

En el presente capítulo presentamos una *Formulación por Índice de Tiempo* (TIF, por sus siglas en inglés) para el problema de Reprogramación de cronograma.

En la práctica, resolver instancias reales para este tipo de problemas suele ser demandante en términos de los tiempos de cómputo. En general, dado que nuestro modelo debe brindar una solución en tiempo real, esto restringe el tamaño de los problemas que se pueden considerar. Dadas estas limitaciones, en muchas ocasiones una opción es, por ejemplo, utilizar solo una parte de la red ferroviaria, cercana al área del bloqueo. Esto puede tener impactos significativos en la calidad de la solución, dado que al no contemplar la totalidad de la red, la solución óptima para una porción de la misma puede ser muy distante de la solución óptima general.

Bajo este contexto, nos interesa analizar si es posible mejorar los resultados de la formulación Orig estudiada en detalle en el Capítulo 3. Esta es la pregunta de investigación fundamental de nuestro trabajo, sobre la cual incluimos un estudio teórico en el presente capítulo. En el Capítulo 5 de experimentación y resultados realizamos un estudio empírico comparando ambas formulaciones.

La estrategia principal para lograr nuestro objetivo consiste en desarrollar una formulación de PLE alternativa que posea una relajación PL más ajustada. Para obtenerla, modelamos las restricciones de tiempo entre los eventos sin utilizar desigualdades *big-M*, las cuales se encuentran presentes en Orig y son conocidas por producir relajaciones PL débiles en general. Para cumplir este propósito, extendemos el enfoque planteado en Dash et al. [4] para el TSPTW para nuestro problema de Reprogramación de cronograma. En nuestro caso, a diferencia del problema mencionado, los eventos no deben ser visitados obligatoriamente (los trenes pueden cancelarse), cada evento puede utilizar más de un recurso (y por lo tanto surgen diferentes restricciones de tiempo por cada uno) y adicionalmente se deben modelar restricciones particulares para cada tipo de recurso.

Es importante mencionar que también se consideraron otras opciones adicionales como estrategias para mejorar los resultados de Orig. En particular, nuestra nueva formulación TIF provee una estructura mucho más rica la cual permite analizar distintas opciones de preprocesamiento y enfocar el estudio de desigualdades válidas para ser incorporadas al algoritmo *Branch-and-Cut*. Por otro lado, creemos que el estudio de heurísticas para mejorar las cotas primales también puede mejorar considerablemente los resultados. Sobre estas cuestiones, incluimos en el Capítulo 6 de conclusiones y trabajo futuro una explicación más detallada.

El capítulo se organiza de la siguiente manera. En la Sección 4.1 se detallan los nuevos conjuntos de variables introducidos, y las adaptaciones y extensiones realizadas sobre la formulación Orig utilizando las nuevas variables. En la Sección 4.2 se incluye la formulación TIF completa. Luego, en la Sección 4.3 se incluye la demostración del principal resultado teórico de nuestro trabajo, i.e., que TIF posee una mejor relajación lineal que Orig. Finalmente, en la Sección 4.4 mostramos como el conjunto de soluciones óptimas de nuestra formulación TIF es equivalente al de Orig.

4.1. Nuevos conjuntos de variables

Antes de introducir los nuevos conjuntos de variables, primero extendemos la notación introducida en las secciones previas, basada en su mayoría en el trabajo de Veelenturf et al. [12].

Sea $R = \{\text{tr, est, mr}\}$ el conjunto de los tipos de recursos, es decir, vías de tramos, vías de estaciones y material rodante. Notamos como $R^-(e), R^+(e) \subseteq R$ a los tipos de recursos potencialmente requeridos o liberados por el evento $e \in E_{tren}$ respectivamente. Recordemos que algunas transferencias de recursos son implícitas, y por lo tanto estos dos conjuntos pueden ser diferentes para un evento dado. Por ejemplo, un arribo requiere el recurso de vía de estación pero no libera dicho recurso, su liberación es implícita hacia su salida próxima.

Dado un tipo de recurso $r \in R^-(e)$, podemos reemplazar la notación $C \in A^-(e)$ utilizada en las secciones previas como $A_r^-(e) \in A^-(e)$. Una definición análoga es considerada para los conjuntos $R^+(e)$ y $A^+(e)$. Esta nueva notación permite explicitar los recursos de los subconjuntos de actividades, característica que utilizamos para definir la formulación TIF.

Aún más, tal como vimos en la Subsección 3.4.5, dada una actividad $a = (e, f)$ relativa a un recurso r , puede suceder que $a \in A_r^+(e)$ pero al mismo tiempo $a \notin A_r^-(f)$. En base a esto, definimos el conjunto $A_r^{hacia}(f)$ como

$$A_r^{hacia}(f) = A_r^-(f) \cup \{a = (e, f) \in A_r^+(e) : e \in E\}$$

A partir de este conjunto podemos determinar si un recurso r pertenece a $R^-(f)$, i.e., si sucede que $A_r^{hacia}(f) \neq \emptyset$. De esta manera, consideramos que r es un recurso potencialmente requerido por un evento f si posee al menos una actividad en su conjunto de entrada o si es alimentado a través del conjunto de salida de otro evento e . Análogamente definimos

$$A_r^{desde}(e) = A_r^+(e) \cup \{a = (e, f) \in A_r^-(f) : f \in E\}$$

Y luego $r \in R^+(e)$ sii $A_r^{desde}(e) \neq \emptyset$. Vale aclarar que en la Subsección 3.4.5 también observamos que dada una actividad $a = (e, f)$, si sucede que $a \in A_r^-(f)$, entonces $a \in A_r^+(e)$. Luego, el conjunto $A_r^{desde}(e)$ es en realidad equivalente a $A_r^+(e)$.

Dado un evento $e \in E_{tren}$, sabemos que sus posibles horarios en el nuevo cronograma están limitados por su horario original q_e y su máximo retraso d_e . Llamaremos *ventana de tiempo* de e a los distintos instantes de tiempo en los cuales el evento e puede ser programado en el nuevo cronograma, y la notaremos W_e , i.e., $W_e = \{t : q_e \leq t \leq q_e + d_e\}$.

Dados dos eventos $e, f \in E_{tren}$ y un tipo de recurso $r \in R^-(e)$ tal que $a = (f, e) \in A_r^{hacia}(e)$, para cada $t \in W_e$ definimos $I_f^e(t, r)$ como el conjunto de los posibles instantes de tiempo en los cuales el evento f puede suceder, si al transferirle a e el recurso r , éste queda disponible exactamente en el instante t . Formalmente, definimos

$$I_f^e(t, r) = \{\tau \in W_f : a = (f, e) \in A_r^{hacia}(e), \text{ máx}\{\tau + L_a, q_e\} = t\}. \quad (4.1)$$

Estos conjuntos son análogos a los definidos en Dash et al. [4] para derivar la TIF para el TSPTW. Notar algunas propiedades sobre el conjunto $I_f^e(t, r)$:

- Dados $t, t' \in W_e, t \neq t' \Rightarrow I_f^e(t, r)$ e $I_f^e(t', r)$ conforman conjuntos disjuntos

- $I_f^e(q_e, r) = \{\tau \in W_f : a = (f, e) \in A_r^{hacia}(e), \tau + L_a \leq q_e\}$
- Si $t > q_e \Rightarrow |I_f^e(t, r)| \leq 1$
- Si $q_f + L_a > q_e + d_e \Rightarrow I_f^e(t, r) = \emptyset \forall t \in W_e$

A continuación, detallamos los tres nuevos conjuntos de variables introducidos para desarrollar la formulación de PLE alternativa.

4.1.1. Binarización de variables de tiempo de eventos

Recordemos que estamos considerando un horizonte de tiempo discreto, por lo tanto, podemos identificar los posibles instantes de tiempo en los cuales los eventos pueden ocurrir.

En base a esto, definimos x_e^t como una variable binaria que toma valor 1 si el evento $e \in E_{tren}$ es programado en el instante $t \in W_e$. Formalmente, las variables enteras x_e originales que indican el tiempo para el evento $e \in E_{tren}$, pueden reescribirse de la siguiente manera:

$$x_e = \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau \quad (4.2)$$

Para que las nuevas variables x_e^t modelen correctamente el tiempo del evento e en el nuevo cronograma, debemos asegurar que e sea programado en un único instante de tiempo. Además, como dijimos, las nuevas variables son binarias. Luego, incorporamos a TIF las siguientes igualdades relativas a las variables x_e^t :

$$\sum_{t=q_e}^{q_e+d_e} x_e^t = 1 \quad \forall e \in E_{tren} \quad (4.3)$$

$$x_e^t \in \{0, 1\} \quad \forall e \in E_{tren}, t \in W_e \quad (4.4)$$

En nuestra nueva formulación TIF podemos prescindir de las variables x_e originales. Para cada evento $e \in E_{tren}$, en lugar de trabajar con una única variable entera que cumple $q_e \leq x_e \leq q_e + d_e$, utilizamos $d_e + 1$ variables binarias x_e^t .

Las variables de tiempos de eventos x_e aparecen en la formulación Orig en la función objetivo:

$$\sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{tren}} \mu_e (x_e - q_e)$$

y en las desigualdades:

$$x_f - x_e \geq L_a \quad \forall a = (e, f) \in A_{tren} \quad (3.4)$$

$$x_e \geq \tau_2 z_a \quad \forall a = (e_{tr}^{\tau_2+}, e) \in A_{tr}^+(e_{tr}^{\tau_2+}) \quad (3.16)$$

Para eliminar las variables x_e de la formulación TIF, reescribimos la función objetivo utilizando la decomposición en instantes de tiempos (4.2):

$$\sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{tren}} \mu_e \left(\sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau - q_e \right)$$

Las desigualdades (3.4) se pueden reescribir como:

$$\sum_{t=t'}^{q_e+d_e} x_e^t \leq \sum_{t=\max(t'+L_a, q_f)}^{q_f+d_f} x_f^t \quad \forall a = (e, f) \in A_{tren}, t' \in W_e \quad (4.5)$$

Asimismo, podemos reescribir las desigualdades (3.16) como:

$$\sum_{t=q_e}^{\tau_2-1} x_e^t + z_a \leq 1 \quad \forall a = (e_{tr}^{\tau_2+}, e) \in A_{tr}^+(e_{tr}^{\tau_2+}) \quad (4.6)$$

Además de las restricciones mencionadas, las variables originales x_e también aparecen en la formulación Orig en las desigualdades (3.2), (3.3) y (3.9). Las desigualdades (3.9) son modeladas en Orig mediante una restricción *big-M*, eliminarlas comprende nuestro objetivo inicial, con lo cual a continuación veremos como reescribirlas. En cuanto a las desigualdades (3.2) y (3.3), podemos eliminar las mismas de la formulación TIF.

Debido a las nuevas desigualdades (4.3), vemos que ningún evento puede ser programado anteriormente a su horario original q_e . De esta manera, evitamos incluir en TIF las desigualdades (3.2).

En cuanto a las desigualdades (3.3), podemos separar el análisis en dos casos. Dado un evento $e \in E_{tren}$, si su tren asociado no es cancelado (i.e. $y_{t_e} = 0$), podemos reescribir a (3.3) como:

$$x_e \leq q_e + d_e \quad \forall e \in E_{tren}$$

Luego, nuevamente en base a las restricciones (4.3) sabemos que esto siempre se cumple. En caso de que el tren del evento e sea cancelado, podemos reescribir a (3.3) como:

$$x_e \leq q_e \quad \forall e \in E_{tren}$$

Al no incluir las restricciones (3.3) en TIF, podríamos obtener una solución para la cual dado un evento $e \in E_{tren}$ suceda que $y_{t_e} = 1$ y $x_e^t = 1, t > q_e$. Sin embargo, debemos notar que en este caso la solución será penalizada tanto por la cancelación del tren t_e como por el retraso del evento e . Se puede ver fácilmente que si $y_{t_e} = 1$, entonces no se impone ninguna restricción de tiempo sobre los eventos del tren t_e , más allá de los tiempos de viaje y espera relativos a las actividades de tren. Luego, dada una solución factible \tilde{x} de TIF, se pueden ajustar los tiempos de los eventos de t_e de manera independiente de los eventos restantes en la red. Como el horario original de cada evento se encuentra dentro de su ventana de tiempo, podemos armar una nueva solución \bar{x} en la cual asignamos $x_e^{q_e} = 1$

para cada evento e del tren t_e . De esta manera, el valor de la función objetivo para la nueva solución \bar{x} será mejor que el de la solución anterior \tilde{x} , y además \bar{x} cumple la desigualdad (3.3) de Orig.

En base a esto, eliminar la desigualdad (3.3) de TIF no afecta la equivalencia entre el conjunto de soluciones óptimas de ambas formulaciones.

4.1.2. Reescritura de variables de actividad por índice de tiempo

Utilizando una idea similar que para las variables de eventos, definimos las variables binarias z_a^t relativas a la actividad $a = (e, f) \in A_r^+(e)$, las cuales toman valor 1 si el evento e ocurre en el instante de tiempo $t \in W_e$ y transfiere el recurso r a f . La intuición es que la variable de actividad original, z_a , es decompuesta sobre los posibles tiempos de inicio del evento de origen e .

Para garantizar que las nuevas variables modelen correctamente esta relación entre las actividades y sus eventos de origen, agregamos a TIF las siguientes desigualdades:

$$\sum_{a \in A_r^+(e)} z_a^t \leq x_e^t \quad \forall e \in E_{tren}, r \in R^+(e), t \in W_e \quad (4.7)$$

Vemos como dado un evento $e \in E_{tren}$, si el mismo traspasa el recurso r mediante alguna de sus actividad de salida en el instante $t \in W_e$, se fuerza a que sea programado a partir de dicho instante. Notar como las desigualdades se agrupan por cada recurso $r \in R^+(e)$, esto es así dado que el evento e podría realizar el traspaso de más de un tipo de recurso.

En realidad, dada la semántica definida para la actividad z_a^t , el conjunto de desigualdades (4.7) debería ser definido como una igualdad. En ese caso, si sucede que $x_e^t = 1$, alguna de las actividades de salida de e debe activarse en el instante t . El problema detrás de esto es que dado un evento $e \in E_{tren}$, aunque el mismo posea un conjunto de actividades de salida no vacío para el recurso r (la desigualdad no se define en caso contrario), podría suceder que ninguna de esas actividades se utilice, es decir, que el evento e sea el último evento en utilizar la unidad correspondiente del recurso r . En ese caso, estaríamos forzando $x_e^t = 0 \forall t \in W_e$. Por otro lado, dado que la función objetivo penaliza cada unidad de retraso, en caso de que $z_a^t = 1$, el modelo fuerza a que e sea programado en el instante t y no después, que es lo que buscamos. Por dicho motivo, trabajamos en TIF con las desigualdades (4.7).

Creemos igualmente que plantear las desigualdades (4.7) como igualdades generaría mejores resultados, al disminuir el conjunto de soluciones factibles. Una manera de lograr esto, es agregando al modelo *eventos de destino de inventario*, los cuales modelarían el hecho de que un recurso deja de ser utilizado. De esta manera, en caso de que el evento e no traspase el recurso r a ningún otro evento de tren, realiza un traspaso a un evento de destino de inventario mediante una actividad incluida en $A_r^+(e)$, y de esa manera se logra el comportamiento buscado.

Para incluir estos eventos especiales deben tenerse en cuenta algunas cuestiones, a saber:

- Se deben incluir en la red evento-actividad las actividades correspondientes hacia los eventos de destino de inventario. Las mismas se deben agregar en los conjuntos $A_r^+(e)$ si el evento e traspasa unidades de tipo r .

- El modelo debe forzar a que, en caso de que un evento $e \in E_{tren}$ no traspase su unidad de recurso r a ningún otro evento de tren, las nuevas actividades sean utilizadas. La idea detrás de esto es modificar las desigualdades (3.7) por igualdades.
- En caso de definir (4.7) como una igualdad, habría que considerar el hecho de que el tren t_e del evento e puede ser cancelado, y en dicho caso sus actividades se anulan.

Dejamos esta opción planteada para trabajo a futuro como una posibilidad de mejora para el modelo TIF.

Debemos modelar correctamente la relación entre las nuevas variables z_a^t y las variables de actividad z_a originales. Además, al igual que x_e^t , las nuevas variables son binarias. Luego, incorporamos a TIF las siguientes igualdades:

$$z_a = \sum_{t=q_e}^{q_e+d_e} z_a^t \quad \forall a = (e, f) \in A, e \in E_{tren} \quad (4.8)$$

$$z_a^t \in \{0, 1\} \quad \forall a = (e, f) \in A, e \in E_{tren}, t \in W_e \quad (4.9)$$

Es importante notar que ambas igualdades se definen para las actividades $a = (e, f)$ tales que $e \in E_{tren}$. Esto se debe a que las nuevas variables z_a^t se refieren al instante de tiempo t de la ventana W_e . Para las actividades $a' = (e, f)$ tales que $e \in E_{inv}$, utilizamos únicamente las variables z_a originales.

En cuanto a la igualdad (4.8), notar como la misma refleja el hecho de que si la actividad a es utilizada, solo una variable z_a^t es activada para $t \in W_e$. Esto se condice con el hecho de que los eventos deben ser programados en un único instante de tiempo.

4.1.3. Variables de liberación de recurso

Dado que cada evento puede requerir más de un tipo de recurso, necesitamos tener en cuenta el tiempo en que cada recurso se encuentra listo para ser utilizado. Luego, definimos para cada $e \in E_{tren}$, $r \in R^-(e)$ y $t \in W_e$ las variables binarias $r w_e^t$ que toman valor 1 si en el instante de tiempo t el recurso r pasa a estar disponible para ser utilizado por el evento e .

Agregamos a TIF los siguientes conjuntos de desigualdades:

$$\sum_{t=q_e}^{q_e+d_e} r w_e^t + y_{t_e} = 1 \quad \forall e \in E_{tren}, r \in R^-(e) \quad (4.10)$$

$$\sum_{t' \geq t} r w_e^{t'} \leq \sum_{t' \geq t} x_e^{t'} \quad \forall e \in E_{tren}, r \in R^-(e), t \in W_e \quad (4.11)$$

$$r w_e^t \in \{0, 1\} \quad \forall e \in E_{tren}, r \in R^-(e), t \in W_e \quad (4.12)$$

Las desigualdades (4.10) modelan el hecho de que dado un evento $e \in E_{tren}$ y un recurso r potencialmente requerido, el mismo pasa a estar disponible en un único instante

de tiempo de la ventana de e . También se tiene en cuenta que si el tren del evento e es cancelado (i.e. $y_{t_e} = 1$) entonces ningún recurso puede estar disponible para uso por parte de e .

Las desigualdades (4.11) establecen que un evento e puede ser programado una vez que todos los recursos que requiere se encuentran disponibles.

Recordemos la definición del conjunto $I_f^e(r, t)$ vista en (4.1). En base a dicha definición y utilizando los nuevos conjuntos de variables ${}^r w_e^t$ podemos reescribir la desigualdad (3.9) del modelo Orig por:

$$\sum_{\substack{(f,e) \in A_r^{hacia}(e), \\ f \in E_{tren}}} \sum_{\tau \in I_f^e(r, t)} z_{(f,e)}^\tau = {}^r w_e^t \quad \forall e \in E_{tren}, r \in R^-(e), t \in (q_e, q_e + d_e] \quad (4.13)$$

Intuitivamente, si al evento $e \in E_{tren}$ se le libera el recurso r en el instante t , debe suceder que dicho recurso sea provisto por un evento f anterior. Dicho evento será programado en algún instante de su ventana de tiempo respetando el tiempo de seguridad de la actividad (f, e) (considerado en $I_f^e(r, t)$) dado que la variable $z_{(f,e)}^\tau$ fuerza el nuevo horario del evento f por las desigualdades (4.7). Luego, dado que el evento e debe ser programado una vez que posea sus recursos disponibles (desigualdades (4.11)), podemos concluir que en caso de que f le traspase una unidad de recurso r a e , sus horarios respetarán el tiempo de seguridad $L_{(f,e)}$ requerido, que es lo que se modela originalmente mediante las desigualdades (3.9).

Para terminar de modelar correctamente el traspaso de recursos, debemos tener en cuenta el hecho de que el evento e podría obtener su unidad de recurso r desde un evento de inventario y no desde otro evento de tren anterior. Para los casos en que un evento e obtenga su recurso de esta manera, consideraremos que el mismo se encuentra disponible desde un inicio, i.e. ${}^r w_e^{q_e} = 1$, dado que no hay restricciones de tiempo sobre los eventos de inventario. En base a esto, definimos una desigualdad similar a (4.13) pero solo para el instante de tiempo q_e . La desigualdad (4.13) solo considera los instantes de tiempo restantes de W_e .

$$\sum_{\substack{(f,e) \in A_r^{hacia}(e), \\ f \in E_{tren}}} \sum_{\tau \in I_f^e(r, q_e)} z_{(f,e)}^\tau + \sum_{\substack{(f,e) \in A_r^{hacia}(e), \\ f \in E_{inv}}} z_{(f,e)} = {}^r w_e^{q_e} \quad \forall e \in E_{tren}, r \in R^-(e) \quad (4.14)$$

Al tener en cuenta las actividades desde eventos $f \in E_{inv}$, utilizamos las variables originales $z_{(f,e)}$ dado que f no posee un horario en el cronograma.

Debemos aclarar que, tal como vimos en el Capítulo 3, en general existe un único $f \in E_{inv}$ tal que $a = (f, e) \in A_r^{hacia}(e)$, dado que para un recurso r existe un único evento de inventario que puede alimentar a e con una unidad de r . Sin embargo, como la disponibilidad de vías de tramos en la sección del bloqueo se modela con dos eventos de inventario distintos como vimos en la Sección 3.5.5, utilizamos una sumatoria en la desigualdad (4.14) y consideramos así todos los casos posibles.

En base a los nuevos conjuntos de variables introducidos y en particular, a partir las últimas desigualdades descriptas, eliminamos las desigualdades *big-M* incluidas en Orig mediante las restricciones (3.9).

4.2. Formulación TIF

A continuación incluimos la formulación TIF completa.

$$\text{Min } \sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{tren}} \mu_e \left(\sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau - q_e \right)$$

$$\text{s.a. (3.6) - (3.8), (3.11) - (3.15)}$$

$$\sum_{t=q_e}^{q_e+d_e} x_e^t = 1 \quad \forall e \in E_{tren} \quad (4.3)$$

$$x_e^t \in \{0, 1\} \quad \forall e \in E_{tren}, t \in W_e \quad (4.4)$$

$$\sum_{t=t'}^{q_e+d_e} x_e^t \leq \sum_{t=\max(t'+L_a, q_f)}^{q_f+d_f} x_f^t \quad \forall a = (e, f) \in A_{tren}, t' \in W_e \quad (4.5)$$

$$\sum_{t=q_e}^{\tau_2-1} x_e^t + z_a \leq 1 \quad \forall a = (e_{tr}^{\tau_2+}, e) \in A_{tr}^+(e_{tr}^{\tau_2+}) \quad (4.6)$$

$$\sum_{a \in A_{tr}^+(e)} z_a^t \leq x_e^t \quad \forall e \in E_{tren}, r \in R^+(e), t \in W_e \quad (4.7)$$

$$z_a = \sum_{t=q_e}^{q_e+d_e} z_a^t \quad \forall a = (e, f) \in A, e \in E_{tren} \quad (4.8)$$

$$z_a^t \in \{0, 1\} \quad \forall a = (e, f) \in A, e \in E_{tren}, t \in W_e \quad (4.9)$$

$$\sum_{t=q_e}^{q_e+d_e} r w_e^t + y_{te} = 1 \quad \forall e \in E_{tren}, r \in R^-(e) \quad (4.10)$$

$$\sum_{t' \geq t} r w_e^{t'} \leq \sum_{t' \geq t} x_e^{t'} \quad \forall e \in E_{tren}, r \in R^-(e), t \in W_e \quad (4.11)$$

$$r w_e^t \in \{0, 1\} \quad \forall e \in E_{tren}, r \in R^-(e), t \in W_e \quad (4.12)$$

$$\sum_{\substack{(f,e) \in A_{tr}^{hacia}(e), \\ f \in E_{tren}}} \sum_{\tau \in I_f^c(r,t)} z_{(f,e)}^\tau = r w_e^t \quad \forall e \in E_{tren}, r \in R^-(e), t \in (q_e, q_e + d_e] \quad (4.13)$$

$$\sum_{\substack{(f,e) \in A_{tr}^{hacia}(e), \\ f \in E_{tren}}} \sum_{\tau \in I_f^c(r,q_e)} z_{(f,e)}^\tau +$$

$$\sum_{\substack{(f,e) \in A_{tr}^{hacia}(e), \\ f \in E_{inv}}} z_{(f,e)} = r w_e^{q_e} \quad \forall e \in E_{tren}, r \in R^-(e) \quad (4.14)$$

Notar como algunas desigualdades incluidas en TIF provienen de la formulación Orig. Las mismas se corresponden a las desigualdades (3.6) - (3.8) que modelan el traspaso de recursos entre eventos, las desigualdades (3.11) y (3.12) que modelan situaciones particulares para cada tipo de recurso, y las desigualdades (3.13) - (3.15) necesarias para modelar correctamente el bloqueo.

4.3. Relajación lineal TIF vs Orig

Nos enfocamos ahora en la relación entre la relajación PL de Orig y TIF.

En una primera instancia, incluimos en TIF un conjunto de desigualdades que no son necesarias para que la formulación sea válida, dado que toda solución entera óptima las cumple. Sin embargo, su inclusión simplifica el análisis al comparar las relajaciones lineales de ambas formulaciones.

Dado un evento $e \in E_{tren}$, agregamos a TIF las siguientes desigualdades

$$y_{te} \leq x_e^{qe} \quad (4.15)$$

La formulación Orig cumple esta restricción a partir de las desigualdades (3.3). En la Sección 4.1.1 vimos como se eliminan de la formulación TIF las variables x_e utilizando en su lugar las variables x_e^t . En dicho análisis, vimos como cualquier solución factible de TIF que no cumpla (4.15) se puede modificar por una solución con mejor valor de la función objetivo y que cumpla las desigualdades (4.15).

En segundo lugar, asumimos que un preprocesamiento básico es aplicado tanto sobre la formulación Orig como sobre la formulación TIF con el objetivo de eliminar variables que no pueden ser factibles en el modelo.

Dada una actividad $a = (e, f)$ tal que $q_e + L_a > q_f + d_f$, asumimos que dicha actividad no es incluida en la red evento-actividad. Para la formulación TIF, también asumimos que si $\tau + L_a > q_f + d_f$, para $\tau \in W_e$ entonces $z_a^\tau = 0$. Intuitivamente, estamos eliminando las actividades $a = (e, f)$ que, en caso de estar activas, violarían el retraso máximo permitido para el evento f .

La inclusión del preprocesamiento mencionado simplifica el estudio teórico. En el Capítulo 5 de experimentación y resultados analizaremos su impacto en mayor detalle.

Sean \hat{z}_{orig} y \hat{z}_{tif} los valores de las relajaciones PL de las formulaciones Orig y TIF respectivamente. Sean también P_{Orig} y P_{TIF} los conjuntos de soluciones asociados con las formulaciones Orig y TIF, y P_{Orig}^{LP} y P_{TIF}^{LP} los politopos asociados con su correspondiente relajación PL. Recordemos que estamos considerando que TIF incluye las desigualdades (4.15) como parte de la formulación.

Demostraremos primero las siguientes observaciones, las cuales son utilizados luego durante la demostración principal.

Observación 1. Dadas $f \in E_{tren}$ y $r \in R^-(f)$, entonces

$$\sum_{\tau=q_f}^{q_f+d_f} (\tau - q_f) x_f^\tau \geq \sum_{\tau=q_f}^{q_f+d_f} (\tau - q_f)^\tau w_f^r$$

Demostración. La desigualdad se obtiene sumando las desigualdades (4.11) para $t = q_f + 1, \dots, q_f + d_f$. \square

Observación 2. Dado un evento $e \in E_{tren}$, un recurso $r \in R^-(e)$, un instante $t \in W_e$ y una actividad $a = (f, e) \in A_r^{hacia}(e)$ tal que $f \in E_{tren}$, la siguiente desigualdad es válida

$${}^r w_e^t = \sum_{\substack{(g,e) \in A_r^-(e) \\ g \in E_{tren}}} \sum_{\tau \in I_g^e(t,r)} z_{(g,e)}^\tau \geq \sum_{\tau \in I_f^e(t,r)} z_{(f,e)}^\tau$$

Demostración. Para $\tau \in (q_e, q_e + d_e]$ la desigualdad se obtiene considerando la restricción (4.13) para el evento e y el recurso r y restringiendo la sumatoria sólo sobre el evento f . Para $\tau = q_e$ la desigualdad se obtiene considerando la restricción (4.14) de una manera similar y además descartando la sumatoria sobre los eventos de inventario. \square

Observación 3. Sean $e, f \in E_{tren}$ y $r \in R^-(f)$ tal que $a = (e, f) \in A_r^{hacia}(f)$. Luego, la siguiente igualdad se mantiene

$$\sum_{t=q_f}^{q_f+d_f} \sum_{\tau \in I_e^f(t,r)} z_{(e,f)}^\tau = \sum_{\tau=q_e}^{q_e+d_e} z_{(e,f)}^\tau = z_a$$

Demostración. Recordemos que por definición, $\forall t \in W_f, I_e^f(t, r) \subseteq W_e$. Para cada instante de tiempo $t \in W_f$, se consideran los posibles tiempo de inicio $\tau \in I_e^f(t, r)$ para la actividad $a = (e, f)$, incluyendo aquellos en los que la transferencia del recurso termina antes del horario original del evento f , i.e. $\tau + L_a \leq q_f$. Además, dado que la expresión del lado izquierdo considera todos los instantes de tiempo de la ventana de f , si existe un instante de tiempo $\tau \in W_e$ el cual no pertenece a $I_e^f(t, r)$ para ningún $t \in W_f$, entonces se debe a que desde τ no es posible respetar el retraso máximo permitido del evento f , i.e. $\tau + L_a > q_f + d_f$. Pero recordemos que para estos casos estamos asumiendo el preprocesamiento $z_a^\tau = 0$. Por consiguiente, podemos incluir dichos términos en la sumatoria derecha y la igualdad se mantiene. Finalmente, podemos reemplazar esta sumatoria por la variable z_a en base a las igualdades (4.8). \square

A continuación enunciamos el resultado principal respecto a las formulaciones Orig y TIF.

Proposición 1. $\hat{z}_{orig} \leq \hat{z}_{tif}$

Demostración. La idea detrás de la demostración es mostrar que $P_{TIF}^{LP} \subseteq P_{Orig}^{LP}$. Si esto sucede, $\hat{z}_{tif} \in P_{Orig}^{LP}$ y por consiguiente $\hat{z}_{orig} \leq \hat{z}_{tif}$.

Sea $\hat{x} \in P_{TIF}^{LP}$ una solución (fraccionaria) factible de la relajación PL de TIF, necesitamos mostrar que \hat{x} es también factible en P_{Orig}^{LP} . Debemos notar que TIF tiene muchas desigualdades en común con Orig, i.e., aquellas incluidas en Orig en las cuales las variables x_e no son consideradas explícitamente. Luego, el problema se reduce a mostrar que \hat{x} satisface las desigualdades (3.2), (3.3), (3.9), (3.4) y (3.16) de Orig.

Como mencionamos en la Sección 4.1.1, la formulación TIF no posee las variables x_e incluidas en las desigualdades enumeradas. Luego, para que \hat{x} sea factible en P_{Orig}^{LP} necesitamos transformar la solución para que este compuesta por las variables utilizadas en

Orig. Dicha transformación se realiza mediante la igualdad (4.2) que vincula las variables x_e^t con las variables x_e . Por otro lado, las variables y_t y z_a se mantienen equivalentes, mientras que las variables z_a^t y $r w_e^t$ simplemente son descartadas.

A continuación analizamos cada una de las desigualdades mencionadas.

1. *Desigualdades (3.2)*. Comenzamos mostrando que \hat{x} satisface las desigualdades (3.2). Por (4.2) y (4.3) tenemos que

$$x_e = \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^t \geq \sum_{\tau=q_e}^{q_e+d_e} q_e x_e^t = q_e.$$

2. *Desigualdades (3.3)*. Primero reescribimos las desigualdades (3.3) como

$$x_e \leq q_e + (1 - y_{t_e})d_e.$$

Sabemos que \hat{x} satisface las desigualdades (4.15) agregadas a TIF, que establecen que $x_e^{q_e} = 1$ si el tren t_e es cancelado, para soluciones enteras. Reescribiendo las desigualdades (4.3) tenemos que

$$y_{t_e} \leq x_e^{q_e} = 1 - \sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau,$$

y por lo tanto

$$\sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau \leq 1 - y_{t_e}. \quad (4.16)$$

Usando esta desigualdad y analizando la definición de las variables x_e en (4.2), tenemos que

$$\begin{aligned} x_e &= \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau = q_e x_e^{q_e} + \sum_{\tau=q_e+1}^{q_e+d_e} \tau x_e^\tau \leq q_e x_e^{q_e} + \sum_{\tau=q_e+1}^{q_e+d_e} (q_e + d_e) x_e^\tau \\ &= q_e x_e^{q_e} + q_e \sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau + d_e \sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau \\ &\leq q_e \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau + d_e \sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau = q_e + d_e \sum_{\tau=q_e+1}^{q_e+d_e} x_e^\tau \end{aligned}$$

$$\text{por (4.16)} \leq q_e + d_e(1 - y_{t_e}).$$

3. *Desigualdades (3.9)*. A continuación analizamos las desigualdades (3.9) siguiendo una idea similar a la utilizada en Dash et al. [4]. Sean $e, f \in E_{tren}$, $r \in R^-(f)$ y $a = (e, f) \in A_r^{hacia}(f)$. Reescribimos la desigualdad como

$$x_e - M_a(1 - z_a) + L_a \leq x_f,$$

y recordemos que $M_a = L_a + (q_e + d_e - q_f)$. Definimos $\bar{x}_e^t = x_e^t - z_a^t$, y notamos que

$$\sum_{\tau=q_e}^{q_e+d_e} \bar{x}_e^\tau = \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau - \sum_{\tau=q_e}^{q_e+d_e} z_a^\tau = 1 - z_a.$$

Utilizando esta igualdad, podemos reescribir las variables x_e como

$$\begin{aligned} x_e &= \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau = \sum_{\tau=q_e}^{q_e+d_e} \tau \bar{x}_e^\tau + \sum_{\tau=q_e}^{q_e+d_e} \tau z_a^\tau \leq (q_e + d_e) \sum_{\tau=q_e}^{q_e+d_e} \bar{x}_e^\tau + \sum_{\tau=q_e}^{q_e+d_e} \tau z_a^\tau \\ &\leq (q_e + d_e)(1 - z_a) + \sum_{\tau=q_e}^{q_e+d_e} \tau z_a^\tau \end{aligned} \quad (4.17)$$

Reemplazando la variable x_e por esta expresión, obtenemos la siguiente cota superior para el lado izquierdo de la desigualdad (3.9):

$$\begin{aligned} x_e - M_a(1 - z_a) + L_a &= x_e - (L_a + q_e + d_e - q_f)(1 - z_a) + L_a \\ &= x_e + L_a z_a - (q_e + d_e)(1 - z_a) + q_f(1 - z_a) \\ \text{por (4.17)} &\leq L_a z_a + q_f(1 - z_a) + \sum_{\tau=q_e}^{q_e+d_e} \tau z_a^\tau \\ \text{por (4.8)} &= q_f(1 - z_a) + \sum_{\tau=q_e}^{q_e+d_e} (\tau + L_a) z_a^\tau. \end{aligned} \quad (4.18)$$

Ahora obtendremos una expresión que es una cota inferior de x_f , es decir, del lado derecho de la desigualdad (3.9). Para mostrar los resultados intermedios, nos basamos en las observaciones 1-3 analizadas anteriormente.

$$\begin{aligned} x_f &= \sum_{\tau=q_f}^{q_f+d_f} \tau x_f^\tau = \sum_{\tau=q_f}^{q_f+d_f} (q_f + \tau - q_f) x_f^\tau = q_f + \sum_{\tau=q_f}^{q_f+d_f} (\tau - q_f) x_f^\tau \\ \text{(por Observación 1)} &\geq q_f + \sum_{\tau=q_f}^{q_f+d_f} (\tau - q_f)^r w_f^\tau \\ \text{(por Observación 2)} &\geq q_f + \sum_{\tau=q_f}^{q_f+d_f} (\tau - q_f) \sum_{\tau_0 \in I_e^f(\tau, r)} z_{(e,f)}^{\tau_0} \\ \text{(por Observación 3)} &\geq q_f - q_f z_a + \sum_{\tau=q_f}^{q_f+d_f} \tau \sum_{\tau_0 \in I_e^f(\tau, r)} z_{(e,f)}^{\tau_0} \end{aligned} \quad (4.19)$$

La Observación 3 relaciona la sumatoria sobre la ventana de tiempo de f con la sumatoria sobre la ventana de e . Además, dado $\tau \in W_f$, sea $\tau_0 \in I_e^f(\tau, r)$, sabemos que se cumple que $\tau \geq \tau_0 + L_a$. Luego, podemos reemplazar la expresión (4.19) por

$$x_f \geq q_f(1 - z_a) + \sum_{\tau_0=q_e}^{q_e+d_e} (\tau_0 + L_a) z_{(e,f)}^{\tau_0}. \quad (4.20)$$

Como podemos ver, la expresión (4.20) es equivalente a (4.18). Luego, dicha expresión actúa por un lado como una cota superior del lado izquierdo de la desigualdad (3.9) y por otro como una cota inferior de su lado derecho. Por dicho motivo, \hat{x} satisface la desigualdad (3.9).

4. *Desigualdades* (3.4). Inicialmente reescribimos la desigualdad de la siguiente manera

$$x_e + L_a \leq x_f$$

Tengamos en cuenta que estas desigualdades se definen para actividades $a = (e, f) \in A_{\text{tren}}$. Los valores de q_e y q_f son los horarios del cronograma original, los cuales dependen del tiempo de viaje o espera según corresponda. Inicialmente, asumimos que en el nuevo cronograma los tiempos mínimos de viaje o espera (i.e. L_a) son iguales o mayores al cronograma original, por ejemplo, debido a la menor disponibilidad de recursos de la red. Luego, sucede que $q_e + L_a \geq q_f$.

Sumamos la desigualdad (4.5) para cada $t = q_e \dots q_e + d_e$ y obtenemos

$$\sum_{t=q_e}^{q_e+d_e} \sum_{\tau=t}^{q_e+d_e} x_e^\tau \leq \sum_{t=q_e}^{q_e+d_e} \sum_{\tau=t+L_a}^{q_f+d_f} x_f^\tau. \quad (4.21)$$

Analizando el lado izquierdo de la desigualdad, tenemos que

$$\sum_{t=q_e}^{q_e+d_e} \sum_{\tau=t}^{q_e+d_e} x_e^\tau = \sum_{\tau=q_e}^{q_e+d_e} (\tau - q_e + 1) x_e^\tau = \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau + (-q_e + 1) \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau = x_e - q_e + 1. \quad (4.22)$$

En cuanto al lado derecho de (4.21), suponemos que $q_e + d_e + L_a < q_f + d_f$ y separamos la sumatoria de la siguiente manera

$$\begin{aligned} \sum_{t=q_e}^{q_e+d_e} \sum_{\tau=t+L_a}^{q_f+d_f} x_f^\tau &= \sum_{\tau=q_e+L_a}^{q_e+d_e+L_a} (\tau - (q_e + L_a) + 1) x_f^\tau + \sum_{\tau=q_e+d_e+L_a+1}^{q_f+d_f} (d_e + 1) x_f^\tau, \\ &\leq \sum_{\tau=q_e+L_a}^{q_e+d_e+L_a} (\tau - (q_e + L_a) + 1) x_f^\tau + \sum_{\tau=q_e+d_e+L_a+1}^{q_f+d_f} (\tau - (q_e + L_a) + 1) x_f^\tau, \\ &= \sum_{\tau=q_e+L_a}^{q_f+d_f} (\tau - (q_e + L_a) + 1) x_f^\tau. \end{aligned} \quad (4.23)$$

La expresión obtenida en (4.23) es equivalente a la que se obtiene en caso de que $q_e + d_e + L_a \geq q_f + d_f$, quedando ambos escenarios descriptos. Luego, reemplazando (4.22) y (4.23) en la desigualdad (4.21) obtenemos

$$\begin{aligned}
x_e - q_e + 1 &\leq \sum_{\tau=q_e+L_a}^{q_f+d_f} (\tau - (q_e + L_a) + 1)x_f^\tau, \\
(\text{por } q_e + L_a \geq q_f) &\leq \sum_{\tau=q_f}^{q_f+d_f} (\tau - (q_e + L_a) + 1)x_f^\tau, \\
&= \sum_{\tau=q_f}^{q_f+d_f} \tau x_f^\tau + (-q_e - L_a + 1) \sum_{\tau=q_f}^{q_f+d_f} x_f^\tau, \\
&= x_f - q_e - L_a + 1.
\end{aligned}$$

Llegando a la expresión buscada.

También podría suceder que en el nuevo cronograma, los tiempos de viaje o espera sean menores a los originales. Para estos casos, sucede que $q_e + L_a < q_f$.

En particular, en caso de que $q_f > q_e + d_e + L_a$, la desigualdad (3.4) se cumple trivialmente dado que

$$x_e + L_a \leq q_e + d_e + L_a < q_f \leq x_f$$

Luego, nos enfocamos en el caso donde $q_e + L_a < q_f \leq q_e + d_e + L_a$.

Inicialmente, multiplicamos la desigualdad (4.5) para $t = q_e$ por $q_e + L_a$, obteniendo como resultado

$$\sum_{\tau=q_e}^{q_e+d_e} (q_e + L_a)x_e^\tau \leq \sum_{\tau=q_f}^{q_f+d_f} (q_e + L_a)x_f^\tau. \quad (4.24)$$

También agregamos las desigualdades (4.5) para $t = q_e + 1 \dots q_e + d_e$, las cuales sumadas a la desigualdad (4.24) dan como resultado

$$\sum_{\tau=q_e}^{q_e+d_e} (q_e + L_a)x_e^\tau + \sum_{t=q_e+1}^{q_e+d_e} \sum_{\tau=t}^{q_e+d_e} x_e^\tau \leq \sum_{\tau=q_f}^{q_f+d_f} (q_e + L_a)x_f^\tau + \sum_{t=q_e+1}^{q_e+d_e} \sum_{\tau=\max\{t+L_a, q_f\}}^{q_f+d_f} x_f^\tau. \quad (4.25)$$

Analizamos cada lado de la desigualdad por separado. Primero, para el lado izquierdo

podemos ver que

$$\begin{aligned} \sum_{\tau=q_e}^{q_e+d_e} (q_e + L_a)x_e^\tau + \sum_{t=q_e+1}^{q_e+d_e} \sum_{\tau=t}^{q_e+d_e} x_e^\tau &= \sum_{\tau=q_e}^{q_e+d_e} q_e x_e^\tau + \sum_{t=q_e+1}^{q_e+d_e} \sum_{\tau=t}^{q_e+d_e} x_e^\tau + L_a \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau \\ \text{por (4.2) y (4.3)} &= \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau + L_a \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau = x_e + L_a. \end{aligned}$$

Ahora analizamos el lado derecho de la desigualdad (4.25). En particular, nos enfocamos en el segundo término de la suma.

Recordemos que por suposición $q_e + 1 \leq q_f - L_a \leq q_e + d_e$. Luego, podemos reescribir el término de la siguiente manera:

$$\sum_{t=q_e+1}^{q_e+d_e} \sum_{\tau=\max\{t+L_a, q_f\}}^{q_f+d_f} x_f^\tau = \sum_{t=q_e+1}^{q_f-L_a} \sum_{\tau=q_f}^{q_f+d_f} x_f^\tau + \sum_{t=q_f-L_a+1}^{q_e+d_e} \sum_{\tau=t+L_a}^{q_f+d_f} x_f^\tau. \quad (4.26)$$

Analizamos cada parte por separado:

a) El primer término se puede reescribir como

$$\sum_{t=q_e+1}^{q_f-L_a} \sum_{\tau=q_f}^{q_f+d_f} x_f^\tau = (q_f - q_e - L_a) \sum_{\tau=q_f}^{q_f+d_f} x_f^\tau.$$

b) Para el segundo término, el límite superior de la primera sumatoria puede ser reemplazado por $q_f + d_f - L_a$. En caso de que $q_e + d_e > q_f + d_f - L_a$ la nueva expresión es equivalente, dado que los términos excluidos se anulan sobre la sumatoria interna. En caso de que $q_e + d_e \leq q_f + d_f - L_a$, se consideran más términos en la sumatoria interna, y de esa manera se satisface la desigualdad. Además, al considerar la sustitución $t' = t + L_a$, obtenemos la siguiente expresión

$$\begin{aligned} \sum_{t=q_f-L_a+1}^{q_e+d_e} \sum_{\tau=t+L_a}^{q_f+d_f} x_f^\tau &\leq \sum_{t=q_f-L_a+1}^{q_f+d_f-L_a} \sum_{\tau=t+L_a}^{q_f+d_f} x_f^\tau \\ &= \sum_{t'=q_f+1}^{q_f+d_f} \sum_{\tau=t'}^{q_f+d_f} x_f^\tau. \end{aligned}$$

Combinando estas dos expresiones con (4.26) y reemplazando en el lado derecho de

(4.25) obtenemos finalmente

$$\begin{aligned}
x_e + L_a &\leq \sum_{\tau=q_f}^{q_f+d_f} (q_e + L_a)x_f^\tau + (q_f - q_e - L_a) \sum_{\tau=q_f}^{q_f+d_f} x_f^\tau + \sum_{t'=q_f+1}^{q_f+d_f} \sum_{\tau=t'}^{q_f+d_f} x_f^\tau \\
&= \sum_{\tau=q_f}^{q_f+d_f} q_f x_f^\tau + \sum_{t'=q_f+1}^{q_f+d_f} \sum_{\tau=t'}^{q_f+d_f} x_f^\tau \\
&= \sum_{\tau=q_f}^{q_f+d_f} \tau x_f^\tau = x_f.
\end{aligned}$$

lo que completa la demostración para el último caso.

5. *Desigualdades* (3.16). La formulación TIF modela estas desigualdades mediante las restricciones (4.6). Combinando estas últimas con las desigualdades (4.3), tenemos que

$$\sum_{\tau=q_e}^{\tau_2-1} x_e^\tau + z_a \leq 1 = \sum_{\tau=q_e}^{q_e+d_e} x_e^\tau$$

obteniendo como resultado la siguiente expresión

$$z_a \leq \sum_{\tau=\tau_2}^{q_e+d_e} x_e^\tau.$$

A partir de esta desigualdad, podemos llegar a la expresión buscada

$$\tau_2 z_a \leq \tau_2 \sum_{\tau=\tau_2}^{q_e+d_e} x_e^\tau \leq \sum_{\tau=q_e}^{\tau_2-1} \tau x_e^\tau + \sum_{\tau=\tau_2}^{q_e+d_e} \tau_2 x_e^\tau \leq \sum_{\tau=q_e}^{q_e+d_e} \tau x_e^\tau = x_e.$$

A partir de estos cinco puntos, podemos ver que la solución (fraccionaria) $\hat{x} \in P_{\text{TIF}}^{LP}$ es también factible para P_{Orig}^{LP} , y por lo tanto la proposición $\hat{z}_{\text{orig}} \leq \hat{z}_{\text{tif}}$ queda demostrada. \square

4.4. Equivalencia del conjunto de soluciones

En esta última sección, mostramos que el conjunto de soluciones óptimas de ambas formulaciones es equivalente.

Primero, recordemos como en el análisis de la sección anterior, trabajamos con la formulación TIF con las desigualdades (4.15) agregadas al modelo, al cual llamaremos TIF + (4.15). En base a dicho análisis, sabemos que todo punto que vive en TIF + (4.15) también vive en Orig. Además, sabemos que dada una solución entera factible de TIF, la misma se puede modificar por otra solución entera que pertenece a TIF + (4.15) y que tiene una función objetivo mejor o igual. En base a eso, cualquier solución óptima de TIF

pertenece también a TIF + (4.15) y por consiguiente se puede transformar a un punto en Orig.

Habiendo demostrado esto, debemos mostrar también que toda solución óptima que pertenece a Orig se encuentra incluida en TIF. Lo haremos en base a la siguiente proposición.

Proposición 2. *Sea $\hat{x} \in P_{Orig}$ entonces $\hat{x} \in P_{TIF}$*

Esquema de Demostración. La idea de la demostración es la misma que para la Proposición 1, i.e., dada una solución $\hat{x} \in P_{Orig}$, debemos asegurarnos que \hat{x} cumpla las restricciones de TIF que no se encuentren incluidas en Orig.

Tal como comentamos al analizar la Proposición 1, ambos modelos utilizan distintas variables, con lo cual debemos transformar la solución $\hat{x} \in P_{Orig}$ a una solución equivalente que utilice las variables del modelo TIF. Dicha transformación no es tan sencilla como la inversa, dado que el modelo TIF posee las variables por índice de tiempo que no están presentes en Orig.

Dada una solución $\hat{x} \in P_{Orig}$, la misma estará compuesta por variables y_t para cada tren $t \in T$, x_e para cada evento $e \in E_{tren}$ y z_a para cada actividad $a \in A$. Para generar una solución válida para TIF, necesitamos definir los valores de cada una de las variables presentes en dicha formulación. Analizamos como hacerlo por cada tipo de variable.

1. *Variables $y_t \forall t \in T$.* Se definen igual que el valor de y_t de \hat{x} .
2. *Variables $z_a \forall a \in A$.* Se definen igual que el valor de z_a de \hat{x} .
3. *Variables $x_e^t \forall e \in E_{tren}, t \in W_e$.* Dado el valor de x_e de \hat{x} , si sucede que $x_e = t$, entonces se define $x_e^t = 1$. Si por otro lado $x_e \neq t$, se define $x_e^t = 0$.
4. *Variables $z_a^t \forall a = (e, f) \in A, e \in E_{tren}, t \in W_e$.* Dado el valor de z_a de \hat{x} , si sucede que $z_a = 0$, entonces se define $z_a^t = 0 \forall t \in W_e$. En caso de que $z_a = 1$, se analiza el valor de x_e de \hat{x} . Si $x_e = t$, entonces se define $z_a^t = 1$. Si por otro lado $x_e \neq t$, se define $z_a^t = 0$.
5. *Variables $r w_e^t \forall e \in E_{tren}, r \in R^-(e), t \in W_e$.* Finalmente, dado que podemos definir el valor de z_a y z_a^t en base a los pasos mencionados, podemos utilizar dichas variables para definir los valores de $r w_e^t$ en base a las igualdades (4.13) y (4.14) de TIF.

Una vez establecido como realizar la transformación, necesitamos mostrar que el punto transformado cumple las restricciones de TIF que no se encuentran presentes en Orig, i.e., que cumple las desigualdades (4.3) - (4.14).

Se puede ver fácilmente que se cumplen cada una de las restricciones. Varias de ellas se pueden derivar algebraicamente a partir del método de transformación mencionado. En particular, las igualdades (4.13) y (4.14) se comprueban trivialmente a partir de la definición de $r w_e^t$. Por dicho motivo, no incluimos la explicación detallada para cada desigualdad. Notar que a diferencia de la demostración que relaciona las relajaciones lineales de ambas formulaciones, en este caso el análisis se realiza sobre soluciones enteras y no fraccionarias.

En base al resultado de la Proposición 2, podemos concluir que el conjunto de soluciones óptimas de ambos modelos es equivalente.

5. Experimentación y Resultados

En el presente capítulo detallamos los experimentos llevados a cabo para comparar, observando distintas métricas, el comportamiento de nuestra formulación alternativa TIF respecto de la formulación Orig en la que basamos nuestro trabajo. Para esto, utilizamos un algoritmo *Branch-and-Cut* para resolver ambos modelos. El objetivo es analizar de manera empírica las hipótesis formuladas y los resultados teóricos obtenidos en el Capítulo 4.

Se presenta un esquema de experimentación que se organiza de la siguiente manera. La Sección 5.1 describe el caso de estudio a tratar, a partir del cual construimos las instancias que utilizamos en nuestros experimentos. En la Sección 5.2 incluimos un repaso de los distintos parámetros que intervienen en el problema, cuáles son sus implicancias sobre nuestro modelo, cómo se interrelacionan, y cuáles utilizaremos para nuestros experimentos. Luego, en 5.3 describimos los escenarios de bloqueos considerados para la generación de instancias. La Sección 5.4 describe brevemente las distintas versiones de preprocesamiento consideradas. En 5.5, se incluye el estudio realizado en la búsqueda de la mejor configuración del algoritmo *Branch-and-Cut* para la formulación TIF. Previo a la comparación de ambas formulaciones, se incluye en la Sección 5.6 un breve apartado sobre errores numéricos relativos a nuestra experimentación. Finalmente, en 5.7 se muestran los resultados computacionales comparando las formulaciones Orig y TIF.

Todos los experimentos fueron realizados bajo un entorno de pruebas con un Intel Core i7, 16GB RAM, sistema operativo Ubuntu 16.04 LTS, utilizando C++ y el paquete de optimización CPLEX versión 12.6.3.

5.1. Caso de estudio: Línea Mitre

Una de las problemáticas que enfrentamos a la hora de encarar nuestra experimentación fue la imposibilidad de disponer de *benchmarks* de acceso público para utilizar como datos de comparación.

En el trabajo de Veelenturf et al. [12] se utilizan instancias generadas a partir de la red de trenes holandesa. Dicha información corresponde a una empresa privada de la cual no pudimos obtener los datos correspondientes.

El caso de estudio seleccionado como fuente de datos para nuestro trabajo corresponde a la red de trenes de la **Línea Mitre**, cuyos datos son de acceso público¹.

La Línea Mitre se encuentra enmarcada dentro de la red de servicios del área metropolitanas de Buenos Aires. La misma cuenta con una red de 192km de extensión, conecta 56 estaciones y posee 38 formaciones operativas las cuales operan 317 servicios diarios.

Es la segunda línea de mayor extensión (y cantidad de estaciones y formaciones) solo por detrás de la línea Roca. Une la zona norte y noroeste del Gran Buenos Aires con el centro de la Ciudad Autónoma de Buenos Aires, más precisamente el barrio de Retiro. La Figura 5.1 muestra su mapa de recorridos.

¹ <http://www.trenmitre.com.ar/horarios.php>



Fig. 5.1: Línea Mitre - Mapa de recorridos

Buscando acotar el conjunto de datos para que el mismo no sea muy extenso en una primera instancia, decidimos utilizar únicamente el ramal **Retiro-Tigre**. Esto nos agrega otra simplificación, que es el hecho de que para este ramal la red es lineal (las estaciones forman una línea, no hay desvíos posibles ni ciclos.) Este tipo de red posee la misma estructura que las consideradas en el trabajo de Louwerse y Huisman [9]. Uno de los objetivos como trabajo futuro es realizar una serie de experimentos con redes con estructuras más complejas. Detallamos algunas ideas al respecto en el Capítulo 6.

El ramal Retiro-Tigre cuenta con 17 estaciones. Cada una de ellas cuenta con dos vías y cada par de estaciones está unido por tramos de también dos vías. Es el ramal más utilizado de la Línea Mitre. En 2015, del total de 41.5 millones de pasajeros (con boleto pago), 26.7 millones viajaron en este ramal.

Para nuestros experimentos, tomamos el cronograma utilizado de Lunes a Viernes hábiles, el cual está compuesto de 162 servicios. Cada servicio demora 60 minutos desde una cabecera a la otra, y parten con una frecuencia de 14 minutos. La Figura 5.2 muestra un extracto del cronograma para los trenes que viajan desde Retiro hacia Tigre².

En cuanto al material rodante, al igual que en el trabajo de Veulenturf et al. [12], solo consideramos la cantidad mínima de unidades disponibles al inicio del día para garantizar la circulación de los trenes en operatoria normal. Es decir que no son consideradas unidades de repuesto. Esta cantidad mínima está determinada por 10 composiciones de material rodante en la estación cabecera Tigre. Esto es así dado que el cronograma indica que los trenes que parten desde Tigre comienzan su recorrido mucho antes (aprox. 1:15hrs), con lo cual alimentan con material rodante a la estación de Retiro una vez que terminan su recorrido.

² El cronograma de la Figura 5.2 difiere sutilmente de los datos mencionados (y utilizados en nuestros experimentos) dado que el mismo corresponde al período 2016 y los datos utilizados corresponden al período 2015.

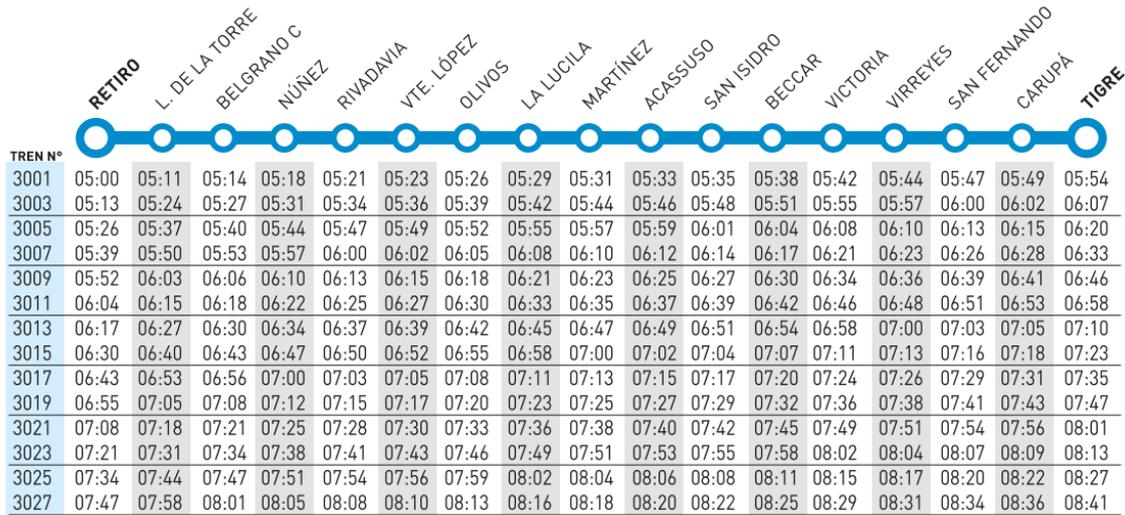


Fig. 5.2: Ramal Retiro-Tigre. Extracto del Cronograma 2016 para trenes viajando desde Retiro hacia Tigre, utilizado en días Lunes a Viernes hábiles.

5.2. Parámetros

Como vimos en detalle en el Capítulo 3, hay numerosos parámetros que intervienen en nuestro modelo. Algunos de ellos corresponden a eventos y otros a actividades de nuestra red evento-actividad.

En cuanto a los parámetros relacionados con eventos, el horario original q_e de cada evento e queda determinado a partir del cronograma del ramal Retiro-Tigre, del cual incluimos un extracto en la Figura 5.2. La capacidad i_e de los eventos de inventario también queda determinada a partir de la descripción realizada en 5.1 de nuestra red.

Otro parámetro relativo a los eventos que debe ser definido es el mayor retraso permitido d_e para cada evento e , cuyo valor tiene una gran relevancia. Valores chicos para d_e implican menos opciones de reprogramación, obligando a cancelar más trenes. Por otro lado, valores grandes dan lugar a muchas más opciones, generando problemas en principio más difíciles, pudiendo causar un fuerte impacto en los tiempos de cómputo. En los trabajos previos de Louwerse y Huisman [9] y Veelenturf et al. [12] se puede ver la gran influencia de este parámetro en los tiempos de ejecución. Además, como vimos en el Capítulo 4, los nuevos conjuntos de variables introducidas para la formulación TIF dependen fuertemente del máximo retraso, por lo que al aumentar su valor, aumentará el número de variables y restricciones con el que trabajará dicho modelo.

Para nuestros experimentos, utilizaremos distintos valores para d_e , a saber: 2, 3, 5 y 7 minutos. Valores superiores para éste parámetro incurren en tiempos de cómputo muy elevados que dificultan el análisis de los resultados. Al mismo tiempo, valores mucho más grandes no son convenientes dentro de un contexto de reprogramación en tiempo real, donde retrasos mayores pueden tener un impacto muy negativo en la calidad del servicio para los usuarios. Consideramos que los valores elegidos nos brindan una buena muestra, nos interesa sin embargo como trabajo a futuro evaluar valores mayores para d_e . Vale destacar que los valores utilizados son similares a los seleccionados en Veelenturf et al. [12]. Más adelante, al abordar los distintos experimentos, estudiaremos las implicancias de la

elección de este parámetro.

En cuanto a los parámetros relativos a las actividades, utilizamos como tiempos mínimos de viaje de cada actividad (i.e. L_a) a los tiempos de viaje del cronograma 5.2. Luego, para el nuevo cronograma estamos asumiendo que un tren no puede recorrer un tramo entre estaciones más rápidamente que en el programa original, ni tampoco forzamos que el tiempo de viaje tenga que ser mayor. Asimismo, en el cronograma 5.2 los arribos y posteriores salidas en una estación se encuentran programados en el mismo horario. Luego, todos los valores L_a relativos a actividades de espera en una estación en nuestro modelo serán 0, permitiendo el mismo comportamiento que en el programa original.

Los parámetros L_a de las actividades de recursos, es decir, los tiempos de seguridad, son valores que dependen de las características de la infraestructura de la red y de las restricciones operativas. Dichos datos no son de acceso público para nuestro caso de estudio. Luego, tomamos como valores para estos parámetros los mismos que se utilizan en Veelenturf et al. [12]. Para vías de tramos el tiempo de seguridad es de 2 minutos para trenes operando en una misma dirección y de 0 minutos en dirección contraria. Para el traspaso de vías de estación, el tiempo de seguridad es de 2 minutos. En cuanto al material rodante, para las actividades de traspaso directo (sin pasar por el patio de maniobras) el L_a es de 5 minutos, mientras que para las actividades a través del patio de maniobras el tiempo de seguridad es de 10 minutos.

Finalmente, debemos considerar las dos penalidades presentes en nuestro modelo, pasadas en la función objetivo, la penalización λ_t por cancelación del tren t y la penalización μ_e por unidad (i.e. minuto) de retraso del evento e .

Para mantener un alto nivel de servicio, uno de los objetivos es operar el mayor número posible de trenes. Luego, cancelar un tren es penalizado con un valor mucho mayor que un retraso de un evento.

Para las penalizaciones por retraso μ_e , consideramos una penalidad de valor 1 por cada unidad (minuto) de retraso con respecto al cronograma original.

En cuanto a las cancelaciones, primero consideramos el tiempo total de recorrido de un servicio, el cual en nuestro caso es de 60 minutos. Tomaremos para nuestros experimentos tres valores de penalización por cancelación. Un valor bajo, que corresponde a tomar una vez el tiempo total del recorrido, i.e. $\lambda_t = 60$, un valor intermedio, en el cual penalizamos con 25 veces el tiempo del recorrido, i.e. $\lambda_t = 1500$, y finalmente una relación alta de cancelación respecto del retraso, en el cual tomamos 50 veces el tiempo de recorrido, i.e. $\lambda_t = 3000$.

5.3. Escenarios de bloqueo

A partir de nuestro caso de estudio y de los parámetros descriptos, generamos distintos escenarios de bloqueo con el objetivo de conformar un conjunto de instancias para realizar nuestros experimentos.

Uno de los aspectos claves del bloqueo es la duración del mismo. Como vimos en la Sección 3.5, la duración de nuestro bloqueo determinará la cantidad de eventos (y por consiguiente de actividades) a incluir en nuestra red evento-actividad. Luego, es de esperar que el valor de este parámetro tenga un fuerte impacto en los tiempos de cómputo.

Para los valores de los instantes de tiempo, utilizamos la unidad de minutos del día para simplificar. Como valor de inicio del bloqueo (i.e. τ_1), tomaremos siempre 800 (13:20hs). Vale destacar que en dicho horario las salidas de ambas cabeceras ya se encuentran normalizadas, saliendo servicios desde ambas estaciones con la misma frecuencia, distinto a como sucede al principio del día. Decidimos trabajar con tres duraciones distintas de bloqueos, alterando el valor del parámetro τ_2 en 850, 900 y 1000. Esto nos da una duración de 50, 100 y 200 minutos respectivamente.

También utilizamos distintos valores de τ_3 . Recordemos que este parámetro indica el tiempo límite (posterior a τ_2) en que debe realizarse la transición desde el nuevo cronograma al original. Esto implica que a partir de τ_3 ningún tren puede cancelarse, ningún evento puede retrasarse, y todas las unidades de material rodante para el funcionamiento del cronograma original deben estar disponibles. Los valores utilizados son: 10, 30, 50 y 100 minutos.

La ubicación del bloqueo (i.e. el tramo de vías que afecta) es otro dato a considerar. Las cuatro opciones consideradas son: Belgrano-Nuñez, Olivos-La Lucila, San Isidro-Beccar y San Fernando-Carupá.

Finalmente, para cada escenario de bloqueo descrito, consideramos la opción de bloqueo parcial (una vía bloqueada) y de bloqueo total (ambas vías bloqueadas).

A partir de la variación de estos parámetros del bloqueo y de la variación de los retrasos máximos y la penalización por cancelación mencionadas en la sección previa, se generan un total de 1152 instancias con las cuales realizaremos nuestros experimentos. Llamaremos al conjunto de instancias de prueba MITRE.

Es importante remarcar que los experimentos realizados con el conjunto MITRE generado corresponden a un primer abordaje empírico sobre las formulaciones. Creemos que estos experimentos nos otorgan un buen panorama inicial a partir del cual se puede construir un estudio más robusto en próximas etapas, expandiendo el conjunto de instancias y herramientas utilizadas.

5.4. Preprocesamiento

El objetivo del preprocesamiento es fijar o eliminar algunas de las variables de nuestro modelo buscando obtener un formulación equivalente (en el sentido de tener el mismo conjunto de soluciones óptimas), la cual sea más ajustada y nos permita de esa manera obtener mejores resultados.

Para este experimento preliminar, realizamos un análisis básico sobre un subconjunto de las instancias del conjunto MITRE.

El análisis se enfocó principalmente en el preprocesamiento relacionado a las actividades de recursos, buscando explotar la precedencia temporal entre los eventos que relacionan. En la Sección 3.4, se definen las actividades para cada tipo de recurso. Sin embargo, si analizamos algunas de las desigualdades de nuestro modelo, podemos ver como varias de las actividades definidas se pueden eliminar de la red evento-actividad.

Estudiamos tres opciones posibles de preprocesamiento:

- Dada una actividad $a = (e, f)$, las desigualdades (3.9) nos dicen que, si la misma es utilizada, entonces el evento que recibe el recurso (f) no puede obtenerlo de un

evento (e) que ocurre después (i.e. si $z_a = 1$ entonces $x_e \leq x_f$). Este es el mismo concepto que utilizamos en las figuras de la Sección 3.4 en las cuales no incluimos estas actividades para simplificar el entendimiento.

Luego, dada una actividad $a = (e, f)$, si sucede que $q_e > q_f + d_f$ entonces la actividad no se incluye en nuestro modelo. Llamamos a esta opción de preprocesamiento pre_1 .

- La opción pre_1 se puede mejorar si tenemos en cuenta el tiempo de seguridad L_a de la actividad. Es decir, sea $a = (e, f)$, si sucede que $q_e + L_a > q_f + d_f$ entonces la actividad no se incluye. Este es el primer preprocesamiento mencionado en la Sección 4.3, donde comparamos las relajaciones lineales de las formulaciones TIF y Orig. Para actividades con valor alto de L_a , como lo es el traspaso del material rodante a través del patio de maniobras, este preprocesamiento puede dejar afuera muchas más actividades que la opción anterior. Llamamos a este preprocesamiento pre_2 .
- Lo visto para pre_2 se puede extender para la formulación TIF. Aprovechando que la formulación posee variables de actividad por cada unidad de tiempo, dado $t \in W_e$, si sucede que $t + L_a > q_f + d_f$ entonces podemos forzar a que $z_a^t = 0$. A esta tercera opción la llamamos pre_3 . Este es el segundo preprocesamiento mencionado en la Sección 4.3.

En base a nuestros resultados preliminares, pudimos ver que pre_1 tiene un gran impacto en la cantidad de actividades que se eliminan de la red evento-actividad N , como era de esperar. Asimismo, pre_2 introdujo una mejora no despreciable respecto a dicha opción. Ambas opciones, al disminuir considerablemente la cantidad de actividades incluidas en nuestro red, y por consiguiente la cantidad de variables con la que trabajan los modelos, mostraron una fuerte disminución en los tiempos de cómputo.

La opción pre_3 , contrario a lo que esperábamos, no tuvo ningún impacto en los experimentos realizados. Vale recordar que esta opción aplica solo para la formulación TIF. De esta manera, si bien este preprocesamiento fue útil a la hora de realizar un estudio teórico sobre las formulaciones, en base a los resultados obtenidos y para mantener el mismo preprocesamiento para ambas formulaciones, se decidió utilizar la versión pre_2 para todos los experimentos posteriores. Al realizar la experimentación comparando ambas formulaciones en la Sección 5.7, analizaremos si el resultado teórico entre las relajaciones de ambos modelos se mantiene a nivel empírico sin utilizar el preprocesamiento pre_3 en TIF.

Como comentario final sobre el preprocesamiento, sabemos que hay mucho espacio para profundizar sobre este tema.

En Ascheuer et al.[1] se resuelve una variante del clásico TSPTW, con ventanas de tiempo similares a las W_e de nuestro modelo. En dicho trabajo se muestran extensiones que se pueden aplicar tanto a nivel de precedencia temporal entre eventos como también sobre cómo ajustar las ventanas de tiempos W_e fijando valores de x_e^t .

El problema de TSPTW difiere de nuestro problema en algunos aspectos (por ejemplo, al poder cancelar trenes no todos los eventos deben ser visitados, como sí lo son los clientes en TSPTW) por lo que es posible que algunas de las ideas que exploran no apliquen para nuestro caso. Sin embargo, creemos es de gran interés estudiar las mismas y adaptarlas para nuestra situación particular. Será uno de los temas pendientes a considerar como trabajo a futuro.

5.5. Configuración Algoritmo *Branch-and-Cut*

En la siguiente sección detallaremos los experimentos llevados a cabo con el objetivo de buscar la mejor configuración posible para nuestro algoritmo *Branch-and-Cut* sobre la formulación TIF. Como introdujimos en el Capítulo 2, dicho algoritmo posee distintos pasos que se pueden resolver mediante diversos enfoques.

Para nuestro trabajo, nos concentramos en tres aspectos del mismo. Primero realizamos un análisis básico del algoritmo de resolución de la relajación en el nodo raíz y de la estrategia de selección de nodo, el cual se describe en la siguiente subsección. Luego, realizamos un estudio más detallado sobre la regla de selección de *branching*.

5.5.1. Algoritmo de resolución nodo raíz y Selección de nodo

Para resolver la relajación de nuestro problema en el nodo raíz, CPLEX por defecto utiliza el algoritmo de *Simplex*. Como mencionamos en el Capítulo 2, en muchos problemas de PLE una parte no despreciable del tiempo de cómputo se dedica a calcular la primera relajación de nuestro problema. Luego, el objetivo de nuestra experimentación fue analizar si utilizar el algoritmo de *Barrier* en vez del algoritmo de *Simplex* para el cálculo de la relajación tiene un impacto positivo en los tiempos de cómputo.

Procedimos entonces a medir los tiempos de cómputo para el cálculo de la relajación en el nodo raíz para nuestro conjunto de instancias MITRE.

Los resultados arrojaron que el impacto del cambio del algoritmo es prácticamente nulo, con pequeñas variaciones a favor de uno u otro en algunas instancias. Por dicho motivo, decidimos mantener el comportamiento por defecto y utilizar *Simplex*.

Como fue explicado en el Capítulo 2, al recorrer el árbol de subproblemas activos, el algoritmo *Branch-and-Cut* debe tener definida una estrategia de selección del próximo nodo a explorar. El paquete de optimización CPLEX utiliza por defecto la estrategia de *mejor cota primero*. Además de la opción de *búsqueda en profundidad*, CPLEX también posee la opción de *mejor estimación*, en la cual se estima la mejor solución entera que se obtendría en cada nodo y se selecciona el nodo de mejor valor.

Realizamos algunos experimentos preliminares sobre nuestro conjunto de instancias para comprobar cuál de las opciones se adapta mejor a nuestro problema. Se consideraron dos métricas. Por un lado el tiempo total de cómputo para resolver el problema de PLE. Por otro lado, para las instancias que no terminaron, se observó la métrica *final gap*. Esta métrica considera la diferencia relativa entre la mejor solución entera y la mejor cota dual obtenida hasta la terminación del algoritmo. Vale mencionar que el tiempo límite utilizado fue de 10 minutos.

Al igual que para el algoritmo de resolución en el nodo raíz, los resultados no son demasiado interesantes con lo cual no incluimos gráficos o tablas sino que describimos brevemente los comportamientos observados.

La estrategia de *búsqueda en profundidad* dió resultados notoriamente peores, con lo cual fue descartada. También pudimos observar que *mejor estimación* arroja resultados sutilmente mejores que *mejor cota primero*. Si bien no tenemos certezas, atribuimos este comportamiento a que la segunda dedica un poco más de tiempo en calcular la mejor relajación de los subárboles cada vez que tiene que decidir el próximo nodo a explorar, por lo que localmente hay un costo mayor en esa decisión, el cual no termina valiendo la pena. Sin embargo, como la mejora en las métricas mencionadas es casi despreciable, y la

misma solo se presenta en un subconjunto de las instancias, decidimos quedarnos con el valor por defecto que utiliza CPLEX, es decir, la estrategia de *mejor cota primero*.

5.5.2. Branching por variables

A diferencia del estudio de las configuraciones anteriores, el análisis de la regla de *branching* resultó mucho más interesante.

Cada vez que nuestro algoritmo *Branch-and-Cut* encuentra una solución que no cumple con las restricciones de integralidad, debe subdividir el problema a partir de una de las variables que no cumple dicha restricción. Por defecto, CPLEX elige entre cualquiera de las variables que no cumpla la restricción de integralidad para realizar el *branching*, según lo considere mejor. Sin embargo, puede resultar conveniente determinar prioridades de algunos tipos de variables sobre otras, de manera que al realizar el *branching* siempre primero por esas variables, se mejoren los resultados obtenidos. Por ejemplo, uno de los objetivos buscados en muchos problemas es lograr que el árbol de enumeración se mantenga lo más balanceado posible, dado que dicha característica suele tener buenos resultados en la práctica. Por las características particulares de cada problema, puede resultar favorable priorizar el *branching* de cierto tipo de variables por sobre otras.

En nuestra formulación TIF tenemos 5 tipos de variables, las cuales todas son enteras, a saber: $y_t, x_e^t, z_a, z_a^t, {}^r w_e^t$. Luego, además de la estrategia por defecto de CPLEX, utilizamos 5 estrategias más, una por cada tipo de variable de nuestro modelo, en la cual dicha variable es priorizada sobre el resto a la hora de realizar el *branching*.

Comparamos para estas 6 versiones el tiempo total de cómputo, y la cantidad de instancias resueltas sobre el total de nuestro conjunto MITRE. Vale aclarar que para este experimento, el tiempo límite de corrida fue de 20 minutos.

| Por defecto | y_t | z_a | x_e^t | z_a^t | ${}^r w_e^t$ |
|-------------|-------------|-------|---------|---------|--------------|
| 1083 | 1083 | 841 | 1032 | 1063 | 1034 |

Tab. 5.1: Cantidad de instancias resueltas por cada opción de *branching* sobre el conjunto MITRE, compuesto por 1152 instancias.

Como se puede apreciar en la Tabla 5.1, la única opción que puede resolver una cantidad similar a la versión por defecto (de hecho, es la misma cantidad) es el *branching* por y_t . Para el resto de las opciones, se está por debajo. Además, el resto de las opciones también posee peores resultados en la métrica de tiempo de cómputo, con lo cual fueron descartadas.

A continuación, en la Figura 5.3 vemos la comparación del tiempo de cómputo para las dos opciones consideradas. Sólo estamos considerando en esta ocasión las instancias de MITRE en las cuales ambas opciones terminaron por optimalidad, que conforman un total de 1077 instancias.

En base a los resultados arrojados en la Figura 5.3, vemos que la opción de priorizar siempre el *branching* primero por la variable y_t que sobre el resto, arroja mejores resultados incluso que la estrategia por defecto de CPLEX. Sin embargo, la mejora es solo considerable cuando el parámetro d_e toma valor 2, mientras que para el resto de los valores siempre se encuentra por debajo del 10%. Por este motivo, creemos que si bien existe una mejora, por el momento es conveniente seguir manteniendo el *branching* por defecto de CPLEX debido a que los resultados podrían estar sobreajustados a nuestro conjunto de instancias.

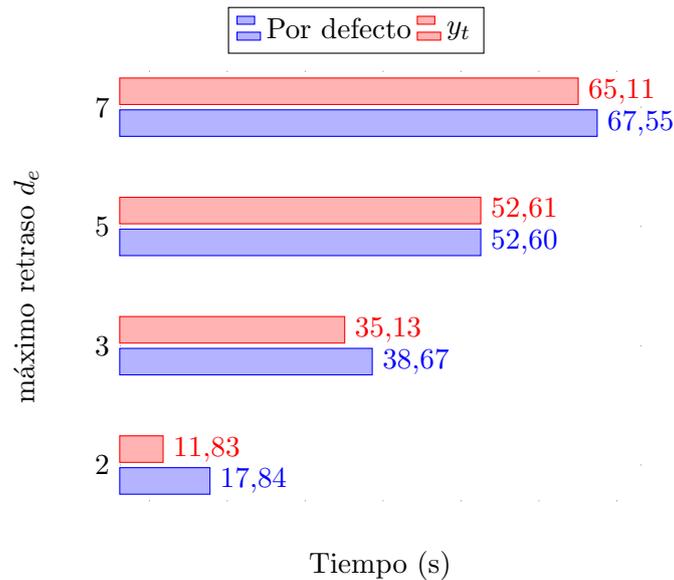


Fig. 5.3: Comparación de tiempo de cómputo opción por defecto vs y_t . Los resultados se muestran agrupados por el parámetro d_e . Para cada valor, se reporta el promedio de la métrica.

Vale mencionar que verificamos que al medir la varianza del tiempo de cómputo, las diferencias entre las distintas opciones fueran similares. Esto es importante dado que el promedio podría ocultarnos el hecho de que para una formulación los valores se encuentren cerca de la media mientras que para la otra se encuentren muy dispersos en un extremo y el otro.

Uno de los objetivos como trabajo a futuro es ampliar el conjunto de instancias de experimentación a considerar. En caso de que los resultados sobre la opción de y_t se mantengan, o incluso mejoren, nos dará un buen indicio para utilizar dicha regla por sobre la versión por defecto.

Antes de avanzar a la próxima sección, debemos primero analizar nuevamente los resultados vistos en la Tabla 5.1. Nos llamó la atención la gran diferencia entre la cantidad de instancias resueltas por la opción de z_a y el resto de las opciones. Analizando en detalle un motivo para esta diferencia, comparando distintas métricas, notamos en particular que dicha versión tiene resultados extremadamente distintos al resto para la métrica de cantidad de nodos enumerados por el algoritmo *Branch-and-Cut*.

La Figura 5.4 muestra la métrica de cantidad de nodos enumerados para las opciones por defecto, y_t y z_a . El resto de las opciones no fueron consideradas ya que los resultados son similares tanto a y_t como a la opción por defecto. Es importante mencionar que para cada instancia, la métrica de cantidad de nodos esta normalizada por su tiempo de ejecución. De esta manera, evitamos resultados engañosos al comparar instancias en donde las distintas opciones toman distinto tiempo en resolver, y por consiguiente sería de esperar que enumeran distinta cantidad de nodos.

Como se puede ver en la Figura 5.4, la diferencia es muy amplia. Lo interesante es que si bien la opción por z_a tiene un tiempo de cómputo mayor al resto (en promedio entre 2.5 y 3.5 veces más), la diferencia en esta métrica es mucho mayor aún. En un principio

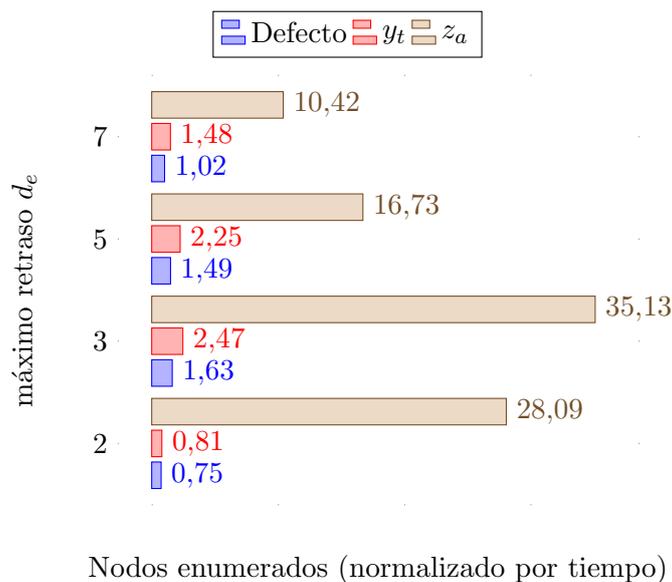


Fig. 5.4: Comparación de nodos enumerados (normalizado por tiempo) para opción por defecto vs y_t vs z_a . Los resultados se muestran agrupados por el parámetro d_e . Para cada valor, se reporta el promedio de la métrica.

no estaba claro el por qué de este comportamiento de la versión z_a por sobre el resto. La teoría que manejamos fue la siguiente: si z_a enumera una cantidad de nodos mucho mayor al resto en el mismo tiempo, esto se puede deber a que para z_a , el tiempo de cómputo en **cada** nodo es mucho menor.

Para comprobar nuestra teoría, analizamos la métrica de cantidad de iteraciones de *Simplex* que se realizan en cada instancia. Si bien no podemos conocer el tiempo de cómputo en cada nodo, si tomamos la cantidad total de iteraciones de *Simplex*, y lo normalizamos por la cantidad de nodos enumerados por el algoritmo de *Branch-and-Cut*, esto nos da una buena aproximación de procesamiento (o tiempo) en cada nodo. En base a esto, esperamos lo siguiente: si para una instancia determinada la opción de z_a enumera una cantidad de nodos (normalizada por tiempo) mayor al resto, también debería realizar una cantidad menor de iteraciones de *Simplex* en cada nodo, y además las proporciones de ambas diferencias deberían ser similares.

Para tener una primera intuición, analizamos esta hipótesis para algunas instancias particulares dentro del conjunto MITRE. Para realizar la comparación contra z_a , utilizamos la opción de branching por y_t , ya que como dijimos anteriormente, el resto de las opciones se comportan de manera similar a nivel enumeración de nodos. La Tabla 5.2 muestra los resultados para tres instancias particulares.

Como se puede ver en las últimas dos columnas de cada opción de *branching*, la opción por z_a siempre enumera más nodos que y_t , y al mismo tiempo esta última realiza más iteraciones de *Simplex* en cada nodo. Esto se condice con nuestra teoría de que z_a recurre a menos iteraciones de *Simplex* en cada nodo, y de ahí su aumento en la enumeración en el *Branch-and-Cut*.

Nos interesa además verificar que la relación en la cual difieren es similar, ya que eso avalaría aun más nuestra teoría, mostrando una relación más explícita de los resultados.

| Nro Inst | y_t | | | | | z_a | | | | |
|----------|--------|-----|-------|-------|--------|--------|-------|---------|-------|--------|
| | $t(s)$ | n | it | n/t | it/n | $t(s)$ | n | it | n/t | it/n |
| 1 | 8 | 11 | 24036 | 1.3 | 2185 | 32 | 1111 | 93045 | 34.7 | 84 |
| 2 | 10 | 74 | 53408 | 7.4 | 722 | 311 | 38468 | 1958604 | 123.6 | 51 |
| 3 | 27 | 12 | 50208 | 0.4 | 4184 | 72 | 105 | 140328 | 1.4 | 1336 |

Tab. 5.2: Comparación de las opciones de *branching* y_t y z_a sobre instancias particulares del conjunto MITRE. t corresponde a tiempo (en segundos), n a nodos e it a iteraciones de *Simplex*. Las últimas dos columnas de cada opción muestran la cantidad de nodos (normalizada por tiempo) y la cantidad de iteraciones de simplex por nodo.

Luego, computamos la relación entre los nodos enumerados por tiempo de y_t respecto de z_a y la relación entre la cantidad de iteraciones de *Simplex* por nodo de z_a respecto de y_t . La Tabla 5.3 muestra dicha relación para estas mismas instancias.

| Nro Inst | $\%n/t - y_t$ vs z_a | $\%it/n - z_a$ vs y_t |
|----------|------------------------|-------------------------|
| 1 | 3.96 % | 3.83 % |
| 2 | 5.98 % | 7.05 % |
| 3 | 30.48 % | 31.94 % |

Tab. 5.3: Diferencia de enumeración de nodos por tiempo y de iteraciones de *Simplex* por nodo para las opciones de *branching* y_t y z_a , sobre las instancias particulares del conjunto MITRE de la Tabla 5.2.

Como se puede apreciar, la diferencia entre ambas métricas es muy similar. De hecho, si calculamos el promedio de dichas métricas para el conjunto de todas nuestras instancias, vemos que mientras la cantidad de nodos por tiempo de y_t respecto de z_a representa el **38.69 %**, la cantidad de iteraciones por nodo de z_a respecto de y_t representa el **38.58 %**, arrojando una diferencia promedio prácticamente igual. Luego, al comprobar que estas tendencias se mantienen para el total de las instancias MITRE, creemos que las mismas explican los resultados observados en la Figura 5.4 sobre la diferencia en la cantidad de nodos enumerados.

Para finalizar nuestro análisis al respecto, debemos decir que si bien en base a los últimos resultados mostrados, confiamos en que nuestra teoría es acertada, existe la posibilidad de que haya otros factores involucrados. Nos queda como debate abierto comprender por qué la opción por z_a requiere menos iteraciones de Simplex en cada nodo para calcular la relajación. Asimismo, es interesante ver cómo el hecho de que al branchear por z_a el algoritmo de *Branch-and-Cut* pueda ir recorriendo el subárbol de problemas de una manera mucho más rápida, esto no termina arrojando buenos resultados a nivel de la métrica de tiempo de cómputo general como vimos anteriormente, donde las dos opciones en consideración son la opción por defecto e y_t .

5.6. Errores Numéricos

A continuación, incluimos una breve descripción de algunos aspectos de índole numérico con los cuales nos topamos al tratar con las formulaciones Orig y TIF utilizando el paquete de optimización CPLEX.

Al trabajar con ambos modelos sobre nuestro conjunto de instancias MITRE, muchas veces sucede que la función objetivo toma valores muy elevados.

Para poder decidir si el algoritmo *Branch-and-Cut* encontró la solución óptima, CPLEX se basa en la métrica del MIP gap. Esta métrica corresponde a la diferencia relativa entre la mejor cota primal y la mejor cota dual conocidas hasta el momento. Cuando el MIP gap toma valor 0 (utilizando una tolerancia para hacer el cálculo), el algoritmo termina. Es decir, sea \underline{z} la mejor cota dual conocida, \bar{z} la mejor cota primal conocida y tol el valor de tolerancia. CPLEX determina que el algoritmo termina cuando:

$$\frac{|\bar{z} - \underline{z}|}{\bar{z}} < tol$$

El problema de trabajar con valores altos del funcional es que al verificar si el MIP gap está por debajo del valor de tolerancia, si éste último no es lo suficientemente pequeño, puede suceder que la solución encontrada no sea realmente la solución óptima de nuestro problema, dado que en realidad ambas cotas no poseen el mismo valor y existen soluciones enteras entre las mismas.

Para determinar el nivel de precisión, CPLEX utiliza los parámetros internos EPGAP y EPAGAP, los cuales controlan el valor de la tolerancia contra la cual se compara el MIP gap. Es decir, a menor valor de EPGAP y EPAGAP, más precisión tendremos. CPLEX por defecto utiliza como valor de tolerancia $1e^{-4}$. Notamos que si aumentamos la precisión para alguna de las instancias de nuestro conjunto MITRE, efectivamente la solución óptima varía. Haciendo distintas pruebas, corroboramos que el valor de tolerancia $1e^{-8}$ es el mayor valor de tolerancia con el cual pudimos obtener la solución óptima en todas nuestras instancias.

Es importante destacar que el hecho de utilizar valores de tolerancia más pequeños, implica que nuestro algoritmo *Branch-and-Cut* demora más tiempo en terminar. Esto muchas veces sucederá en situaciones donde con un valor de tolerancia más alto, igualmente se encontraría la solución óptima. Sin embargo, si trabajáramos con valores mayores, obtendríamos en algunos casos soluciones que al interpretar sus variables nos indicarían, por ejemplo en el caso de las variables x_e , horarios de los eventos en el nuevo calendario que no se corresponden con la solución ideal. Es decir, terminar la ejecución del algoritmo antes de encontrar la solución óptima real implicaría luego tomar decisiones subóptimas en base a esos resultados. Por este motivo, creemos que es importante trabajar con una tolerancia adecuada.

Por otro lado, notamos que para algunas instancias en particular del conjunto MITRE, las soluciones óptimas para Orig y TIF varían levemente. Para dichos casos, TIF brinda un valor menor (i.e. mejor) que Orig. Sin embargo, en base a lo visto en el Capítulo 4, sabemos que el conjunto de soluciones óptimas de ambos modelos es equivalente.

Otro parámetro que utiliza CPLEX y que también es relativo a cuestiones numéricas es NUMERCALEMPHASIS. Si bien la documentación al respecto no incluye demasiada información, la misma describe que en caso de activar el parámetro, se enfatiza la pre-

cisión en problemas numéricamente inestables o difíciles. Por defecto, este parámetro no está activado. Luego, corrimos nuevamente las instancias mencionadas con el parámetro activado para la formulación Orig y de esa manera obtuvimos exactamente los mismos resultados que para TIF.

Si bien el uso de este parámetro para la formulación Orig resuelve el problema mencionado, debe destacarse que el hecho de activarlo tiene un impacto no despreciable en los tiempos de cómputo, por el costo de la realización de los cálculos con una precisión aumentada. Dado que la diferencia entre las soluciones óptimas solo se presentó para unas pocas instancias, decidimos correr los experimentos subsiguientes **sin** utilizar este parámetro para Orig. Sin embargo, en caso de utilizar esta formulación en un contexto real, es posible que sea necesario utilizarlo para considerar los resultados válidos. En caso de observar una situación similar sobre un conjunto más amplio de instancias, se puede plantear a esta característica de formular problemas numéricamente más estables como una ventaja de la formulación TIF por sobre Orig.

5.7. Comparación Orig vs TIF

Incluimos en esta última sección de experimentación, la comparación entre ambas formulaciones.

Para comenzar el análisis, haremos foco en lo que fue nuestro estudio teórico en el Capítulo 4, la mejora en la relajación de la formulación TIF por sobre Orig. Para esto, estudiaremos la métrica *LP gap*. Esta métrica determina la diferencia relativa entre la relajación de nuestro problema y el valor de la solución óptima. Es decir, sea z^* el óptimo de nuestro problema original y z_{LP}^* el óptimo de nuestra relajación, calculamos el *LP gap* como:

$$\frac{|z^* - z_{LP}^*|}{z^*}$$

Como algunas instancias terminan su ejecución sin encontrar el óptimo, para esos casos calculamos el LP gap contra la mejor solución conocida por ambas formulaciones, para comparar siempre la diferencia contra el mismo valor.

La función objetivo de ambas formulaciones posee en su segundo término la constante $\sum_{e \in E_{tren}} -\mu_e q_e$. Luego, al resolver nuestras instancias con el paquete de optimización CPLEX, dicha constante no es incluida como parte del funcional sino que se suma una vez obtenida la solución de CPLEX para generar la solución óptima real. Vale aclarar que para reportar los valores de la métrica LP gap, la misma es computada en base al funcional real, y no al que utiliza internamente CPLEX, el cual tiene una magnitud mayor al no tener la constante restada. Esta aclaración es importante dado que el hecho de que CPLEX cuente con un valor de LP gap muy pequeño no significa que el LP gap real también lo sea, por la diferencia en la magnitud de ambos valores. Por este mismo motivo se utilizan los valores de tolerancia mencionados en la Sección 5.6, para evitar resultados distorsionados por el hecho de trabajar con una función objetivo de mayor valor.

Recordemos que en la Sección 4.3 agregamos a TIF las desigualdades (4.15), las cuales a pesar de no ser necesarias para que el modelo sea válido (el conjunto de soluciones óptimas es equivalente), fueron incluidas para simplificar la demostración de que TIF posee una mejor relajación lineal. Nos interesa validar nuestro resultado teórico con resultados experimentales sobre la formulación sin las desigualdades (4.15), i.e., si TIF posee un valor

mayor de la relajación que Orig, y por consiguiente un menor LP gap, y estudiar también cuál es la magnitud de dicha diferencia.

Para los experimentos realizados en esta sección, al igual que para los de la regla de *branching*, el límite de tiempo de ejecución seleccionado fue de 20 minutos. Dicho límite se debe a que valores mayores no son considerados viables dado que nuestro modelo debe brindar una solución en un contexto de tiempo real. Cabe destacar que se realizaron experimentos preliminares con límites menores, pero los resultados para las distintas métricas consideradas fueron similares.

En la Figura 5.5 mostramos los resultados de nuestros experimentos para la métrica mencionada.

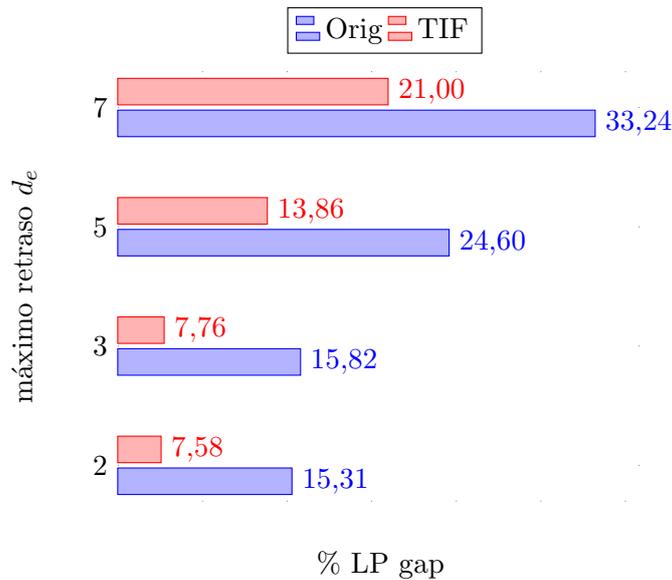


Fig. 5.5: Comparación % LP gap Orig vs TIF. Los resultados se muestran agrupados por el parámetro d_e . Para cada valor, se reporta el promedio de la métrica.

Como se puede apreciar, los resultados fueron los esperados. Podemos ver cómo nuestra formulación TIF muestra una mejora que ronda aproximadamente entre el 35 % y el 50 % del valor del LP gap. Vale mencionar que, al igual que sobre los experimentos de regla de *branching*, verificamos que al medir la varianza, los resultados fueron similares.

Otro dato interesante a notar es el hecho de que para ambas formulaciones, a mayor valor del parámetro d_e , mayor es el valor del LP gap. Esto se corresponde con el hecho de que dicho valor tiene un alto impacto en la dificultad de las instancias. Notar que esto no ocurre únicamente para TIF, donde el impacto de d_e es directo sobre la cantidad de variables, sino que Orig también se ve afectado de manera similar.

Cuando el máximo retraso permitido aumenta, las instancias comienzan con un valor de la relajación más lejano al de la solución óptima. Esto podría dificultar la terminación del algoritmo, aspecto que analizaremos más adelante.

Tal como fue detallado en la Sección 3.5.2, otro parámetro a considerar de gran impacto en la dificultad de las instancias es el tamaño del bloqueo. El mismo determina la cantidad de eventos a considerar en nuestra red evento-actividad, y en consecuencia la cantidad de

variables y restricciones de los modelos. En la Figura 5.6 se muestran los resultados para la métrica de LP gap agrupados por la duración del bloqueo (i.e. $\tau_2 - \tau_1$), que en nuestro conjunto de instancias MITRE queda determinado por la variación del parámetro τ_2 dado que τ_1 es fijo.

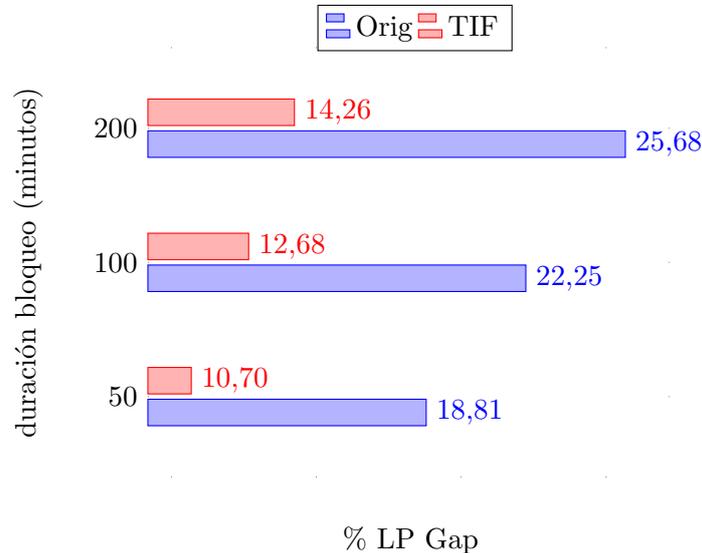


Fig. 5.6: Comparación % LP Gap Orig vs TIF. Los resultados se muestran agrupados por la duración del bloqueo (en minutos). Para cada valor, se reporta el promedio de la métrica.

Podemos ver cómo nuestra formulación sigue manteniendo la misma tendencia de mejora respecto de Orig, entre un 35 % y 50 %, si vemos los resultados agrupados por el tamaño del bloqueo. También se puede apreciar claramente el impacto que tiene este parámetro, al igual que d_e , en la dificultad de las instancias, y cómo el incremento del LP gap en función del mismo parece ser proporcional para ambas formulaciones.

Además de lo mencionado sobre el LP gap, es interesante analizar la métrica de cantidad de nodos enumerados por el algoritmo *Branch-and-Cut* por cada formulación. La Figura 5.7 muestra el valor de esta métrica para nuestro conjunto de instancias, con las dos opciones de agrupación vistas para el LP gap.

Como se puede apreciar, la diferencia es muy grande, de alrededor de dos órdenes de magnitud. Si bien aún no incluimos el análisis sobre los resultados a nivel del tiempo de cómputo, como veremos más adelante, el mismo se encuentra en el mismo orden de magnitud para ambas formulaciones. Es decir, aún cuando ambas formulaciones tienen tiempos de ejecución similares, Orig realiza una enumeración de nodos muy superior a TIF.

Creemos que la mejora en la relajación de TIF y el hecho de que trabaje con una formulación más ajustada, permite obtener mejores cotas duales y por consiguiente encontrar el óptimo enumerando una cantidad de nodos mucho menor.

Por otro lado, podemos ver cómo los valores de ambas métricas crecen según se incrementa la complejidad de las instancias, es decir, según crecen los parámetros d_e y τ_2 . Al trabajar con instancias de mayor tamaño, el algoritmo de *Branch-and-Cut* debe enumerar una mayor cantidad de nodos hasta su terminación. Esto se condice también con los

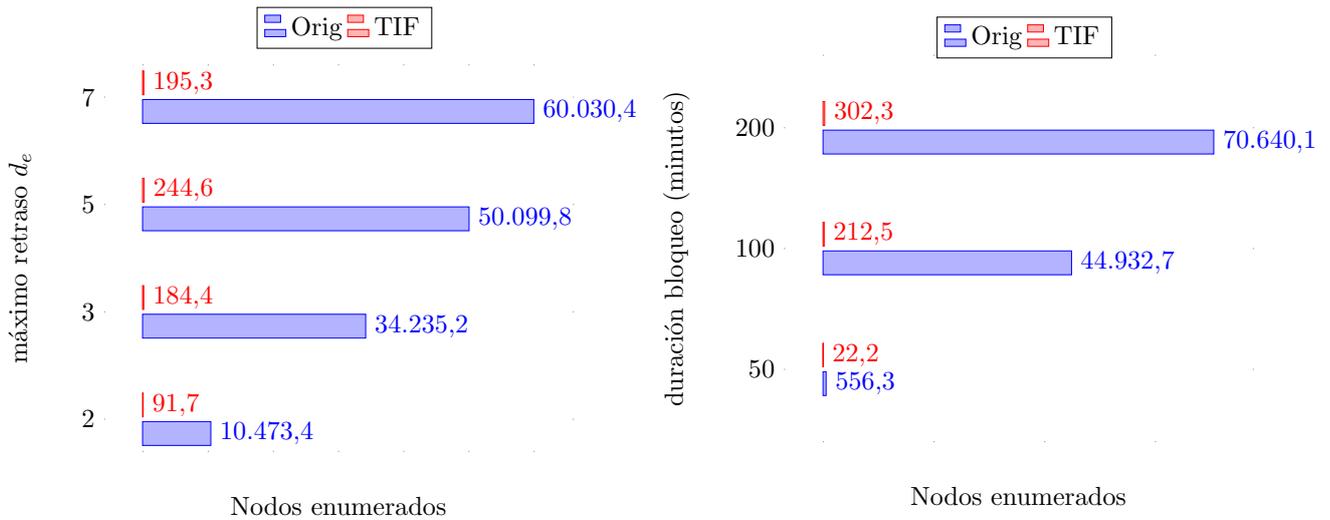


Fig. 5.7: Comparación de nodos enumerados Orig vs TIF. Los resultados se muestran agrupados por el parámetro d_e (izquierda) y la duración del bloqueo (derecha). Para cada valor, se reporta el promedio de la métrica.

resultados observados sobre el LP gap, donde observamos que al incrementar los valores de los parámetros d_e y τ_2 , partimos con un valor de la relajación más lejano al óptimo.

Es interesante notar el caso particular para la formulación TIF donde la cantidad de nodos enumerados para $d_e = 7$ decrece respecto de $d_e = 5$. Creemos que esto se debe al fuerte impacto del parámetro d_e en dicha formulación, el cual incide sobre la cantidad de variables y desigualdades de la misma, aumentando la dificultad de resolver la relajación lineal del problema. Esto tiene impacto principalmente en el cálculo de la relajación en el nodo raíz, que debe calcularse desde el comienzo, pero también en el cálculo de la relajación de cada subproblema, y por consiguiente, la enumeración de nodos del árbol de *Branch-and-Cut* disminuye.

Esto debería verse reflejado en una disminución de la cantidad de instancias resueltas para $d_e = 7$ respecto de $d_e = 5$. Si no fuera así, si bien puede haber otros factores involucrados, sería de esperar que al tratar un problema más difícil se enumeren más nodos, tal como comentamos anteriormente.

| d_e | 2 | 3 | 5 | 7 |
|---------------------------|-----|-----|-----|-----|
| Instancias resueltas Orig | 276 | 252 | 242 | 238 |
| Instancias resueltas TIF | 278 | 275 | 268 | 262 |

Tab. 5.4: Cantidad de instancias resueltas por cada formulación para cada opción del parámetro d_e . Para cada valor de d_e , hay un total de 288 instancias.

En la Tabla 5.4 podemos ver la cantidad de instancias resueltas por la formulación TIF según el valor de d_e . Para cada opción de dicho parámetro, hay un total de 288 instancias del conjunto MITRE. Como podemos ver, para la opción de $d_e = 7$, TIF resuelve menos instancias que para $d_e = 5$, lo que se condice con lo esperado. Pensamos que para dichas

instancias que no resuelve para $d_e = 7$ pero sí para $d_e = 5$, el tiempo de ejecución de las relajaciones es muy elevado, lo que termina disminuyendo la cantidad de nodos enumerados considerablemente, y en consecuencia que el promedio para el conjunto de instancias con $d_e = 7$ quede por debajo de $d_e = 5$. Otro factor que podría incidir al respecto es que trabajar con $d_e = 7$ podría facilitar la obtención de una solución factible en algunas instancias, y que por dicho motivo se terminen enumerando menos nodos en promedio.

Como comentario final sobre esta métrica, debemos aclarar que si bien nuestra formulación TIF enumera una cantidad de nodos mucho menor, esto no significa que el tiempo de cómputo también sea inferior a Orig. Esto se debe a que el tiempo de procesar cada nodo para nuestra formulación puede ser muy superior al tiempo de cómputo en cada nodo para Orig, al tratarse de un modelo con menor cantidad de variables y restricciones. Sin embargo, el resultado no deja de ser valioso.

Creemos que los resultados anteriores sobre LP gap y cantidad de nodos enumerados son muy interesantes. Los mismos motivan el análisis para seguir mejorando estos resultados aún más, por ejemplo, haciendo un estudio de desigualdades válidas, buscando aumentar la calidad sobre las cotas duales. Incluimos algunos detalles sobre estas ideas en el Capítulo 6 de conclusiones y trabajo futuro.

Para finalizar la comparación entre ambas formulaciones nos concentramos en el análisis a nivel del tiempo de cómputo. Como mencionamos en la introducción del Capítulo 4, nuestra investigación se enfocó principalmente en la mejora a nivel de la relajación, la cual fue abordada en base al estudio de la métrica del LP gap. Como comentamos en el Capítulo 1, el contexto de aplicación de nuestro problema es en tiempo real, con lo cual el tiempo de ejecución es un aspecto clave a considerar. Por dicho motivo, es interesante estudiar cómo se comportan ambas formulaciones al respecto y analizar si los resultados favorables que fueron observados para la formulación TIF inciden positivamente en ésta métrica.

Como pudimos corroborar, nuestra formulación posee una mejor relajación, con lo cual el algoritmo *Branch-and-Cut* inicia con una mejor cota dual. Esto podría mejorar los tiempos de cómputo en el caso de encontrar una solución integral óptima, al estar más cerca de probar optimalidad y así poder terminar. Además, al manejarnos con una formulación más ajustada, esto generará mejores relajaciones en cada nodo del árbol, con lo cual se incrementan también las posibilidades de podar el árbol en base a la cota primal. Por otro lado, nuestra formulación TIF trabaja con una cantidad de variables mucho mayor que Orig, esto puede tener un gran impacto en los tiempos de cómputo, por el costo de resolver cada subproblema asociado. Bajo este escenario, es posible que la desventaja inicial que posee Orig respecto del LP gap sea compensada al enumerar más rápidamente.

Previo a analizar los resultados, es importante tener en cuenta la cantidad de instancias resueltas por optimalidad (i.e. sin llegar al límite de 20 minutos) por cada formulación sobre el total de instancias MITRE. Recordemos que nuestro conjunto MITRE está compuesto por 1152 instancias.

Mientras la formulación Orig resuelve 1008 instancias (87.5% del total), nuestra formulación TIF logra resolver 1083 (94% del total). Es decir, nuestra formulación es capaz de resolver más del 6% del total de instancias que la formulación Orig. Consideramos que este resultado es de suma importancia. Aún en el caso en que nuestra formulación obtenga peores resultados que Orig sobre el resto de las instancias, puede ser preferible utilizar TIF

igualmente, dado que resuelve por optimalidad una mayor cantidad. Otra posibilidad, en el caso de que los tiempos de ejecución de TIF sean muy superiores a los de Orig sobre las instancias que esta última si resuelve, es utilizar ambos modelos en paralelo.

Analicemos ahora los resultados para nuestro conjunto MITRE, la Figura 5.8 muestra el promedio del tiempo de cómputo para las 1152 instancias.

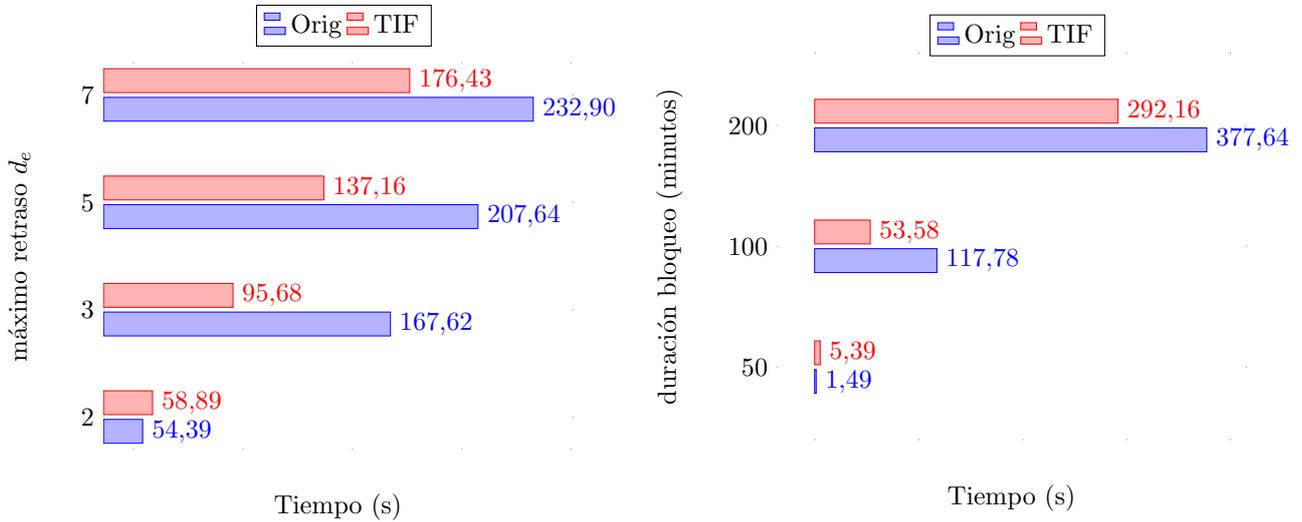


Fig. 5.8: Comparación de tiempo Orig vs TIF. Los resultados se muestran agrupados por el parámetro d_e (izquierda) y la duración del bloqueo (derecha). Para cada valor, se reporta el promedio de la métrica.

Podemos observar que si tomamos el promedio sobre todas las instancias, nuestra formulación tiene un tiempo de cómputo menor que Orig. Esta ventaja es particularmente importante en las instancias de mayor complejidad ($d_e = 7, \tau_2 - \tau_1 = 200$), dado que en dichos casos los tiempos de ejecución se acercan a valores restrictivos para el uso en tiempo real. Además, para estas instancias de mayor dificultad, TIF también logra resolver una mayor cantidad de instancias que Orig, como podemos ver en la Tabla 5.4 incluida anteriormente y en la Tabla 5.5 que agrupa las instancias resueltas por duración del bloqueo.

| Duración bloqueo | 50 | 100 | 200 |
|---------------------------|-----|-----|-----|
| Instancias resueltas Orig | 384 | 360 | 264 |
| Instancias resueltas TIF | 384 | 383 | 316 |

Tab. 5.5: Cantidad de instancias resueltas Orig vs TIF para cada opción de duración del bloqueo. Para cada opción de duración, hay un total de 384 instancias.

Como se puede observar, la ventaja en la cantidad de instancias resueltas de TIF por sobre Orig no se da solo en las instancias de menor tamaño, sino que se mantiene para todas las instancias, y en particular para las de mayor dificultad.

Si bien los resultados de la Figura 5.8 son interesantes, en la misma estamos calculando el promedio del tiempo de cómputo sobre todas las instancias. En particular, como

hemos analizado, hay un subconjunto que TIF logra resolver pero Orig no, llegando al tiempo límite de 1200 segundos e impactando fuertemente sobre el promedio. En base a esto, incluimos en la Figura 5.9 una comparación de tiempos teniendo en cuenta solo las instancias en las cuales ambas formulaciones terminan por optimalidad. Vale aclarar que estas instancias son en total 1002, representando un 87% del total.

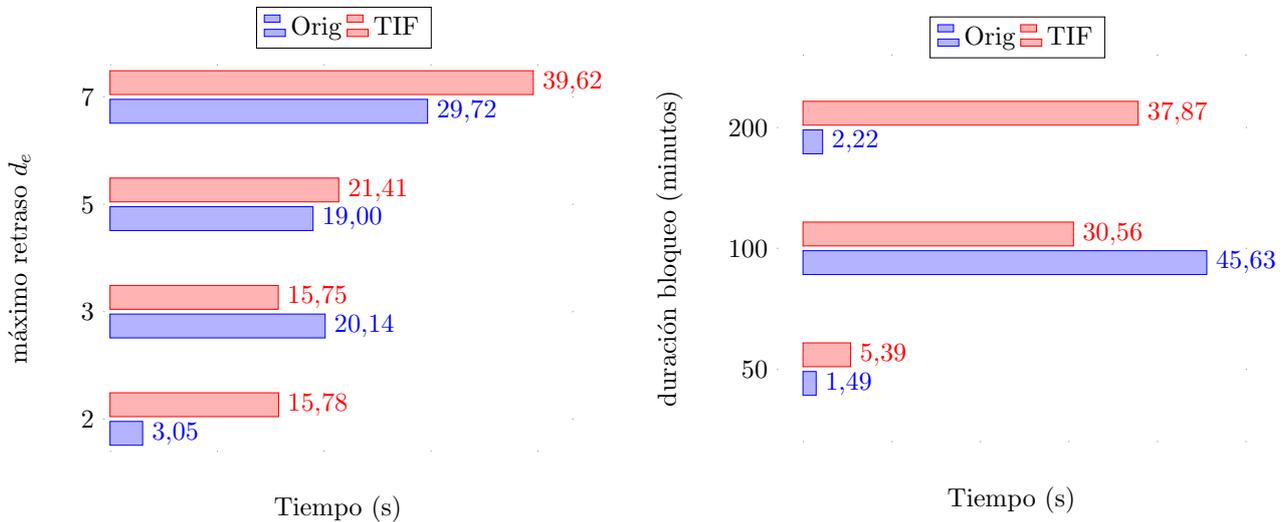


Fig. 5.9: Comparación de tiempo Orig vs TIF para las instancias de MITRE en que ambas formulaciones terminan por optimalidad. Los resultados se muestran agrupados por el parámetro d_e (izquierda) y la duración del bloqueo (derecha). Para cada valor, se reporta el promedio de la métrica.

Como podemos ver, si eliminamos de nuestro análisis a las instancias en donde alguna de las formulaciones no logra encontrar el óptimo, los resultados cambian notoriamente. En general, sobre estas instancias que podemos considerar más fáciles de resolver, Orig obtiene mejores resultados. Igualmente, podemos ver cómo para ciertos casos ($d_e = 3$ y $\tau_2 - \tau_1 = 100$) TIF obtiene mejores resultados incluso cuando nos limitamos sobre este conjunto.

Algo que llamó nuestra atención fue la gran diferencia en tiempo de cómputo para el caso donde el bloqueo tiene 200 minutos de duración. Si bien la métrica para TIF toma valores esperados (crece según crece τ_2), el resultado inesperado fue el bajo tiempo de cómputo para Orig.

Lo primero que validamos fue la cantidad de instancias sobre las cuales se toma el promedio, es decir, de las instancias con bloqueo de 200 minutos, cuantas son terminadas por ambas formulaciones. Las mismas corresponden a 258 de las 384 instancias de ese grupo. De esta manera, vemos que el promedio no se toma sobre un número de instancias demasiado bajo. Por otro lado, verificamos que restringiendo el estudio a las instancias resueltas por ambas formulaciones, TIF sigue manteniendo una diferencia similar en cuanto al LP gap, también para el grupo de $\tau_2 - \tau_1 = 200$.

Lo que si creemos da un buen indicio de lo que sucede, es la cantidad de nodos enumerados sobre estas instancias. Como vimos anteriormente, la diferencia entre ambas formulaciones es muy grande para éste métrica. Sin embargo, si miramos los resultados solo sobre las instancias resueltas por ambas formulaciones, como se muestra en la Figura 5.10,

vemos que mientras para $\tau_2 - \tau_1 = 50, 100$, Orig mantiene la tendencia de una enumeración mucho mayor, para $\tau_2 - \tau_1 = 200$ ambas formulaciones muestran valores similares, incluso con TIF con un promedio mayor.

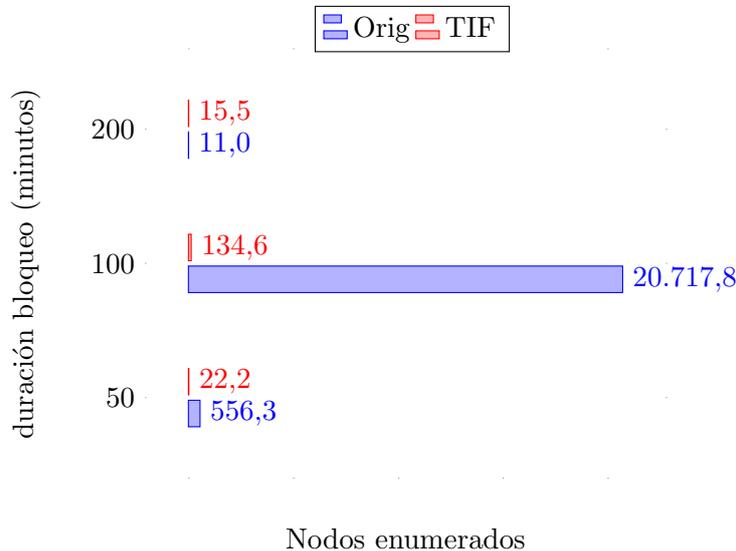


Fig. 5.10: Comparación nodos enumerados Orig vs TIF para las instancias de MITRE en que ambas formulaciones terminan por optimalidad. Los resultados se muestran agrupados por la duración del bloqueo. Para cada valor, se reporta el promedio de la métrica.

Vemos entonces que para estas instancias, Orig aún cuando posee un LP gap mayor que TIF, logra resolver el problema enumerando muy pocos nodos. Como para Orig el peso de resolver cada relajación es mucho menor, esto termina explicando la disminución en los tiempos de cómputo que observamos en la Figura 5.9.

Para poder conocer el motivo de este comportamiento deberíamos hacer un análisis detallado sobre este conjunto de instancias, dado que con resultados agregados es muy difícil poder validar nuestras conjeturas. Sin embargo, nuestra hipótesis es que para este conjunto de instancias de bloqueo de 200 minutos en las cuales ambas formulaciones terminan por optimalidad, Orig logra encontrar cotas primales de muy buena calidad rápidamente y eso le permite cerrar el procesamiento en pocos nodos. Por otro lado, vimos cómo nuestra formulación tarda mucho más tiempo en resolver estas instancias. Sin embargo, del total de 384 instancias de bloqueo de mayor tamaño logra resolver 316 instancias, mientras que Orig solo resuelve 264, tal como vimos en la Tabla 5.5. Creemos que estas instancias que Orig no logra resolver pero TIF si, corresponden a instancias difíciles (por ejemplo, con un valor de $d_e = 7$) dentro del conjunto de bloqueo de 200 minutos, en donde TIF explota su formulación más ajustada y en base a cotas duales más fuertes obtiene mejores resultados. Creemos que las 258 instancias de bloqueo de mayor tamaño que ambas formulaciones resuelven corresponden a instancias más simples, en donde Orig explota su rápida enumeración para obtener mejores resultados que TIF.

En base a lo anterior, uno de los experimentos planteados como trabajo a futuro es la contrastación de esta hipótesis analizando en detalle el recorrido del árbol de *Branch-and-Cut* y las cotas obtenidas progresivamente para las instancias mencionadas.

Para cerrar el estudio sobre el tiempo de cómputo, recopilamos primero el hecho de

que TIF logra resolver más instancias y además, si miramos los tiempos sobre el total de nuestro conjunto de instancias MITRE, vemos que obtenemos mejores resultados. Mirando las diferencias entre las figuras 5.8 y 5.9, vemos como al limitar el estudio sobre las instancias en las cuales ambas formulaciones terminan por optimalidad, Orig muestra mejores resultados. Este es uno de los puntos donde nos interesa enfocar nuestro trabajo futuro, buscando obtener mejores resultados también para estos casos. Tenemos en mente distintas técnicas de PLE para afrontar este tema, que incluyen tanto heurísticas como el estudio de desigualdades válidas, para mejorar la calidad de las cotas primales y duales respectivamente. Incluimos algunos detalles sobre estas ideas en el Capítulo 6 de conclusiones y trabajo futuro. De todas maneras, creemos que es significativo el hecho de que los tiempos de ambos modelos son comparables y factibles para su aplicación en el contexto planteado.

Para cerrar el capítulo de resultados, volvemos a remarcar que todos los experimentos aquí realizados nos marcan tendencias interesantes, pero conforman una primera etapa dentro de un estudio empírico donde hay mucho más terreno por abarcar. El objetivo final de este tipo de modelos es el de poder brindar una solución a problemas en instancias variadas de la vida real.

En este sentido, uno de nuestros objetivos como trabajo a futuro es realizar una serie de experimentos sobre redes ferroviarias de distinto tipo. En el Capítulo 6 incluimos más precisión sobre estas cuestiones.

6. Conclusiones y trabajo futuro

El problema de Gestión de Interrupciones en redes ferroviarias es un problema complejo, pero con un alto potencial de uso en la práctica. Al trabajar a nivel operacional, las limitaciones en los tiempos de cómputo agregan un desafío extra comparado a otros tipos de problemas (e.g. Planificación Táctica).

El problema de Reprogramación de Cronograma comprende una de las etapas fundamentales del mismo, ya que es la etapa inicial en la cual se determinan los nuevos horarios de los servicios, y de la cual dependen las próximas etapas.

Nuestra contribución consistió en extender el trabajo de Veelenturf et al. [12], proponiendo una nueva formulación de PLE para este problema la cual llamamos TIF, dado que incorpora la dimensión del tiempo al modelo.

Como característica principal, podemos destacar que nuestra formulación posee una relajación lineal más ajustada que la formulación original. Al respecto, enunciamos dicha relación en una proposición.

Asimismo, la nueva formulación TIF posee una estructura mucho más rica la cual brinda mucho espacio para aplicar técnicas de PLE.

Se realizó una completa experimentación comparando ambos modelos sobre un caso de estudio de una red ferroviaria del area Metropolitana de Buenos Aires. Sobre estos experimentos, realizados sobre un total de 1152 instancias, podemos destacar

- Nuestra formulación es capaz de mejorar considerablemente el % LP gap, mostrando una mejora aproximada entre 35 % y 50 % dependiendo el caso.
- Nuestra formulación logra resolver por optimalidad aproximadamente un 6.5 % más de instancias que la formulación original.
- Comparando los tiempos de cómputo sobre el total de instancias, TIF obtuvo mejores resultados en promedio. Limitando el análisis a instancias donde ambos modelos resuelven por optimalidad, el modelo original presenta mejores resultados, pero incluso sobre dicho subconjunto nuestra formulación presenta mejores resultados en algunos casos.
- En general, pudimos observar que ambos modelos son comparables y factibles para su aplicación en la práctica.

Finalmente, otro aporte del trabajo es la explicación minuciosa realizada sobre distintos aspectos del modelo para que el mismo capture correctamente escenarios particulares del problema. En particular, sobre las extensiones realizadas en la red evento-actividad y el modelo de PLE para contemplar el bloqueo de vías. Creemos que este es un valioso aporte dada la complejidad del problema y el hecho de que no encontramos este nivel de detalle en la literatura.

En relación a este punto, la dificultad del problema y la necesidad de desgranar distintos detalles del mismo derivó en que algunas de las direcciones en las cuales deseamos enfocar nuestro estudio no puedan ser encaradas en esta etapa. A continuación mencionamos los principales ejes de estudio para considerar a futuro.

Análisis Cualitativo Consideramos que para continuar el estudio sobre el problema es crucial contar con herramientas de visualización. Poder visualizar los cronogramas generados y compararlos contra los originales permitirá comprender la implicancia de parámetros clave como la ubicación y duración del bloqueo. También posibilitará analizar cambios en parámetros potencialmente configurables como el retraso máximo permitido o la penalización por cancelación y en base a esto estudiar como mejorar la calidad de la solución obtenida. En este sentido, otra cuestión deseada es poder contar con métricas a nivel cualitativo con las cuales medir nuestra solución. Por ejemplo, conocer niveles deseados de cancelaciones o retrasos, y comparar bajo este aspecto nuestra solución contra soluciones implementadas mediante planes de contingencia. Creemos que estos temas son de suma importancia para avanzar sobre fases más concretas, pensando en una futura aplicación en la práctica.

Instancias y datos Nos interesa realizar una serie de experimentos sobre un caso de estudio distinto al utilizado en el Capítulo 5. En particular, apuntamos a trabajar con una red no lineal como lo es el ramal Retiro-Tigre. Por otro lado, un objetivo es poder obtener datos reales de bloqueos para probar nuestro modelo utilizando los mismos en lugar de utilizar los escenarios simulados. Otra información que deseamos disponer son los datos operativos de la infraestructura, los mismos no son de carácter público por lo que en nuestros experimentos tomamos los estándares de la literatura en su lugar.

Extensiones al modelo Tal como comentamos al inicio del Capítulo 3, en Veelenturf et al. [12] se incluyen algunas extensiones sobre el problema que no fueron consideradas en nuestro trabajo para abocarnos sobre otros aspectos. En este sentido, nos interesa incorporar dichas características a nuestra formulación TIF, y analizar como se comportan ambos modelos al tenerlas en cuenta. Entre las principales extensiones a tener en cuenta se destaca el traspaso regular de material rodante, que modela como ciertos conjuntos de servicios conforman una *serie* que utiliza las mismas formaciones de material rodante de manera regular. Asimismo, se desea poder manejar distintos tipos de material rodante dependiendo el tipo de servicio (e.g. Servicio Eléctrico, Servicio Diesel). Otra extensión consiste en permitir el desvío de servicios por nuevos recorridos y por último incorporar la característica del *balanceo* del cronograma, donde se busca que la cantidad de trenes cancelados en ambos sentidos sea similar. La incorporación de estas extensiones permitiría aplicar el modelo en una mayor cantidad de escenarios.

En cuanto a un aspecto más teórico, pensando en posibles extensiones sobre la formulación TIF, podemos destacar

Desigualdades válidas Tenemos en mente distintas familias de desigualdades válidas para incorporar al *Branch-and-Cut* aprovechando la estructura por índice de tiempo de TIF. Por ejemplo, una idea es explotar la relación entre el número de recursos de un evento de inventario y la cantidad de eventos usando dichos recursos en un instante de

tiempo dado. Sabemos que la cantidad de estos eventos no puede superar la capacidad del inventario. Incorporando algunas definiciones y utilizando las variables x_e^t y z_a^t se pueden incorporar estas desigualdades, con el objetivo de mejorar aún más la calidad de las cotas duales obtenidas.

Preprocesamiento Otra manera de aprovechar la estructura de TIF es añadiendo reglas de preprocesamiento. Una posibilidad concierne al ajuste de las ventanas de tiempo W_e , estudiada en Ascheuer et al. [1] para el TSPTW. Si bien a diferencia del clásico problema en nuestro caso tenemos múltiples recursos y no todos los eventos deben ser visitados (por la cancelación de trenes), creemos que las ideas de dicho trabajo se pueden adaptar a nuestro problema. En dicho caso, fijando los valores de las variables x_e^t de TIF se pueden modificar fácilmente los límites de la ventana del evento e .

Heurísticas Nos interesa estudiar heurísticas para incorporar al *Branch-and-Cut*. El aporte de nuestra formulación TIF parte de la mejora en la relajación lineal y, por consiguiente, de las cotas duales obtenidas. Luego, consideramos que desarrollar una heurística para obtener cotas primales de buena calidad puede mejorar los resultados considerablemente. Por otro lado, tal como analizamos en la experimentación, pudimos ver que sobre la mayor parte de las instancias la formulación original tiene en promedio mejores tiempos de cómputo. Una teoría que tenemos es que dicha formulación, aprovechando su rápida enumeración de nodos, obtiene cotas primales de buena calidad en poco tiempo, contrario a lo que sucede con TIF. En caso de confirmarse esta tendencia, una posibilidad para generar una heurística es correr en primera instancia el modelo original, obtener una solución factible de buena calidad, y luego utilizarla como solución inicial para TIF. La idea es aprovechar que TIF tiene una relajación más ajustada, y de esa manera explotar los beneficios de ambas formulaciones.

Bibliografía

- [1] N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut. *Mathematical Programming*, 2001.
- [2] N. Besinovic, V. Cacchiani, T. Dollevoet, R. M. Goverde, D. Huisman, M. P. Kidd, L. G. Kroon, E. Quaglietta, J. Rodriguez, P. Toth, et al. Integrated decision support tools for disruption management, 2015.
- [3] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63(0):15–37, 2014.
- [4] S. Dash, O. Günlük, A. Lodi, and A. Tramontani. A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 24(1):132–147, 2012.
- [5] T. Dollevoet, D. Huisman, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar. Application of an iterative framework for real-time railway rescheduling. *Computers & OR*, 78:203–217, 2017.
- [6] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Society*, 64:275–278, 1958.
- [7] M. Grötschel, M. Jünger, and G. Reinelt. Calculating exact ground states of spin glasses: A polyhedral approach. pages 325–353. 1987.
- [8] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [9] I. Louwerse and D. Huisman. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, 235(3):583–593, 2014.
- [10] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.*, 33(1):60–100, Feb. 1991.
- [11] M. Schachtebeck and A. Schöbel. To wait or not to wait—and who goes first? delay management with priority decisions. *Transportation Science*, 44(3):307–321, 2010.
- [12] L. P. Veelenturf, M. P. Kidd, V. Cacchiani, L. G. Kroon, and P. Toth. A Railway Timetable Rescheduling Approach for Handling Large-Scale Disruptions. *Transportation Science*, 2015.
- [13] L. A. Wolsey. *Integer programming*. Wiley-Interscience, New York, NY, USA, 1998.