



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

REALIZACIÓN DE UN MÉTODO DE GENERACIÓN DE COLUMNAS PARA
RESOLVER UN PROBLEMA OPERACIONAL INTEGRADO DE PRODUCCIÓN Y
DISTRIBUCIÓN APLICADO AL PROCESO DE QUIMIOTERAPIAS A DOMICILIO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

PABLO ALBERTO HERNÁNDEZ MUÑOZ

PROFESOR GUÍA:
JAIME ZÚÑIGA CASTRO

MIEMBROS DE LA COMISIÓN:
CHARLES THRAVES CORTÉS-MONROY
RODOLFO URRUTIA URIBE

Este trabajo ha sido parcialmente financiado por INRIA

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL
POR: PABLO ALBERTO HERNÁNDEZ MUÑOZ
FECHA: DICIEMBRE 2019
PROF. GUÍA: JAIME ZÚÑIGA CASTRO

REALIZACIÓN DE UN MÉTODO DE GENERACIÓN DE COLUMNAS PARA RESOLVER UN PROBLEMA OPERACIONAL INTEGRADO DE PRODUCCIÓN Y DISTRIBUCIÓN APLICADO AL PROCESO DE QUIMIOTERAPIAS A DOMICILIO

Este informe presenta una descripción del trabajo conjunto realizado con el centro de investigación INRIA en Lille, Francia bajo la supervisión de Maxime Ogier y Diego Cattaruzza. Dado que el cáncer representa una de las principales causas de muerte tanto en Francia como en Chile, se plantea avanzar en el desarrollo de una herramienta de ayuda a la toma de decisión en la realización de quimioterapias a domicilio. Debido a la alta perención de los medicamentos utilizados en dicho proceso, se plantea el desafío de integrar la producción de dichos medicamentos con el problema de trazado de rutas de vehículos. Dicho problema corresponde a un Problema Operacional Integrado (POI).

Luego de una revisión bibliográfica del estado del arte de la resolución de POIs, se describe e implementa una metodología de *Branch-and-Price* basada en el paradigma de generación de columnas para la resolución del problema de distribución. Posteriormente, se describe el problema de producción de los medicamentos utilizados por los tratamientos de quimioterapia y se propone un modelo integrado en el cual se resuelven ambos problemas de forma conjunta. Las columnas que deben ser generadas para la resolución del POI, corresponden, por un lado a rutas de distribución realizadas por los enfermeros, los cuales visitan a los pacientes y, por otro lado a programas de producción que deben realizar los químicos farmacéuticos, a los cuales se les asignan tareas de producción de medicamentos.

Con el objetivo de evaluar la calidad de la solución propuesta, se proponen algoritmos de etiquetado para generar columnas acordes al nuevo modelo planteado y se generan resultados numéricos a partir de las instancias de Solomon. Ambos algoritmos son resueltos de forma heurística limitando la cantidad de etiquetas por nodo y se resuelven de forma exacta al final del método de generación de columnas.

Para el caso del algoritmo de generación de rutas, 18 de las 29 instancias de 100 nodos cada una fueron resueltas de forma exacta dentro de 300 segundos. El algoritmo de generación de programas de producción, en 300 segundos, sólo fue capaz de resolver 4 de 29 instancias constituidas de los 15 primeros nodos de cada una de las instancias de Solomon. Debido a lo anterior, el POI fue resuelto tomando en cuanto solamente los 15 primeros nodos. De las 29 instancias probadas, 17 pudieron ser resueltas dentro de un tiempo límite de 3600 segundos.

El algoritmo de etiquetado modificado para el trazado de rutas dentro de un contexto integrado resulta eficiente y es posible utilizarlo para instancias grandes. No obstante, el bajo rendimiento del algoritmo de *pricing* de programas utilizado durante el método de generación de columnas, resulta una limitación para la resolución del POI. Otros enfoques para resolver el problema de generación de programas deben ser estudiados para entregar una solución apta para un caso real.

ABSTRACT OF THE DISSERTATION TO OBTAIN
THE DEGREE OF INDUSTRIAL ENGINEER
BY: PABLO ALBERTO HERNÁNDEZ MUÑOZ
DATE: DECEMBER 2019
SUPERVISOR(S): JAIME ZÚÑIGA CASTRO

REALIZACIÓN DE UN MÉTODO DE GENERACIÓN DE COLUMNAS PARA
RESOLVER UN PROBLEMA OPERACIONAL INTEGRADO DE PRODUCCIÓN Y
DISTRIBUCIÓN APLICADO AL PROCESO DE QUIMIOTERAPIAS A DOMICILIO

The current dissertation presents the work performed in INRIA research centre located in Lille, France under the supervision of professors Maxime Ogier and Diego Cattaruzza. Since cancer is one of the main causes of death in France as well as in Chile, the development of a decision support model for the at-home chemotherapy process is described in this report. Due to short life constraints of the drugs used for this purpose, it is suggested the integration of the production process of these drugs with the vehicle routing problem. This problem corresponds to an Integrated Operational Problem (IOP).

After a literature review of the state of the art for solving IOPs, a Branch-and-Price methodology based on the column generation paradigm is described and implemented for solving the distribution problem. Subsequently, the problem of drug production used by chemotherapy treatments is described and an integrated model is proposed in which both problems are solved simultaneously. The columns generated during the resolution of the IOP, correspond, on the one hand, to distribution routes performed by the nurses who visit the patients and, on the other hand, to production schedules that must be assigned to the pharmacists, to whom are assigned the tasks of drug production.

In order to evaluate the quality of the proposed solution, two labelling algorithms are proposed to generate columns according to the new model. Numerical results are generated from Solomon instances. Both algorithms are solved limiting the quantity of labels per node making it a heuristic method. They are both solved exactly at the end of the column generation method.

In the case of the route generation algorithm, 18 of the 29 instances of 100 nodes each were solved within 300 seconds. The production schedule generation algorithm, within 300 seconds, was only able to solve 4 of 29 instances of 15 nodes. Due to the above, the IOP was solved considering only the first 15 nodes. Among the 29 tested instances, 17 of them were solved within a time limit of 3600 seconds.

The modified labelling algorithm for route generation within an integrated context resulted very efficient and may be used for large instances. However, the low performance of the pricing algorithm for scheduling used during the column generation method is a limitation for the resolution of the IOP. Different approaches to solve the scheduling problem should be studied to deliver a suitable solution for a real case.

Al pueblo chileno, por su incansable energía para luchar por una vida digna.

Agradecimientos

Esta memoria, por su carácter internacional, no puedo haber sido posible sino gracias al apoyo y la disposición de quienes trabajan en las dos instituciones que me han formado como ingeniero: la FCFM de la Universidad de Chile y la École Centrale Paris, y en la institución que me acogió como pasante a lo largo de la realización de esta investigación: el centro de investigación del INRIA en Lille. Particularmente, quisiera agradecer a: Macarena Zárate, sub jefa docente del DII de la FCFM y a Mónica Valdebenito, secretaria de la Unidad de Titulación del DII de la FCFM, por haber hecho posible esta encrucijada administrativa. A Fernando Ordóñez, antiguo director del DII, por haber apoyado mi postulación al INRIA. A mi profesor guía, Jaime Zúñiga por su compromiso con esta memoria y a mi co-guía, Charles Thraves, por sus comentarios y correcciones de este trabajo. A Christophe Gallon, director de la organización de la escolaridad de la École Centrale Paris, quien siempre estuvo disponible para el envío de los documentos legales necesarios en Francia. A Maxime Ogier y Diego Cattaruzza, investigadores del grupo INOCS del INRIA, quienes idearon esta memoria, y que estuvieron siempre presentes cuando tenía dudas o la inspiración no se hacía presente. Y, desde luego, a todo el equipo de INOCS por sus consejos, almuerzos, risas y cervezas juntos, en especial a Pablo Escalona y a Luis Salazar por hacerme sentir como en casa estando tan lejos de ella.

No menos importante, es el apoyo que recibí por parte de mi familia y de mis amigos. Sin lugar a dudas, este trabajo no se hubiera realizado sino fuera por la mediación de Rubén, quien me puso en contacto con la gente del INRIA, gracias por esas incontables salidas franco-parlantes. También a Claudio, por haber logrado juntos una memoria en el extranejero. Quisiera agradecer a mi madre y a mi padre, por su incondicional apoyo, tanto financiero como emocional en todas mis aventuras fuera del país. A mi hermana, titulada del DII, quien me ha ayudado con los procesos administrativos. Agradezco a mis compañeros de generación, Felipe, Franco, Memo, Edward, Álvaro, Martín, Seba M. y Seba O. por las juntas *geek* y por hacer más ameno mi paso por la facultad. A mis compañeros en el último año de estudios, Javiera, Dennis, Camila y Gabo, por ayudarme a readaptarme al ritmo Beauchefiano. A todos mis compañeros de la École Centrale Paris de The K-Brass Band, porque aún con la distancia y el tiempo siguen siendo amigos con la que puedo contar. A toda la gente que conocí en Lille mientras realizaba mi pasantía y que me recibieron calurosamente. Finalmente, quisiera agradecer a mis tres amigas parisinas más cercanas, por hacerme sentir bien acogido en la ciudad luz, a Coralie por las fiestas en sus casa, a Anne por esta amistad bilingüe y en especial a Louise, por todo lo que vivimos juntos, tanto en Francia como en Chile, por tu apoyo, tu comprensión y tu energía.

Tabla de Contenido

Índice de Tablas	vii
Índice de Ilustraciones	viii
Introducción	1
1. Problema operacional de las quimioterapias a domicilio	3
1.1. Planteamiento del Problema	3
1.2. Justificación del Problema	4
1.2.1. Ejemplo simplificado a baja escala	5
1.3. Objetivo General	8
1.4. Objetivos Específicos	8
2. Revisión bibliográfica del problema operacional de producción y distribución	9
2.1. Trazado de Rutas de Vehículos	9
2.2. Problema Operacional Integrado	10
2.3. Metodología	11
2.4. Alcances	12
3. Modelo del problema operacional integrado	13
3.1. Descripción del problema de distribución	13
3.1.1. Branch-and-Price para la resolución del VRPTW	14
3.2. Descripción del problema de producción	17
3.2.1. Modelo de programación lineal	19
3.3. Modelo integrado	20
3.3.1. Dual del modelo integrado	22
4. Un método de generación de columnas para resolver el POI	24
4.1. El problema de distribución	24
4.1.1. Algoritmo de etiquetado	25
4.1.2. Generación de columnas	27
4.2. Integración del problema de producción	28
4.2.1. Modificación del algoritmo de etiquetado de rutas	28
4.2.2. Algoritmo para la resolución del problema de pricing de producción	34
5. Resultados	38

5.1. Instancias de prueba	38
5.2. Resolución de sub-problemas	39
5.2.1. Ejecución del algoritmo de etiquetado modificado para el trazado de rutas factibles	40
5.2.2. Ejecución del algoritmo de etiquetado modificado para la generación de programas factibles	41
5.3. Resolución del Problema Operacional Integrado	42
5.3.1. POI para instancias de 10 nodos	43
5.3.2. POI para instancias de 15 nodos	45
Conclusión y trabajo futuro	47
Siglas	49
Glosario	50
Bibliografía	51
A. Soluciones del ejemplo en sección 1.2.1	53
B. Notation Table	54
C. Algorithms	56
C.1. VRPTW Initial Columns	56
D. Resultados de la implementación del etiquetado de rutas	57
D.0.1. Resolución del sub-problema	57
D.0.2. Resolución del MP relajado	58
D.0.3. Resolución del MP entero	58
E. Proofs	60
E.1. Modified algorithm start time proof	60
F. IOP routing pricing problem results	66
G. IOP scheduling pricing problem results	69

Índice de Tablas

1.1. Comparación de soluciones para el ejemplo 1.2.1	7
5.1. Intervalos de generación de valores duales aleatorios	39
5.2. Tiempo de resolución del algoritmo modificado de trazado de rutas	40
5.3. Tiempo de resolución del algoritmo modificado de generación de programas .	41
5.4. Resultados agrupados de la resolución del POI	43
5.5. Resultados de la resolución del POI para 10 nodos	44
5.6. Rutas para R104	45
5.7. Programas para R104	45
5.8. Resultados de la resolución del POI para 15 nodos	46
A.1. Solución 1: Enfermero	53
A.2. Solución 1: Farmacéutico	53
A.3. Solución 2: Enfermero	53
A.4. Solución 2: Farmacéutico	53
B.1. Notations	55
D.1. Tiempo de resolución promedio para el ESPPRC	57
D.2. Tiempo de resolución promedio para el VRPTW relajado	58
D.3. Tiempo de resolución promedio para el VRPTW	59
F.1. Resolution times for the routing IOP pricing problem with 25 clients.	66
F.2. Resolution times for the routing IOP pricing problem with 50 clients.	67
F.3. Resolution times for the routing IOP pricing problem with 100 clients.	68
G.1. Resolution times for the scheduling IOP pricing problem with 10 clients.	69
G.2. Resolution times for the scheduling IOP pricing problem with 12 clients.	70
G.3. Resolution times for the scheduling IOP pricing problem with 15 clients.	71

Índice de Ilustraciones

1.1. Grafo no dirigido representando un ejemplo simple del problema de quimioterapias a domicilio	6
4.1. Ejemplo de ruta con tiempos de visita más tempranos	30
4.2. Ejemplo de ruta con tiempos de visita más tardíos	31
4.3. Ejemplo de ruta con tiempos de visita óptimos	31
4.4. Ejemplo de programa con tiempos de asignación tempranos	36
4.5. Ejemplo de programa, con mejora en la calidad de la solución	36
4.6. Ejemplo de programa, con tiempos de asignación óptimos	36

Introducción

Según un estudio realizado por *Santé publique France* [4], hoy en día el cáncer constituye la principal causa de muerte en Francia. Durante el año 2013, correspondió al 28,8 % de las causas de muerte y para el año 2017, según un reporte entregado por el *Institut National du Cancer* [3], el número estimado de personas fallecidas por algún tipo de cáncer ascendió a 150.000.

Actualmente, los principales tratamientos que existen contra este tipo de enfermedades corresponden a: Cirugías, Quimioterapias y Radioterapias. De estos tratamientos, en el año 2015 la quimioterapia en Francia correspondió a un 39 % de la actividad hospitalaria total en el sector oncológico, siendo el segundo tratamiento más utilizado contra este tipo de enfermedades, luego de la cirugía.

Las hospitalizaciones ligadas al diagnóstico, tratamiento y seguimiento del cáncer representan 5,9MM € de gastos anuales para los establecimientos MCO (medicina, cirugía y obstetricia). Dichas sumas corresponden a 2.538.329 sesiones o consultas y a 258.429 estadias en el hospital en 2016. Además del sector MCO, se le debe sumar el sector de hospitalización a domicilio (HAD). Si bien desde el año 2011 este sector ha presentado un aumento en sus atenciones en Francia, según un estudio del *Institut National du Cancer* [1], en el año 2014 representaba solamente el 5 % del total de quimioterapias realizadas durante todo el año. Cabe destacar que las personas que reciben este tratamiento en HAD, tienen una edad mediana más avanzada (66 años) que en el sector MCO (64 años).

Con respecto a Chile, según datos entregados por el *Intituto Nacional de Estadísticas* [2], el cáncer constituye la segunda causa de muerte en todo el territorio nacional, con un 25 % del total de las causas de muerte, después de las enfermedades del sistema circulatorio, las cuales representan un 27,1 % de estas. De acuerdo al reciente plan entregado por el Ministerio de Salud del Gobierno de Chile [11], se proyecta que el cáncer pase a ser la principal causa de muerte en Chile dentro de los próximos 10 años.

De esta forma, tanto en Francia como en Chile, actualmente el cáncer representa una cantidad bastante considerable de muertes al año y ambos Estados han concentrado grandes esfuerzos, tanto legislativos como monetarios, para tratar de combatir este problema de salud. En el mundo, el cáncer es causa de millones de muertes al año y las comunidades científicas de todo el planeta han dedicado gran parte de sus recursos en mejorar tanto los tratamientos oncológicos como la gestión de los centros asistenciales que tratan con este tipo de enfermedades.

Sumado a lo anterior, el área de investigación de operaciones y cadena de producción se ha interesado bastante en los problemas de gestión dentro del área de salud y diversos artículos haciendo mención a esto son posibles de encontrar en la literatura académica. Uno de los principales problemas de los que se trata en la literatura es el problema de distribución. Dicho problema corresponde al modelamiento mediante redes de distribución de las rutas necesarias para entregar asistencia a los pacientes. Por otro lado, el problema de cadena de producción para el caso de la quimioterapia posee variadas restricciones, lo cual hace aún más difícil obtener una referencia, dentro del estado del arte actual, que satisfaga las necesidades del problema de las quimioterapias.

Desde hace 2 décadas el problema integrado de cadena de producción y de distribución ha permitido obtener notables reducciones de costo [13], a diferencia de los problemas resueltos de manera clásica, los cuales abordan los problemas de producción y de distribución por separado. De esta forma, este informe detalla las razones para abordar el problema de las quimioterapias en el sector HAD como un problema integrado así como una revisión al estado del arte actual de este tipo de problemas.

El presente trabajo fue realizado en las inmediaciones del centro de investigación *Institut National de Recherche en Informatique et Automatique* (INRIA), en su sede ubicada en Lille, Francia. La investigación fue realizada bajo la supervisión de los profesores Maxime Ogier y Diego Cattaruzza, quienes forman parte del grupo *Integrated Optimization with Complex Structure* (INOCS) y que guiaron el desarrollo continuo del trabajo que se presenta a lo largo de este informe.

Capítulo 1

Problema operacional de las quimioterapias a domicilio

Dentro del contexto de las quimioterapias en el sector HAD en Francia, se busca entregar una herramienta que aporte tanto al bienestar de los pacientes¹ como al de los trabajadores del área de la salud. Hasta el día de hoy, los establecimientos de salud no cuentan con ninguna herramienta que optimice los tiempos del proceso de quimioterapias a domicilio, por lo cual se plantea desarrollar un algoritmo que permita disminuir los tiempos de realización de este proceso.

1.1. Planteamiento del Problema

Hasta el día de hoy, no existen herramientas de ayuda a la decisión aptas para mejorar el proceso de quimioterapias en HAD, por lo que aún se realiza a muy pequeña escala (sólo un 5% del total de quimioterapias durante el año 2014). Es por esto que se plantea realizar un estudio de dicho problema operacional y poder entregar una herramienta que ayude a disminuir los tiempos del proceso de quimioterapias a domicilio.

El proceso de quimioterapias en HAD estudiado consta de tres grandes etapas que se presentan a continuación:

1. El **hospital** solicita la fabricación de los medicamentos que se utilizarán para las quimioterapias,
2. Los **químico farmacéuticos** fabrican los medicamentos solicitados durante la etapa anterior, dentro del mismo establecimiento médico,
3. Luego de reunir los medicamentos, los **enfermeros** realizan las quimioterapias en el domicilio del paciente.

¹El género masculino es utilizado a lo largo de este informe para denotar de forma indistinta al grupo de personas que juegan un rol dentro del problema abordado.

Dentro de este proceso se pueden distinguir dos problemas diferentes. El primero corresponde a un problema de producción de medicamentos, en el cual se desea producir a tiempo los medicamentos necesarios para la realización de las quimioterapias. El segundo es el problema de distribución, correspondiente a un problema de optimización estudiado desde hace ya varias décadas y teniendo múltiples variantes dependiendo de las características del problema a tratar. Ambos problemas son conocidos dentro del ámbito de la ingeniería y ciencias, sin embargo, para este caso particular, se deben tomar en cuenta las siguientes restricciones:

- El enfermero debe esperar a que todos los medicamentos estén listos antes de realizar la ruta.
- Los medicamentos son altamente perecibles (duración del orden de algunas horas), por lo que no basta con determinar tan solo la cantidad, sino que también se debe considerar el momento en el cual producirlos.

Dadas las restricciones mencionadas anteriormente, es necesario integrar el problema de producción de los medicamentos con el problema de distribución. De esta forma, se hablará de un Problema Operacional Integrado (POI) para referirse a la resolución conjunta de ambos problemas íntimamente ligados. Una descripción detallada del proceso de fabricación de medicamentos, así como una solución al problema de producción de medicamentos dentro de un contexto de quimioterapias en MCO, se puede encontrar el trabajo realizado por Y. Kergosien [19].

1.2. Justificación del Problema

Los tratamientos de quimioterapia a domicilio son relativamente nuevos y, en lo que respecta a la literatura, aún no existen suficientes fuentes de información que permitan determinar con exactitud los beneficios o desventajas de la realización de quimioterapias en el domicilio de los pacientes. A principios de los años 2000 recién se estaba llegando a un consenso en la factibilidad de que los tratamientos puedan realizarse fuera de un establecimiento de salud [6] y se discutían aún sus posibles beneficios a futuro.

La evidencia muestra que, en las evaluaciones en que prima lo cualitativo frente a lo cuantitativo, estas sí concuerdan en que los pacientes se sienten beneficiados por la posibilidad de seguir sus tratamientos oncológicos en la comodidad de sus casas [8]. Sin embargo, es necesario tomar en cuenta que, tanto las evaluaciones cuantitativas como cualitativas, dependen del contexto social de cada región en la que se realizan los tratamientos. En trabajos más recientes, como el realizado por el *National Institute for Health Research* en Inglaterra el año 2015 [7], la variable regional es analizada, pero sin avanzar más en lo que ya se planteado hasta ahora. No obstante, dicho trabajo confirma de forma concreta los beneficios comunitarios que tiene la quimioterapia en HAD.

Es importante recalcar que, si bien los resultados económicos disponibles actualmente en su mayoría no son comparables y una gran parte de estos son desestimados por ser considerados como estudios sesgados, el desarrollo de una herramienta de soporte para la realización de quimioterapias a domicilio significa un avance para una evaluación económica más certera.

De forma general, la hospitalización a domicilio ha probado ser una buena alternativa para la disminución de costos en los hospitales y para la reducción de la congestión [5]. La entrega de modelos de programación lineal mixta para resolver problemas de optimización ligados a la HAD aportan al funcionamiento tanto administrativo como financiero de los establecimientos de salud. Siguiendo este enfoque, se estima que la entrega de herramientas que permitan controlar los procesos dentro de esta área aporta tanto a una evaluación fiable de los costos de operación de los hospitales como al desarrollo social de las comunidades ligadas a dichos procesos.

Para entender de mejor manera el problema es necesario detallar ciertos puntos. Primero que todo, la mayoría de los medicamentos tienen una duración de solo algunas horas, las cuales pueden variar dependiendo del tipo de medicamento. La producción de dichos medicamentos es bastante compleja y, para el caso de los medicamentos que se entregan a través de inyecciones intravenosas, para ciertos medicamentos se recomienda realizar el tratamiento dentro de las 4 horas luego de haber sido fabricados [10]. En segundo lugar, se debe tomar en cuenta que la planificación es diaria, es decir que los horarios de producción y las rutas a seguir son calculados para cada día. Esto implica que cualquier herramienta entregada para ayudar a la gestión de las quimioterapias en HAD debe ser lo suficientemente eficiente como para entregar una solución en un tiempo razonable dentro de este horizonte de tiempo.

1.2.1. Ejemplo simplificado a baja escala

Para ejemplificar la razón por la cual es necesario tratar este problema como un POI, se plantea un ejemplo a muy baja escala en el cual se consideran tan solo 3 pacientes que deben ser tratados con un mismo medicamento. La producción de los medicamentos se realiza en el hospital y se desestiman los tiempos de carga entre el fin de la producción y el empaquetado en el vehículo. Los datos para dicho ejemplo se presentan a continuación:

- Tiempo de estabilidad del medicamento = 6 horas ²
- Duración del tratamiento = 2 horas
- Tiempo de producción = 2 horas
- Jornada de 6AM a 12AM
- 3 pacientes
- 1 enfermero
- 1 químico farmacéutico

Para simplificar el problema de producción, se considera que los medicamentos pueden ser producidos en paralelo por el mismo farmacéutico. Para el caso real se debe tomar en cuenta la ocupación de los operadores. Otra simplificación importante es que para una aplicación real los pacientes tienen tiempos de disponibilidad limitados o “ventanas de tiempo”, dentro de los cuales se debe realizar el tratamiento.

²El **tiempo de estabilidad** de un medicamento corresponde a la duración del medicamento luego de haber sido producido. Pasado el tiempo de estabilidad, el medicamento comienza a perder sus características químicas e incluso puede volverse nocivo.

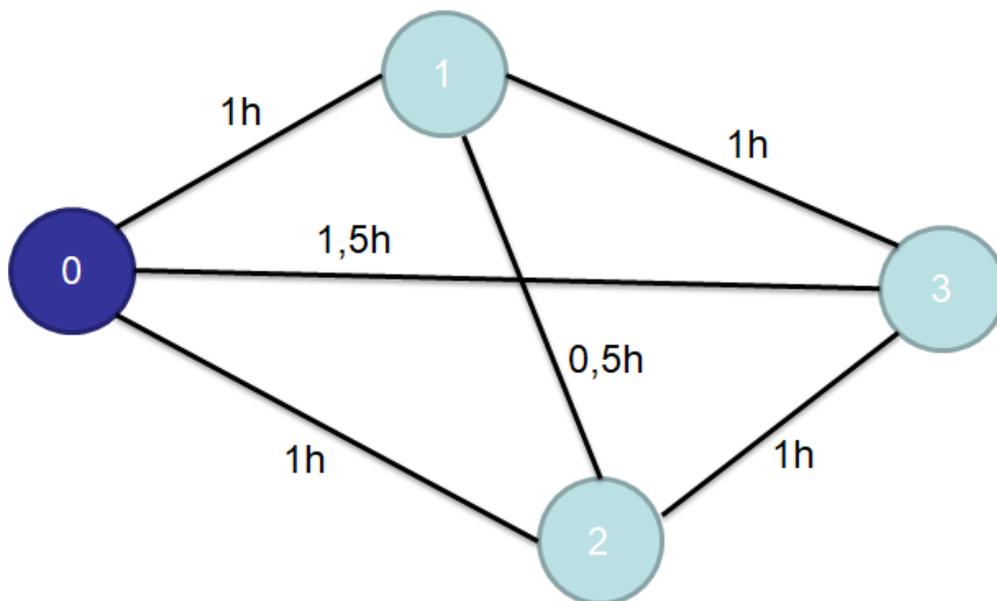


Figura 1.1: Grafo no dirigido representando un ejemplo simple del problema de quimioterapias a domicilio

En la Figura 1.1 se observa un grafo no dirigido que entrega una representación del problema de distribución de las quimioterapias en HAD. Sean V_i con $i \in \{0, 1, 2, 3\}$, los nodos del grafo que representan al hospital y a los domicilios de los pacientes. Sea el nodo V_0 el hospital, es decir, el nodo origen del problema a resolver. Sea E_{ij} el arco que une los nodos V_i y V_j , donde $(i, j) \in \{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ e $i \neq j$. Sea c_{ij} el costo de utilizar el arco E_{ij} , el cual corresponde al número sobre el arco correspondiente en la Figura 1.1.

En el caso en que cada uno de los problemas sea resuelto por separado, para el problema de producción, el farmacéutico produce los 3 medicamentos de forma conjunta al principio de su jornada laboral. Para el problema de distribución, este corresponde al problema del vendedor viajero, en el cual se busca reducir el tiempo total de la ruta recorriendo todos los nodos del grafo y partiendo y terminando en el nodo origen, en este caso, el hospital.

Como el único enfermero debe cubrir los 3 tratamientos que se modelan en la Figura 1.1, el farmacéutico produce los 3 medicamentos de forma paralela antes de que el enfermero pueda partir, esto para simplificar el problema de producción. Así, a las **6AM** el químico farmacéutico comienza la producción de los medicamentos. A las **8AM** se termina la fabricación de los medicamentos y el enfermero puede trazar su ruta. Mediante un algoritmo voraz del vecino más próximo, es posible encontrar el camino más corto³ para cubrir todos los nodos. Esta aproximación corresponde a tomar los arcos con menor costo desde el origen y hasta cubrir todos los nodos del grafo.

A partir del nodo V_0 es posible tomar ya sea el arco E_{01} ó el arco E_{02} . Por orden de procesamiento, se toma el arco E_{01} , el cual tiene un costo de $c_{01} = 1$ hora, por lo que el enfermero llega al domicilio del paciente a las **9AM**, hora en la cual el tiempo de estabilidad

³En la mayoría de los casos, la aproximación del vecino más próximo no encuentra la solución óptima, sin embargo, basta con encontrar todos los caminos posibles dentro del grafo para comprobar que, para este ejemplo simplificado, dicho algoritmo sí encuentra la solución óptima.

restante de los medicamentos es de 5 *horas*. La realización de la quimioterapia tiene una duración de 2 *horas*, por lo tanto el enfermero puede recién emprender la próxima ruta a las **11AM**, hora en la cual los 2 medicamentos restantes tienen un tiempo de estabilidad de 3 *horas*. El camino próximo corresponde al arco E_{12} , llegando, el enfermero, a las **11:30AM**, dejando un tiempo de estabilidad de 2,5 *horas* para los 2 medicamentos. Luego de realizar el tratamiento, a las **1:30PM** el tiempo de estabilidad del último medicamento es de tan solo 0,5 *hora* y el costo c_{23} para llegar al último nodo es de 1 *hora*, por lo cual el enfermero debe volver al hospital a pedir un nuevo medicamento y el medicamento perecido es desechado.

Incluso siendo un problema simplificado y a muy baja escala, la solución óptima no es tan evidente como se podría esperar. Siguiendo el ejemplo anterior, pero esta vez integrando el problema de producción con el problema de distribución, se tienen dos soluciones factibles que son interesantes de analizar. A saber, una primera solución factible corresponde la ruta $E_{01}, E_{12}, E_{20}, E_{03}$ con las horas de producción **6AM** y **12PM**, la segunda solución corresponde a la ruta $E_{01}, E_{10}, E_{02}, E_{23}$ con las horas de producción **6AM** y **10AM**. El desarrollo de ambas soluciones se puede encontrar detallado en el anexo A.

Resultado	Solución 1	Solución 2
Ocupación Farmacéutico	4 <i>horas</i>	4 <i>horas</i>
Ocupación Enfermero	10 <i>horas</i>	10 <i>horas</i>
Término Farmacéutico	2PM	12PM
Término Enfermero	6PM	6PM

Tabla 1.1: Comparación de soluciones para el ejemplo 1.2.1

Como se puede ver en la Tabla 1.1, si bien ambas soluciones presentan resultados que, en términos del tiempo de ocupación, consideran la misma cantidad de horas (10 *horas* para el enfermero y 4 *horas* para el farmacéutico), el químico farmacéutico se desocupa 2 *horas* antes en la solución 2 con respecto a la solución 1. Esto último ejemplifica que el problema estudiado no es del todo evidente y que tanto el objetivo a minimizar como el algoritmo de resolución requieren una investigación de las posibilidades de resolución actuales.

El ejemplo anterior da cuenta de que este problema debe ser abordado con la integración tanto de la producción como de la distribución, pues la producción no puede ser iniciada sin conocer la ruta que el enfermero realizará y, a su vez, la ruta debe considerar las horas de producción para poder trazarse. Además, dada la integración de ambos problemas, a la hora de escalar el problema para poder resolverlo para un caso de la vida real, este se hace enormemente complejo por lo cual muchos de los métodos de resolución conocidos actualmente no resultan eficientes.

De esta forma, por un lado se hace necesaria una herramienta que ayude a la toma de decisiones del proceso de quimioterapias en HAD. Lo anterior implica la utilización de conocimientos del área de la ingeniería y ciencias para poder entregar una solución que mejore el proceso estudiado de forma eficiente. Por otro lado y tal como se observó anteriormente, el problema matemático y computacional a resolver es extremadamente complejo y las soluciones óptimas, utilizando distintos métodos de resolución, no son para nada evidentes cuando se habla de un POI, lo cual determina la actividad de investigación durante el trabajo a realizar.

1.3. Objetivo General

Dado el contexto presentado en las secciones precedentes, se busca minimizar el tiempo de realización del proceso de quimioterapias a domicilio. Para lograr esto, se busca entregar una función objetivo que integre la minimización de los tiempos de producción en conjunto con los tiempos de trazado de rutas. La magnitud para la cual se desea resolver este problema integrado es del orden de 100 pacientes con sus respectivos 100 medicamentos que deben ser atendidos dentro de un horizonte de tiempo de 1 día de jornada laboral.

En vista de la complejidad del problema, existen metodologías heurísticas para la resolución de problemas de optimización. Dentro de las más utilizadas se encuentran los métodos de generación de columnas, los cuales permiten descomponer el problema de forma dinámica con el objetivo de tomar en cuenta solo un subconjunto de las variables del problema maestro conservando la calidad de la solución.

Considerando lo anterior, el objetivo general del trabajo de título corresponde a “Generar un algoritmo basado en el paradigma de generación de columnas para resolver un POI que permita minimizar los tiempos de ocupación tanto de los químicos farmacéuticos como de los enfermeros”.

1.4. Objetivos Específicos

De forma más específica, dentro del área de investigación operativa, es necesario caracterizar matemáticamente el problema para luego implementar un programa computacional que sea capaz de resolver dicho el modelo matemático propuesto. En esta línea, se pueden listar 4 objetivos específicos que encuadran el desarrollo general de esta investigación.

- Definir la naturaleza de cada uno de los problemas tratados. En un principio para el problema de distribución correspondiente a un problema de trazado de rutas y en segundo lugar el problema de producción de los medicamentos.
- Integrar matemáticamente los problemas de producción y distribución para el problema de quimioterapias en HAD.
- Entregar una descomposición adecuada que permita resolver el problema maestro correspondiente a un POI con una cantidad exponencial de variables.
- Evaluar si las columnas del algoritmo de generación de columnas deben representar solo un sub-problema de alguno de los dos problemas o si es necesario tomar en cuenta una parte de cada uno de los problemas estudiados.

Capítulo 2

Revisión bibliográfica del problema operacional de producción y distribución

Dentro de las áreas de la ingeniería industrial, el presente tema de memoria está enmarcado en el área de investigación de operaciones. El estado del arte actual permite resolver los problemas de producción y los problemas de distribución separadamente de forma eficiente. Sin embargo, cuando se integran ambos problemas los modelos matemáticos son extremadamente complejos, y no pueden ser resueltos por algoritmos de resolución comerciales de forma eficiente. Para facilitar la lectura, una descripción de las siglas utilizadas a lo largo de este trabajo, así como un glosario se pueden encontrar al final de este informe.

2.1. Trazado de Rutas de Vehículos

Sentando las bases del problema a resolver, el problema de trazado de rutas de vehículos o VRP, por sus siglas en inglés, es un problema *NP-Hard* [20], lo que quiere decir que algoritmos exactos son eficientes solamente para problemas a muy baja escala. Metodologías heurísticas y/o meta-heurísticas son a menudo recomendadas para aplicaciones más prácticas, pues los problemas de la vida real son en general de mayor escala.

En lo que respecta al estado del arte del VRP, si bien el problema tiene distintas variantes, no existen muchos problemas integrados al problema de producción con productos que poseen una vida útil acotada. Considerando el criterio de clasificación para los problemas de trazado de rutas descrito en un artículo hecho por investigadores de la Universidad Católica de Lovaina [9], se puede tener un primer acercamiento de los modelos matemáticos necesarios para resolver el problema tratado. Así, una primera clasificación del problema estudiado se presenta a continuación:

- **Nodos:** Al conocer la cantidad y direcciones de los pacientes, los nodos se consideran como determinísticos, es decir, conocemos la cantidad de nodos de forma previa.
- **Origen:** Como todos los enfermeros deben partir desde el hospital, entonces se habla de *single origin*, pues solo posee un único nodo origen que representa al hospital.

- *Load Split*: Esto corresponde a que los enfermeros tienen la posibilidad de volver al nodo origen para recuperar medicamentos y continuar la ruta.
- Ventana de Tiempo: Más comúnmente llamada *Time Windows* por su nombre en inglés. Se considera que el problema presentado tiene ventanas de tiempo correspondientes a la realización de los tratamientos.
- Horizonte de Tiempo: Se espera realizar una planificación de las quimioterapias de forma diaria, por lo que se habla de *single period*, pues la resolución del problema toma en cuenta la programación de un solo día de tratamientos a domicilio.

Dadas estas informaciones, es posible tener un primer planteamiento del problema abordado con la variante de múltiples viajes del problema de trazado de rutas de vehículos con ventanas de tiempo (MTVRPTW por sus siglas en inglés). La característica de múltiples viajes o *multiple trips*, corresponde al hecho de que el *Load Split* está permitido. Una solución basada en un algoritmo de *Branch-and-Price* (que utiliza un paradigma de generación de columnas) se presenta en el trabajo de F. Hernandez [16], sin embargo, las restricciones del tiempo de estabilidad de los medicamentos no son tomadas en cuenta por esta solución.

2.2. Problema Operacional Integrado

Tal como se explica en la sección 1.2, el problema abordado debe ser tratado como un POI. En la literatura existen escasas fuentes que se refieran a la resolución de este problema y la mayoría de los modelos matemáticos postulados son extremadamente complejos, pudiendo resolverse tan solo a muy baja escala. En las últimas 2 décadas, en búsqueda de reducir costos, algunos problemas integrados han aparecido en la literatura, como es caso por ejemplo del problema de producción y distribución [23] que combina el problema de *lot sizing* con el de VRP. Si bien, el problema anterior no se relaciona con el de quimioterapias en HAD, pues considera la gestión de inventario y de producción por lotes, da cuenta de la complejidad de las soluciones que se desean plantear.

Un artículo que se acerca en mayor medida al problema estudiado corresponde a la descomposición para la resolución de un POI entregado por Y. Kergosien [18]. Dicho problema también se encuentra dentro del contexto de la realización de quimioterapias y entrega una formulación integrada del problema de producción en conjunto con el de VRP. La principal diferencia con este modelo es que tan solo considera un único vehículo para la realización de los viajes, por lo que es necesario extenderlo para que pueda ser aplicado al problema estudiado. Una segunda diferencia es el tipo de *Time Window*, pues en el artículo citado anteriormente, tan solo se toma en cuenta la entrega de los productos y no así la realización del tratamiento por el repartidor (para el caso estudiado correspondería al enfermero).

2.3. Metodología

La creación de modelos matemáticos resueltos a través de la programación lineal representa la principal herramienta para abordar un problema de optimización. Si bien, una formulación de programación lineal entera mixta es necesaria para el mejor entendimiento del problema, para efectos prácticos no es una condición suficiente para poder resolver el POI. Por esta razón, es necesario entregar una formulación extendida con el fin de generar una descomposición adecuada del problema y entregar una solución de forma eficiente. Dicha formulación extendida [22] puede considerar tanto la introducción de nuevas variables como la relajación o introducción de ciertas restricciones que ayuden a estructurar de mejor manera el problema abordado.

Como se mencionó en la sección 1.2, se busca entregar un algoritmo basado en el paradigma de generación de columnas, el cual se puede ver explicado en detalle en el trabajo de J. Desrosiers [12]. El principio que guía a este tipo de metodologías es que cada variable del problema está asociada a una columna dentro de la matriz de restricciones. Esta noción de columna es evidente para ciertos problemas clásicos como el de distribución, en el cual la mayoría de las veces una columna representa la ruta que debe realizar el conductor. No obstante, para la resolución de un POI, la noción de columna no es para nada evidente ya que la información contenida en una ruta no considera los tiempos de producción que minimizan el tiempo de ocupación de los farmacéuticos. De forma recíproca, la información entregada por la producción no necesariamente permite encontrar las rutas que minimizan el tiempo de trayecto cubriendo todos los tratamientos. Finalmente, de forma conjunta, la cantidad de rutas permitidas no es necesariamente igual a la cantidad de farmacéuticos disponibles, por lo que no se puede optar por una integración en cadena entre la producción y la distribución.

Los modelos matemáticos y algoritmos descritos en el presente trabajo son implementados en el lenguaje de programación Java para la obtención de resultados empíricos que evidencien el desempeño de la solución propuesta. Dicho lenguaje es escogido por razones del balance que ofrece entre velocidad de cálculo y facilidad de la implementación en sí. En términos de velocidad de cálculo, siendo un lenguaje compilado, Java resulta ser más eficaz frente a otros lenguajes interpretados como Python. A pesar de que Java es un lenguaje de mayor nivel que C y C++, estos últimos presentan una mayor dificultad en la implementación del código, lo cual retrasaría la realización de pruebas de los algoritmos elaborados a lo largo del presente trabajo y su posterior ajuste.

Para la resolución de modelos de programación lineal propuestos a lo largo de la realización de este trabajo, el *solver* comercial CPLEX es utilizado. Dicho software ha sido durante años líder en el mercado de las herramientas de optimización, por lo que es una opción confiable para estos efectos además de ofrecer una interfaz de programación para el lenguaje Java. Otros *solver* orientados a la resolución de problemas lineales pueden ser utilizados para resolver los modelos propuestos, sin embargo la licencia de CPLEX es entregada de forma gratuita por las instituciones que apoyan la realización de la presente memoria y la evaluación del software utilizado para resolver el problema no entra dentro de los objetivos definidos en el capítulo 1. Por lo tanto, el software escogido corresponde a la mejor opción en términos financieros y de eficacia.

2.4. Alcances

La presente investigación necesita de un estudio acabado de los problemas de trazado de rutas así como de los problemas integrados de producción-distribución. El alcance de este estudio considera la formulación de un problema integrado de producción-distribución, sin embargo la variante de *load split* no es tomada en cuenta dada la complejidad que agrega a la resolución del problema, por lo cual el problema de distribución sólo se trata un VRPTW en vez del MTVRPTW mencionado en la sección 2.1.

La formulación derivada de la primera etapa de investigación, correspondiente a la resolución del problema de trazado de rutas, es reformulada con el objetivo de desarrollar una formulación extendida del problema operacional integrado. Para esta última formulación propuesta, se estudia una descomposición adecuada del problema, que permita desarrollar un algoritmo que entregue una solución al problema de las quimioterapias en HAD.

Capítulo 3

Modelo del problema operacional integrado

En el presente capítulo se entrega un modelo matemático para la resolución del POI. De aquí en adelante, para denotar la cantidad de enfermeros necesarios para la realización de quimioterapias se hablará de **vehículos**, de la misma forma, para hacer mención a la cantidad de químicos farmacéuticos necesarios para la producción de los medicamentos, se hablará de **operadores**.

En un principio se define el problema de distribución mediante un modelo de *set-covering*. Para entender la aproximación utilizada para solucionar este problema de forma numérica, se entrega una explicación del problema del camino unitario más corto con restricciones de recursos (ESPPRC por sus siglas en inglés). Una correcta comprensión de este problema es necesario para la explicación del modelo de *set-covering* que resuelve el problema de trazado de rutas de vehículos con ventanas de tiempo (VRPTW). Dado el problema de distribución, se describe el problema de producción de medicamentos y su relación con el problema de distribución, para luego entregar una propuesta del modelo integrado de producción y distribución como extensión del primer modelo de distribución descrito.

3.1. Descripción del problema de distribución

Tal como se menciona en la sección 2.4, el problema de distribución es modelado como un VRPTW como se describe a continuación. Dados un origen v_0 , un conjunto de pacientes $V = \{v_1, \dots, v_n\}$ y una flota de vehículos U , el objetivo del problema es encontrar el conjunto de rutas de duración total mínima empezando desde el origen y terminando en el mismo lugar, tal que cada uno de los pacientes es visitado exactamente una vez por cualquiera de los vehículos. Cada visita debe respetar las disponibilidades del paciente, el vehículo puede esperar al paciente si llega de forma temprana, sin embargo, las llegadas tardías no están permitidas. Una descripción formal de este problema puede encontrarse en diferentes artículos académicos, para mayor información, el lector es referido al trabajo de Kallehauge, B. [17].

3.1.1. Branch-and-Price para la resolución del VRPTW

Un método de *Branch-and-Price* (BnP) es un híbrido entre un método clásico de *Branch-and-Bound* (BnB) y uno de generación de columnas. En efecto, el método de BnP explora un árbol de candidatos a solución numerados, tal como se hace en el método de BnB, este último intenta buscar mejores soluciones a medida que recorre el árbol del conjunto de soluciones a través de la relajación lineal de las restricciones del modelo. Dado que diversos conjuntos de soluciones o “ramas” exploradas por el algoritmo no son parte de la solución óptima, un método de generación de columnas es utilizado por el método de BnP con el fin de generar sólo las soluciones que podrían ayudar a la obtención de una mejor solución en cada ramificación del árbol de soluciones.

Con respecto al método de generación de columnas mencionado en el párrafo anterior, este trabaja de la siguiente manera dentro del algoritmo de BnP. Primero, se realiza una reformulación del problema original, dicha formulación se denomina Problema Maestro (ó *Master Problem* (MP) en inglés), para el presente trabajo el *MP* corresponde al modelo de *set-covering*, en el cual cada columna corresponde a una ruta realizada por un vehículo. Para cada iteración, en lugar de resolver el problema entero, una relajación lineal es resuelta para obtener una cota inferior de la solución óptima. En un principio, un conjunto de columnas es entregado al *MP* y, a partir de la solución obtenida de la relajación lineal, los valores duales son introducidas a un sub-problema llamado el problema de *pricing*, el cual en este caso corresponde a un ESPPRC. El sub-problema es resuelto y entrega una columna correspondiente a una ruta de un vehículo, así, la ruta es introducida en el conjunto de rutas factibles y se continúa a la iteración siguiente resolviendo nuevamente la relajación lineal del *MP*.

El algoritmo de generación de columnas se detiene cuando no encuentra ninguna columna que mejore la solución del problema maestro, esto quiere decir que el sub-problema entregue un costo reducido no negativo. Una vez que esto ocurre, el *MP* entero es resuelto entregando una cota superior de la solución óptima del problema original.

Una explicación más extensiva de las razones por las cuales un enfoque de BnP es usado para resolver este problema puede ser encontrada en el tutorial académico de D. Feillet [14]. Para el propósito de esta memoria, sólo la descomposición del problema maestro y el sub-problema tratado por el algoritmo de generación de columnas son presentados.

Problema Maestro (*MP*)

El modelo de *set-covering* se describe a continuación. Sea Ω el conjunto de rutas factibles, es decir, el conjunto de caminos que empiezan y terminan en el origen, satisfaciendo las restricciones de capacidad y ventanas de tiempo, visitando a lo más una vez cada paciente. Sea c_k el costo de la ruta $r_k \in \Omega$. Sea $a_{ik} = 1$ si la ruta $r_k \in \Omega$ visita al paciente $v_i \in V$ y 0 si no. Se definen las siguientes variables de decisión:

$$x_k = \begin{cases} 1 & \text{si la ruta } r_k \text{ es seleccionada} \\ 0 & \text{si no.} \end{cases}, \quad (\forall r_k \in \Omega)$$

El VRPTW se describe a través del siguiente modelo:

$$\text{minimizar } \sum_{r_k \in \Omega} c_k x_k \quad (3.1)$$

sujeto a

$$\sum_{r_k \in \Omega} a_{ik} x_k \geq 1 \quad (\forall v_i \in V), \quad (3.2)$$

$$\sum_{r_k \in \Omega} x_k \leq U, \quad (3.3)$$

$$x_k \in \{0, 1\} \quad (\forall r_k \in \Omega). \quad (3.4)$$

No es difícil observar que el conjunto de rutas factibles Ω es extremadamente grande, por lo cual resulta muy difícil de ser resuelto por un *solver* comercial y de generar de forma eficiente las rutas, lo cual tomaría un tiempo considerable en la resolución, incluso luego de haber generado la totalidad de las rutas.

Es importante observar que la restricción (3.2) requiere que todos los pacientes sean visitados, no obstante no asegura que sólo sean visitados una vez. Los supuestos que validan esta relajación son, primero que todo, que las distancias entre los nodos satisfacen la desigualdad triangular, esto es, $\forall(i, j, k) v_i, v_j, v_k \in V, c_{ij} \leq c_{ik} + c_{kj}$, lo cual implica que al suprimir un nodo de una ruta factible el costo sólo puede ser igual o menor que el costo anterior. El segundo supuesto es que, dado el sub-problema ESPPRC y dada la desigualdad triangular anterior, la restricción (3.2) permite encontrar soluciones factibles más rápidamente pero tendiendo siempre hacia la unicidad de las visitas puesto que las rutas entregadas por el problema de *pricing* siempre tendrán visitas unitarias, facilitando la reducción de la cantidad de nodos visitados.

Dual del *MP*

Para la obtención de nuevas rutas que ayuden a mejorar la solución obtenida en cada iteración, es necesario observar el problema dual del modelo de *set-covering* descrito anteriormente en esta sección. De esta forma, se definen las siguientes variables duales del *MP*:

- Sean α_i ($\forall v_i \in V$) las variables duales asociadas las restricciones (3.2).
- Sea g la variable dual asociada a la restricción (3.3).

El dual es descrito a continuación,

$$\text{maximizar } \sum_{v_i \in V} \alpha_i + U \times g \quad (3.5)$$

sujeto a

$$\sum_{v_i \in V} a_{ik} \alpha_i + g \leq c_k \quad (\forall r_k \in \Omega), \quad (3.6)$$

$$\alpha_i \geq 0 \quad (\forall v_i \in V), \quad g \leq 0. \quad (3.7)$$

A partir de las restricciones (3.6) es posible definir el costo reducido de las rutas que serán introducidas al conjunto Ω durante la resolución del *MP* relajado. La expresión del costo reducido \hat{c}_k para una ruta en particular r_k corresponde a la siguiente,

$$\hat{c}_k = c_k - g - \sum_{v_i \in V} a_{ik} \alpha_i. \quad (3.8)$$

Problema de *pricing*

Para generar las rutas factibles del modelo de *set-covering*, *i.e.* columnas, un ESPPRC es resuelto a partir de los valores duales entregados por la relajación lineal del *MP*. El problema de *pricing* se describe a continuación. Sea $G = (V, E)$ el grafo dirigido de la instancia a resolver, donde $V = \{v_1, \dots, v_n\} \cup \{v_0, v_t\}$ incluyendo un nodo v_0 como el origen y un nodo v_t como el nodo de destino, el cual, para este problema, corresponde a un duplicado del nodo origen v_0 , y E el conjunto de arcos. Para cada arco $(v_i, v_j) \in E$ un costo c_{ij} y una distancia $dist_{ij} \geq 0$ están asociados. Hacer una diferencia entre el costo y la distancia entre los nodos es necesaria para la posterior introducción de los valores duales y el cálculo del costo reducido de la ruta.

Se consideran dos recursos: el tiempo y la demanda. En conjunto con cada nodo $v_i \in V$ están asociados dos valores no negativos: a_i y b_i , tales que el consumo del recurso de tiempo a través del camino que va desde v_0 a v_i es restringido para que pertenezca al intervalo $[a_i, b_i]$. Si el consumo del recurso del tiempo es menor a a_i cuando el camino llega a v_i , este es puesto en a_i . Un tiempo de servicio $serv_i$ requerido por el nodo v_i también es considerado en el cálculo del consumo del recurso de tiempo. Finalmente, a cada nodo v_i también se asigna un consumo de demanda d_i . Las restricciones de capacidad son representadas definiendo intervalos $[0, Q]$ en los nodos, donde Q es el límite de capacidad.

El objetivo del problema de *pricing* es minimizar el costo reducido definido en la ecuación 3.8. El término c_k corresponde al costo total de la ruta r_k , el cual es igual a la suma de los costos de los arcos utilizados por la ruta. A esto se le deben restar los valores duales entregados por la última resolución del *MP* relajado, tal como se muestra en la ecuación mencionada. Dado que el valor de g es simplemente una constante para el sub-problema, entonces se tiene que $\min\{\hat{c}_k\} + g = \min\{\hat{c}_k + g\}$. En otras palabras minimizar \hat{c}_k corresponde a minimizar $\hat{c}_k + g$, por lo cual se puede optar por minimizar esta última expresión en la que se anula el valor de g . Habiendo definido el problema de *pricing*, una descripción formal del modelo se presenta a continuación.

Sean las siguientes variables de decisión,

$$z_{ij} = \begin{cases} 1 & \text{si el arco } (v_i, v_j) \text{ es utilizado} \\ 0 & \text{si no} \end{cases}, \quad (\forall (v_i, v_j) \in E),$$

$$s_i = \text{tiempo en el que el nodo } v_i \text{ es visitado}, \quad (\forall v_i \in V).$$

Formalmente, el modelo de programación lineal del ESPPRC descrito anteriormente es representado de la siguiente forma:

$$\text{minimizar } \sum_{(v_i, v_j) \in E} (c_{ij} - \alpha_j) z_{ij} \quad (3.9)$$

sujeto a

$$\sum_{v_j \in V} z_{ij} - \sum_{v_j \in V} z_{ji} = \begin{cases} 1 & \text{si } i = 0 \\ -1 & \text{si } i = t \\ 0 & \text{si no} \end{cases} \quad (\forall v_i \in V), \quad (3.10)$$

$$\sum_{v_j \in V} z_{pj} = 1, \quad (3.11)$$

$$\sum_{v_i \in V} z_{it} = 1, \quad (3.12)$$

$$s_i + serv_i + dist_{ij} - s_j + M z_{ij} \leq M \quad (\forall v_i \in V \setminus v_0), \quad (3.13)$$

$$\sum_{(v_i, v_j) \in E} d_i z_{ij} \leq Q, \quad (3.14)$$

$$z_{ij} \in \{0, 1\} \quad (\forall v_i \in V, \forall v_j \in V), \quad s_i \in [a_i, b_i] \quad (\forall v_i \in V). \quad (3.15)$$

donde M es un número grande que puede ser definido como el total del horizonte de tiempo del problema, es decir, si el horizonte de tiempo es 1 día, M se define igual a dicha cantidad.

Las restricciones (3.10) a (3.12) aseguran la naturaleza unitaria de las visitas a los nodos empezando desde el nodo origen v_0 y terminando en el nodo de destino v_t . Las restricciones (3.13) y (3.14) verifican las restricciones de utilización de los recursos de tiempo y demanda respectivamente.

3.2. Descripción del problema de producción

La preparación de los medicamentos necesarios para la quimioterapia es modelada como un problema de programación de n **tareas**, correspondientes a los n pacientes que deben

ser atendidos, cada uno, con un medicamento específico. Una **tarea** corresponde al proceso de fabricación de un medicamento en particular. Dichas tareas deben ser realizadas por una cantidad W de operadores uniformes que trabajan de forma paralela, quienes corresponden a los químicos farmacéuticos. Para simplificar el problema, se considera que todos los operadores trabajan a una misma tasa constante de producción. Cada una de las n tareas es asignada a un operador en particular en un tiempo de producción determinado en el cual el operador debe iniciar dicha tarea. Esto último debe considerar las restricciones de programación a las que se deben acotar los tiempos de producción de cada uno de los medicamentos.

Formalmente, considerando un conjunto de n tareas J , cada tarea $j_i \in J$, con $i \in \{1, \dots, n\}$, está caracterizada por el tiempo de estabilización del producto y el tiempo de producción. Sin embargo, como consecuencia de las ventanas de tiempo durante las cuales se debe tratar al paciente, una fecha de lanzamiento y una fecha de entrega son introducidas como se describe a continuación. Considerando $dist_{k,i}$ como el tiempo necesario para llegar desde el nodo $v_k \in V$ hasta el nodo $v_i \in V$, para cada tarea $j_i \in J$ se tienen las siguientes características:

- Sea st_i el **tiempo de estabilización** del producto como resultado de la realización de la tarea j_i ,
- Sea pt_i el **tiempo de procesamiento** de la tarea j_i ,
- Se define $rt_i = \max(0, a_i - dist_{0,i})$ la **fecha de lanzamiento** de la tarea j_i ,
- Se define $ft_i = b_i - dist_{0,i}$ la **fecha de entrega** de la tarea j_i .

Es necesario notar que si $ft_i < pt_i$ entonces el problema no tiene solución, ya que existe un paciente para el cual no es posible producir el medicamento antes del tiempo final de su ventana de tiempo.

Cada operador procesa sin interrupción sólo una tarea j_i a la vez, y cada tarea, una vez asignada, sólo puede ser procesada por un operador. Para simplificar el modelo, se considera que los W operadores trabajan a una misma tasa y que el tiempo de procesamiento es equivalente a la duración p_i de la tarea j_i . También se toma por supuesto que no existe tiempo de preparación entre la realización de dos tareas distintas consecutivas programadas para un mismo operador y que al final de la agrupación de las tareas en lotes no existe tiempo de espera entre esta acción y el inicio de la ruta. Estas simplificaciones son discutidas más adelante en el presente informe.

El problema de producción consiste en programar cada una de las n tareas entre los W operadores respetando los tiempos de entrega y de inicio de producción además de las restricciones de procesamiento de los operadores. El objetivo es encontrar la asignación de tareas que minimiza la duración del programa de tareas asignado para el conjunto de operadores. La duración de un programa de tareas, asignado a un operador en particular, corresponde al tiempo transcurrido entre el inicio de la primera tarea programada hasta el fin de la última tarea programada.

3.2.1. Modelo de programación lineal

A continuación, se describe el modelo de programación lineal entera mixta correspondiente al problema de producción descrito en esta sección. Se consideran, además de las tareas que deben ser programadas, dos tareas ficticias con parámetros nulos, fecha de lanzamiento igual a 0 y fecha de entrega infinita, que asimilen respectivamente a una tarea “inicial” j_0 y una tarea “final” j_t , con lo que se redefine el conjunto J incluyendo ambas tareas.

Sean las siguientes variables de decisión,

$$\pi_{ik}^l = \begin{cases} 1 & \text{si la tarea } j_i \text{ es asignada justo antes de la tarea } j_k \text{ al operador } l \\ 0 & \text{si no} \end{cases},$$

$$(\forall (j_i, j_k) \in J^2, \quad \forall l \in \{1, \dots, W\})$$

$$w_i^l = \text{tiempo en el que la tarea } j_i \text{ es asignada al operador } l \\ (\forall j_i \in J, \quad \forall l \in \{1, \dots, W\}).$$

$$\psi_l = \text{duración del programa de tareas del operador } l \\ (\forall l \in \{1, \dots, W\}).$$

El objetivo del problema es minimizar la suma de las duraciones de los programas de los operadores.

$$\text{minimizar } \sum_{l=1}^W \psi_l \tag{3.16}$$

El problema está sujeto a las siguientes restricciones. Primero que todo, cada una de las tareas debe estar asignada a algún operador.

$$\sum_{l=1}^W \sum_{j_k \in J} \pi_{ki}^l = 1, \quad (\forall j_i \in J \setminus \{j_0, j_t\}). \tag{3.17}$$

Luego, sólo una tarea puede ser la tarea inicial. De la misma forma, sólo una tarea puede ser la tarea final asignada a un operador.

$$\sum_{j_i \in J} \pi_{0i}^l = 1, \quad (\forall l \in \{1, \dots, W\}). \tag{3.18}$$

$$\sum_{j_i \in J} \pi_{it}^l = 1, \quad (\forall l \in \{1, \dots, W\}). \tag{3.19}$$

Dado que las tareas no pueden ser procesadas simultáneamente, se tienen las siguientes restricciones de precedencia.

$$\sum_{j_k \in J} \pi_{ik}^l - \sum_{j_k \in J} \pi_{ki}^l = \begin{cases} 1 & \text{si } i = 0 \\ -1 & \text{si } i = t \\ 0 & \text{si no} \end{cases} \quad (\forall j_i \in J, \forall l \in \{1, \dots, W\}). \quad (3.20)$$

A continuación, se tienen las restricciones de no interrupción durante la ejecución de las tareas.

$$w_i^l + pt_i - w_k^l + M\pi_{ik}^l \leq M \quad (\forall (j_i, j_k) \in J^2, \forall l \in \{1, \dots, W\}). \quad (3.21)$$

La duración de los programas asignados a cada operador corresponde al tiempo transcurrido entre el inicio de la primera tarea programada hasta el final de la última tarea programada. Considerando que las tareas ficticias j_0 y j_t tienen tiempos de procesamiento pt_0 y pt_t iguales a 0, dado que las restricciones (3.21) también son válidas para ambas tareas ficticias, se definen las siguientes restricciones sobre la duración de los programas.

$$\psi_l \geq w_t^l - w_0^l \quad (\forall l \in \{1, \dots, W\}). \quad (3.22)$$

Finalmente, se define la naturaleza de las variables de decisión de la siguiente forma.

$$\begin{aligned} \pi_{ik}^l &\in \{0, 1\} & (\forall (j_i, j_k) \in J^2, \forall l \in \{1, \dots, W\}) \\ w_i^l &\in [rt_i, ft_i] & (\forall j_i \in J, \forall l \in \{1, \dots, W\}) \\ \psi_l &\geq 0 & (\forall l \in \{1, \dots, W\}). \end{aligned} \quad (3.23)$$

Lo interesante de plantear este modelo es que su formulación se asemeja bastante a la formulación de un VRPTW. De esta forma, se puede definir el concepto de **programa factible** como la sucesión de tareas asignadas a un operador en particular respetando las ventanas de tiempo de cada una de las tareas. A partir de la definición anterior, se puede modelar el problema con el objetivo de encontrar el conjunto de programas de mínima duración con inicio en la tarea “inicial” y término en la tarea “final” que cubran la realización de todas las tareas.

3.3. Modelo integrado

Basado en el modelo de *set-covering* descrito en la sección 3.1.1. Para el problema de distribución, a cada ruta generada $r_k \in \Omega$ se le asocia un costo correspondiente a la duración necesaria para la realización de dicha ruta. Sean las siguientes variables de decisión.

$$x_k = \begin{cases} 1 & \text{si la ruta } r_k \text{ es elegida} \\ 0 & \text{si no.} \end{cases}, \quad (\forall r_k \in \Omega)$$

Además, se define el parámetro u_{ik} ($\forall v_i \in V, \forall r_k \in \Omega$) que toma el valor del tiempo en el que la ruta $r_k \in \Omega$ visita al nodo $v_i \in V$. El parámetro u_{ik} toma el valor de 0 si el nodo v_i no es visitado por la ruta r_k . De la misma forma, se define el parámetro ω_k ($\forall r_k \in \Omega$) correspondiente al tiempo en que la ruta $r_k \in \Omega$ inicia su recorrido. Finalmente, de recuerda al lector que el parámetro a_{ik} ($\forall v_i \in V, \forall r_k \in \Omega$) toma el valor de 1 si la ruta $r_k \in \Omega$ visita al paciente $v_i \in V$ y 0 si no.

Un modelo de *set-covering* es también propuesto para resolver el problema de distribución. Esto dado las semejanzas con el problema de trazado de rutas descritas en la sección 3.2.1. Para este caso, se define Ψ como el conjunto de programas factibles que cumplan con las restricciones del problema de producción descritas en la sección precedente. Así, definiendo $p_l \in \Psi$ un programa factible, se definen las siguientes variables de decisión.

$$y_l = \begin{cases} 1 & \text{si el programa } p_l \text{ es seleccionado} \\ 0 & \text{si no.} \end{cases}, \quad (\forall p_l \in \Psi)$$

Asociados a cada programa factible se tienen los siguientes parámetros asociados. En primer lugar, b_{il} toma el valor de 1 si el programa $p_l \in \Psi$ tiene asignado la tarea $j_i \in J$ y 0 si no. En segundo lugar, w_{il} corresponde al tiempo en el que la tarea j_i comienza dentro del programa p_l , este parámetro toma el valor de 0 en caso de que la tarea no esté incluida en el programa.

Para simplificar la notación y evitar confusiones entre los conjuntos de pacientes V y de tareas J , dado el número n de pacientes, se define el siguiente conjunto de índices $I = \{1, \dots, n\}$, tal que para todo $i \in I$ se tiene asociado un paciente $v_i \in V$ y una tarea $j_i \in J$ correspondiente al medicamento necesario para v_i . Considerando las restricciones de los tiempos de estabilización y las restricciones de precedencia, se propone el siguiente modelo.

$$\text{minimizar } \sum_{r_k \in \Omega} c_k x_k + \sum_{p_l \in \Psi} \psi_l y_l \quad (3.24)$$

sujeto a

$$\sum_{r_k \in \Omega} a_{ik} x_k = 1 \quad (\forall i \in I), \quad (3.25)$$

$$\sum_{r_k \in \Omega} x_k \leq U, \quad (3.26)$$

$$\sum_{p_l \in \Psi} b_{il} y_l = 1 \quad (\forall i \in I), \quad (3.27)$$

$$\sum_{p_l \in \Psi} y_l \leq W, \quad (3.28)$$

$$\sum_{r_k \in \Omega} u_{ik} x_k \leq \sum_{p_l \in \Psi} (w_{il} + st_i) b_{il} y_l \quad (\forall i \in I), \quad (3.29)$$

$$\sum_{r_k \in \Omega} a_{ik} \omega_k x_k \geq \sum_{p_l \in \Psi} w_{il} y_l \quad (\forall i \in I), \quad (3.30)$$

$$x_k \in \{0, 1\} \quad (\forall r_k \in \Omega), \quad y_l \in \{0, 1\} \quad (\forall p_l \in \Psi). \quad (3.31)$$

La función objetivo (3.24) considera la minimización de la duración del conjunto de rutas en la primera suma de izquierda a derecha. En el término de la derecha se considera la minimización de la duración del conjunto de programas. Entre ambas expresiones se busca minimizar los tiempos totales de realización de las quimioterapias a domicilio.

Las restricciones (3.25) requieren que cada paciente sea visitado exactamente una vez. En lo que respecta a la restricción (3.26), esta indica la cantidad máxima de vehículos permitidos. Las restricciones (3.27) y (3.28) requieren que todas las tareas para cada uno de los productos sean asignadas exactamente una vez a un operador y que la cantidad de programas no sobrepase la cantidad de operadores disponibles. Las restricciones (3.29) requieren que el producto sea distribuido durante su tiempo de estabilización y finalmente, las restricciones (3.30) requieren que el producto esté listo antes del inicio de la ruta que distribuye dicho producto.

Nota. La igualdad en las restricciones (3.25) y (3.27) es necesaria para que las restricciones (3.29) y (3.30) sean válidas. Si se dejara la relajación de mayor o igual, entonces las restricciones mencionadas no asegurarían que el producto que respeta los tiempos de estabilidad sea producido antes del inicio de alguna de las rutas que distribuye dicho producto. De forma recíproca, un producto que haya sido terminado antes del inicio de la ruta que lo distribuye podría no ser el producto que respeta los tiempos de estabilidad para esa ruta en particular. De esta forma, de ahora en adelante, en vez de hablar de *set-covering*, se hablará de un modelo de *set-partitioning*.

3.3.1. Dual del modelo integrado

El problema dual del modelo integrado propuesto en esta sección tiene las siguientes variables duales,

- α_i ($\forall i \in I$) correspondientes a las restricciones (3.25),
- g correspondiente a la restricción (3.26),
- β_i ($\forall i \in I$) correspondientes a las restricciones (3.27),
- h correspondiente a la restricción (3.28),
- δ_i ($\forall i \in I$) correspondientes a las restricciones (3.29),
- ε_i ($\forall i \in I$) correspondientes a las restricciones (3.30).

Considerando esta notación, la formulación del problema dual es la siguiente.

$$\text{maximizar } \sum_{v_i \in V} \alpha_i + U \times g + \sum_{i \in I} \beta_i + W \times h \quad (3.32)$$

sujeto a

$$\sum_{v_i \in V} a_{ik} \alpha_i + g + \sum_{v_i \in V} u_{ik} \delta_i + \sum_{v_i \in V} a_{ik} \omega_k \varepsilon_i \leq c_k \quad (\forall r_k \in \Omega), \quad (3.33)$$

$$\sum_{v_i \in V} b_{il} \beta_i + h - \sum_{v_i \in V} (w_{il} + st_i) b_{il} \delta_i - \sum_{v_i \in V} w_{il} \varepsilon_i \leq \psi_l \quad (\forall p_l \in \Psi), \quad (3.34)$$

$$\alpha_i \in \mathbb{R}, \beta_i \in \mathbb{R}, \delta_i \leq 0, \varepsilon_i \geq 0 \quad (\forall i \in I), g \leq 0, h \leq 0. \quad (3.35)$$

Para efectos de la resolución a través de un método de generación de columnas, los siguientes costos reducidos son definidos.

$$\hat{c}_k = c_k - g - \sum_{v_i \in V} a_{ik} \alpha_i - \sum_{v_i \in V} u_{ik} \delta_i - \sum_{v_i \in V} a_{ik} \omega_k \varepsilon_i \quad (3.36)$$

$$\hat{\psi}_l = \psi_l - h - \sum_{v_i \in V} b_{il} \beta_i + \sum_{v_i \in V} (w_{il} + st_i) b_{il} \delta_i + \sum_{v_i \in V} w_{il} \varepsilon_i \quad (3.37)$$

Dado que para el problema dual se tienen dos tipos de restricción diferentes, un tipo de restricción correspondiente a las rutas de distribución y otro tipo de restricción correspondiente a los programas de producción, se tienen entonces dos sub-problemas distintos a resolver. Los costos reducidos definidos anteriormente, corresponden a las funciones objetivo de cada uno de los sub-problemas de trazado de rutas y de asignación de tareas respectivamente, que deben ser minimizadas respetando las ventanas de tiempo y las condiciones de precedencia para el caso de los programas de tareas. Lo anterior significa el principal cambio a tomar en cuenta con respecto a la resolución del algoritmo de *pricing* descrito en la sección 3.1.1.

Para denotar el origen de los valores duales tratados por los problemas de *pricing*, en lo que sigue, se denominan **duales de cobertura** a los valores α_i y β_i , asimismo, para los valores δ_i se habla de **duales de estabilidad** y para los valores ε_i de **duales de precedencia**, esto para todo $i \in I$.

Capítulo 4

Un método de generación de columnas para resolver el POI

Dado que dos problemas diferentes han sido tratadas hasta ahora, el modelo es presentado en dos secciones principales de la forma en que se han tratado a lo largo de la realización del presente trabajo. La primera sección está dedicada a la resolución del problema de distribución usando un enfoque de generación de columnas. Esta sección está dividida en dos sub-secciones: 1. Un algoritmo exacto para resolver un ESPPRC es presentado como el problema de *pricing* a resolver durante la resolución del problema de distribución, 2. Un método de generación de columnas que resuelve el problema de trazado de rutas donde el problema maestro corresponde a un VRPTW.

La segunda sección principal corresponde a la integración del problema de producción con el problema de distribución tratado en la sección anterior. Como consecuencia de esta integración, el sub-problema que genera las rutas para el *MP* es modificado con el fin de encontrar costo reducido mínimo correspondiente al modelo de la sección 3.3. Posteriormente se describe el algoritmo utilizado para resolver el sub-problema correspondiente a la generación de programas de producción. Una tabla de notación se puede encontrar en el anexo B para facilitar la comprensión de las ecuaciones y definiciones introducidas en este capítulo.

4.1. El problema de distribución

Para generar las columnas del problema, las cuales, para este caso, corresponden a rutas factibles, se propone la resolución de un ESPPRC, tal como se explicó anteriormente. Por sus características, el modelo de programación lineal mixta descrito en la sección 3.1.1 puede tomar una cantidad de tiempo bastante considerable si se resuelve directamente por un *solver* comercial. Dado que este problema debe ser resuelto diversas veces por el algoritmo de generación de columnas para encontrar las columnas de costo reducido mínimo para el *MP*, un algoritmo más eficiente es estudiado como una alternativa a la utilización de un *solver* comercial.

4.1.1. Algoritmo de etiquetado

El algoritmo escogido para resolver este problema es el algoritmo de etiquetado propuesto por D. Feillet [15]. Este último tiene la particularidad de tomar en cuenta una cantidad indefinida de recursos en vez de sólo calcular tiempo y/o demanda, lo cual puede ser beneficioso para la futura integración del problema de producción. Otra de sus ventajas, es que al final de su ejecución, se obtiene no sólo la ruta de menor costo reducido, sino que también un conjunto de rutas factible que pueden aportar a alimentar más rápidamente el conjunto Ω del MP relajado.

Formalmente, una etiqueta λ_i^* es un camino desde el origen v_0 hasta un nodo específico $v_i \in V$, que puede ser definida por un costo c_* y un conjunto del consumo de recursos (d_*^1, \dots, d_*^L) donde L es el número de recursos. Tal como se explica en la sección 3.1.1, cada uno de los recursos está restringido a un intervalo de consumo $[a_i^l, b_i^l]$ asociado al nodo v_i .

Para mejorar la eficiencia del algoritmo algunas reglas de “dominación” son definidas para descartar aquellas etiquetas que no son necesarias para encontrar la solución óptima.

Definición 4.1 Sean λ_i^a y λ_i^b dos etiquetas definidas por sus costos y recursos $(c_a, d_a^1, \dots, d_a^L)$ y $(c_b, d_b^1, \dots, d_b^L)$ desde el nodo origen v_0 hasta el nodo v_i . Se dice que λ_i^a **domina** λ_i^b si, solo si $c_a \leq c_b$, $d_a^l \leq d_b^l$, $\forall l \in \{1, \dots, L\}$ y $\lambda_i^a \neq \lambda_i^b$.

La importancia de esta definición es que puede ser fácilmente adaptada para las necesidades del problema en particular. Esto quiere decir que algunos recursos pueden o no ser tomados en cuenta para mejorar los tiempos de resolución, así como para la realización de métodos heurísticos.

Para la ejecución del algoritmo en sí, se utiliza el principio de “dominancia” para limitar la cantidad de etiquetas totales y el de “extensión” de las etiquetas para generar caminos parciales dentro del grafo hasta llegar el depósito v_t . Este último principio se define a continuación.

Definición 4.2 Sea c_{ij} el costo del arco (v_i, v_j) y sea la etiqueta $\lambda_i^* = (c_*, d_*^1, \dots, d_*^L)$. Se define $\lambda_j^{*'} = (c_{*l}, d_{*l}^1, \dots, d_{*l}^L)$ como la **extensión** de la etiqueta λ_i^* al nodo v_j , tal que $c_{*l} = c_* + c_{ij}$ y $\forall l \in \{1, \dots, L\}$, $d_{*l}^l = d_*^l + e_j^l$, donde $e_j^l \geq 0$ corresponde al consumo del recurso l por el nodo v_j .

Des esta forma, se comienza por una etiqueta inicial $\lambda_0 = (\hat{c}_0, 0, \dots, 0)$ que se va propagando a través de los arcos del grafo, ramificándose hasta generar diferentes caminos para llegar al depósito v_t .

Otra noción importante propuesta por D. Feillet [15], es la de *nodos inalcanzables*, que fue añadida como uno de los recursos dentro de las etiquetas. La definición de esta noción se presenta a continuación.

Definición 4.3 Para cada etiqueta $\lambda_i^* = (c_*, d_*^1, \dots, d_*^L)$ que va desde el origen v_0 hasta el el nodo $v_i \in V$, un nodo v_k se considera **inalcanzable** si este ya ha sido visitado por el camino

de la etiqueta λ_i^* ó si existe un recurso $l \in \{1, \dots, L\}$, tal que $d_*^l + e_k^l > b_k^l$, donde b_k^l es el límite superior del consumo del recurso l en el nodo v_k .

Así, como recurso en cada etiqueta se introduce un vector binario de nodos inalcanzables (V^1, \dots, V^n) , donde $V^k = 1$ si el nodo v_k es “inalcanzable”, 0 si no. Además, para facilitar el cálculo de las reglas de dominación, se introduce el recurso $u = \sum_{v_i \in V} V^i$.

Limitación de etiquetas

La cantidad de etiquetas generadas por el algoritmo aumenta de forma exponencial a medida que aumenta la cantidad de nodos, por lo tanto, es necesario reducir este aumento en cada extensión para mejorar el rendimiento del algoritmo de etiquetado. Basándose en la definición 4.1, durante el funcionamiento del algoritmo de resolución del ESPPRC, las etiquetas “dominadas” son desechadas, limitando así en primera instancia la cantidad de etiquetas asignadas a cada nodo. Es importante recalcar que, siguiendo el principio de optimalidad de Bellman, esta acción no compromete el resultado óptimo del algoritmo.

Dado que el algoritmo de etiquetado es ejecutado múltiples veces durante la resolución del *MP*, la limitación de etiquetas por dominación no es suficiente para una resolución a gran escala. Por esta razón, se propone un método heurístico que limita la cantidad de etiquetas asignadas a cada nodo a 100. La elección de las etiquetas que se eliminan es realizada por la magnitud de sus costos, es decir, para una cantidad $100 + s$ de etiquetas sobre un nodo, se eliminan las s etiquetas con mayor costo. Es necesario recalcar que los costos de algunos arcos son negativos, por lo cual esta heurística no asegura el resultado óptimo de la solución. No obstante, dado que el objetivo del problema de *pricing* es generar columnas de costo reducido negativo, se utiliza esta variante hasta encontrar un costo reducido no negativo. En caso de encontrar una columna de costo reducido no negativo, se elimina el límite de etiquetas y se resuelve el problema de forma exacta para asegurar el término del algoritmo, en caso de que la resolución exacta entregue un costo reducido negativo se vuelve a limitar la cantidad de etiquetas y se continúa de la forma precedente.

Implementación del sub-problema

Utilizando la notación de 3.1.1, para la ejecución del algoritmo de etiquetado se definen los siguientes recursos con sus respectivos cálculos para la “extensión” de las etiquetas.

- Tiempo: $t_j = \max(t_i + serv_i + dist_{ij}, a_j)$,
- Demanda: $q_j = q_i + d_i$,
- Nodos Inalcanzables: $u_j \geq u_i + 1$ (luego se recalculan según el consumo de recursos).
Nótese que $V_j^k \geq V_i^k$ ($\forall v_k \in V$).

Asociadas a cada una de las restricciones del problema se tienen distintas variables duales. Más concretamente, luego de resolver una vez el *MP* relajado, para las restricciones 4.3 se tienen valores duales de cobertura α_i para cada uno de los nodos $v_i \in V$. Dichos valores

duales son introducidos en el cálculo del costo de cada una de las etiquetas, para lo cual se redefine el costo de los arcos E del grafo como se muestra a continuación.

$$c'_{ij} = c_{ij} - \alpha_j, \quad (\forall (v_i, v_j) \in E) \quad (4.1)$$

De esta forma, los costos y el consumo de recursos de las etiquetas son calculados de forma iterativa a partir de una primera etiqueta en el origen de costo reducido $\hat{c}_0 = 0$ y consumo nulo de recursos. El algoritmo de generación de columnas se detiene una vez que el sub-problema, resuelto de forma exacta, entrega una ruta de costo reducido mínimo no negativo.

4.1.2. Generación de columnas

Para inicializar el algoritmo de generación de columnas es necesario tener una primera solución factible de la cual se extraerán los primeros valores duales para la resolución del problema de *pricing*. Dado que al inicio del algoritmo es posible que no se encuentre una solución factible que cubra todos los nodos respetando las restricciones de flota de vehículos, se introduce una variable correspondiente al número de vehículos extra, la cual penalizada por un número grande M_Ω , permitiendo así entregar soluciones que no cumplan necesariamente el límite de vehículos. Sea la siguiente variable de decisión,

$\gamma =$ cantidad de vehículos extra.

Utilizando la misma notación utilizada en la sección 3.1.1, el MP queda de la siguiente manera:

$$\text{minimizar } \sum_{r_k \in \Omega} c_k x_k + M_\Omega \times \gamma \quad (4.2)$$

sujeto a

$$\sum_{r_k \in \Omega} a_{ik} x_k \geq 1 \quad (\forall v_i \in V), \quad (4.3)$$

$$\sum_{r_k \in \Omega} x_k - \gamma \leq U, \quad (4.4)$$

$$x_k \in \{0, 1\} \quad (\forall r_k \in \Omega), \quad \gamma \in \mathbb{N}. \quad (4.5)$$

Para la relajación lineal del MP las restricciones 4.5 se cambian por las siguientes:

$$x_k \in [0, 1] \quad (\forall r_k \in \Omega), \quad \gamma \geq 0. \quad (4.6)$$

Es importante recalcar que este cambio en el MP no afecta en la definición del del costo reducido, ya que la introducción de la variable γ corresponde a agregar la siguiente restricción

al problema dual:

$$-g \leq M_\Omega, \quad (4.7)$$

donde se recuerda que g corresponde a la variable dual asociada a la restricción (4.4) que asegura la penalización del exceso de vehículos.

Distintas técnicas existen para entregar un primer conjunto de columnas que permita agilizar las primeras iteraciones del algoritmo. Para este trabajo, se compararon dos soluciones iniciales distintas: la primera consistente en rutas individuales para cada nodo en las cuales n vehículos visitan a uno de los n pacientes y luego vuelve al nodo origen. La segunda solución corresponde a un algoritmo de elaboración propia donde cada nodo se ordena de forma creciente según el inicio de su ventana de tiempo y luego se crean rutas considerando la capacidad límite de los vehículos. Este algoritmo se encuentra descrito en el anexo C.1.

4.2. Integración del problema de producción

Siguiendo la misma lógica utilizada para resolver el problema de distribución, el exceso de operadores usado por el conjunto de solución es penalizado por un número grande M_Ψ . Por consiguiente, la siguiente variable de decisión debe ser introducida al modelo,

$$\eta = \text{exceso de la cantidad operadores.}$$

La función objetivo del modelo integrado descrita en la expresión 3.24 queda entonces definida por la siguiente expresión,

$$\text{minimizar } \sum_{r_k \in \Omega} c_k x_k + M_\Omega \times \gamma + \sum_{p_l \in \Psi} \psi_l y_l + M_\Psi \times \eta \quad (4.8)$$

En consecuencia, la introducción de las variables de decisión γ y η implican que la restricción (3.26) que limita la cantidad de vehículos debe ser reemplazada por la restricción (4.4) y, similarmente, la restricción (3.28) que limita la cantidad de operadores es reemplazada por la siguiente expresión que penaliza el exceso de operadores en vez de limitarlos,

$$\sum_{p_l \in \Psi} y_l - \eta \leq W, \quad (4.9)$$

Finalmente, en lo que respecta a la resolución de la relajación lineal del MP , las restricciones (3.31) son reemplazadas por las siguientes restricciones:

$$x_k \in [0, 1] \quad (\forall r_k \in \Omega), \quad y_l \in [0, 1] \quad (\forall p_l \in \Psi), \quad \gamma \geq 0, \quad \eta \geq 0. \quad (4.10)$$

4.2.1. Modificación del algoritmo de etiquetado de rutas

Para esta nueva formulación que integra tanto el problema de trazado de rutas como el problema de producción, aún es necesario generar rutas de costo reducido negativo que

cumplan con las restricciones descritas en la sección 3.1.1, con la salvedad de que para este caso no es necesario tomar en cuenta la demanda, pues las soluciones factibles del *MP* consideran que necesariamente el fármaco demandado por el cliente es producido y está disponible para su entrega, por lo que la restricción (3.14) es desechada. Esto último no afecta en gran medida la ejecución del algoritmo, pues lo único que se debe modificar es que el recurso de la demanda ya no debe ser tomado en cuenta ni para la extensión ni para las reglas de dominación de las etiquetas.

No obstante, la función objetivo a minimizar por dicho algoritmo sí tiene cambios considerables y, desafortunadamente, esto hace que el algoritmo de etiquetado tal como es descrito en la sección anterior, no necesariamente encuentre la solución óptima a este problema. Esto es debido principalmente a que en la expresión del costo reducido de las rutas en la ecuación 3.36, la función objetivo depende de los horarios de visitas a los pacientes y del horario en el que se inicia la ruta en cuestión. Por lo tanto, es necesario realizar algunas modificaciones al algoritmo de etiquetado.

Para las explicaciones de las modificaciones introducidas, se toma una etiqueta cualquiera λ_k^* correspondiente a una ruta parcial de $k + 1$ nodos comenzando desde el nodo origen v_0 . Los nodos visitados por la ruta son enumerados de forma consecutiva siguiendo el orden respectivo de sus visitas del 0 al k , donde la visita 0 corresponde al nodo origen, por lo cual la ruta queda definida por la sucesión $\{0, 1, \dots, k\}$. A partir de dicha etiqueta se analizan los tiempos de visita de los nodos y el tiempo de inicio que minimizan el costo reducido de la ruta estudiada.

La nueva función objetivo

Dado que los valores duales de precedencia ε_i ($\forall i \in I$) son positivos, pero el valor los duales de estabilidad δ_i ($\forall i \in I$) es negativo, se define el parámetro $\mu_i = -\delta_i$, $\forall i \in I$ para facilitar la comprensión de las demostraciones de las expresiones matemáticas expuestas a continuación, al tratar solamente con parámetros positivos acompañando las variables de decisión temporales s_i , $\forall i \in I$ y s_0 . Siguiendo la misma notación utilizada a lo largo de este informe, el costo reducido que define a la etiqueta λ_k^* queda expresado de la siguiente forma:

$$c_k^* = \sum_{i=1}^k c_{i-1,i} - \sum_{i=1}^k \alpha_i + \sum_{i=1}^k s_i^* \mu_i - s_0^* \sum_{i=1}^k \varepsilon_i. \quad (4.11)$$

Los dos primeros términos, de izquierda a derecha, corresponden al costo reducido de las etiquetas para al algoritmo de etiquetado original y está compuesto solamente de parámetros fijos para el sub-problema. Los dos términos de la derecha corresponden a la introducción de las variables duales del POI y, como se puede notar, dependen del tiempo en que los nodos son visitados por la ruta, el cual corresponde a una variable de decisión del ESPPRC. Debido a las restricciones (3.13) que definen el ESPPRC, estas variables poseen una interdependencia que hace menos evidente el cálculo de los valores que minimizan el costo de la ruta.

Con respecto al sentido físico de la función objetivo del problema de *pricing* de rutas, se puede observar que las variables duales de estabilidad ($\mu_i \geq 0$ en este caso) requieren

que el paciente i sea visitado de forma temprana para asegurar que el medicamento sea utilizado dentro de su tiempo de estabilidad. En cambio, las variables duales de precedencia (ε_i) requieren que la ruta comience de forma tardía para que los medicamentos estén listos antes del inicio de la ruta, representado por el término s_0^* en la ecuación 4.11.

Tiempos de visita óptimos

Se busca encontrar los tiempos de visita que minimizan el costo reducido de la ruta λ_k^* . Para separar la parte dependiente de los tiempos de visita de la parte fija del costo reducido, se define φ_k^* el **costo fijo** de la expresión 4.11 dado por:

$$\varphi_k^* = \sum_{i=1}^k c_{i-1,i} - \sum_{i=1}^k \alpha_i.$$

Este costo fijo no varía al cambiar los valores de los tiempos de visita, por lo cual se puede ignorar para el cálculo del costo reducido mínimo para una ruta dada.

Tomando en cuenta un ejemplo simple en el cual se consideran 4 nodos visitados, con sus respectivas ventanas de tiempo y variables duales, se comienza desde las visitas más tempranas de cada nodo, como se muestra en la Figura 4.1, en la cual cada flecha corresponde al tiempo necesario para llegar de un nodo al siguiente y cada cuadrado representa un nodo de la ruta. El intervalo que se encuentra en la parte superior de cada cuadrado corresponde a la ventana de tiempo de cada nodo, mientras que la tupla en la parte inferior de cada cuadrado corresponde a las variables duales de estabilidad y de precedencia respectivamente de izquierda a derecha. Se define un tiempo de servicio de valor 5 para los 4 nodos presentes en el ejemplo. De izquierda a derecha, se enumeran los nodos del 1 al 4, siendo el 0 el nodo origen no representado en la figura.

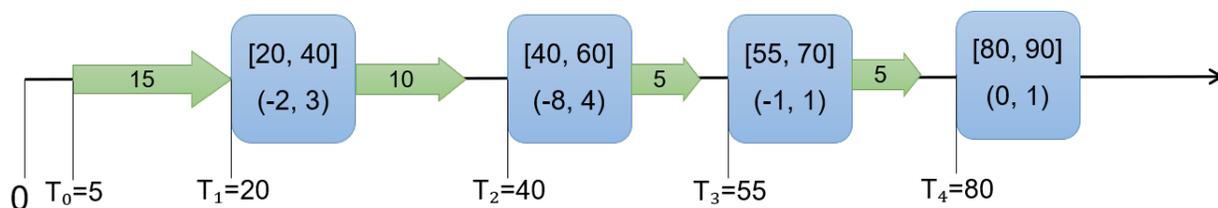


Figura 4.1: Ejemplo de ruta con tiempos de visita más tempranos

El costo variable $\sum_{i=1}^k s_i^* \mu_i - s_0^* \sum_{i=1}^k \varepsilon_i$ de la ruta representada en la Figura 4.1 corresponde a $20 \times 2 + 40 \times 8 + 55 \times 1 + 80 \times 0 - 5 \times 9 = \mathbf{370}$. Esta ruta corresponde a los tiempos de visita más tempranos. Es importante notar que el tiempo de inicio de la ruta es fijado a 5, ya que de todas formas se opta por no dejar tiempo de espera al inicio de la ruta.

El **tiempo de espera** de un nodo corresponde a la diferencia entre el tiempo en que el vehículo llega a destino y el inicio del tiempo de servicio. Se denota σ_i el tiempo que debe esperar el vehículo antes de atender al nodo i . Para el ejemplo anterior, se tiene:

- $\sigma_1 = 0$
- $\sigma_2 = 5$
- $\sigma_3 = 5$
- $\sigma_4 = 15$

El caso en que se desea minimizar el tiempo de espera de la ruta se puede observar en la Figura 4.2. Este corresponde al caso del inicio de la ruta más tardío posible. Para este caso el costo de la ruta corresponde al siguiente: $40 \times 2 + 55 \times 8 + 65 \times 1 + 80 \times 0 - 20 \times 9 = 405$. Este costo es mayor al costo de la Figura 4.1 y es debido al aumento en el tiempo de visita del nodo 2, el cual tiene un valor dual de estabilidad bastante alto comparado con los valores duales de precedencia de la ruta.

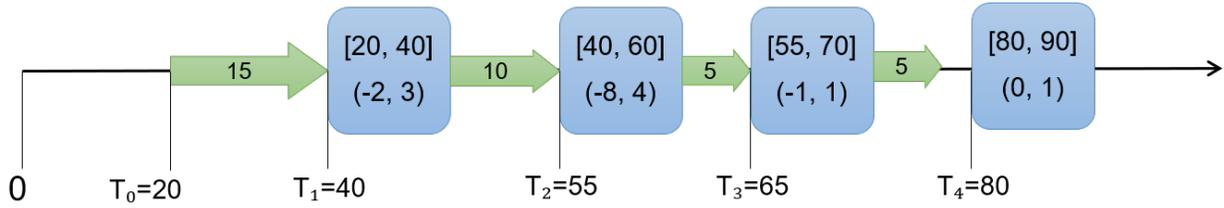


Figura 4.2: Ejemplo de ruta con tiempos de visita más tardíos

Se debe notar que el factor que acompaña al tiempo de inicio de la ruta corresponde a la suma de los valores duales de precedencia de la ruta $\sum_{i=1}^k \varepsilon_i$, que para este ejemplo es igual a 9. Se busca entonces, reducir los tiempos de espera mientras el valor de la suma de los valores duales de precedencia sea mayor o igual a la suma de los valores duales de estabilidad de los nodos precedentes. De forma gráfica, se puede observar que para el primer nodo de la sucesión, el valor dual de precedencia corresponde a 2, lo cual es menor al factor que acompaña a s_0^* , por lo cual es preferible seguir reduciendo los tiempos de espera. En el caso del segundo nodo, la suma de los valores de estabilidad de 1 y 2 es igual a 10, por lo cual no se sigue avanzando el tiempo de inicio de la ruta y se escoge el tiempo más temprano para el resto de los nodos.

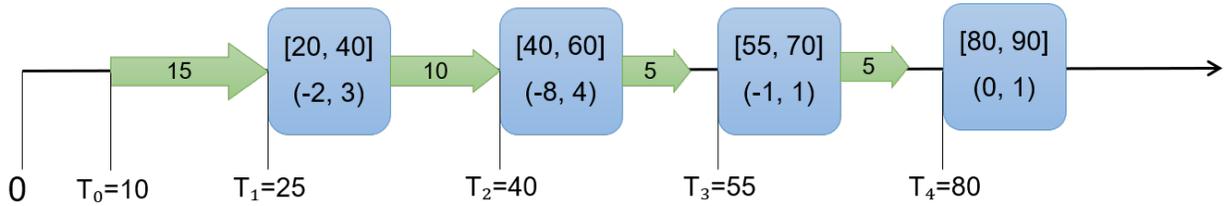


Figura 4.3: Ejemplo de ruta con tiempos de visita óptimos

El costo variable de la ruta de que se observa en la Figura 4.3 corresponde al siguiente: $25 \times 2 + 40 \times 8 + 55 \times 1 + 80 \times 0 - 10 \times 9 = 335$ que es efectivamente menor que los costos calculados anteriormente. De esta forma, es posible definir un método para encontrar los tiempos de visita que minimizan el costo reducido de una ruta en particular.

Definición 4.4 Para una ruta correspondiente a una sucesión finita de nodos v_1, \dots, v_k , se denomina **nodo de quiebre** al primer nodo l de la ruta λ_k^* tal que $\sum_{i=1}^k \varepsilon_i < \sum_{i=1}^l \mu_i$.

La intuición detrás de la definición anterior está en que, para la expresión del costo reducido original dada por la ecuación 4.11, para encontrar el mínimo costo se busca reducir los términos s_i que poseen factores μ_i positivos. En cambio, se busca aumentar el término s_0 ya que posee un factor negativo $-\sum_{i=1}^k \varepsilon_i$, tal como se puede observar en la ecuación 4.11. Sin embargo, al aumentar este último termino en algún momento las variables s_i comenzarán a aumentar de forma sucesiva. Dicho *trade-off* se quiebra en el momento en que se encuentra un nodo que cumpla con la definición 4.4. En términos simples, el tiempo de inicio de la ruta aumentará mientras el factor que acompaña el aumento de s_0 sea mayor o igual al aumento de los factores que acompañan a los nodos que se mueven de forma conjunta con el tiempo de inicio de la ruta.

De esta forma, cada vez que se genera una nueva ruta se debe encontrar el nodo de quiebre. Una vez que este es encontrado, es posible calcular directamente el tiempo de inicio s_0^* de la ruta y luego los tiempos de visita de los nodos corresponden a los tiempos más tempranos dado el tiempo de inicio.

Por otro lado, se tiene que el término s_0^* está acotado a su vez por las siguientes restricciones,

$$s_0^* \leq b_i - \tau_i^*, \quad (\forall i \in \{1, \dots, k\}). \quad (4.12)$$

Donde τ_i^* corresponde al **tiempo estricto** de la ruta parcial entre 0 e i definido por:

$$\tau_i^* = \sum_{j=1}^i serv_{j-1} + dist_{j-1,j}.$$

Se recuerda al lector que el término $serv_j$ es el tiempo de servicio requerido por el nodo $v_j \in V$ y que el término $dist_{j-1,j}$ corresponde al tiempo necesario para ir desde el nodo v_{j-1} al nodo v_j .

Con el fin de simplificar la notación para los cálculos posteriores, se definen las siguientes funciones que fijan cotas para el cálculo del valor de s_0^* ,

- $\bar{A}_i^* = \max_{j=1}^i a_j - \tau_j^*$
- $B^* = \min_{i=1}^k b_i - \tau_i^*$

Teniendo en cuenta estas definiciones es posible demostrar que, para una etiqueta λ_k^* , el tiempo de inicio que define el costo reducido mínimo para dicha ruta parcial, se define como:

$$s_0^* = \begin{cases} \min\{\bar{A}_l^*, B^*\} & \text{si existe un nodo de quiebre } l > 0 \\ B^* & \text{si no.} \end{cases}$$

A partir del cual es posible calcular el resto de los tiempos de visita de la ruta de forma recurrente.

Desde el punto de vista práctico del algoritmo de generación de rutas, para evitar tener que comenzar el recorrido de los nodos desde la mitad de la secuencia, se modifica la definición de nodo de quiebre dada la siguiente propiedad.

Propiedad 1. $\sum_{i=1}^k \varepsilon_i < \sum_{i=1}^l \mu_i$ sí, solo si $\sum_{i=1}^k \mu_i - \varepsilon_i > \sum_{i=l}^k \mu_i$.

Esta propiedad viene directamente del complemento de la condición descrita en la Definición 4.4.

De la misma forma, se nombra **diferencia dual** a la expresión

$$\Lambda^* = \sum_{i=1}^k \mu_i - \varepsilon_i.$$

Esto permite llevar el cálculo de un término que toma en cuenta ambas variables duales a lo largo de la ejecución del algoritmo. Además, esto permite calcular directamente la condición para encontrar el nodo de quiebre a partir del último nodo ingresado en la secuencia de la ruta. Un desarrollo extensivo de las demostraciones de lo expuesto puede ser encontrado en el anexo E.1.

Extensión y dominación de etiquetas

Para agilizar la extensión de las rutas durante la ejecución del algoritmo se utilizan las siguientes propiedades:

Propiedad 2. Si $l \in \{1, \dots, k\}$ es tal que $\Lambda^* > \sum_{j=l}^k \mu_j$ entonces,

$$\Lambda^* > \sum_{j=i}^k \mu_j \quad (\forall i \geq l)$$

Propiedad 3. Si $l \in \{1, \dots, k\}$ es tal que $\Lambda^* \leq \sum_{j=l}^k \mu_j$ entonces,

$$\Lambda^* \leq \sum_{j=i}^k \mu_j \quad (\forall i \leq l)$$

Las propiedades 2 y 3 permiten detener el recorrido de los nodos una vez encontrada la condición del nodo de quiebre.

Para ajustar aún más los cálculos del algoritmo, se denota **factor de estabilidad** a la expresión

$$\Delta^* = \sum_{i=1}^k \mu_i \sum_{j=1}^i \tau_j^*.$$

Este factor corresponde al mínimo valor al cual se puede reducir el aporte que entrega los tiempos de visita de los nodos de la ruta. Esto quiere decir que el valor de $\sum_{i=1}^k s_i^* \mu_i$ será de al menos el valor del factor de estabilidad. La diferencia entre ambos valores dependerá del tiempo de inicio y de los tiempos de espera de la ruta.

Por lo tanto, durante la ejecución del algoritmo de etiquetado modificado, además de los recursos ya definidos en la sección 4.1.1, se añaden los recursos listados a continuación.

- Costo fijo φ_k^* ,
- Tiempo estricto φ_k^a ,
- Factor de estabilidad Δ^* ,
- Diferencia dual Λ^* ,
- Tiempo de inicio de la ruta s_0^* .

Es necesario recalcar que, a excepción del tiempo estricto, estos recursos **no cumplen** con la desigualdad triangular debido a que los valores duales entregados por el *MP* pueden tomar cualquier valor entre nodos conectados entre sí. Debido a esto, ya no es intuitivo utilizar la noción de “recurso” para referirse a estas cantidades. Por lo tanto, estos parámetros que definen cada etiqueta, para lo que resta del presente informe, se denominan “cantidades características”.

Para las reglas de dominación, se utiliza la definición 4.1, con la salvedad de que, para el tiempo de inicio de la ruta este es comparado de forma inversa, es decir, que para que una etiqueta λ^a domine a otra etiqueta λ^b , entonces $s_0^a \geq s_0^b$. Por otro lado, las cantidades características del costo fijo y el factor de estabilidad son cantidades fijas dentro del cálculo del costo reducido de la etiqueta, por lo que se compara la suma de ambos.

Definición 4.5 Sean λ_i^a y λ_i^b dos etiquetas definidas por sus costos y cantidades características $(c_a, T_a, u_a, V_a^1, \dots, V_a^n, \varphi_i^a, \tau_i^a, \Delta^a, \Lambda^a, s_0^a)$ y $(c_b, T_b, u_b, V_b^1, \dots, V_b^n, \varphi_i^b, \tau_i^b, \Delta^b, \Lambda^b, s_0^b)$ desde el nodo origen v_0 hasta el nodo v_i . Se dice que λ_i^a **domina** λ_i^b si, solo si, $\lambda_i^a \neq \lambda_i^b$ y

- $c_a \leq c_b$,
- $T_a \leq T_b$,
- $u_a \leq u_b$,
- $V_a^k \leq V_b^k \forall k \in \{1, \dots, n\}$,
- $\varphi_i^a + \Delta^a \leq \varphi_i^b + \Delta^b$,
- $\Lambda^a \leq \Lambda^b$,
- $s_0^a \geq s_0^b$.

La explicación detrás de la definición 4.5, está en el concepto de nodos inalcanzables. El hecho de que, para que una etiqueta domine a otra, los nodos inalcanzables para la etiqueta dominante deben ser también inalcanzables para la etiqueta dominada, puede ser visto como la oportunidad que una etiqueta posee para disminuir su costo. De esta forma, se comparan los valores que influyen en el cálculo del costo reducido de la etiqueta y las etiquetas que presentan tanto mayor costo como menor oportunidad de reducción del costo son dominadas por aquellas de mayor posibilidades de reducción del costo.

4.2.2. Algoritmo para la resolución del problema de pricing de producción

Dado que el problema de producción es modelado de forma similar que el problema de distribución, mediante un enfoque basado en el modelo de *set-partitioning*, se plantea resolver el

sub-problema para la generación de programas factibles mediante el algoritmo de etiquetado planteado a lo largo del presente informe. Esto ya que las condiciones de factibilidad de los programas corresponden a las mismas necesarias para las rutas, esto es, que los tiempos de visita o de programación estén dentro de las ventanas de tiempo.

Se considera una etiqueta cualquiera ρ_l correspondiente a un programa parcial de l nodos. Los nodos asignados por el programa son enumerados de forma consecutiva siguiendo el orden respectivo de sus asignaciones del 1 al l , donde la asignación 1 corresponde al primer nodo asignado, por lo cual el programa queda definido por la sucesión $\{1, \dots, l\}$. A partir de esta etiqueta se analizan los tiempos de asignación de los nodos que minimizan el costo reducido del programa en cuestión. El costo reducido que se desea minimizar corresponde al siguiente.

$$\hat{\psi}_l = w_l + st_l - w_1 - \sum_{i=1}^l \beta_i - \sum_{i=1}^l (w_i + st_i)\mu_i + \sum_{i=1}^l w_i \varepsilon_i \quad (4.13)$$

Se puede definir el **costo fijo** θ_l del programa de la siguiente forma.

$$\theta_l = st_l - \sum_{i=1}^l st_i - \sum_{i=1}^l \beta_i$$

Por lo cual se busca minimizar la parte variable del costo reducido de la ruta correspondiente a:

$$\phi_l = w_l - w_1 + \sum_{i=1}^l w_i(\varepsilon_i - \mu_i) \quad (4.14)$$

Al analizar la ecuación 4.14, se puede observar que se busca encontrar los tiempos de asignación que minimizan ϕ_l , los cuales a su vez pueden poseer tanto factores positivos, que corresponde a un tiempo de asignación lo más temprano posible, como factores negativos, correspondientes a tiempos de asignación lo más tarde posible. La dificultad para encontrar estos tiempos de asignación está en las condiciones de factibilidad de los programas, ya que estos tiempos deben respetar el orden de asignación, esto es, la no interrupción de la tarea precedente y estar dentro de las ventanas de tiempo.

Como ejemplo del razonamiento que se debe seguir para encontrar los tiempos de asignación óptimos, se pueden tomar los mismos nodos utilizados anteriormente, los cuales se pueden observar en la Figura 4.4. El intervalo que se encuentra en la parte superior de cada cuadrado corresponde a la ventana de tiempo de cada nodo, mientras que la tupla en la parte inferior de cada cuadrado corresponde a las variables duales de estabilidad y de precedencia respectivamente de izquierda a derecha. Se define un tiempo de procesamiento de valor 5 para los 4 nodos presentes en el ejemplo.

Para el nodo 1, se puede notar que el factor que acompaña a la variable de decisión del tiempo de asignación es igual a $3 - 2 - 1 = 0$. El -1 se debe a que es el primer nodo de la secuencia. Para la generación de programas, se considera que el factor 0 corresponde a una asignación temprana, por lo que el primer nodo es asignado en el inicio de su ventana de tiempo.

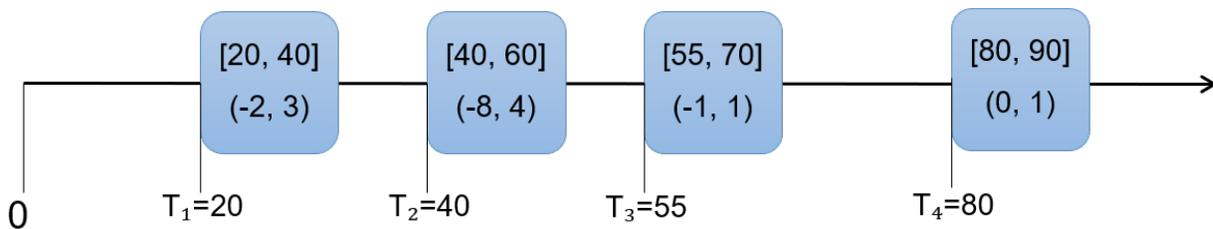


Figura 4.4: Ejemplo de programa con tiempos de asignación tempranos

Para el nodo 2, se tiene que el factor que lo acompaña es un factor negativo igual a $4 - 8 = -4$, por lo tanto se atrasa el tiempo de su asignación como se muestra en la Figura 4.5.

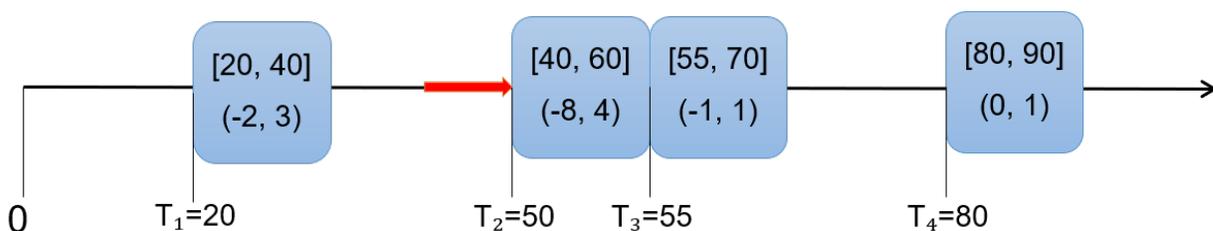


Figura 4.5: Ejemplo de programa, con mejora en la calidad de la solución

Con respecto al nodo 3, el factor que lo acompaña es igual a $1 - 1 = 0$, sin embargo, como el nodo 2 tiene un factor negativo, este último va a atrasar el tiempo de asignación del nodo 3 hasta llegar al final de la ventana de tiempo del nodo precedente como se muestra en la Figura 4.6.

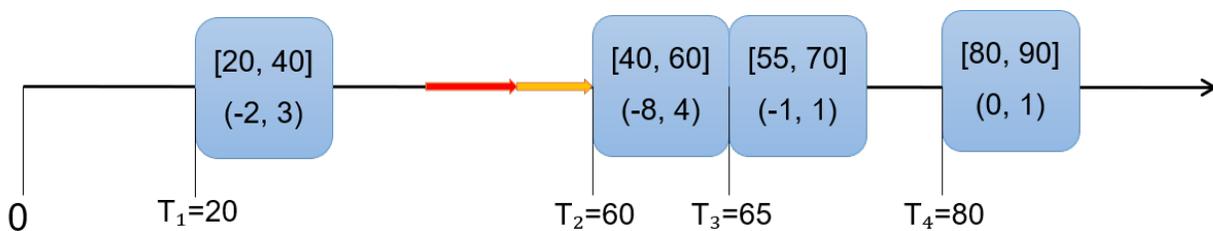


Figura 4.6: Ejemplo de programa, con tiempos de asignación óptimos

Finalmente, el nodo 4 tiene un factor igual a $1 - 0 + 1 = 2$, dado que es el último nodo de la secuencia. De esta forma, el tiempo es fijado al inicio de su ventana de tiempo y no afecta a los tiempos fijados anteriormente.

Con respecto a la complejidad del cálculo de los tiempos de asignación, para los nodos que están a los extremos este cálculo es constante. Desafortunadamente, para el resto de los nodos, se deben encontrar los nodos con los cuales se mueve de forma conjunta, es decir, que al disminuir o aumentar su tiempo de asignación los otros nodos también disminuyen o aumentan el suyo, y luego se tienen que actualizar los tiempos de cada uno de dichos nodos conjuntos. De esta forma, la complejidad del cálculo del tiempo de asignación es de $\mathcal{O}(n^2)$.

Recorrido hacia atrás y extensión de etiquetas

Dadas las características del problema, se busca asignar al principio del programa las tareas que posean un mayor factor que el resto de los nodos de la secuencia. Siguiendo la misma lógica, se busca asignar al final de la secuencia, aquellas tareas que posean un factor negativo. De esta forma, el algoritmo de etiquetado se inicia en un nodo depósito ficticio y las tareas se van asignando antes de las tareas asignadas en el programa hasta llegar al nodo origen ficticio. Esto permite obtener menores costos de forma más rápida, al asignar los nodos con factores negativos directamente al final de la secuencia.

No obstante, es necesario tener en consideración que el factor del último nodo agregado cambia cada vez que la etiqueta es extendida, ya que este deja de ser el primer nodo asignado. Esto hace mucho más complejo el proceso de etiquetado del algoritmo, por lo cual se introduce el concepto de **costo abierto**. Dicho costo corresponde a la siguiente expresión:

$$\phi_l + w_0 = w_l + \sum_{i=1}^l w_i(\varepsilon_i - \mu_i)$$

Este parámetro permite al algoritmo tener un valor comparable del costo entre las distintas etiquetas sin tener que recalcular el costo nodo por nodo cada vez que crea una nueva etiqueta. De todas formas, los tiempos de visita deben ser recalculados en cada extensión, sin embargo el recorrido de los nodos puede ser detenido una vez que el nuevo nodo ingresado no afecta en los tiempos de los nodos precedentes.

Finalmente, dada la complejidad de la función objetivo del problema de *pricing* de los programas, no se definen reglas de dominación exactas. Esto dado que, incluso calculando el costo real del programa, este se ve afectado durante el proceso de extensión al cambiar el factor que acompaña al tiempo de asignación del último nodo añadido. Una clasificación según el costo abierto es utilizada para la implementación de este algoritmo y se compararan los tiempos de asignación finales así como los tiempos de asignación de los primeros nodos de la secuencia (que corresponden a los últimos nodos añadidos). Sin embargo, esta clasificación de etiquetas no es exacta y corresponde a una regla de dominación heurística, por lo cual la eficacia del algoritmo no está asegurada.

Capítulo 5

Resultados

Esta sección está organizada como se describe a continuación. Primeramente se describen las instancias utilizadas para probar la eficiencia del modelo propuesto para la resolución del POI. Luego, se presentan los resultados obtenidos mediante una simulación del algoritmo de etiquetado modificado descrito en la sección 4.2.1 utilizado para resolver el problema de *pricing* para la generación de nuevas rutas que se entregan al *MP* relajado. Posteriormente, se muestran los resultados del algoritmo de etiquetado modificado para la entrega de programas factibles descrito en la sección 4.2.2. Finalmente, se resuelve el POI a través del *solver* CPLEX introduciendo el modelo descrito en la sección 3.3. Esto último consiste en resolver el *MP* con las modificaciones que permiten la entrega de soluciones que sobrepasan el límite de operadores, tal como se describe en la sección 4.2, siguiendo el enfoque de BnP.

5.1. Instancias de prueba

Las pruebas fueron realizadas sobre las instancias R, C y RC de Solomon [21], las cuales pueden ser encontradas libremente en internet. Estas instancias corresponden a simulaciones de problemas de distribución. Todas las instancias contienen 100 puntos en un plano de 2 dimensiones entre los cuales es posible calcular la distancia euclidiana, esta última fue utilizada como el tiempo necesario para recorrer los arcos entre cada uno de los nodos. Asociados a cada nodo, se tienen un tiempo inicial y un tiempo final de la ventana de tiempo, así como un tiempo de servicio y una demanda. La diferencia entre los tipos de instancia está en el método utilizado para generar los datos. Para las instancias R, estas fueron generadas de forma aleatoria, las instancias C corresponden a agrupaciones de nodos, finalmente las instancias RC son una mezcla de ambos datos, aleatorios y agrupados.

Las instancias mencionadas en el párrafo anterior son pre-procesadas, tomando en cuenta las ventanas de tiempo de cada uno de los nodos para eliminar los arcos no factibles dentro del grafo. De esta forma, se reduce la cantidad de arcos mejorando en el rendimiento de los algoritmos probados. Es importante destacar que el tiempo del proceso anterior no es incluido en ninguno de los resultados presentados en esta sección, ya que solo representa un tiempo fijo dependiendo de la cantidad de nodos de la instancia.

En conjunto con el pre-procesamiento anterior, se generan instancias correspondientes al problema de producción asociadas a cada una de las instancias de distribución de Solomon. Para la simulación del problema de producción se tomó en cuenta un solo tipo de medicamento para todos los pacientes de cada instancia. Para las instancias R y RC, dado que el horizonte de tiempo entre 230 y 240 unidades de tiempo, se tomó en cuenta un tiempo de estabilización igual 100 unidades y un tiempo de producción de 10 unidades. Dado que para las instancias de tipo C el horizonte de tiempo es de aproximadamente 1200 unidades de tiempo, para estas se tomó en cuenta un tiempo de estabilización de 400 unidades y un tiempo de producción de 20 unidades.

Los valores de prueba se eligieron considerando que la realización de quimioterapias se realizan durante el día, por lo cual, tomando un horizonte de 12 horas y considerando que los medicamentos con tiempo de estabilización más cortos son de entre 4 y 6 horas [10], se tiene que los medicamentos que restringen en mayor medida la realización de las rutas tienen un tiempo de estabilización de entre $1/4$ y $1/2$ el horizonte de tiempo dado. El tiempo de producción fue calculado en base a las distancias y las ventanas de tiempo del grafo, es importante notar que este tiempo puede variar para el caso real, el cual depende del lugar de producción (para este caso se considera dentro del hospital) y de la especialización de los químico farmacéuticos.

5.2. Resolución de sub-problemas

Con el fin de probar la eficiencia de los algoritmos, se simula primeramente la ejecución de los algoritmos que resuelven los sub-problemas para la generación de columnas. Para esto, se entregan de forma aleatoria valores duales emulando las variables que el *MP* relajado entrega a los sub-problemas. Para la presente simulación se utilizaron distintos intervalos para cada uno de los tipos de variables como se resume en la siguiente tabla.

Dual Variable	Interval
α	[-50, 50]
β	[-50, 50]
δ	[-2, 0]
ε	[0, 2]

Tabla 5.1: Intervalos de generación de valores duales aleatorios

De la Tabla 5.1, las variables duales de cobertura α y β corresponden a valores enteros y las variables de estabilidad δ y de precedencia ε a valores continuos. Dichos valores fueron generados basándose en la observación de la ejecución del algoritmo de generación de columnas en pruebas primitivas, las cuales fueron realizadas con algoritmos de fuerza bruta a muy pequeña escala.

Con respecto a los resultados obtenidos mediante las simulaciones, todos los tiempos mostrados están en **milisegundos** y aproximados a la décima. Los tiempos corresponden al tiempo transcurrido durante la ejecución del algoritmo y es medido a través de la función *System.nanoTime()* de Java.

5.2.1. Ejecución del algoritmo de etiquetado modificado para el trazado de rutas factibles

Para la ejecución del algoritmo modificado descrito en la sección 4.2.1, el tiempo de resolución es fijado a un máximo de 300 segundos, ya que el sub-problema debe ser resuelto varias veces dentro del método de BnP. Considerando la necesidad de encontrar soluciones factibles de la forma más eficiente posible, se compara el rendimiento del algoritmo exacto con una resolución heurística limitada a 100 etiquetas, como se encuentra descrito en la sección 4.1.1. Se realizaron pruebas para 3 tamaños distintos de instancia, tomando los primeros 25 nodos de las instancias de prueba, tomando los primeros 50 y finalmente con las instancias completas de 100 nodos.

Los resultados agrupados por tipo de instancia y cantidad de nodos de estas pruebas se pueden observar en la Tabla 5.2. El detalle puede ser encontrado en el anexo F. Es importante notar que, para este algoritmo, 13 de las 87 instancias probadas no consiguieron ser resueltas dentro de los 300 segundos.

Instance Type	Nodes	Time	Heuristics Time	Cost Gap	Solved	Total
R	25	27,6	21,9	0,00 %	12	12
C	25	17.849,1	27,4	0,00 %	9	9
RC	25	5,5	5,3	0,00 %	8	8
AVG	25	5.552,31	19,03	0,00 %	29	29
R	50	2.535,9	171,7	0,08 %	12	12
C	50	8.951,1	56,6	2,01 %	7	9
RC	50	342,1	57,4	0,00 %	8	8
AVG	50	3.549,1	108,0	0,55 %	27	29
R	100	54.432,4	692,2	1,25 %	5	12
C	100	8.227,8	374,5	0,51 %	6	9
RC	100	16.169,9	590	1,02 %	7	8
AVG	100	25.513,00	568,06	0,97 %	18	29
Total	-	9.676,72	185,03	0,44 %	74	87

Tabla 5.2: Tiempo de resolución del algoritmo modificado de trazado de rutas

Los resultados promedio que se muestran en la Tabla 5.2, sólo toman en cuenta aquellas instancias que fueron resueltas dentro del tiempo límite. La columna *Heuristics Time* corresponde al tiempo de resolución del algoritmo con una cantidad de limitada de etiquetas por nodo y la columna *Cost Gap* corresponde al porcentaje de la diferencia entre el costo entregado por el algoritmo exacto y el costo entregado por el algoritmo heurístico sobre el costo exacto. Se puede observar que la pérdida en la calidad de las soluciones del algoritmo heurístico es, en promedio, menor al 1 %.

En general, el método de trazado de rutas resulta bastante eficiente para instancias de hasta 50 nodos. Para el caso de 100 nodos, el algoritmo no logra resolver 11 de las 29 instancias en menos de 300 segundos, sin embargo, para estas instancias el método heurístico encuentra soluciones de menor costo para la mayoría de los casos, y en los demás casos la solución heurística tiene una diferencia de costo de menos del 2 %.

5.2.2. Ejecución del algoritmo de etiquetado modificado para la generación de programas factibles

Tal como fue fijado para el algoritmo de trazado de rutas, el tiempo de ejecución para el algoritmo de generación de programas fue limitado a 300 segundos. Los resultados se comparan con la resolución heurística del algoritmo limitado a 200 etiquetas por cada nodo. La elección de duplicar la cantidad límite de etiquetas con respecto al algoritmo de trazado de rutas es debido a que el problema de producción, por su construcción, posee ventanas de tiempo más amplias que el problema de distribución, por lo cual se tiene una mayor cantidad de arcos en el grafo. Asimismo, el desempeño del algoritmo se ve afectado tanto por la densidad de los grafos como por la cantidad de cálculos que se deben realizar durante la ejecución del algoritmos, específicamente durante la etapa de extensión de las etiquetas. Es importante destacar que la clasificación de las etiquetas para este algoritmo se realiza considerando el costo abierto y no el costo real, ya que este último es calculado al final del algoritmo. Debido a la complejidad del algoritmo, las pruebas fueron realizadas tan solo para los 10 primeros nodos de cada instancia, luego para los primeros 12 y finalmente para los primeros 15 nodos.

Instance Type	Nodes	H. Time	Cost Gap	Feasible	Generated	Solved	Total
R	10	121	0,10 %	1.023	57.804	12	12
C	10	45	0,18 %	848	34.921	9	9
RC	10	77,13	0,09 %	938	63.853	8	8
AVG	10	85,10	0,12 %	945	52.371	29	29
R	12	173	-2,73 %	1.414	211.502	2	12
C	12	96	-3,14 %	957	123.932	5	9
RC	12	158	-7,57 %	1.230	200.236	1	8
AVG	12	144,83	-4,19 %	1.221	181.217	8	29
R	15	279	-5,80 %	1.330	214.194	1	12
C	15	398	-10,25 %	2.837	202.977	2	9
RC	15	406	-6,50 %	1.137	220.593	1	8
AVG	15	351,00	-7,37 %	1.745	212.478	4	29
Total	-	193,64	-3,81 %	1.304	148.689	41	87

Tabla 5.3: Tiempo de resolución del algoritmo modificado de generación de programas

En la Tabla 5.3 se resumen los resultados de la simulación del algoritmo de generación de programas factibles. Dado que tan solo se lograron resolver 41 de las 87 instancias en menos de 300 segundos, el tiempo promedio del algoritmo exacto no es incluido en esta tabla resumen, ya que más de la mitad de los resultados se encuentran truncados a 300 segundos. El detalle de estas experiencias se puede encontrar en el anexo G.

La eficiencia de este algoritmo resulta limitante para la resolución del POI, sobretodo al compararse con su símil de trazado de rutas. Sin embargo, al igual que en los resultados del algoritmo de *pricing* para las rutas, el método que limita la cantidad de etiquetas resulta en una diferencia en el costo de menos del 5% en cada una de las instancias resueltas en menos de 300 segundos y, en la mayoría de las instancias no resueltas, encuentra una solución con un costo de hasta un 21% menor que el método exacto.

5.3. Resolución del Problema Operacional Integrado

El método para resolver el POI corresponde a uno de BnP y sigue las mismas características que el descrito en la sección 3.1.1. El algoritmo comienza con un conjunto de rutas y programas unitarios, como los descritos en la sección 4.1.2. Si bien se realizaron experimentos entregando un conjunto de rutas distinto al unitario, este primer conjunto no siempre asegura la existencia de una combinación factible con los programas de producción, por lo cual se optó finalmente por generar conjuntos unitarios tanto para las rutas como para los programas de forma de asegurar una primera solución factible, esto habiendo incluido en la función objetivo las variables de exceso de vehículos y operadores como se describe en la sección 4.2, estas variables son penalizadas con un número grande igual al máximo horizonte de tiempo de las instancias de prueba.

En cada iteración se resuelve el *MP* relajado del POI y luego ambos sub-problemas se resuelven de forma heurística, con un límite de 100 etiquetas para el algoritmo de trazado de rutas y un límite de 200 etiquetas para el de generación de programas, y se ingresan todas las rutas y todos los programas que posean un costo reducido negativo. Una vez que un sub-problema entrega un costo reducido mínimo no negativo, este se resuelve de forma exacta. El método se detiene una vez que ambos sub-problemas resueltos de forma exacta entregan un costo reducido mínimo no negativo, entregando una cota inferior del óptimo. Posteriormente, se resuelve el problema entero que entrega una cota superior del óptimo. Dadas las limitaciones del problema de *pricing* de generación de programas, se realizaron pruebas para 10 nodos con 6 vehículos y 6 operadores y para 15 nodos con 10 vehículos y 10 operadores. El método de BnP se limitó a 3600 segundos, en cuanto a la resolución de los sub-problemas se limitó a 600 segundos, finalmente, la resolución del *MP* entero fue limitada a 300 segundos.

En el caso en que el método de generación de columnas haya sido detenido antes de haber encontrado columnas de costo reducido no negativo, no se obtiene cota inferior. Sin embargo, se resuelve de todas formas el *MP* entero entregando una solución entera al problema operacional integrado. Para el caso en que una cota inferior haya sido encontrada, se calcula una diferencia entre ambas cotas mediante la siguiente fórmula $Gap = \frac{CotaSup.-CotaInf.}{CotaInf.}$, dicha diferencia puede interpretarse como el porcentaje de cercanía de la solución entera encontrada con la solución óptima. Los tiempos de cálculo se muestran en **segundos** y corresponden al tiempo transcurrido durante la ejecución del algoritmo medido a través de la función *System.nanoTime()* de Java.

En la Tabla 5.4 se muestran los resultados promedio agrupados por tipo de instancia y por cantidad de nodos. Para 10 nodos, las 29 instancias de prueba pudieron ser resueltas dentro del tiempo límite y teniendo un tiempo promedio de 23,1 segundos, lo cual se enmarca dentro un tiempo aceptable de resolución. Al contrario, para instancias de 15 nodos tan solo 17 pudieron ser resueltas. Sin embargo, dentro de las instancias R la mayoría de las instancias fueron resueltas y la calidad de la solución se encuentra dentro de un *Gap* del 11,35 % para este tipo de instancias. Es importante tomar en cuenta que las soluciones que sobrepasan la cantidad de vehículos o de operadores tienen un alto costo de penalización, por lo cual el cálculo del *Gap* puede elevarse considerablemente para casos en que no exista solución entera que cumpla con este límite dentro de las columnas generadas.

Type	Nodes	Gap	E. Time	Cols. R	Cols. S	Iter.	Solved	Total
R	10	7,60 %	11,7	1.752	148	6,8	12	12
C	10	284,18 %	42,3	1.539	452	12,0	9	9
RC	10	265,74 %	18,7	2.418	330	12,1	8	8
AVG	10	164,65 %	23,1	1.869,5	292,8	9,9	29	29
R	15	11,35 %	2.209,2	8.520	289	8,0	11	12
C	15	17,03 %	2.942,0	3.910	742	10,9	3	9
RC	15	20,85 %	3.491,6	6.171	487	11,0	3	8
AVG	15	14,03 %	2.790,4	6.441,4	484,2	9,7	17	29

Tabla 5.4: Resultados agrupados de la resolución del POI

Otro resultado interesante es la cantidad de columnas generadas por cada resolución de los problemas de *pricing*. La columna *Cols. R* corresponde a la cantidad de rutas de costo reducido negativo y la columna *Cols. S* a la cantidad de programas de costo reducido negativo generadas durante la resolución del POI. Se puede observar que la cantidad de programas generados es al rededor de un 10 % la cantidad de rutas generadas. Por otro lado, la cantidad de iteraciones es similar para instancias de 10 y 15 nodos, por lo cual se puede afirmar que el tiempo promedio de cada iteración es abiertamente mayor para 15 nodos en comparación con las instancias de 10 nodos. A continuación se analizan los resultados detallados para las instancias de 10 y 15 nodos.

5.3.1. POI para instancias de 10 nodos

Los resultados detallados para cada instancia de prueba se encuentran en la Tabla 5.5. las columnas *D. Density* y *P. Density* muestran, respectivamente para los problemas de distribución y de producción, el porcentaje de arcos con respecto a la cantidad de nodos de cada instancia. La columna *LB* corresponde a la cota inferior del costo, si esta fue encontrada, mientras que la columna *UB* a la cota superior. En caso de haber encontrado una cota inferior, se calcula el *Gap* mediante la fórmula descrita anteriormente. Las columnas *Routes* y *Sched.* muestran respectivamente la cantidad de rutas y programas utilizados en la solución entera, mientras que *Cols. R* y *Cols. S* corresponden a la cantidad de rutas y programas de costo reducido negativo generados mediante la resolución respectiva de los sub-problemas.

Primeramente, se puede observar que no existe correlación directa entre la densidad de los grafos de distribución y producción y el tiempo de resolución del algoritmo de generación de columnas. Por lo tanto, dichas diferencias en el tiempo de resolución puede deberse a factores que no han sido medidos durante la ejecución de las experiencias. En segundo lugar, y tal como se mencionó en la descripción de la Tabla 5.4, se observa que para las instancias C102, C103, C104 y RC104, el costo de la solución entera es severamente penalizado debido al exceso de rutas y/o programas. Si bien el resultado promedio del *Gap* calculado se ve afectado directamente por este fenómeno, al observar los valores del *Gap* para las instancias C que cumplen con la cantidad límite de rutas y programas, se puede notar que estos están al rededor del 15 % y, para el caso de las instancias RC, el *Gap* se encuentra alrededor de un 25 % llegando incluso a un 50,7 % para la instancia RC108.

Instance	D. Density	P. Density	LB	UB	Gap	E. Time	Routes	Sched.	Cols. R.	Cols. S.	Iter.
R101	45,45 %	93,64 %	365,6	385,1	5,33 %	1,0	4	5	881	83	6
R102	84,55 %	100,00 %	330,9	356,0	7,60 %	3,0	4	6	630	144	6
R103	84,55 %	100,00 %	330,9	356,0	7,60 %	2,8	4	6	630	144	6
R104	93,64 %	100,00 %	301,4	341,9	13,44 %	7,9	4	6	1065	185	6
R105	52,73 %	97,27 %	358,6	366,6	2,23 %	3,8	4	6	1186	90	6
R106	88,18 %	100,00 %	319,2	341,9	7,11 %	11,2	4	6	1762	137	7
R107	88,18 %	100,00 %	319,2	341,9	7,11 %	9,4	4	6	1762	137	7
R108	95,45 %	100,00 %	301,4	329,2	9,22 %	15,7	3	6	2895	174	9
R109	65,45 %	100,00 %	336,3	358,3	6,53 %	36,0	3	5	4482	142	6
R110	80,00 %	100,00 %	322,7	335,6	4,00 %	36,0	3	5	3507	178	8
R111	89,09 %	100,00 %	315,8	353,1	11,83 %	7,7	4	3	1492	144	6
R112	100,00 %	100,00 %	301,4	329,2	9,23 %	5,6	3	6	729	223	8
AVG					7,60 %	11,7	3,7	5,5	1.752	148	6,8
C101	59,09 %	82,73 %	284,4	327,0	14,99 %	1,8	3	6	1085	364	10
C102	90,91 %	99,09 %	269,3	2.872,1	966,47 %	79,4	5	8	1942	465	13
C103	90,91 %	99,09 %	269,3	2.872,1	966,47 %	79,9	5	8	1942	465	13
C104	97,27 %	99,09 %	267,1	1.691,5	533,22 %	81,9	7	6	1655	824	21
C105	59,09 %	85,45 %	283,1	327,0	15,49 %	1,0	3	6	670	341	8
C106	59,09 %	84,55 %	283,9	324,4	14,26 %	2,5	3	6	1340	428	10
C107	60,00 %	87,27 %	281,8	322,6	14,48 %	1,8	3	5	909	410	9
C108	68,18 %	92,73 %	280,8	328,2	16,86 %	6,4	3	6	688	348	9
C109	75,45 %	97,27 %	277,5	320,1	15,36 %	126,5	3	3	3623	423	15
AVG					284,18 %	42,3	3,9	6,0	1.539	452	12,0
RC101	58,18 %	96,36 %	315,9	410,2	29,83 %	2,5	4	5	274	151	6
RC102	90,00 %	100,00 %	271,9	338,8	24,59 %	29,0	3	6	4479	397	15
RC103	90,00 %	100,00 %	271,9	338,8	24,59 %	29,1	3	6	4479	397	15
RC104	96,36 %	100,00 %	267,2	5.404,6	1922,95 %	16,0	5	10	829	327	11
RC105	77,27 %	100,00 %	292,1	351,2	20,21 %	20,1	3	6	671	260	11
RC106	75,45 %	100,00 %	299,1	341,6	14,21 %	19,1	3	6	4222	241	9
RC107	88,18 %	100,00 %	276,8	384,2	38,82 %	20,4	3	5	2861	365	13
RC108	96,36 %	100,00 %	267,5	403,2	50,70 %	13,4	4	4	1525	505	17
AVG					265,74 %	18,7	3,5	6,0	2.418	330	12,1
TOTAL					164,65 %	23,1	3,7	5,8	1.869,5	292,8	9,9

Tabla 5.5: Resultados de la resolución del POI para 10 nodos

Si bien tanto los vehículos como los operadores fueron limitados a un valor igual a 6, el promedio de operadores de 5,8 es bastante cercana a dicho límite, mientras que la cantidad promedio de rutas es de 3,7, por lo cual es bastante menor al límite. Esto se debe directamente al costo definido para los programas y, por consiguiente, de la función objetivo del problema de *pricing* de programas. Esto se explica dado que el costo de un programa que posee dos tareas programadas de forma consecutiva, sin tiempos de espera entre ambas, tiene el mismo costo que la suma de los costos de dos programas unitarios que tengan programados separadamente ambas tareas. Esto último no es válido para el cálculo en el costo de las rutas, ya que para cada ruta se tiene un costo asociado a “salir del origen” y otro costo asociado a “volver al origen”.

Finalmente, se muestra la solución entera de la instancia R104 en las Tablas 5.6 y 5.7. Este resultado es incluido para ejemplificar la distribución de nodos que siguen la mayoría de las soluciones enteras encontradas al término del algoritmo de generación columnas. Se puede notar que, debido al tipo de función objetivo propuesto, las rutas se encuentran mejor balanceadas que los programas. Esto quiere decir que los nodos se distribuyen de forma más uniforme en las rutas que en los programas, en estos últimos tan solo se tiene un programa con 5 tareas de forma consecutiva y los demás programas con una sola tarea.

Cost	Route
22,2	6
86,2	10, 7, 8, 5
63,2	2, 4
70,3	1, 9, 3

Tabla 5.6: Rutas para R104

Cost	Schedule
10	6
10	7
10	8
10	9
10	10
50	5, 2, 1, 4, 3

Tabla 5.7: Programas para R104

5.3.2. POI para instancias de 15 nodos

Al igual que en la sección precedente, la Tabla 5.8 presenta la misma información descrita para la Tabla 5.5. Es importante notar que tan solo para 17 de las 29 instancias de 15 nodos se pudo encontrar una cota inferior dentro del tiempo límite. Para los demás casos, dentro de las columnas *LB* y *Gap* se ingresó un guión (-) para denotar la falta de estos valores. Se recuerda al lector que los tiempos de ejecución se encuentran expresados en segundos.

De forma general, se puede notar que, al igual que para las instancias de 10 nodos, no se puede observar una relación directa entre la densidad de los grafos y el tiempo de ejecución. Sin embargo, dado que para 12 instancias el tiempo se encuentra truncado, el valor del tiempo de ejecución no es una medida totalmente fiable. Por otra parte, para las instancias C y RC los grafos de las instancias de distribución más densos coinciden con aquellas instancias en las que no se logró obtener una cota inferior.

Con respecto a la cantidad de rutas y programas de la solución entera, para el caso de las instancias de 15 nodos, solo 2 instancias fueron penalizadas por exceso de programas, a diferencia de las 4 instancias con exceso en las instancias de 10 nodos. Con respecto al promedio de rutas y programas utilizados, estos siguen la misma tendencia. La cantidad promedio de programas tiende a la cantidad límite de 10 operadores, en cambio la cantidad de rutas tiende a ser bastante menor al límite de 10 vehículos, para la mayoría de las instancias.

Instance	D. Density	P. Density	LB	UB	Gap	E. Time	Routes	Sched.	Cols. R.	Cols. S.	Iter.
R101	40,42 %	91,25 %	546,4	598,8	9,59 %	1.900,7	8	10	26205	117	6
R102	75,42 %	97,50 %	482,1	514,7	6,76 %	1.225,2	5	7	5895	297	8
R103	78,33 %	98,33 %	481,8	522,8	8,51 %	1.228,6	5	8	4371	300	8
R104	84,58 %	98,33 %	450,3	520,1	15,51 %	3.818,3	6	10	5010	380	11
R105	49,17 %	95,00 %	524,0	567,6	8,31 %	2.179,1	6	10	9985	182	6
R106	79,58 %	98,33 %	474,3	566,0	19,34 %	1.081,8	7	10	721	282	7
R107	84,17 %	99,17 %	-	494,0	-	3.763,7	5	9	10237	369	10
R108	89,58 %	99,17 %	436,1	502,6	15,26 %	1.265,5	10	10	18063	434	8
R109	65,42 %	100,00 %	478,3	500,0	4,53 %	3.279,1	5	8	7546	261	8
R110	83,75 %	100,00 %	443,8	493,4	11,17 %	1.877,6	5	9	4248	318	8
R111	82,92 %	99,17 %	461,1	520,0	12,78 %	2.428,3	6	9	9206	293	9
R112	100,00 %	100,00 %	430,5	487,1	13,15 %	2.463,2	5	10	758	232	7
AVG					11,35 %	2.209,2	6,1	9,2	8.520	289	8,0
C101	54,17 %	85,00 %	488,0	548,1	12,31 %	147,8	4	8	1107	545	11
C102	81,67 %	97,50 %	-	721,4	-	3.819,4	8	10	3960	844	11
C103	85,00 %	97,92 %	-	586,1	-	3.964,7	6	8	5588	862	10
C104	90,00 %	98,33 %	-	1.888,4	-	4.044,1	8	11	3443	1177	13
C105	57,08 %	88,75 %	480,8	545,9	13,54 %	2.627,3	4	10	3098	576	11
C106	55,42 %	85,83 %	488,0	611,2	25,24 %	327,1	5	8	3158	631	12
C107	59,17 %	90,83 %	-	575,9	-	3.851,5	5	10	4813	557	10
C108	66,25 %	95,42 %	-	609,7	-	4.035,5	6	9	4480	720	11
C109	75,83 %	98,33 %	-	609,5	-	3.660,5	6	10	5540	768	9
AVG					17,03 %	2.942,0	5,8	9,3	3.910	742	10,9
RC101	53,33 %	95,00 %	427,1	499,3	16,91 %	1.078,3	4	10	4659	204	10
RC102	77,50 %	98,33 %	-	495,7	-	3.973,6	4	8	6004	606	11
RC103	82,92 %	99,17 %	-	623,2	-	3.861,5	6	10	4764	509	12
RC104	87,50 %	99,17 %	-	480,9	-	4.103,2	4	10	6286	581	12
RC105	70,00 %	98,33 %	448,9	551,0	22,74 %	3.412,7	5	9	4056	373	11
RC106	68,75 %	100,00 %	385,2	473,4	22,89 %	3.835,8	4	10	18116	374	10
RC107	89,17 %	100,00 %	-	468,7	-	3.763,3	4	6	2620	472	10
RC108	97,92 %	100,00 %	-	1.981,1	-	3.904,5	8	11	2864	779	12
AVG					20,85 %	3.491,6	4,9	9,3	6.171	487	11,0
TOTAL					14,03 %	2.790,4	5,7	9,2	6.441,4	484,2	9,7

Tabla 5.8: Resultados de la resolución del POI para 15 nodos

Conclusión y trabajo futuro

A partir del problema de distribución de quimioterapias a domicilio, se plantea un modelo integrado de resolución mediante un método de *Branch-and-Price* que toma en cuenta tanto la producción de medicamentos como la realización de quimioterapias. Este modelo, requiere de la generación de dos tipos de columnas, un tipo de columna correspondiente a rutas de realización de quimioterapias y el otro tipo de columna a programas de producción de medicamentos. El Problema Maestro (*MP*) se encarga de encontrar las rutas y programas que cumplen con las restricciones de los tiempos de estabilidad de los medicamentos y con las restricciones de producción previa al inicio de la distribución del medicamento. Por consiguiente, el objetivo general planteado en la sección 1.3 de generar un algoritmo que permita resolver el POI en un contexto de quimioterapias a domicilio es cumplido. Los objetivos específicos descritos en la sección 1.4 son cumplidos con ciertas limitaciones que se describen a continuación.

Debido las restricciones de estabilidad y precedencia, el *MP* debe ser tratado en base a un modelo de *set-partitioning*, el cual considera restricciones de igualdad en vez de las restricciones de mayor ó igual de un modelo de *set-covering* para asegurar la cobertura de los pacientes. Esta diferencia se traduce en que una menor cantidad de conjuntos de rutas y programas sean parte de las soluciones factibles del POI dentro de un modelo de *set-partitioning*, por lo que no solo la resolución del *MP* entero toma más tiempo, sino que la diferencia entre las cotas inferior y superior de la solución tienden a ser mayores. Una formulación que sea más permisiva con respecto a la asociación de rutas y programas, ayudaría a mejorar los resultados de la solución entera, sin embargo esto debe ser realizado respetando las restricciones de estabilidad y de precedencia para cada una de las tareas asignadas y para cada uno de los pacientes visitados.

Con respecto a la eficiencia de los algoritmos de *pricing* que resuelven los problemas que generan las columnas, se puede notar que los tiempos de resolución del algoritmo de asignación de tareas es mucho mayor al del trazado de rutas, sobre todo para instancias más grandes. Debido a esta diferencia, el algoritmo de asignación de tareas resulta muy limitante, tanto para la generación de rutas como para la ejecución del algoritmo que resuelve el POI. Esto es debido, por un lado a la falta de una definición de dominación exacta entre las etiquetas, y por otro lado, a la densidad del grafo de producción. Dado que las ventanas de tiempo del problema de producción son bastante más holgadas que en el problema de distribución, se tiene una mayor cantidad de etiquetas generadas durante la ejecución del algoritmo de asignación de tareas en comparación con el de trazado de rutas. Debido a esto, es necesario explorar otros enfoques para la generación de programas para un operador. Si bien

en la literatura académica existen métodos de asignación de tareas para un solo operador, la función objetivo de este problema difiere considerablemente. Es necesario encontrar un algoritmo de secuenciamiento que minimice el tiempo del flujo total y adaptarlo integrando los valores duales.

Los valores del *Gap* para las instancias que fueron resueltas dentro del tiempo límite, son bastante altos. Para la mayoría de las instancias estos valores se encuentran por sobre el 10%. Considerando los resultados que consideran tan solo el problema de distribución que se incluyen en el anexo D, el valor esperado del *Gap* es de un 5% o menos. Esto se debe, por una parte a lo explicado en el primer párrafo de esta sección, ya que el problema entero, con respecto al problema relajado que encuentra la cota inferior, tiene restricciones mucho menos permisivas y una gran cantidad de conjuntos de rutas y de programas no se consideran factibles para el caso en que se visite el mismo nodo o se produzca el mismo fármaco dos veces. Si bien esto último en un principio puede resultar poco intuitivo al estar repitiendo una quimioterapia dentro del conjunto de solución, esto es posible de corregir rápidamente al descartar una de las repeticiones de algún programa o ruta, lo cual de todas formas aporta en la mejora de la solución, en este caso, en la disminución del tiempo total.

Si bien el método de resolución del POI no tiene un desempeño suficientemente bueno para ser utilizado en un caso real, sí puede ser una referencia para el cálculo de los costos que puede tener la implementación real de un sistema de quimioterapias a domicilio. Diversas modificaciones pueden realizarse al objetivo del modelo para, por ejemplo, encontrar la dotación de enfermeros y químico farmacéuticos mínima necesaria para la realización de quimioterapias a domicilio. Con respecto al problema de distribución se le pueden agregar tiempos de preparación entre una tarea y otra. Esto puede ser realizado ingresando estos tiempos como el costo de recorrer el arco que une dos nodos. Con respecto al problema de distribución, se puede extender a un modelo *Multi-Trip* en caso de que la dotación de enfermeros para la realización de quimioterapias sea muy limitada. Se recuerda que este caso no fue tomado en cuenta dada la complejidad que agrega a la resolución del problema de trazado de rutas.

Desde el punto de vista científico, el planteamiento matemático de los sub-problemas de trazado de rutas y de programación de tareas resulta bastante interesante de analizar. Con respecto al algoritmo de trazado de rutas, este corresponde a un problema donde no solo se tiene un tiempo de inicio variable, sino que también se tienen nodos para los cuales es preferible tener tiempo de espera dentro de la ruta y otros en los que es posible atrasar el tiempo de visita con el fin de atrasar el tiempo de inicio también. Para el problema de programación de tareas, se plantea un problema matemático que puede ser utilizado en otros contextos, en los que se tengan problemas de secuenciamiento en los que ciertos nodos deban ser ingresados a la secuencia de forma tardía, mientras que otros nodos se deban ingresar de forma temprana.

Siglas

BnB *Branch-and-Bound*. 14

BnP *Branch-and-Price*. 10, 14, 38, 40, 42, 47

ESPPRC *Elementary Shortest Path Problem with Resource Constraints*. 13–17, 24, 26, 29

HAD Hospitalización a Domicilio. 1–8, 10, 12

MCO Medicina, Cirugía y Obstetricia. 1, 4

MP *Master Problem*. 14–16, 24–29, 34, 38, 39, 42, 47

MTVRPTW *Multiple Trip Vehicle Routing Problem with Time Windows*. 10, 12

POI Problema Operacional Integrado. 4, 5, 7, 8, 10, 11, 29, 38, 41–43, 47, 48

VRP *Vehicle Routing Problem*. 9, 10

VRPTW *Vehicle Routing Problem with Time Windows*. 12, 13, 15, 20, 24

Glosario

Algoritmo voraz Estrategia de búsqueda por la cual se sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. 6

Costo reducido Cantidad en que debe cambiar el coeficiente de una variable en particular en la función objetivo para que en la solución óptima dicha variable tenga un valor positivo. 14, 16, 24–26, 58

Generación de columnas Técnica utilizada para resolver problemas de programación lineal que consiste en empezar con una cantidad reducida de variables de decisión para luego ir añadiendo de forma iterativa nuevas variables de decisión. 10, 11, 14, 23, 24, 27, 39, 42, 43

Heurística Técnica diseñada para resolver un problema más rápidamente cuando los métodos clásicos son demasiado lentos, o para encontrar una solución aproximada cuando los métodos clásicos fallan en encontrar una solución exacta. 8, 9, 25, 26, 58

Problema dual Corresponde a un problema construido de forma sistemática a partir de un problema denominado “primal”. Ambos problemas se encuentran estrechamente relacionados y poseen propiedades que ayudan a la resolución del problema planteado. 15, 22, 23, 28

Programación lineal Campo de la programación matemática dedicado a maximizar o minimizar (optimizar) una función lineal, denominada función objetivo, de tal forma que las variables de dicha función estén sujetas a una serie de restricciones expresadas mediante un sistema de ecuaciones o inecuaciones también lineales. 11

Relajación lineal Eliminación de todas las condiciones de valores enteros o binarios para las variables de decisión de un problema de programación lineal. 14, 16, 27

Valor dual Es la mejora del valor óptimo si se agrega una unidad adicional al lado derecho de una restricción en particular. 14, 16, 26, 27

Bibliografía

- [1] Situation de la Chimiothérapie des Cancers. Année 2014. Technical report, Institut National du Cancer, Boulogne-Billancourt, France, 2014.
- [2] Estadísticas Vitales. Technical report, Instituto Nacional de Estadísticas, Chile, 2016.
- [3] Les Cancers en France en 2017 L’Essentiel Des Faits et Chiffres. Technical report, Institut National du Cancer, Boulogne-Billancourt, France, 2017.
- [4] L’état de santé de la population en France. Rapport 2017. Technical report, Santé Publique France, Saint-Maurice, France, 2017.
- [5] Daouia Aiane, Adnen El-Amraoui, and Khaled Mesghouni. A new optimization approach for a home health care problem. *Proceedings of 2015 International Conference on Industrial Engineering and Systems Management, IEEE IESM 2015*, (October):285–290, 2016.
- [6] Lucy Boothroyd, Pascale Lehoux, and Québec (Province). Agence d’évaluation des technologies et des modes d’intervention en santé. *Home-based chemotherapy for cancer : issues for patients, caregivers, and the health care system*. Agence d’évaluation des technologies et des modes d’intervention en santé, 2004.
- [7] Mark Corbett, Morag Heirs, Micah Rose, Alison Smith, Lisa Stirk, Gerry Richardson, Daniel Stark, Daniel Swinson, Dawn Craig, and Alison Eastwood. The delivery of chemotherapy at home: an evidence synthesis. *Health Services and Delivery Research*, 3(14):1–182, 4 2015.
- [8] Nicole Crisp, Priscilla M. Koop, Karen King, Wendy Duggleby, and Kathleen F. Hunter. Chemotherapy at home: Keeping patients in their “natural habitat”. *Canadian Oncology Nursing Journal*, 24(2):89–94, 4 2014.
- [9] Nathalie De Jaegere, Mieke Defraeye, and Inneke Van Nieuwenhuyse. *The vehicle routing problem: state of the art classification and review*. PhD thesis, KU Luven, 2014.
- [10] Mário L. de Lemos and Linda Hamata. Stability issues of parenteral chemotherapy drugs. *Journal of Oncology Pharmacy Practice*, 2007.
- [11] Departamento de Manejo Integral del Cáncer y otros Tumores. Plan Nacional de Cáncer 2018-2028. Technical report, Departamento de Manejo Integral del Cáncer y otros Tumores, Chile, 2018.

- [12] Jacques Desrosiers and Marco Lübbecke. A Primer in Column Generation. In *Column Generation*, number May, pages 1–32. 2006.
- [13] Ş.Selçuk Erengüç, N.C. Simpson, and Asoo J. Vakharia. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 115(2):219–236, 6 1999.
- [14] Dominique Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4or*, 8(4):407–424, 2010.
- [15] Dominique Feillet, Pierre DEJAX, Michel Gendreau, and Cyrille Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216–229, 10 2004.
- [16] Florent Hernandez, Dominique Feillet, Rodolphe Giroudeau, and Olivier Naud. Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559, 2016.
- [17] Brian Kallehauge, Jesper Larsen, Oli B.G Madsen, and Marius M. Solomon. *Vehicle Routing Problem with Time Windows*, pages 67–98. Springer US, Boston, MA, 2005.
- [18] Y. Kergosien, M. Gendreau, and J.-C. Billaut. A benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. *European Journal of Operational Research*, 262(1):287 – 298, 2017.
- [19] Y. Kergosien, J. F. Tournamille, B. Laurence, and J. C. Billaut. Planning and tracking chemotherapy production for cancer treatment: A performing and integrated solution. *International Journal of Medical Informatics*, 80(9):655–662, 2011.
- [20] Karel Lenstra, Jan Rinnooy, and H. G. Alexander. Complexity of vehicle routing and scheduling. *Networks*, 11:221–227, 1981.
- [21] Marius Mihai. Solomon. Vehicle routing and scheduling with time window constraints: Models and algorithms. 01 1983.
- [22] François Vanderbeck and Laurence A. Wolsey. Reformulation and Decomposition of Integer Programs. In *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [23] Henrique Watanabe, Rafael Cirino, Willy Soler, and Maristela Oliveira. Solution Methods for the Integrated Production Routing Problem. *Revista de Sistemas de Informação da FSMA*, 19(June):52–60, 2017.

Anexo A

Soluciones del ejemplo en sección 1.2.1

Actividad	Hora	Duración
Arco E_{01}	8AM	1 hora
Tratamiento	9AM	2 horas
Arco E_{12}	11AM	0,5 hora
Tratamiento	11:30AM	2 horas
Arco E_{20}	1:30PM	1 hora
Arco E_{03}	2:30PM	1,5 hora
Tratamiento	4PM	2 horas
Total	6PM	10 horas

Tabla A.1: Solución 1: Enfermero

Actividad	Hora	Duración
Arco E_{01}	8AM	1 hora
Tratamiento	9AM	2 horas
Arco E_{10}	11AM	1 hora
Ruta E_{02}	12PM	1 hora
Tratamiento	12PM	2 horas
Arco E_{23}	2:30PM	1 hora
Tratamiento	4PM	2 horas
Total	6PM	10 horas

Tabla A.3: Solución 2: Enfermero

Actividad	Hora	Duración
Producción	6AM	2 horas
Producción	12PM	2 horas
Total	2PM	4 horas

Tabla A.2: Solución 1: Farmacéutico

Actividad	Hora	Duración
Producción	6AM	2 horas
Producción	10AM	2 horas
Total	12PM	4 horas

Tabla A.4: Solución 2: Farmacéutico

Anexo B

Notation Table

Notation	Definition
n	Number of patients
v_i	A patient
v_0	Depot
e_i	Beginning of the service date for patient v_i
f_i	Due date of the service date for patient v_i
σ_i	The time a vehicle must wait before serving patient v_i
j_i	A drug production job
st_i	The stability time of a product resulting from the realisation of a job
pt_i	The production time of a product assigned to a particular patient
rt_i	Release date for job j_i
dt_i	Due date for job j_i
r_k	A feasible route
c_k	The cost of a route
ω_k	The stating time of a specific route
a_{ik}	Parameter representing if a specific route visits a patient or not
u_{ik}	Time when a route visits a particular patient
p_l	A schedule for a pharmacist
ψ_l	The finishing time of a particular schedule
w_{il}	Time when a specific schedule finishes a particular job
Q	Vehicle capacity limit
U	The number of vehicles
W	The number of available farmacist
M_Ω	A large number to penalise the excess of routes
M_Ψ	A large number to penalise the excess of schedules

Notation	Definition
d_i	Patient's demand
s_i	Time when a patient is served
$dist_{i,j}$	Euclidian distance between two nodes
$serv_i$	Service time
λ	A label corresponding to a vehicle route
L	Number of resources
l	A resource
e_l	A resource consumption
V^k	Whether a node is unreachable or not
u	The number of unreachable nodes
q	Consumption of the demand resource
V	Set of patients
J	Set of drug production jobs
Ω	Set of feasible routes
Ψ	Set of feasible schedules
I	Set of index
x_k	Decision variable to choose whether route r_k is used or not
y_l	Decision variable to choose whether schedule ψ_l is used or not
γ	Decision variable considering the excess of vehicle routes
η	Decision variable considering the excess of pharmacists
α_i	Route node covering dual variable
β_i	Drug production covering dual variable
δ_i	Stability dual variable
ε_i	Precedence dual variable
g	Vehicle float dual variable
h	Available pharmacists dual variable

Tabla B.1: Notations

Anexo C

Algorithms

C.1. VRPTW Initial Columns

Data: a graph $\mathcal{G} = (V, E)$ and an origin v_p .

Result: set of feasible routes covering all the nodes of the given graph.

Initialisation;

routeList $\leftarrow \emptyset$;

visitedNodes $\leftarrow \emptyset$;

while length(visitedNodes) < length(V) **do**

 route \leftarrow initialiseRoute(v_p);

 current $\leftarrow v_p$;

while length(route.unreachableNodes) < length(V) **do**

 successors \leftarrow successorList (current);

 sort(successors);

 next \leftarrow getFirstReachable(route, successors);

 route \leftarrow extend(route, next);

 visitedNodes.add(next);

 current \leftarrow next;

end

 routeList.add(route);

end

Algorithm 1: Initialisation of the column generation method for solving a VRPTW

Anexo D

Resultados de la implementación del etiquetado de rutas

D.0.1. Resolución del sub-problema

Un código en Java basado en el enfoque de etiquetado de rutas descrito en 4.1.1 fue desarrollado y comparado con los tiempos de resolución del modelo de programación lineal mixto descrito en 3.1.1. El *solver* utilizado para dichas pruebas corresponde a CPLEX. Para simular la resolución de un problema de *pricing*, se generaron números entre 0 y 20 de forma aleatoria los cuales se restaron a los costos sobre los arcos de cada instancia y sus costos fueron truncados a una décima, tal como se describe en la experiencia de D. Feillet [15]. Esto último con el objetivo de poder comparar los resultados con dicho trabajo. Diferentes pruebas fueron realizadas considerando sólo los 25 primeros puntos de cada instancia, luego los primeros 50 puntos para finalmente probar las instancias completas. El tiempo de resolución fue limitado a 600 segundos.

Type	MIP Model [ms]	Labelling [ms]
R	140.844	700
C	117.163	28.866
RC	146.235	144
AVG	133.427,63	10.962,74

Tabla D.1: Tiempo de resolución promedio para el ESPPRC

Como se puede observar en la Tabla D.1 el algoritmo de etiquetado es considerablemente más rápido que la resolución mediante un modelo de programación lineal mixto. Esto último, si bien era lo esperado, la principal razón para la realización de estas pruebas es para comprobar que la solución entregada por el algoritmo de etiquetado es el óptimo (o cercano, en caso de cumplirse el tiempo límite) entregado por CPLEX.

D.0.2. Resolución del MP relajado

Continuando con el siguiente paso para acercarse a la resolución del POI, es necesario resolver el problema de distribución. De esta forma, el método de generación de columnas fue implementado siguiendo la metodología presentada en el capítulo 4. Al final de cada iteración, todas las etiquetas asociadas al nodo terminal v_t de costo reducido negativo son ingresadas al conjunto de rutas factibles Ω .

En un principio se compararon los resultados obtenidos para el *MP* relajado utilizando el método exacto del algoritmo de etiquetado en contraste con el método heurístico limitado a 100 etiquetas descrito en 4.1.1. Un resumen de estos resultados se encuentra en la tabla D.2.

		Exact method			100 labels limit		
Type	Total	Time 1[ms]	Iter. 1	Solved 1	Time 2[ms]	Iter. 2	Solved 2
R-25	12	400	19,83	12	454	20,67	12
C-25	9	9.220	38,22	9	2.540	47,11	9
RC-25	8	3.171	27,86	8	1.287	32,10	8
AVG	29	3.137	28,64	29	1.021	33,29	29
R-50	12	16.392	33,83	12	6.524	38,25	12
C-50	9	21.194	79,88	8	12.345	98,13	8
RC-50	8	5.741	52,75	8	6.618	57,88	8
AVG	29	13.286	55,49	28	6.560	64,75	28
R-100	12	37,615	53.6	7	38.092	72	9
C-100	9	38,511	150.9	7	68.371	184,67	8
RC-100	8	105,943	63.4	7	112,007	94.1	7
AVG	29	30.788	89,29	21	41.742	116,94	24
Total	87	15.737	58	78	16.441	72	81

Tabla D.2: Tiempo de resolución promedio para el VRPTW relajado

A partir de los resultados obtenidos de la resolución mediante el método exacto y el método heurístico, se puede observar que limitando la cantidad de etiquetas, dentro del tiempo límite de 600 segundos se pueden resolver 3 instancias más que en su versión exacta. Con respecto a los tiempos de resolución, el método exacto es más lento para las instancias de 25 y 50 nodos, sin embargo toma menos tiempo para las instancias de 100 nodos. Esto último puede explicarse dado que la versión heurística del algoritmo realiza en promedio más iteraciones hasta encontrar la solución óptima del *MP* relajado. Al observar los resultados resumidos se puede observar que, en promedio, ambas versiones sólo tienen una diferencia del orden de 1 segundo.

D.0.3. Resolución del MP entero

Para la resolución del problema completo se limitó el tiempo de resolución de la generación de columnas a 600 segundos, si esta última termina dentro del tiempo estipulado, se obtiene una cota inferior del óptimo del problema. La flota de vehículos es 25 para todas las instancias,

esto simplemente corresponde a una cota suficientemente grande para encontrar el costo mínimo sin restringir la cantidad de rutas en un principio. Con respecto a la resolución del problema entero este es resuelto por CPLEX y su resolución se limita a 90 segundos. La solución obtenida de este último problema, haya o no terminado dentro del tiempo límite, entrega una cota superior de la solución. Los resultados resumidos se encuentran en la tabla D.3.

Nodes	Found LB	Best UP	Gap	Rel. Gap	Time [ms]	Cols.	Iter.	Sol. Nodes
25	29	29	0,009	0	307	748	32	25
50	28	29	0,048	0	24.909	1.948	56	50
100	25	23	0,033	0,008	151.606	5.423	118	103
Total	82	81	0,030	0,003	58.941			

Tabla D.3: Tiempo de resolución promedio para el VRPTW

La columna *Found LB* corresponde a la cantidad de instancias en donde el algoritmo de generación de columnas pudo terminar dentro de 600 segundos, la columna *Best UP* corresponde a la cantidad de instancias donde CPLEX pudo terminar la resolución del modelo de *set-covering* dentro de 90 segundos, la columna *Gap* representa a la diferencia entre la cota superior y la cota inferior como porcentaje de la cota inferior de la generación de columnas, la columna *Rel. Gap* representa la misma diferencia anterior pero esta vez de la resolución entera del modelo de programación lineal. Finalmente, la columna *Sol. Nodes* corresponde al promedio de la cantidad de nodos visitados por la solución entera que entrega la cota superior.

Anexo E

Proofs

E.1. Modified algorithm start time proof

Let us consider, for example, a label λ_k^* corresponding to a route from the origin v_0 to node k . To simplify indices, let us assume the route is a succession of enumerated nodes starting from 0 to k . The cost of the route is given by the following expression.

$$c_k^* = \sum_{i=1}^k c_{i-1,i} - \sum_{i=1}^k \alpha_i + \sum_{i=1}^k s_i^* \mu_i - s_0^* \sum_{i=1}^k \varepsilon_i. \quad (\text{E.1})$$

Due to positive factors μ_i , service times of the minimum cost of route λ_k^* are given by the following recursive expression:

$$s_i^* = \text{máx}\{s_{i-1}^* + \text{serv}_{i-1} + \text{dist}_{i-1,i}, a_i\}. \quad (\text{E.2})$$

Therefore, only the value of s_0^* is needed to calculate the visit times s_i^* that minimise the cost of label λ_k^* . To find the minimum cost starting time s_0^* , some properties and definitions should be explained first.

Property 1. s_i^* depends on s_0^* if and only if $s_{j-1}^* + \text{serv}_{j-1} + \text{dist}_{j-1,j} \geq a_j \quad \forall j \leq i$.

DEMOSTRACIÓN. From equation E.2 it is easy to notice that, s_i^* depends on the last visited node service time if $s_{i-1}^* + \text{serv}_{i-1} + \text{dist}_{i-1,i} \geq a_i$. Inductively, for each service time s_i^* this statement needs to be true until $i = 1$, where $s_1^* = s_0^* + \text{dist}_{0,1}$. \square

Therefore, the notion of **breaking node** can be defined as follows.

Definición E.1 For a label λ_k^* , the first node $l \in \{1, \dots, k\}$ in the succession of visited nodes whose service time s_l^* is equal to its time window lower bound a_l is called **breaking node**.

Without loss of generality, service times for all $i < l$ are equal to the following expression.

$$s_i^* = s_0^* + \sum_{j=1}^i dist_{j-1,j} + serv_{j-1}$$

To simplify notation, the following terms are defined. Let φ_k^* be the **fixed cost** given by:

$$\varphi_k^* = \sum_{i=1}^k c_{i-1,i} - \sum_{i=1}^k \alpha_i.$$

Let τ_i^* be the **strict time** of a partial path from 0 to i for all $i \in \{1, \dots, k\}$ given by:

$$\tau_i^* = \sum_{j=1}^i serv_{j-1} + dist_{j-1,j}.$$

The minimum cost of the route may be written in terms of a breaking node l , the fixed cost φ_k^* and strict times τ_i^* .

$$c_k^* = \varphi_k^* + \sum_{i=l}^k s_i^* \mu_i + \sum_{i=1}^{l-1} \mu_i (\tau_i^* + s_0^*) - s_0^* \sum_{i=1}^k \varepsilon_i. \quad (\text{E.3})$$

Moreover, from equation E.2, it can easily be proven that, the minimum cost of the route has a lower bound.

$$\begin{aligned} c_k^* &\geq \varphi_k^* + \sum_{i=1}^k \mu_i (\tau_i^* + s_0^*) - s_0^* \sum_{i=1}^k \varepsilon_i, \\ c_k^* &\geq \varphi_k^* + \sum_{i=1}^k \mu_i \tau_i^* + s_0^* \sum_{i=1}^k (\mu_i - \varepsilon_i). \end{aligned} \quad (\text{E.4})$$

Also from equation E.2, the notion of **waiting time** can be defined.

Definición E.2 Let σ_i^* be the waiting time for a node i .

$$\sigma_i^* = \begin{cases} a_i - (s_{i-1}^* + serv_{i-1} + dist_{i-1,i}) & \text{if } a_i \geq s_{i-1}^* + serv_{i-1} + dist_{i-1,i} \\ 0 & \text{otherwise.} \end{cases}$$

This expression comes from the fact that if the vehicle visits a node i before its time window start time a_i , then the vehicle will have to wait, hence the name.

Due to the recursive nature of waiting time, service times for each node may be written in terms of waiting times.

$$s_i^* = s_0^* + \tau_i^* + \sum_{j=1}^i \sigma_j^* \quad (\text{E.5})$$

DEMOSTRACIÓN. The service time for a node i will be at least the minimum time the vehicle can get to that particular node $s_0^* + \tau_i^*$. In addition to the last, the time the vehicle has already waited must be added to get the exact service time. \square

Replacing equation E.5 in equation E.3.

$$\begin{aligned} c_k^* &= \varphi_k^* + \sum_{i=l}^k \mu_i (s_0^* + \tau_i^* + \sum_{j=1}^i \sigma_j^*) + \sum_{i=1}^{l-1} \mu_i (\tau_i^* + s_0^*) - s_0^* \sum_{i=1}^k \varepsilon_i, \\ c_k^* &= \varphi_k^* + \sum_{i=l}^k \mu_i \sum_{j=1}^i \sigma_j^* + \sum_{i=1}^k \mu_i (\tau_i^* + s_0^*) - s_0^* \sum_{i=1}^k \varepsilon_i, \\ c_k^* &= \varphi_k^* + \sum_{i=1}^k \mu_i \tau_i^* + \sum_{i=l}^k \mu_i \sum_{j=1}^i \sigma_j^* + s_0^* \sum_{i=1}^k (\mu_i - \varepsilon_i). \end{aligned} \quad (\text{E.6})$$

Notice that σ_j^* and the breaking node l depend on the value of s_0^* . A higher value of s_0^* might cause a higher breaking node and it will mean a lower value for the set of σ_j^* . From the first property, it can be observed that $\sigma_j^* = 0 \forall j < l$. This comes from the definition of breaking node.

Waiting times σ_i^* may be written as a recursive function depending on s_0 (which is a variable, but s_0^* is the optimal value) taking into account equation E.5.

$$\sigma_i^*(s_0) = \begin{cases} a_i - s_0 - \tau_i^* - \sum_{j=1}^{i-1} \sigma_j^* & \text{if } a_i \geq s_0 + \tau_i^* + \sum_{j=1}^{i-1} \sigma_j^* \\ 0 & \text{otherwise.} \end{cases} \quad (\text{E.7})$$

Let us study the term $\Gamma^*(s_0) = \sum_{i=l}^k \mu_i \sum_{j=1}^i \sigma_j^*(s_0)$. Firstly, only waiting times from l and higher may be taken into account, $\Gamma^*(s_0) = \sum_{i=l}^k \mu_i \sum_{j=l}^i \sigma_j^*(s_0)$. Moreover, Γ^* is a piecewise continuous function depending on s_0 .

$$\Gamma^*(s_0) = \begin{cases} \sum_{i=1}^k \mu_i \sum_{j=1}^i \sigma_j^*(s_0) & \text{if } 0 \leq s_0 \leq a_1 - \tau_1^* \\ \sum_{i=2}^k \mu_i \sum_{j=2}^i \sigma_j^*(s_0) & \text{if } a_1 - \tau_1^* \leq s_0 \leq a_2 - \tau_2^* \\ \vdots & \\ \sum_{i=l}^k \mu_i \sum_{j=l}^i \sigma_j^*(s_0) & \text{if } a_{l-1} - \tau_{l-1}^* \leq s_0 \leq a_l - \tau_l^* \\ \vdots & \\ \mu_k \sigma_k^*(s_0) & \text{if } a_{k-1} - \tau_{k-1}^* \leq s_0 \leq a_k - \tau_k^* \\ 0 & \text{otherwise.} \end{cases} \quad (\text{E.8})$$

It is important to observe not all intervals defined in the $\Gamma^*(s_0)$ function are actually valid, since the starting value of a time window might be lower than the value of its precedent node.

Nevertheless, they do not influence the calculation of the values of the function. Indeed, if an interval is said not to be valid, this means $a_{l-1} - \tau_{l-1}^* > a_l - \tau_l^*$, then its waiting time σ_l^* will always be equal to 0, therefore this interval can be pulled out from the definition of the function and the lower bound of the next interval should be replaced by the precedent upper bound value resulting in the following interval $a_{l-1} - \tau_{l-1}^* \leq s_0 \leq a_{l+1} - \tau_{l+1}^*$. If the interval is still not valid, the same action must be repeated with the following node until the last node, from which the function will only be equal to 0.

Introducing $\Gamma^*(s_0)$ function into equation E.6,

$$c_k^*(s_0) = \varphi_k^* + \sum_{i=1}^k \mu_i \tau_i^* + \Gamma^*(s_0) + s_0 \sum_{i=1}^k (\mu_i - \varepsilon_i). \quad (\text{E.9})$$

Equation E.9 allows the definition of a marginal cost from a change in the value s_0 . Notice the dependence on s_0 has a negative factor in $\Gamma^*(s_0)$.

$$c_k^{*'} = \begin{cases} \sum_{i=1}^k (\mu_i - \varepsilon_i) - \sum_{i=1}^k \mu_i & \text{if } 0 \leq s_0 \leq a_1 - \tau_1^* \\ \vdots & \\ \sum_{i=1}^k (\mu_i - \varepsilon_i) - \sum_{i=l}^k \mu_i & \text{if } a_{l-1} - \tau_{l-1}^* \leq s_0 \leq a_l - \tau_l^* \\ \vdots & \\ \sum_{i=1}^k (\mu_i - \varepsilon_i) - \mu_k & \text{if } a_{k-1} - \tau_{k-1}^* \leq s_0 \leq a_k - \tau_k^* \\ \sum_{i=1}^k (\mu_i - \varepsilon_i) & \text{otherwise.} \end{cases} \quad (\text{E.10})$$

The explanation of the marginal cost is that for every unit start time s_0 grows, the cost will be reduced from $\sum_{i=1}^k \varepsilon_i$, however, this grow in s_0 will have the same effect for each of the nodes having direct dependence on this value. These dependent nodes, as it have been proven, are all nodes visited before the breaking node, hence the cost grows at a rate $\sum_{i=1}^{l-1} \mu_i = \sum_{i=1}^k \mu_i - \sum_{i=l}^k \mu_i$. For a better understanding, the expression at the right is prioritised over the compact expression at the left.

From the marginal cost E.10, the notion of **dual difference** Λ^* is defined.

$$\Lambda^* = \sum_{i=1}^k (\mu_i - \varepsilon_i) \quad (\text{E.11})$$

Since time windows for each node have maximum times, the starting time of the route s_0^* is bounded by the following constraints.

$$s_0^* \leq b_i - \tau_i^*, \quad (\forall i \in \{1, \dots, k\}). \quad (\text{E.12})$$

To simplify notation, the following values are defined:

- $\bar{A}_i^* = \max_{j=1}^i a_j - \tau_j^*$

- $B^* = \min_{i=1}^k b_i - \tau_i^*$

Having defined all the concepts needed to find the optimal value of the starting time s_0^* , in the following, it is proven the optimal value.

$$s_0^* = \begin{cases} \min\{\bar{A}^*_l, B^*\} & \text{if there is a breaking node } l > 0 \\ B^* & \text{otherwise.} \end{cases}$$

DEMOSTRACIÓN. Since the cost c_k^* of the route, as it is shown in equation E.9, is a piecewise linear function depending on the value of the starting time. As a linear function, the optimal value of the starting time will be given by the extreme values s_0 can have. This means the value of s_0 will be the lowest possible for a marginal cost $c_k^{*'} > 0$, on the contrary, s_0 will be the highest possible for $c_k^{*'} \leq 0$.

Let us observe $c_k^{*'}(s_0)$ is, in fact, an increasing function. Indeed the term Λ^* is fixed for all intervals of the marginal cost and only the term $\sum_{i=l}^k \mu_i$ depending on the interval where s_0 is. Let us have two different start time values $s_0^a \geq 0$ and $s_0^b \geq 0$, such that $s_0^a < s_0^b$. The value of the marginal cost for each start time are equal to $c_k^{*'}(s_0^a) = \Lambda^* - \sum_{i=l(s_0^a)}^k \mu_i$ and $c_k^{*'}(s_0^b) = \Lambda^* - \sum_{i=l(s_0^b)}^k \mu_i$ respectively. Since $s_0^a < s_0^b$, by definition of the marginal cost, $l(s_0^a) \leq l(s_0^b)$. Considering the values μ_i are all equal to 0 or positive, $\sum_{i=l(s_0^a)}^k \mu_i \geq \sum_{i=l(s_0^b)}^k \mu_i$. Therefore, $c_k^{*'}(s_0^a) \leq c_k^{*'}(s_0^b)$.

Property 2. If $l \in \{1, \dots, k\}$ is such that $\Lambda^* > \sum_{j=l}^k \mu_j$ then,

$$\Lambda^* > \sum_{j=i}^k \mu_j \quad (\forall i \geq l)$$

Property 3. If $l \in \{1, \dots, k\}$ is such that $\Lambda^* \leq \sum_{j=l}^k \mu_j$ then,

$$\Lambda^* \leq \sum_{j=i}^k \mu_j \quad (\forall i \leq l)$$

Both properties, 2 and 3, are trivial from the fact that $\mu_j \geq 0$. These properties are helpful to see that only the first node in the succession of route λ_k^* , such that $\Lambda^* > \sum_{j=l}^k \mu_j$ must be taken into account to find the minimal value of c_k^* .

Noticing that $\forall s_0 \leq a_1 - \tau_1^*$, $c_k^{*'}(s_0) \leq 0$, if there exists such node $l \in \{2, \dots, k\}$, where $\Lambda^* \leq \sum_{j=l-1}^k \mu_j$ and $\Lambda^* > \sum_{j=l}^k \mu_j$, then the value of s_0 minimising the cost c_k^* is the maximum possible value for the interval $[a_{l-2} - \tau_{l-2}^*, a_{l-1} - \tau_{l-1}^*]$, which actually is the lowest possible value for the interval $[a_{l-1} - \tau_{l-1}^*, a_l - \tau_l^*]$. As it has been explained before in the definition of Γ^* function, for the intervals to be valid, some of them are pulled out from the definition. To avoid problems with this definition, then the value of s_0^* can be defined by the terms of the lower bound of the interval where l is assigned \bar{A}^*_{l-1} . Thus, $s_0^* = \bar{A}^*_{l-1}$, which is consistent with the definition of Γ^* and the marginal cost.

If there is no visited node satisfying the previous statement, then s_0^* is higher than the last defined interval lower bound, which is \bar{A}^*_k . Since the marginal cost in this range of values is equal to Λ^* , then s_0^* will take the lowest possible value if $\Lambda^* > 0$, which is the lower bound of the interval. On the contrary, it will take the highest possible value if $\Lambda^* \leq 0$, which is equal to B^* . Since the value s_0^* is always bounded by the value of B^* , it must be taken into account in the definition of the optimal value.

□

Anexo F

IOP routing pricing problem results

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Gap	Feasible	Generated
R101	34,46 %	26,79 %	-127,22	67	-127,22	58	0,00 %	43	166
R102	61,69 %	31,92 %	-194,35	22	-194,35	14	0,00 %	81	682
R103	76,31 %	32,46 %	-200,04	43	-200,04	26	0,00 %	100	1517
R104	83,85 %	33,03 %	-328,55	67	-328,55	50	0,00 %	76	2677
R105	45,54 %	28,72 %	-139,62	1	-139,62	0	0,00 %	46	227
R106	69,23 %	32,22 %	-218,41	9	-218,41	8	0,00 %	128	1051
R107	82,15 %	32,02 %	-227,78	21	-227,78	21	0,00 %	130	1836
R108	88,62 %	32,64 %	-328,55	63	-328,55	45	0,00 %	77	3113
R109	62,77 %	28,68 %	-155,36	2	-155,36	2	0,00 %	87	487
R110	80,62 %	30,53 %	-181,77	6	-181,77	7	0,00 %	100	902
R111	79,54 %	32,30 %	-256,91	10	-256,91	12	0,00 %	117	1310
R112	99,38 %	30,80 %	-243,92	20	-243,92	20	0,00 %	143	1730
AVG	72,01 %	31,01 %	-216,87	27,6		21,9	0,00 %	94,0	1.308,2
C101	49,23 %	59,06 %	-1.302,37	1	-1.302,37	1	0,00 %	111	518
C102	69,54 %	52,21 %	-1.416,33	120	-1.416,33	26	0,00 %	369	5936
C103	84,00 %	51,47 %	-1.631,20	4.534	-1.631,20	72	0,00 %	855	39488
C104	91,69 %	49,50 %	-2.275,46	155.901	-2.275,46	117	0,00 %	1229	175659
C105	51,54 %	59,10 %	-1.382,53	2	-1.382,53	2	0,00 %	166	806
C106	49,69 %	59,44 %	-1.425,28	1	-1.425,28	1	0,00 %	99	510
C107	53,54 %	59,20 %	-1.485,63	2	-1.485,63	1	0,00 %	122	642
C108	65,85 %	55,84 %	-1.651,49	12	-1.651,49	8	0,00 %	296	2005
C109	74,15 %	54,15 %	-1.728,18	69	-1.728,18	19	0,00 %	512	4435
AVG	65,47 %	55,55 %	-1.588,72	17.849,1		27,4	0,00 %	417,7	25.555,4
RC101	41,85 %	27,57 %	-67,20	0	-67,20	0	0,00 %	35	239
RC102	60,92 %	29,29 %	-133,95	2	-133,95	2	0,00 %	60	653
RC103	78,00 %	28,80 %	-133,95	7	-133,95	9	0,00 %	74	1207
RC104	84,77 %	28,13 %	-267,64	17	-267,64	14	0,00 %	36	1758
RC105	55,69 %	30,39 %	-129,88	1	-129,88	1	0,00 %	61	505
RC106	58,62 %	28,61 %	-90,52	1	-90,52	1	0,00 %	48	444
RC107	77,38 %	29,82 %	-115,56	4	-115,56	5	0,00 %	69	837
RC108	94,31 %	28,38 %	-192,99	12	-192,99	10	0,00 %	55	1457
AVG	68,94 %	28,87 %	-141,46	5,5		5,3	0,00 %	54,8	887,5
TOTAL				5.552,31		19,03	0,00 %	183,6	8.717,1

Tabla F.1: Resolution times for the routing IOP pricing problem with 25 clients.

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Gap	Feasible	Generated
R101	32,16 %	23,90 %	-222,71	59	-222,71	64	0,00 %	152	722
R102	55,53 %	27,05 %	-449,98	376	-449,98	139	0,00 %	643	11870
R103	71,06 %	27,76 %	-449,98	1.439	-449,98	201	0,00 %	695	25855
R104	89,45 %	26,35 %	-528,77	10.204	-528,77	468	0,00 %	653	75224
R105	40,71 %	24,08 %	-266,88	5	-266,88	4	0,00 %	231	1248
R106	62,00 %	27,20 %	-449,98	489	-449,98	78	0,00 %	722	15994
R107	75,76 %	27,48 %	-458,46	1.648	-458,46	214	0,00 %	703	31182
R108	92,12 %	26,35 %	-528,77	14.720	-524,00	511	0,90 %	639	80558
R109	58,67 %	24,00 %	-312,24	26	-312,24	17	0,00 %	399	2870
R110	78,16 %	24,28 %	-510,33	79	-510,33	46	0,00 %	450	5524
R111	78,20 %	26,83 %	-458,46	1.019	-458,46	158	0,00 %	745	20109
R112	99,10 %	25,01 %	-518,50	367	-518,50	160	0,00 %	491	11069
AVG	0,7	0,3		2.535,9		171,7	0,08 %	543,6	23518,8
C101	46,43 %	37,84 %	-2.485,94	32	-2.485,94	14	0,00 %	457	2783
C102	65,45 %	35,53 %	-4.421,78	60.248	-4.282,75	106	3,14 %	2456	127871
C103	81,65 %	36,31 %	-4.257,79	329.036	-4.515,35	379	-6,05 %	2898	339296
C104	95,76 %	34,44 %	-4.217,63	309.761	-4.894,12	1316	-16,04 %	1672	583765
C105	49,73 %	38,33 %	-3.041,33	75	-3.041,33	28	0,00 %	778	5128
C106	47,80 %	37,82 %	-2.994,64	71	-2.994,64	26	0,00 %	613	3942
C107	53,10 %	38,04 %	-3.243,66	158	-3.243,66	39	0,00 %	1123	7258
C108	63,88 %	36,59 %	-3.305,92	422	-3.215,09	59	2,75 %	1441	11593
C109	74,31 %	35,57 %	-3.710,58	1.652	-3.408,30	124	8,15 %	1872	23203
AVG	0,6	0,4		8.951,1		56,6	2,01 %	1248,6	25968,3
RC101	31,61 %	19,98 %	-167,01	3	-167,01	2	0,00 %	154	999
RC102	49,80 %	18,98 %	-349,00	135	-349,00	31	0,00 %	623	6231
RC103	66,63 %	16,89 %	-349,00	352	-349,00	67	0,00 %	625	11107
RC104	86,67 %	14,12 %	-349,00	1.708	-349,00	172	0,00 %	813	24542
RC105	46,08 %	19,74 %	-287,00	29	-287,00	16	0,00 %	499	3346
RC106	46,31 %	19,56 %	-190,83	15	-190,83	11	0,00 %	304	2501
RC107	69,80 %	16,12 %	-279,42	122	-279,42	51	0,00 %	628	7331
RC108	89,14 %	14,56 %	-316,79	373	-316,79	109	0,00 %	742	13221
AVG	60,75 %	17,49 %		342,1		57,4	0,00 %	548,5	8.659,8
TOTAL				3.549,1		108,0	0,55 %	727,8	19.751,1

Tabla F.2: Resolution times for the routing IOP pricing problem with 50 clients.

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Gap	Feasible	Generated
R101	32,09 %	15,55 %	-378,15	123	-378,15	133	0,00 %	359	3073
R102	54,86 %	19,00 %	-561,88	437035	-551,84	673	1,79 %	3955	438596
R103	71,32 %	19,49 %	-551,58	301672	-551,58	1.490	0,00 %	2417	413330
R104	86,84 %	18,94 %	-476,57	301533	-580,72	3.007	-21,85 %	3500	520929
R105	41,38 %	15,58 %	-476,98	180	-476,98	84	0,00 %	928	8966
R106	61,78 %	18,77 %	-579,39	304480	-578,92	457	0,08 %	3377	386766
R107	76,54 %	19,39 %	-548,13	307017	-578,92	882	-5,62 %	2263	482819
R108	89,93 %	18,91 %	-527,52	300619	-602,28	1.512	-14,17 %	2479	541677
R109	58,77 %	17,50 %	-535,64	3851	-535,64	250	0,00 %	2704	35738
R110	78,03 %	18,61 %	-558,19	27346	-558,19	558	0,00 %	4294	112831
R111	77,62 %	19,09 %	-578,92	302358	-578,92	615	0,00 %	3276	450684
R112	99,08 %	18,90 %	-614,40	240662	-575,92	2.436	6,26 %	7524	399577
AVG	61,87 %	17,23 %		54432,4		692,2	1,25 %	3161,8	112.037
C101	43,49 %	17,58 %	-2.577,66	310	-2.577,66	102	0,00 %	1375	11779
C102	62,90 %	17,17 %	-3.224,42	300570	-3.772,12	946	-16,99 %	5156	405325
C103	80,37 %	17,36 %	-2.892,81	300040	-4.154,66	3.074	-43,62 %	3610	546832
C104	92,72 %	16,72 %	-3.214,72	459177	-4.438,07	6.944	-38,05 %	1983	961714
C105	47,59 %	17,50 %	-2.681,43	997	-2.681,43	181	0,00 %	2211	19705
C106	51,42 %	16,45 %	-2.631,83	1374	-2.631,83	288	0,00 %	2218	23979
C107	51,96 %	17,23 %	-2.862,58	1521	-2.862,58	310	0,00 %	2515	25247
C108	61,82 %	16,99 %	-2.962,85	7974	-2.872,10	436	3,06 %	4741	53963
C109	73,80 %	16,31 %	-3.280,98	37191	-3.280,98	930	0,00 %	5518	110650
AVG	55,01 %	17,01 %		8.227,8		374,5	0,51 %	3.096,3	40.887
RC101	35,26 %	11,68 %	-353,11	82	-353,11	65	0,00 %	637	6563
RC102	54,13 %	12,99 %	-474,53	9961	-474,53	391	0,00 %	2170	74801
RC103	70,81 %	12,56 %	-495,97	82090	-495,97	1.019	0,00 %	2511	252902
RC104	86,81 %	11,67 %	-500,10	301948	-501,66	2.929	-0,31 %	1190	526587
RC105	50,14 %	12,84 %	-448,10	1057	-448,10	290	0,00 %	1783	23620
RC106	51,80 %	12,86 %	-435,04	688	-435,04	251	0,00 %	1324	19007
RC107	73,08 %	12,87 %	-448,38	4398	-448,38	613	0,00 %	1937	47876
RC108	91,15 %	12,42 %	-477,36	14913	-443,12	1.501	7,17 %	2443	93280
AVG	60,91 %	12,60 %		16.169,9		590	1,02 %	1.829,3	74.007
TOTAL	59,89 %	15,23 %		25.513,00		568,06	0,97 %	2.645,94	76.697,18

Tabla F.3: Resolution times for the routing IOP pricing problem with 100 clients.

Anexo G

IOP scheduling pricing problem results

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Cost Gap	Feasible	Generated
R101	93,64 %	55,34 %	-1.238,87	431	-1.238,87	166	0,00 %	459	8523
R102	100,00 %	54,55 %	-1.440,66	22.778	-1.440,66	135	0,00 %	1000	75194
R103	100,00 %	54,55 %	-1.440,66	29.420	-1.440,66	108	0,00 %	1000	75194
R104	100,00 %	54,55 %	-1.487,89	20.494	-1.487,89	136	0,00 %	1283	69183
R105	97,27 %	54,21 %	-1.275,64	1.837	-1.275,64	57	0,00 %	539	18687
R106	100,00 %	54,55 %	-1.456,75	27.236	-1.456,75	128	0,00 %	1159	73694
R107	100,00 %	54,55 %	-1.456,75	27.008	-1.456,75	128	0,00 %	1159	73694
R108	100,00 %	54,55 %	-1.495,10	26.458	-1.495,10	88	0,00 %	1382	68560
R109	100,00 %	54,55 %	-1.333,51	15.163	-1.332,93	150	0,04 %	900	48214
R110	100,00 %	54,55 %	-1.339,77	17.475	-1.324,57	122	1,13 %	1104	48806
R111	100,00 %	54,55 %	-1.446,61	23.624	-1.446,61	123	0,00 %	1132	68771
R112	100,00 %	54,55 %	-1.432,64	28.091	-1.432,64	107	0,00 %	1159	65124
AVG				20.001		121	0,10 %	1.023	57.804
C101	82,73 %	49,45 %	-4.441,47	21	-4.441,47	14	0,00 %	272	2123
C102	99,09 %	55,05 %	-5.963,32	38.314	-5.963,32	75	0,00 %	1911	87100
C103	99,09 %	55,05 %	-5.963,32	37.586	-5.963,32	89	0,00 %	1911	87100
C104	99,09 %	55,05 %	-6.703,24	46.787	-6.703,24	60	0,00 %	1687	82961
C105	85,45 %	50,00 %	-4.581,19	90	-4.581,19	23	0,00 %	309	4041
C106	84,55 %	50,54 %	-4.519,98	37	-4.519,98	16	0,00 %	311	2660
C107	87,27 %	51,04 %	-4.637,47	211	-4.637,47	26	0,00 %	305	6284
C108	92,73 %	51,96 %	-4.960,75	797	-4.960,75	39	0,00 %	396	11320
C109	97,27 %	53,27 %	-5.119,43	6.415	-5.034,80	61	1,65 %	533	30704
AVG				14.473		45	0,18 %	848	34.921
RC101	96,36 %	54,72 %	-1.259,28	1.756	-1.259,28	39	0,00 %	484	18329
RC102	100,00 %	54,55 %	-1.420,40	34.914	-1.420,40	71	0,00 %	1067	89908
RC103	100,00 %	54,55 %	-1.420,40	35.055	-1.420,40	74	0,00 %	1067	89908
RC104	100,00 %	54,55 %	-1.451,53	39.277	-1.451,53	80	0,00 %	1320	84568
RC105	100,00 %	54,55 %	-1.372,13	35.199	-1.372,13	73	0,00 %	1241	90331
RC106	100,00 %	54,55 %	-1.333,15	11.516	-1.323,25	118	0,74 %	685	41822
RC107	100,00 %	54,55 %	-1.336,22	10.272	-1.336,22	82	0,00 %	684	37861
RC108	100,00 %	54,55 %	-1.389,74	23.039	-1.389,74	80	0,00 %	956	58097
AVG				23.878,50		77,13	0,09 %	938,00	63.853,00
TOTAL				19.355		85,10	0,12 %	945,34	52.371,07

Tabla G.1: Resolution times for the scheduling IOP pricing problem with 10 clients.

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Cost Gap	Feasible	Generated
R101	92,95 %	62,76 %	-1.306,68	3.823	-1.259,93	196	3,58 %	1056	34236
R102	99,36 %	61,94 %	-1.372,84	303.497	-1.398,11	174	-1,84 %	1668	221761
R103	99,36 %	61,94 %	-1.372,84	303.475	-1.398,11	139	-1,84 %	1668	221761
R104	99,36 %	61,94 %	-1.416,48	307.525	-1.422,08	126	-0,40 %	1881	242753
R105	97,44 %	61,18 %	-1.332,96	37.763	-1.308,39	127	1,84 %	1424	89812
R106	100,00 %	61,54 %	-1.224,10	302.066	-1.407,56	201	-14,99 %	1936	197915
R107	100,00 %	61,54 %	-1.384,14	429.201	-1.407,56	192	-1,69 %	1413	287548
R108	100,00 %	61,54 %	-1.395,82	302.407	-1.419,39	188	-1,69 %	1309	247853
R109	100,00 %	61,54 %	-1.331,46	303.125	-1.350,13	213	-1,40 %	1160	260423
R110	100,00 %	61,54 %	-1.224,93	302.738	-1.350,64	196	-10,26 %	1152	221505
R111	100,00 %	61,54 %	-1.370,62	447.666	-1.393,43	145	-1,66 %	1380	295567
R112	100,00 %	61,54 %	-1.351,85	302.449	-1.383,81	174	-2,36 %	919	216891
AVG						173	-2,73 %	1.413,83	211.502
C101	84,62 %	57,58 %	-5.061,03	353	-5.061,03	28	0,00 %	544	11161
C102	99,36 %	61,94 %	-5.374,75	315.909	-5.643,24	165	-5,00 %	1098	227094
C103	99,36 %	61,94 %	-5.374,75	313.873	-5.643,24	160	-5,00 %	1098	227094
C104	99,36 %	61,94 %	-5.082,26	303.419	-5.731,25	202	-12,77 %	1540	236095
C105	88,46 %	58,70 %	-5.106,44	2.366	-5.070,46	45	0,70 %	703	23858
C106	85,90 %	58,21 %	-5.081,79	658	-5.081,79	35	0,00 %	665	14192
C107	89,74 %	59,29 %	-5.145,78	5.737	-5.109,92	49	0,70 %	845	36273
C108	94,87 %	60,14 %	-5.189,22	138.609	-5.189,22	68	0,00 %	1079	143660
C109	98,08 %	60,78 %	-4.892,23	316.279	-5.230,40	113	-6,91 %	1037	195959
AVG						96	-3,14 %	957	123.932
RC101	97,44 %	61,84 %	-1.315,53	38.232	-1.295,64	103	1,51 %	1349	79788
RC102	100,00 %	61,54 %	-1.254,51	318.456	-1.386,53	174	-10,52 %	924	213037
RC103	100,00 %	61,54 %	-1.254,51	324.245	-1.386,53	173	-10,52 %	924	213037
RC104	100,00 %	61,54 %	-1.263,82	326.931	-1.392,88	155	-10,21 %	1326	213901
RC105	99,36 %	61,94 %	-1.337,78	315.219	-1.358,82	145	-1,57 %	1430	231373
RC106	100,00 %	61,54 %	-1.345,08	308.811	-1.345,08	162	0,00 %	1252	238934
RC107	100,00 %	61,54 %	-1.176,69	300.733	-1.344,18	175	-14,23 %	1306	189778
RC108	100,00 %	61,54 %	-1.182,30	359.027	-1.359,28	177	-14,97 %	1327	222043
AVG						158	-7,57 %	1.229,75	200.236
TOTAL						144,83	-4,19 %	1.221,14	181.217,31

Tabla G.2: Resolution times for the scheduling IOP pricing problem with 12 clients.

Name	Density	Neg. Edges	Cost	Time	H. Cost	H. Time	Cost Gap	Feasible	Generated
R101	86,67 %	62,50 %	-1.907,59	7.804	-1.907,59	254	0,00 %	594	48432
R102	92,50 %	62,61 %	-2.075,24	342.480	-2.120,89	343	-2,20 %	2125	246473
R103	98,33 %	62,71 %	-2.075,24	519.789	-2.120,89	353	-2,20 %	2115	291207
R104	98,33 %	62,71 %	-2.018,96	308.405	-2.129,74	211	-5,49 %	1474	230903
R105	95,00 %	61,40 %	-1.982,28	302.412	-1.957,40	221	1,26 %	932	229073
R106	98,33 %	62,29 %	-1.965,19	300.211	-2.164,20	238	-10,13 %	2200	219203
R107	99,17 %	62,18 %	-1.965,19	331.912	-2.164,20	234	-10,13 %	2115	229840
R108	99,17 %	62,18 %	-2.129,85	301.056	-2.164,20	199	-1,61 %	1274	223135
R109	100,00 %	62,50 %	-1.946,29	420.547	-2.031,49	231	-4,38 %	386	268287
R110	100,00 %	62,50 %	-1.859,68	303.586	-2.108,44	487	-13,38 %	709	179518
R111	99,17 %	62,18 %	-1.973,45	307.074	-2.132,88	214	-8,08 %	1527	194650
R112	100,00 %	62,50 %	-1.917,00	312.270	-2.171,38	365	-13,27 %	514	209604
AVG						279	-5,80 %	1.330,42	214.194
C101	85,00 %	58,33 %	-7.526,97	97.092	-7.526,97	266	0,00 %	6430	130573
C102	97,50 %	62,39 %	-7.733,65	319.130	-9.359,70	591	-21,03 %	1350	230774
C103	97,92 %	62,55 %	-7.733,65	335.493	-9.359,70	566	-21,03 %	1350	237938
C104	98,33 %	62,71 %	-9.024,22	389.496	-9.366,41	710	-3,79 %	867	284431
C105	88,75 %	59,62 %	-6.999,10	305.709	-7.686,67	189	-9,82 %	4520	219087
C106	85,83 %	58,74 %	-7.538,48	131.134	-7.538,48	175	0,00 %	6697	152287
C107	90,83 %	60,55 %	-7.501,39	300.876	-7.775,12	243	-3,65 %	2527	198499
C108	95,42 %	61,57 %	-6.543,64	319.316	-8.075,67	462	-23,41 %	1116	175781
C109	98,33 %	61,86 %	-7.684,34	303.530	-8.416,86	382	-9,53 %	674	197424
AVG						398	-10,25 %	2.837	202.977
RC101	95,00 %	62,28 %	-1.914,07	178.206	-1.914,07	140	0,00 %	736	175284
RC102	98,33 %	62,71 %	-1.774,03	303.893	-2.075,02	373	-16,97 %	1745	221891
RC103	99,17 %	62,61 %	-1.774,03	300.666	-2.075,02	396	-16,97 %	1502	224585
RC104	99,17 %	62,61 %	-2.029,63	300.681	-2.075,02	359	-2,24 %	1275	207743
RC105	98,33 %	62,71 %	-1.978,98	366.886	-2.023,63	315	-2,26 %	1615	260700
RC106	100,00 %	62,50 %	-1.995,52	318.011	-1.971,97	302	1,18 %	684	250581
RC107	100,00 %	62,50 %	-2.068,69	301.680	-2.090,17	587	-1,04 %	398	210402
RC108	100,00 %	62,50 %	-1.861,79	301.701	-2.117,32	773	-13,72 %	1140	213558
AVG						406	-6,50 %	1.136,88	220.593
TOTAL						351,00	-7,37 %	1.744,52	212.478,03

Tabla G.3: Resolution times for the scheduling IOP pricing problem with 15 clients.