Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Un algoritmo de generación de columnas para la gestión de ingresos bajo preferencias de clientes

Alumno: Juan José Miranda Bront
L.U. 537/01
jmiranda@dc.uba.ar

Directores: Dra. Isabel Méndez-Díaz
imendez@dc.uba.ar
Departamento de Computación, FCEyN, UBA

Dr. Gustavo Vulcano
gvulcano@stern.nyu.edu
Stern school of bussines, New York University, New York

Marzo de 2007

Tesis para optar al grado de
Licenciado en Ciencias de la Computación

Agradecimientos

A mis viejos y mi abuela, por el aguante y el apoyo incondicional.

A Lupe, mi amor, mi compañera ideal.

A Isabel y Gustavo, dos mentores, dos guías, dos excelentes personas.

A Seba, mi hermano por elección.

A Min Chih Lin, por su aporte invalorable a este trabajo.

Al grupo de Optimización Combinatoria, por la eterna buena onda.

A todos los que me acompañaron en este camino: Moski, Ari, los Tula, Marlene, Leo, Tomás, el otro Leo, los chicos de la oficina, Javier, ...

Resumen

En los últimos años, ha habido una nueva tendencia a enriquecer los modelos tradicionales de *revenue management* basados en el paradigma de demanda independiente mediante la incorporación del comportamiento selectivo de los clientes. Esta extensión involucra desafíos tanto en el modelado como desde el punto de vista computacional.

Una forma de describir este comportamiento selectivo es asumir que cada cliente pertenece a un *segmento* que se caracteriza por un su respectivo *consideration set*, es decir, un subconjunto de los productos que ofrece la compañía que el cliente considera como potenciales opciones. El cliente elige un producto en particular de acuerdo al criterio *multinomial-logit*, un modelo frecuentmente utilizado en la literatura de *marketing*.

En esta tesis, consideramos el modelo determinístico de programación lineal basado en elección (CDLP) propuesto por Gallego et al. [4], y la heurística de descomposición por programación dinámica propuesta por van Ryzin y Liu [10], y nos centramos en una versión más general de los modelos, donde los clientes pertenecen a segmentos superpuestos. Para resolver el CDLP para redes de tamaño real, implementamos un algoritmo de generación de columnas. Probamos que el subproblema de generación de columnas asociado es NP-Completo, y proponemos una heurística golosa para mejorar la complejidad de un algoritmo exacto. Nuestros resultados computacionales muestran que la heurística es muy efectiva, que el método general presenta un buen potencial para ser aplicado en la práctica y lleva a soluciones de alta calidad.

Abstract

In the last few years, there has been a trend to enrich traditional revenue management models built upon the independent demand paradigm by accounting for customer choice behavior. This extension involves both modeling and computational challenges.

One way to describe choice behavior is to assume that each customer belongs to a *segment*, which is characterized by a *consideration set*, i.e., a subset of the products provided by the firm that a customer views as options. The customer chooses a particular product according to a multinomial-logit criterion, a model widely used in the marketing literature.

In this thesis, we consider the choice-based, deterministic, linear programming model (CDLP) of Gallego et al. [4], and the follow-up dynamic programming (DP) decomposition heuristic of van Ryzin and Liu [10], and focus on the more general version of these models, where customers belong to overlapping segments. To solve the CDLP for real-size networks, we need to develop a column generation algorithm. We prove that the associated column generation subproblem is indeed NP-Complete, and propose a greedy heuristic to overcome the complexity of an exact algorithm. Our computational results show that the heuristic is quite effective, the overall approach has good practical potential and leads to high quality solutions.

Índice general

Capítulo 1

Introducción

La disciplina conocida como *Revenue Management* se ocupa, entre otras cosas, de establecer políticas de control sobre distintos recursos con el objetivo de maximizar los ingresos. Ésta puede ser utilizada en diferentes áreas de servicios, por ejemplo en las aerolíneas, hoteles, ferrocarriles, etc. En este trabajo, nos centramos en generalizar una política de control de recursos que incorpora explícitamente el comportamiento selectivo de los clientes, cuya aplicación está directamente relacionada con la venta de pasajes de aerolíneas.

Este capítulo se encuentra organizado de la siguiente manera. Primero explicamos los conceptos básicos del Revenue Management: qué es (Sección 1.1), cuáles son sus orígenes (Sección 1.2), en qué tipo de negocios es conveniente aplicarlo (Sección 1.3), etc. Luego vemos las nociones generales de algunas de las políticas de control utilizadas tradicionalmente en el *Revenue Management* aplicado a aerolíneas (Sección 1.5) y de los modelos existentes para capturar el comportamiento de los clientes (Sección 1.6). Finalmente, la Sección 1.7 contiene una introducción específica para el problema que resolvemos en este trabajo.

1.1. ¿Qué es *Revenue Management*?

Cualquier vendedor de un producto o servicio tiene que enfrentar algunas decisiones fundamentales relacionadas con el proceso de venta, entre ellas: cuándo vender, a qué precio, y por cuánto tiempo. Todo aquel que alguna vez haya tenido que tomar este tipo de decisiones conoce el grado de incertidumbre que involucran. Lo que queremos es vender en el instante en que las condiciones de mercado son más favorables, pero si no vendemos ahora, ¿estamos seguros que sí lo haremos en el futuro? Además, el precio tiene que ser el correcto: no muy alto, para que existan potenciales compradores; tampoco muy bajo, para no perder una potencial ganancia.

En un entorno típico de negocios, es necesario tomar decisiones relacionadas con la demanda aún más complejas. Por ejemplo, ¿Cómo puede una compañía segmentar a sus clientes ofreciéndoles diferentes términos contractuales y condiciones que exploten los diferentes comportamientos o disponibilidades a comprar sus productos? Si esta segmentación es posible, ¿Qué precios debe asignarle a cada segmento? ¿Cómo deben ajustarse estos precios a través del tiempo debido a factores de temporadas y la demanda observada hasta la fecha para cada producto o servicio? Etc.

La disciplina *Revenue Management* se encarga de este tipo de decisiones relacionadas con la demanda mencionadas anteriormente, y de los procesos y sistemas requeridos para implementarlas. Involucra el manejo de la interfaz de la compañıa con el cliente, con el objetivo de incrementar los ingresos de la misma.

¿Qué es nuevo en *Revenue Management*?

En algún sentido, *Revenue Management* se ocupa de un problema de larga data. Todo vendedor en la historia de la humanidad ha enfrentado decisiones sobre la demanda. Sin embargo, la novedad del *Revenue Management* no son las decisiones en si mismas, sino cómo tomarlas. La verdadera innovación del *Revenue Management* es el *método de toma de decisiones*, y se caracteriza por la escala y la naturaleza operacional del control de decisiones. Esto se debe principalmente a los siguientes factores

1. *Avances científicos en la toma de decisiones*: Formulación de modelos de demanda, comportamiento de clientes y condiciones económicas cuantifican el grado de incertidumbre y restricciones que enfrentan aquellos que deben tomar decisiones. A su vez, permiten estimar los parámetros de los modelos, hacer predicciones en función de datos conocidos y optimizar la toma de decisiones sobre cotización y control de ventas.

2. *Avances en las tecnologías de información*: Las computadoras y las redes de comunicación automatizan las transacciones, capturan y guardan grandes cantidades de datos que luego pueden ser procesados por algoritmos para implementar y automatizar las decisiones de demanda resultantes.

Sin el primero de ellos, no hubiese sido posible modelar en forma precisa situaciones del mundo real (y por lo tanto, decisiones del mundo real). Sin los avances tecnológicos, es casi imposible automatizar procesos de este tipo. Estas dos características combinadas ofrecen una nueva visión a la toma de decisiones.

1.2. Los orígenes del *Revenue Management*

Los orígenes del *Revenue Management* están directamente relacionados con la Ley de Liberación de Aerolíneas de 1978. Con esta ley, el Consejo de Aviación Civil de los Estados Unidos (CAB) perdió el control de las tarifas de los vuelos comerciales, que hasta el momento habían estado estrictamente regulados. La principal consecuencia de este cambio fue que las aerolíneas pudieron cambiar libremente sus tarifas, cronogramas de vuelos y servicios sin la aprobación del CAB. En particular, las principales aerolíneas desarrollaron nuevas redes de vuelos que les permitieron ofrecer sus servicios en muchos más mercados que antes.

Al mismo tiempo, entraron al mercado un nuevo tipo de aerolíneas, llamadas de bajo costo. Estas aerolíneas apuntaban a un público que, si no fuese por los bajos precios, utilizaría otro tipo de transporte (por ejemplo, familias yendo de vacaciones, estudiantes, etc.). El potencial de este público se evidenció en el rápido crecimiento de *PeopleExpress*. Esta aerolínea comenzó en 1981, ofreciendo tarfias de entre un 50 % - 70 % más bajas que las principales aerolíneas. Para el final de 1984, sus ganancias llegaban a los 60 millones de dólares. La combinación entre tarifas bajas y el nuevo público generó un incremento significativo en la demanda de vuelos en los Estados Unidos.

Ante esta situación, las grandes aerolíneas se centraron en aquellos clientes que preferían la calidad del servicio o mejores horarios antes que la tarifa (por ejemplo, viajantes de negocios), pero de todos modos seguían siendo afectadas por las bajas en las ganancias.

American Airlines fue una de las primeras en intentar resolver este problema. Su CEO, Robert Crandall, se dio cuenta de que su aerolínea estaba produciendo asientos con un costo marginal casi nulo y, dado que los vuelos despegaban con asientos vacíos, entonces podía utilizar el excedente de asientos para competir con los nuevos entrantes al mercado. Para esto, identificó dos problemas a resolver. El primero era identificar cuáles eran efectivamente estos asientos, para poder ofrecer productos con tarifas más bajas. El segundo lugar, tenían que asegurarse de no perder a los viajantes de negocios, evitando que compren alguno de los nuevos productos "baratos" que ofrecerían.

El resultado fue una promoción llamada *"American Super Savers"*, basada en ofrecer descuentos en las tarifas para viajeros que comprasen con suficiente anticipación y que aseguraran un mínimo de duración de estadía, además de ofrecer un número fijo y limitado de este tipo de asientos por vuelo. Esta nueva estrategia trajo buenos resultados, pero todavía era posible mejorarlo.

Crandall y su equipo identificaron que en realidad la demanda variaba en función del mercado considerado. En algunos casos, los vuelos salían con gran cantidad de asientos libres que podían ser utilizados para vender productos con tarifas bajas; otros tenían suficiente demanda como para vender productos normales. Entonces desarrollaron un complejo sistema llamado DINAMO (*Dynamic Inventory Allocation and Maintenance Optimizer*) que le permitió tener una actitud muchos más agresiva y competitiva en el mercado.

El sistema fue terminado a comienzos de 1985, y *American Airlines* lanzó una nueva promoción llamada *"Ultimate Super Saver Fares"*, que mejoraba o igualaba las mejores ofertas disponibles en los mercados que *American* ofrecía.

El efecto de esta nueva estrategia fue dramático. *PeopleExpress* fue una de las compañías a las que más afecto esta estrategia de ventas: pasó de obtener sus máximas ganancias en 1984, a perder 160 millones a fines de 1986. En solamente dos años, la compañía se declaró en bancarrota. Como contraparte, *American Airlines* reportó un incremento en los ingresos de 1,400 millones de dólares en tres años gracias a las nuevas estrategias.

Como resultado de esta historia, hoy en día prácticamente todas las aerolíneas cuentan con sistemas de *Revenue Management*, y la disciplina ha alcanzado un alto nivel de madurez.

1.3. Un marco conceptual para *Revenue Management*

¿En qué situaciones es posible aplicar *Revenue Management*? Una posible respuesta es: en cualquier negocio donde las decisiones tácticas sobre la demanda son importantes, y existe una cultura relacionada con la tecnología y el management que permiten implementarlo. A continuación veremos cuando se dan estas condiciones.

El entorno de demanda de la compañía

Para responder esta pregunta, primero veremos un marco conceptual para pensar en el proceso de toma de decisiones. En primer lugar, podemos pensar que, conceptualmente, para una compañía la demanda tiene múltiples dimensiones, entre ellas: los diferentes productos que ofrece, los diferentes clientes a los que están dirigidos, con sus distintas preferencias y comportamientos, y el tiempo. Si bien es posible agregar otras dimensiones, estas tres sirven como ejemplo para mostrar la idea principal.

Algunos problemas de *Revenue Management* fijan alguna de estas dimensiones, con el objetivo de explotar el potencial de las restantes. Por ejemplo, las técnicas utilizadas tradicionalmente en *Airline Revenue Management* se encargan de manejar decisiones de demanda para múltiples productos en una cantidad determinada de períodos de tiempo, y la dimensión correspondiente al cliente no se considera explícitamente. Este tipo de ideas se utiliza para simplificar el problema, con el fin de obtener un modelo implementable en la práctica.

En muchos de estos modelos, se utiliza una visión monopólica de la demanda. Es decir, los métodos no incluyen explícitamente el número de competidores, sus respuestas a las decisiones de la compañía, etc. En realidad, la competencia por la demanda es un factor sumamente importante, y la visión monopólica refleja limitaciones tanto prácticas como teóricas.

Sin embargo, esta visión tiene su justificación. Si bien las interacciones competitivas no se modelan explícitamente, los efectos son capturados implícitamente mediante los constantes ciclos de estimación/predicción de la demanda, y la optimización de los controles.

Relaciones entre las decisiones

Si las decisiones de demanda que afectan el entorno fuesen completamente independientes, el problema de la toma de decisión sería considerablemente más simple. Pero típicamente uno o más de los siguientes tres factores relaciona la demanda a través de las dimensiones.

- *Restricciones de producción conjunta y costos*: Generalmente, múltiples productos comparten capacidades o tienen costos de producción conjuntos. En estos casos, la demanda para diferentes productos, o para un mismo producto en distintos instantes de tiempo, está relacionada. Por ejemplo, aceptar la demanda de un cliente para un producto en particular en un instante específico puede significar dejar de lado la oportunidad de aceptar la demanda para otro producto en un instante posterior. Por ejemplo, en el caso de las aerolíneas, aceptar el pedido por un determinado productos puede significar utilizar los asientos de un pasaje con tarifa más alta que podría venderse en el futuro.

- *Comportamiento selectivo de los clientes*: Aún cuando las restricciones de producción no relacionen las demandas de distintos productos, los clientes suelen hacerlo. Por ejemplo, estos pueden elegir entre productos substitutos al que buscaban originalmente. Como resultado, las decisiones que la compañía hace sobre un producto puede afectar la demanda de otros.
- *Información*: Las decisiones de demanda para los distintos productos, clientes y períodos de tiempo deben tomarse en función de la información que la compañía obtiene. La forma más común de hacer esto es a través del tiempo: la demanda observada se utiliza para estimar y cuantificar la demanda futura.

Condiciones para utilizar *Revenue Management*

Teniendo esta conceptualización del problema, podemos obtener una mirada más profunda sobre las condiciones que deben darse para que el *Revenue Management* sea beneficioso. Estas son:

1. *El precio no es una señal de calidad*: El *Revenue Management* se aplica mejor a productos donde el precio no es un símbolo de estatus ni tampoco es señal de su valor. Por ejemplo, consideremos el caso de las aerolíneas. Mientras que las distintas aerolíneas se posicionan de forma diferente en relación al precio y la calidad, los clientes generalmente no asocian la tarifa de un *ticket* con la calidad del vuelo. Por ejemplo, no esperamos tener un mejor vuelo cuando pagamos $300 de excedente en el pasaje por haberlo comprado dos días antes de la fecha de salida.
2. *Los clientes son heterogéneos*: Mientras más heterogéneos son los clientes, mayor es el potencial de utilizar este factor estrategicamente para mejorar los ingresos. Las aerolíneas son un buen ejemplo de esta característica. Los clientes poseen una amplia variedad de patrones de uso y comportamientos diferentes en términos de cuándo compran y cuán flexibles son sus planes.
3. *La demanda es variable a través del tiempo*: A medida que se incrementan las variaciones en la demanda y hay más incertidumbre sobre la demanda futura, más difíciles son las decisiones, aumentando la posibilidad de tomar malas decisiones. En general, en aquellas industrias donde las decisiones de capacidad y cotización tienen que ser hechas con anterioridad, es de esperar que exista incertidumbre en la demanda.
4. *La producción no es flexible*: La inflexibilidad de los productos genera mayor interacción entre las decisiones de demanda en diferentes puntos del tiempo entre diferentes segmentos de compradores, productos, etc. Nuevamente, la industria de las aerolíneas es un ejemplo donde sucede esto. Las capacidades de los vuelos y los precios de los productos son determinados con anterioridad, independientemente de cuántos clientes efectivamente tomen ese vuelo.

5. *Disponibilidad de datos y sistemas*: Esta es una condición de implementación. Para caracterizar el modelo de demanda, es necesario contar con sistemas que recolecten y procesen la información requerida. En la mayoría de las industrias, esta es una condición factible de realizar, pero intentar hacerlo en aquellas que no con cuentan con al menos una parte de la infraestructura puede ser riesgoso. Nuevamente, las aerolíneas cumplen esta condición. Los procesos de cotización y ventas se encuentran automatizados desde hace más de tres décadas.

6. *La cultura del management no rechaza a los métodos científicos*: Ésta es una restricción subjetiva, pero muy importante para poder aplicar técnicas como las propuestas por el *Revenue Management*. Las aerolíneas, nuevamente, son un buen ejemplo para esta condición. En general, las grandes aerolíneas han estado en contacto con la *Investigación Operativa* para poder resolver problemas de *scheduling* y asignación de tripulaciones desde mucho antes que el *Revenue Management* se pusiera en práctica.

Estas condiciones caracterizan aquellos negocios donde es posible aplicar *Revenue Management*, con la posibilidad de obtener buenos resultados. Además, vimos que la industria de las aerolíneas cumple con todas ellas, convirtiéndola en una de las industrias donde más se aplica este tipo de técnicas.

1.4. Descripción de un sistema de *Revenue Management*

En la práctica, un sistema de *Revenue Management* puede ser separado conceptualmente en distintas etapas o pasos a seguir. Estos son:

1. *Recolección de datos*: En general, recolectar y guardar datos históricos (precios, demandas, etc.).

2. *Estimación y predicción*: En base a los datos recolectados, estimar los parámetros del modelo de demanda. Luego hacer una predicción de la demanda basándose en estos parámetros y otros medidas relevantes, como por ejemplo, las cancelaciones.

3. *Optimización*: Buscar el conjunto óptimo de controles que serán utilizados hasta la próxima reoptimización.

4. *Control*: Aplicar sobre el inventario el control optimizado.

El proceso de *Revenue Management* involucra ciclar a través de estos pasos cada determinados intervalos de tiempo. Estos intervalos dependen de distintos factores, como por ejemplo el volumen de los datos, cuán rápido cambian, el tipo de métodos de predicción y optimización involucrados en las distintas etapas. La Figura 1.1 muestra una descripción de este tipo de sistemas[1].

En particular, en este trabajo nos centramos en la etapa 3: la optimización de los controles.

[1]Extraído de Talluri y van Ryzin [8].

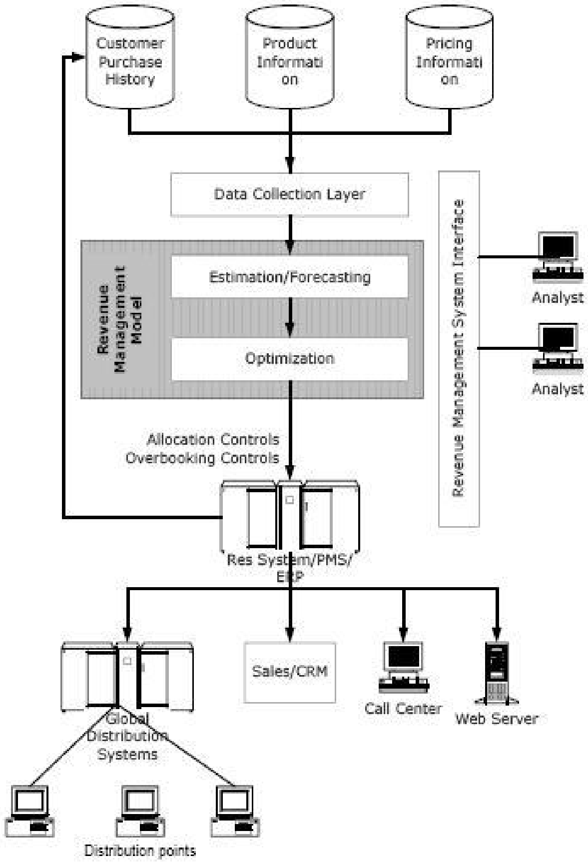


Figura 1.1: Descripción de un sistema de *Revenue Management*.

1.5. Tipos de controles

En el caso particular de las aerolíneas, el problema es manejar las capacidades de un conjunto de vuelos (tramos) en una red del tipo *hub and spoke*, que funciona como punto de conexión y a su vez tiene tráfico local. Un tramo (o recurso) corresponde a un vuelo punto a punto, es decir, un vuelo directo entre dos ciudades. Un producto sobre este tipo de redes se define como una combinación de itinerario (origen-destino, y cuáles son las paradas intermedias) y tarifa (en el siguiente capítulo explicamos en detalle como modelar esto matemáticamente). Luego, distintos productos pueden compartir uno o más tramos, generando una dependencia entre los recursos. En particular, en este trabajo nos centramos en este tipo de redes.

La visión a nivel de red (conocida como *Network Revenue Management*) puede generar beneficios económicos significativos respecto de la versión a nivel de tramo (*single-leg problem*). De hecho, las simulaciones realizadas sobre redes del tipo *hub and spoke* muestran mejoras en los ingresos esperados que van del 0.5 % a valores que superan el 2 %, dependiendo del nivel de congestión de la red. Para empresas que trabajan con márgenes típicamente menores a 10 %, la mejora en las ganancias es significativa.

Sin embargo, la visión a nivel de red agrega cierta complejidad desde el punto de vista práctico, dado que se necesita tomar decisiones considerando un espectro mucho más amplio que el caso de un único tramo.

A continuación veremos las ideas generales de algunas políticas de control, todas ellas aplicadas tradicionalmente en la industria de las aerolíneas. Una explicación detallada de las distintas variantes que se pueden aplicar en cada una de ellas puede verse en Talluri y van Ryzin [8].

Límites de reservas particionados

La idea detrás de esta política de control es simple: reservar un número fijo de asientos sobre cada tramo para cada producto (una combinación de itinerario y clase de tarifa) que es ofrecido. Estas reservas de capacidades no se superponen, en el sentido de que la demanda para un producto solamente tiene acceso a su capacidad y ningún otro producto puede utilizarla.

Esta política tiene algunos defectos. El más importante es que la fragmentación que produce en la capacidad resulta ineficiente (desde el punto de vista de la calidad de la solución) cuando la demanda es estocástica. Por esta razón, estos controles raramente son utilizados en la práctica. Sin embargo, computacionalmente suelen utilizarse en muchos otros métodos aproximados que proveen la entrada a otros tipos de controles.

Niveles de protección anidados

Este tipo de política se define para el caso en el que hay un único tramo. Supongamos que tenemos múltiples productos definidos sobre ese tramo, y que podemos agruparlos jerárquicamente en clases (por ejemplo, por tipos de tarifa). Un nivel de protección especifica una cantidad de asientos que serán reservados para una determinada clase. La propiedad de que sean anidados establece además que una determinada clase tiene acceso a su capacidad reservada y también a las capacidades de las clases de menor jerarquía. Por ejemplo, la clase más alta tiene acceso a toda la capacidad del vuelo.

En la siguiente sección explicamos una aproximación de esta política para el problema a nivel de red.

Controles virtuales anidados

El anidamiento virtual (*virtual nesting*) utiliza reservas de asientos anidadas a nivel de tramo para cada recurso de la red. El anidamiento se realiza sobre un conjunto de *clases virtuales* que no necesariamente corresponden a las tarifas de cada producto. Esta asignación de productos a clases virtuales es conocido como *indexado* y asocia un beneficio a nivel de red para cada producto. Este beneficio se basa en comparar el ingreso del producto con el valor de la capacidad necesario para satisfacer un pedido del mismo, que se lo denomina *costo de oportunidad*. El costo de oportunidad representa la pérdida futura que se generaría al utilizar la capacidad ahora en lugar de reservarla para utilizarla más adelante.

Una vez hecha la asignación, se definen niveles de protección para cada tramo de la red utilizando las clases virtuales y el beneficio a nivel de red de cada producto. Cuando llega un arribo para un determinado producto, se identifican la clase virtual a la cual está asignado sobre cada tramo, y se verifica su disponibilidad de la clase virtual correspondiente sobre los distintos tramos que utiliza. La compra se acepta solamente si la clase se encuentra disponible en todos los casos.

Esta política es muy utilizada en la práctica, y ha dado buenos resultados. Fue introducida por *American Airlines* en 1986, y aún hoy se encuentra operando en muchos sistemas de reservas de pasajes.

Bid prices

La principal diferencia entre las políticas basadas en *bid prices* respecto a las vistas anteriormente radica en que el control se establece en función del ingreso y no de una clase. Un *bid price* determina, para cada tramo, un estimador del costo marginal de la red de consumir la próxima unidad de capacidad. Cuando llega un pedido por un determinado producto, se compara el ingreso instantáneo del mismo (la tarifa) con la suma de los *bid prices* de los recursos que utiliza. Si la tarifa supera esta suma, se acepta el pedido. En caso contrario, se lo rechaza.

1.6. Modelos de demanda

Como vimos en secciones anteriores, un factor muy importante es modelar correctamente la demanda de los clientes. Dependiendo de las características de la industria en la que uno esté aplicando las técnicas de *Revenue Management*, los clientes pueden comportarse de distintas formas.

A continuación describimos las características generales de este modelo, y luego veremos algunos conceptos básicos sobre los modelos de utilidad aleatoria.

Modelo de demanda independiente

El modelo de demanda independiente ha sido un pilar fundamental en el desarrollo del *Revenue Management*, incluídas las políticas de control vistas en la sección anterior. Desde el punto de vista práctico, este modelo hace que las estimaciones y la optimización de los controles se simplifiquen considerablemente.

Este modelo asume que la demanda para cada producto es independiente, y que no está influenciada por los controles aplicados por el vendedor. Además, asume que los clientes arriban en un orden creciente en función de las tarifas de los productos: primero aquellos que prefieren las tarifas más bajas; luego los que prefieren tarifas un poco más altas; etc. Sin embargo, los clientes se comportan de una forma mucho más compleja. En general, suelen elegir productos substitutos en caso de que el que desean no se encuentre disponible, o su decisión puede estar influenciada por los productos que la empresa ofrece en distintos instantes de tiempo.

Por esta razón, en *Revenue Management* es importante incorporar en el modelado el comportamiento selectivo de los clientes.

Generalidades sobre *choice models*

En la práctica, la demanda sobre los productos de una compañía es el resultado de las decisiones de los clientes en distintos instantes del tiempo, como por ejemplo, la elección de un producto sobre otro, esperar para comprar o directamente no hacer la compra, etc. Situaciones de este tipo son las que capturan los *choice models* (modelos de elección). La expansión de las aerolíneas de bajo costo, con su estrategia de saturar el mercado con tarifas similares, generan mayor interés en capturar formalmente el comportamiento selectivo de los clientes.

Considerar este comportamiento selectivo involucra principalmente dos tareas. La primera de ellas es modelar efectivamente la decisión de los clientes en un determinado instante en el tiempo, y también estimar los parámetros que describen esta decisión. La segunda es incorporar estos modelos de demanda a la etapa de optimización de las políticas de control, para que los métodos de control de disponibilidad incorporen explícitamente el comportamiento selectivo.

Existen distintos tipos de *choice models*, entre ellos los denominados modelos de utilidad aleatoria (Ben Akiva y Lerman [3], y Talluri y van Ryzin [8] describen detalladamente distintos tipos de *choice models*). Esta tipo de *choice models* trata al comportamiento asumiendo que las utilidades de los clientes individuales por las distintas alternativas son variables aleatorias, y que los clientes maximizan sus utilidades. Desde el punto de vista de la estimación de los parámetros, este componente aleatorio captura la noción de la incertidumbre del vendedor respecto a cuál es la utilidad de los clientes respecto de los distintos productos. Además, éstos pueden exhibir cierta búsqueda de variedad en la elección, y cambiar sus preferencias a través del tiempo.

El *choice model* que consideramos en este trabajo es el *multinomial logit* (MNL), sobre el que daremos más detalles en la Sección 2.3. Dado que nos centraremos en la etapa de optimización de las políticas de control, asumiremos que la estimación de los parámetros del modelo ya ha sido realizada.

1.7. Acerca de este trabajo

Las políticas de *Revenue Management* basadas en el control de capacidad consisten en administrar la capacidad de un conjunto de recursos en un horizonte finito de tiempo, con el objetivo de maximizar los ingresos. Si bien este tipo de estrategias puede aplicarse en diversas industrias, en este trabajo nos centramos en Revenue Management aplicado a aerolíneas.

Como mencionamos anteriormente, es fundamental incorporar el comportamiento selectivo de los clientes a la hora de modelar. El interés de considerar este comportamiento selectivo en estos métodos tiene sus orígenes en el trabajo de Belobaba y Hopperstand [2], que muestra estudios basados en simulaciones para entender las implicancias del comportamiento selectivo de los clientes respecto de la asunción de demanda independiente. Desde el punto de vista teórico, en Talluri y van

Ryzin [9] se muestra un análisis exacto de la política óptima bajo modelos discretos de elección para el modelo de *Revenue Management* a nivel de tramo. En Zhang y Cooper [14] se analiza la elección de los clientes entre vuelos paralelos con el mismo origen-destino. Utilizando una aproximación basada en simulaciones, pero restringida a una clase paramétrica de políticas (en su caso, controles virtuales anidados), en van Ryzin y Vulcano [11] muestran resultados computacionales promisorios para optimizar políticas de control bajo un modelo de elección muy general.

Un trabajo muy interesante relacionado con la incorporación de los modelos de demanda a la etapa de optimización es el realizado por Gallego et al. [4]. Ellos proponen un modelo de programación lineal que incorpora el comportamiento selectivo de los clientes para el problema de *Revenue Management* a nivel de red. En una continuación del trabajo, van Ryzin y Liu [10] presentan dos alternativas para utilizar la salida del modelo lineal. En ambos trabajos, reportan muy buenos resultados, pero sin embargo los *choice models* que consideran son muy restrictivos. En el primer caso, el modelo de demanda elegido no permite ningún tipo de segmentación (es decir, agrupar en un mismo *segmento* a clientes con preferencias similares) de los clientes. En el segundo, si bien es posible realizar esta segmentación , asume que los productos ofrecidos por la compañía pertenecen a un único segmento.

Motivados por las ideas de Gallego et al. [4], y los resultados obtenidos por van Ryzin y Liu [10], en este trabajo consideramos una extensión de los modelos y métodos propuestos para un escenario más general de *choice model*. En particular, nos centramos en un *choice model* donde los segmentos pueden ser superpuestos, y analizamos las implicancias teóricas y prácticas que trae quitar esta restricción en ambos métodos.

El modelo de programación lineal propuesto tiene un número exponencial de variables, con lo cual desarrollamos un algoritmo de generación de columnas para resolverlo. Probamos que el correspondiente subproblema de generación de columnas es NP-Completo. Considerando el entorno donde éste será utilizado, es fundamental poder resolverlo en forma eficiente. Por esta razón, utilizamos dos técnicas distintas: una heurística golosa para obtener una solución aproximada en forma eficiente, y un modelo lineal entero mixto para resolverlo en forma exacta.

Desde el punto de vista práctico, el principal aporte de este trabajo es la experimentación. Corrimos una serie exhaustiva de tests computacionales, donde comparamos la performance de distintas políticas de control que incorporan la noción del comportamiento selectivo con una basada en el modelo de demanda independiente. También realizamos mediciones de tiempo para cada método, con el fin de evaluar si pueden ser aplicados en la práctica.

El resto de este trabajo se encuentra organizado de la siguiente forma. En el Capítulo 2 introducimos las definiciones básicas utilizadas a lo largo del trabajo, y explicamos la formulación del modelo de programación lineal (Gallego et al. [4]) y la política de control que se basa en su salida (van Ryzin y Liu [10]), especializadas para el MNL. En el Capítulo 3 mostramos cuál es el resultado, desde el punto de vista teórico, de utilizar como *choice model* el MNL con segmentos superpuestos. Los resultados experimentales los exponemos en el Capítulo 4. Finalmente, las conclusiones y el trabajo futuro se encuentran en el Capítulo 5.

Capítulo 2

Modelo General

Como ya explicamos en el capítulo anterior, la disciplina conocida como *Airline Revenue Management* se ocupa de establecer políticas de control sobre las capacidades de una determinada cantidad de vuelos considerados, con el fin de maximizar los ingresos obtenidos. Si bien históricamente los modelos se basaban en el modelo de demanda independiente, los clientes se comportan de una forma mucho más compleja. Por esta razón, resulta indispensable capturar este comportamiento a la hora de modelar. El objetivo de este capítulo es introducir un modelo que considera el comportamiento selectivo de los clientes.

El capítulo se encuentra organizado de la siguiente forma. Comenzamos por enunciar definiciones básicas del área de *Airline Revenue Management* en las secciones 2.1 y 2.3. En esta última explicamos el *choice model* elegido para este trabajo: el *multinomial logit* (MNL). Con estas definiciones, presentamos la formulación exacta que resuelve nuestro problema (Sección 2.2). Debido a limitaciones computacionales, este modelo no puede ser utilizado en la práctica, con lo cual en la Sección 2.4 mostramos la aproximación mediante programación lineal (CDLP) propuesta por Gallego et al.[4], especializada en el caso del MNL. La Sección 2.5 muestra un esquema general sobre cómo resolver esta aproximación, basada en un algoritmo de generación de columnas. Finalmente, debido a que la solución del CDLP presenta cierta ambigüedad, en la Sección 2.6 se explica la heurística de descomposición propuesta por van Ryzin y Liu [10], que permite establecer una política de control de ventas de pasajes basándose en información obtenida de la solución del CDLP.

2.1. Definiciones Básicas

Para introducir nuestro modelo utilizaremos algunas definiciones y notación propuestas en van Ryzin y Liu [10]. La red posee m recursos que pueden ser utilizados por n productos. Las capacidades iniciales y el conjunto de productos son denotados por $c = (c_1, \dots, c_m)$ y $N = \{1, \dots, n\}$ respectivamente. Se define la matriz de incidencia $A = [a_{ij}] \in \{0, 1\}^{m \times n}$, donde $a_{ij} = 1$ si el recurso i es utilizado por el producto j, y $a_{ij} = 0$ en caso contrario. Diremos que A_j, la j-ésima columna de A, es el vector de incidencia del producto j; y la i-ésima fila de A, A^i, es el vector de incidencia para el recurso i. Utilizaremos la notación $i \in A_j$ para decir que el recurso i es utilizado por el producto j; y $j \in A^i$ para indicar que el producto j utiliza el recurso i. El estado de la red está dado por el vector $x^{\mathsf{T}} = (x_1, \ldots, x_m)$ de capacidades remanentes de los recursos. Si una unidad del producto j es vendida, el estado de la red cambia a $x - A_j$. El ingreso obtenido por aceptar un pedido de una unidad del producto j es r_j.

Consideramos períodos de tiempo discretos que son indexados en forma creciente en el tiempo por t, $t = 1, \ldots, T$. Asumimos una discretización temporal de granularidad suficientemente fina de forma tal que la probabilidad de tener mas de un arribo por período es despreciable. Ésto es compatible con un proceso de llegadas Poisson. La probabilidad de que haya un arribo en cada período de tiempo es denominada λ.

La política de control de la aerolínea se ve reflejada mediante el conjunto de productos que ofrece en cada instante del horizonte de reservas. Llamaremos a este conjunto $S \subseteq N$ de productos disponibles *offer set*. Dado un *offer set* S, un comprador elige un producto $j \in S$ con probabilidad $P_j(S)$, donde $P_j(S) = 0$ si $j \notin S$. Llamaremos $P_0(S)$ a la probabilidad de no compra, y por probabilidad total, $\sum_{j \in S} P_j(S) + P_0(S) = 1$. En particular, como se ve generalmente en la literatura de marketing relacionada con *Airline Revenue Management*, estas probabilidades estarán basadas en el modelo *multinomial logit* (MNL), cuyas nociones básicas veremos en la Sección 2.3.

2.2. Modelo de programación dinámica

El problema de decisión consiste en encontrar una política para elegir *offer sets* S en cualquier instante t durante el horizonte de reservas, y puede ser formulado mediante un problema de programación dinámica (DP). Asumiendo que la compañía es neutral al riesgo, y que desea maximizar los ingresos esperados (que llamamos $V_t(x)$), nuestro problema de decisión puede ser expresado mediante la ecuación de Bellman (ver Bellman [1]) como

$$\begin{aligned} V_t(x) &= \max_{S \subseteq N} \left\{ \sum_{j \in S} \lambda P_j(S)(r_j + V_{t+1}(x - A_j)) + (\lambda P_0(S) + 1 - \lambda) V_{t+1}(x) \right\} \\ &= \max_{S \subseteq N} \left\{ \sum_{j \in S} \lambda P_j(S)(r_j - (V_{t+1}(x) - V_{t+1}(x - A_j))) \right\} + V_{t+1}(x) \end{aligned} \tag{2.1}$$

con condiciones de borde

$$\begin{aligned} V_t(0) &= 0, \; t = 1, \ldots, T \\ V_{T+1}(x) &= 0, \; \forall x \geq 0. \end{aligned}$$

La ecuación (2.1) se deduce del hecho de que $\sum_{j \in S} P_j(S) + P_0 = 1$. Esta expresión la obtenemos al plantear que el ingreso esperado máximo en el instante t con capacidades remanentes x está dado por el ingreso esperado de vender una unidad de un producto ofrecido en este instante, sumado al ingreso que se generaría si en lugar de vender ahora lo hacemos en el instante $t+1$. El conjunto que maximice esta expresión es el que debe ofrecer la compañía.

Desde el punto de vista teórico, Talluri y van Ryzin [9] proveen un análisis exacto de la política de control óptima bajo modelos discretos de demanda de elección para el problema (2.1) a nivel de tramo. Entre otras cosas, demuestran que la política óptima es bien simple, y consiste en abrir secuencialmente conjuntos de productos pertenecientes a una familia ordenada llamada *conjuntos eficientes*. Zhang y Cooper [14] analizan la elección de los clientes entre vuelos paralelos con el mismo origen-destino. Su modelo asume que los clientes eligen entre las tarifas similares para vuelos distintos, pero no entre tarifas distintas.

Desafortunadamente, este modelo no es manejable computacionalmente para la mayoría de las redes reales debido a la gran cantidad de estados que posee: $T \times \prod_{i=1}^{m} c_i$. Esta gran cantidad de estados se debe a que el modelo considera todas las combinaciones posibles de capacidades remanentes de los tramos de la red.

2.3. Multinomial Logit

Para modelar el comportamiento de compra de los clientes, seguimos la línea propuesta por van Ryzin y Liu [10], donde las probabilidades serán determinadas mediante el modelo *multinomial logit*, uno de los más utilizados en la literatura relacionada con el *Revenue Management*. Ben-Akiva y Lerman [3] muestran una descripción completa del modelo, incluyendo propiedades y resultados teóricos sobre el mismo.

Bajo las hipótesis de este *choice model*, asumimos que los clientes pertencen a diferentes segmentos de mercado $l = 1, \ldots, L$, y que cada segmento es caracterizado por un *consideration set* $C_l \subseteq N$. Desde el punto de vista de la aerolínea, un cliente que arriba pertenece al segmento l con probabilidad p_l, donde $\sum_{l=1}^{L} p_l = 1$. Luego, cada flujo de arribos de clientes de un segmento l es un Proceso Poisson con tasa $\lambda_l = \lambda p_l$, ya que asumimos un proceso de llegadas Poisson con tasa λ.

La probabilidad de elección de un producto por parte de un cliente perteneciente a un segmento l se encuentra definida en términos de un vector de preferencias, denominado v_l, que indica el valor de preferencia de un cliente para cada producto contenido en C_l. Diremos que v_{l0} es el valor de preferencia de no-compra.

Considerando que la aerolínea ofrece el conjunto S en el instante t, las probabilidades de elección de un cliente se determinan de la siguiente forma: si $P_{lj}(S)$ determina la probabilidad de que un producto j sea elegido, dado que el conjunto ofrecido es S y el cliente pertenece al segmento l, entonces

$$\begin{aligned} P_{lj}(S) &= \mathbb{P}\{\text{elegir} j | S \wedge l\} \\ &= \frac{\mathbb{P}\{\text{elegir} j \wedge S | l\}}{\mathbb{P}\{S | l\}} \\ &= \frac{v_{lj} / \left(\sum_{i \in C_l} v_{li} + v_{l0} \right)}{\left(\sum_{k \in C_l \cap S} v_{lk} + v_{l0} \right) / \left(\sum_{i \in C_l} v_{li} + v_{l0} \right)} \\ &= \frac{v_{lj}}{\sum_{i \in C_l \cap S} v_{li} + v_{l0}} \end{aligned}$$

Si $j \notin C_l \cap S$, entonces $v_{lj} = 0$ (y luego $P_{lj}(S) = 0$).

Definimos $y \in \{0,1\}^n$ como el vector característico del conjunto S. Éste determina qué productos son ofrecidos en cualquier instante t en el horizonte de reservas, de manera tal que si S es el conjunto que se ofrece, $y_j = 1$ si $j \in S$, e $y_j = 0$ en caso contrario. Escribiendo $P_{lj}(S)$ en términos del vector y, obtenemos que

$$P_{lj}(S) = \frac{v_{lj} y_j}{\sum_{i \in C_l} v_{li} y_i + v_{l0}}$$

La principal diferencia de nuestro modelo en relación al considerado por van Ryzin y Liu [10] es que nosotros permitimos que los *consideration sets* correspondientes a segmentos distintos no sean necesariamente disjuntos. Es decir, es posible que $C_l \cap C_{l'} \neq \emptyset$ para $l \neq l'$. A esta versión del *multinomial logit* la llamamos *MNL con segmentos superpuestos*.

Quitar esta restricción sobre los *consideration sets* ofrece mayor flexibilidad a la hora de modelar aunque, como veremos más adelante, modifica la complejidad de la resolución de nuestro problema.

2.4. Modelo de programación lineal

Como ya señalamos en la sección anterior, el modelo de programación dinámica resulta ineficiente desde el punto de vista computacional al aplicarlo a redes de tamaño mediano-grande. Por esta razón, siguiendo la línea propuesta por Gallego et al. [4] (y utilizada también por van Ryzin y Liu [10]), la técnica considerada para evitar esta dificultad es aproximar el problema (2.1) mediante un problema de programación lineal.

Con el fin de obtener la formulación de la aproximación, el *Choice-based Deterministic Linear Programming problem* (CDLP), comenzamos por interpretar $P_j(S)$ como la cantidad determinística (fraccionaria) vendida del producto j cuando el conjunto S es ofrecido y un comprador arriba. Definimos $R(S)$ como el ingreso esperado generado por un comprador que arriba, es decir,

$$R(S) \quad = \quad \sum_{j \in S} r_j P_j(S). \tag{2.2}$$

Dado un arribo, diremos que $Q_i(S)$ es la probabilidad de utilizar una unidad de capacidad sobre el tramo i considerando que el conjunto ofrecido es S. El vector de probabilidades de consumo de capacidades está dado por $Q(S) = AP(S)$, donde $P(S) = (P_1(S), \ldots, P_n(S))$ es el vector de probabilidades de compra para los distintos productos.

El objetivo de esta formulación es determinar la secuencia de *offer sets* $\{S(1), \ldots, S(T)\}$, es decir, qué subconjunto de productos tiene que ofrecer la compañıa en cada instante del horizonte de reservas, de forma tal que se maximicen los ingresos. Llamaremos $t(S)$ a las variables que determinan la cantidad de períodos que el conjunto S es ofrecido, y éstas serán nuestras variables de decisión. En este modelo CDLP, dado que la demanda es determinística y las probabilidades son homogéneas respecto del tiempo, cualquier permutación de la secuencia de *offer sets* generará el mismo ingreso y consumirá la misma capacidad, con lo cual no importa el orden en el que se ofrecen los conjuntos.

Haciendo una relajación más al modelo, permitiremos que las variables $t(S)$ sean continuas. Este hecho podemos interpretarlo de la siguiente forma: si, por ejemplo, para un conjunto S_0, $t(S_0) = 4.5$, entonces S_0 se ofrece durante cuatro períodos y la mitad de otro. Con estas definiciones, la formulación del CDLP es la siguiente

$$\begin{aligned} V^{CDLP} \quad = \quad & \max \sum_{S \subseteq N} \lambda R(S) t(S) \\ & \text{s.t.} \\ & \sum_{S \subseteq N} \lambda Q(S) t(S) \leq c \\ & \sum_{S \subseteq N} t(S) \leq T \\ & t(S) \geq 0 \end{aligned} \tag{2.3}$$

Este modelo busca maximizar el total de ingresos acumulados. En primer lugar, tenemos un conjunto de restricciones que expresa que el consumo total no supera las capacidades iniciales, es decir, no vendemos más asientos de los que tenemos. En segundo lugar, existe una última restricción que establece que el tiempo total que los conjuntos son ofrecidos no excede el largo del horizonte de reservas.

Dado que tenemos una variable $t(S)$ por cada conjunto $S \subseteq N$ (excluyendo el conjunto vacío), el modelo posee un numero exponencial de variables. La forma tradicional de resolver este tipo de problemas en el área de *Investigación Operativa* es mediante la técnica de *generación de columnas*. Es importante destacar que esta formulación se encuentra expresada en forma general, en el sentido de que las probabilidades dependen del *choice model* elegido para caracterizar el comportamiento selectivo de los clientes.

Desde el punto de vista teórico, van Ryzin y Liu [10], entre otras cosas, demuestran que el CDLP es asintóticamente óptimo respecto a la formulación (2.1). Ésto significa que, independientemente del *choice model*, a medida que la capacidad y la demanda son escaladas proporcionalmente, el ingreso obtenido por el CDLP converge al ingreso óptimo de la formulación exacta (2.1). Este resultado brinda el marco teórico necesario que justifica la utilización de la aproximación.

En nuestro caso, los clientes que arriban eligen su producto de acuerdo al MNL, con lo cual tenemos que especializar el CDLP para este *choice model* en particular. Como vimos en la Sección 2.3, el MNL asume que cada cliente pertenece a un determinado segmento l, $l = 1, \ldots, L$. Cada segmento tiene asociado un *consideration set* $C_l \subseteq N$ que lo caracteriza, y los clientes hacen su elección considerando los productos de su segmento que se encuentren disponibles. Desde el punto de vista de la aerolínea, cuando se produce un arribo no es posible determinar con anterioridad a qué segmento pertenece el cliente. Luego, la probabilidad de que un cliente que arriba elija un producto $j \in S$, se encuentra dada por

$$P_j(S) = \sum_{l=1}^{L} p_l P_{lj}(S). \tag{2.4}$$

Al igual que con las probabilidades, primero definimos el ingreso esperado generado por clientes de un segmento en particular, y en función de esta expresión expresamos cuál es el ingreso total esperado. El ingreso esperado generado por un comprador perteneciente a un segmento l en particular dado que el *offer set* es S lo definimos como

$$R_l(S) = \sum_{j \in S} r_j P_{lj}(S) \tag{2.5}$$

y reemplazando esta expresión en (2.2) obtenemos que el ingreso generado por un cliente que arriba (independientemente del segmento al que pertenece) es

$$R(S) = \sum_{l=1}^{L} p_l R_l(S). \tag{2.6}$$

Por último, definimos los vectores de probabilidades de compra dado que el cliente que arriba pertenece a un segmento l como $P_l(S) = (P_{l1}(S), \ldots, P_{ln}(S))$. En función de estos vectores, definimos los vectores de probabilidades de consumo dado que arriba un cliente del segmento l y el *offer set* es S como $Q_l(S) = AP_l(S)$. Luego, el vector de tasas esperadas de consumo para el *offer set* S está dado por

$$Q(S) = \sum_{l=1}^{L} p_l Q_l(S). \tag{2.7}$$

Con estas definiciones podemos expresar la versión general del CDLP para el caso particular en el que las probabilidades se basan en utilizar el MNL como *choice model*. Reemplazando las nuevas definiciones de $R(S)$ y $Q(S)$, (2.6) y (2.7) respectivamente, en la formulación general del CDLP (2.3), y sabiendo que $\lambda_l = \lambda p_l$, el CDLP para este *choice model* en particular, queda formulado como

$$\begin{aligned} V^{CDLP} \quad = \quad & \text{máx} \sum_{l=1}^{L} \lambda_l \sum_{S \subseteq N} R_l(S) t(S) \\ & \text{s.a.} \\ & \sum_{l=1}^{L} \lambda_l \sum_{S \subseteq N} Q_l(S) t(S) \leq c \\ & \sum_{S \subseteq N} t(S) \leq T \\ & t(S) \geq 0 \end{aligned} \tag{2.8}$$

Recordemos que las variables de decisión, $t(S)$, determinan la cantidad de tiempo que el conjunto $S \subseteq N$ es ofrecido durante el horizonte de reservas. Al haber una variable por cada posible *offer set*, la cantidad total de variables es exponencial en función de la cantidad de productos.

2.5. Resolución del CDLP

Como se mencionó anteriormente, este modelo posee $2^n - 1$ variables (una por cada posible subconjunto S no vacío de N), y la técnica elegida para su resolución es generación de columnas. Gallego et al. [4] y van Ryzin y Liu [10] utilizan esta técnica para la resolución del CDLP bajo los distintos *choice models* que consideran, y reportan en sus trabajos resultados computacionalmente promisorios.

El esquema general del algoritmo es el siguiente: comenzamos con un conjunto restringido de columnas (o, equivalentemente, variables) y se resuelve un LP reducido considerando únicamente estas columnas. Luego, verificamos si existe alguna columna que no hayamos considerado cuyo costo reducido sea positivo en función de las variables duales del LP reducido. En caso afirmativo, agregamos una nueva columna con costo reducido positivo al LP reducido y repetimos el procedimiento. En caso contrario, quiere decir que la solución encontrada es óptima.

El problema primal reducido correspondiente al CDLP (2.8), asumiendo que inicialmente consideramos solamente un conjunto limitado de columnas $\mathcal{N} = \{S_1, \ldots, S_k\}$ es

$$\begin{aligned} V^{CDLP-R} \quad = \quad & \text{máx} \sum_{l=1}^{L} \lambda_l \sum_{S \in \mathcal{N}} R_l(S) t(S) \\ & \text{s.a.} \\ & \sum_{l=1}^{L} \lambda_l \sum_{S \in \mathcal{N}} Q_l(S) t(S) \leq c \\ & \sum_{S \in \mathcal{N}} t(S) \leq T \\ & t(S) \geq 0 \end{aligned} \tag{2.9}$$

Para verificar si existe alguna columna con costo reducido positivo que aún no hemos considerado, tenemos que formular el subproblema de generación de columnas. Éste problema corresponde a la búsqueda implícita de una variable con costo reducido máximo. Llamamos π a las variables duales correspondientes a las restricciones de capacidad del CDLP-R, y σ a la correspondiente a la restricción de tiempo. En cada iteración del algoritmo de generación de columnas, para analizar los costos reducidos de las variables, basta considerar el siguiente problema de maximización

$$\max_{S \subseteq N} \Big\{ \lambda R(S) - \lambda \pi^{\mathsf{T}} Q(S) - \sigma \Big\}. \tag{2.10}$$

Dado que σ es un parámetro conocido, resolvemos el problema de maximización y luego comparamos el valor encontrado con σ. Si el valor óptimo de la función (2.10) no es positivo, entonces π y σ son factibles duales y la solución actual encontrada para el problema lineal reducido es el la solución óptima del CDLP original (2.8). En caso contrario, la solución óptima S^* del problema (2.10) determina una nueva columna a ser agregada a la formulación reducida, y se itera nuevamente.

Este problema se encuentra expresado en forma general. Teniendo en cuenta que en este trabajo estamos asumiendo que las probabilidades se encuentran determinadas de acuerdo al modelo MNL, reemplazando las definiciones de $R(S)$ y $Q(S)$ vistas anteriormente y haciendo un poco de álgebra podemos ver que el problema (2.10) puede reescribirse como se muestra a continuación. Una vez hecho el reemplazo, unificamos ambas sumatorias sobre los segmentos y reemplazamos λp_l por su definición, λ_l. Luego expresamos $R_l(S)$ por su definición en términos de $P_{lj}(S)$. Finalmente sacamos $P_{lj}(S)$ como factor común.

$$\begin{aligned}
\max_{S\subseteq N}\left\{\lambda R(S) - \lambda\pi^{\mathsf{T}}Q(S) - \sigma\right\} &= \max_{S\subseteq N}\left\{\lambda R(S) - \lambda\pi^{\mathsf{T}}Q(S)\right\} - \sigma \\
&= \max_{S\subseteq N}\left\{\lambda\sum_{l=1}^{L} p_l R_l(S) - \lambda\pi^{\mathsf{T}}\sum_{l=1}^{L} p_l Q_l(S)\right\} - \sigma \\
&= \max_{S\subseteq N}\left\{\sum_{l=1}^{L}\lambda_l\Big(R_l(S) - \pi^{\mathsf{T}}Q_l(S)\Big)\right\} - \sigma \\
&= \max_{S\subseteq N}\left\{\sum_{l=1}^{L}\lambda_l\Big(\sum_{j\in S} r_j P_{lj}(S) - \pi^{\mathsf{T}}AP_l(S)\Big)\right\} - \sigma \\
&= \max_{S\subseteq N}\left\{\sum_{l=1}^{L}\lambda_l\Big(\sum_{j=1}^{n} P_{lj}(S)(r_j - \pi^{\mathsf{T}}A_j)\Big)\right\} - \sigma.
\end{aligned}$$

Por definición, si un producto j no pertenece al *consideration set* C_l de un segmento l, entonces $v_{lj} = 0$ (y, por lo tanto, $P_{lj}(S) = 0$). Sabiendo esto, y sustituyendo en la expresión anterior $P_{lj}(S)$ por su definición en función del vector característico de S, $y \in \{0,1\}^n$, llegamos a que el subproblema de generación de columnas es equivalente a uno de programación hiperbólica 0-1

$$\begin{aligned}
\max_{S\subseteq N}\left\{\lambda R(S) - \lambda\pi^{\mathsf{T}}Q(S) - \sigma\right\} &= \max_{y\in\{0,1\}^n}\left\{\sum_{l=1}^{L}\lambda_l\Big(\sum_{j=1}^{n}\frac{v_{lj}y_j}{\sum_{i\in C_l} v_{li}y_i + v_{l0}}(r_j - \pi^{\mathsf{T}}A_j)\Big)\right\} - \sigma \\
&= \max_{y\in\{0,1\}^n}\left\{\sum_{l=1}^{L}\lambda_l\frac{\sum_{j\in C_l}(r_j - A_j^{\mathsf{T}}\pi)v_{lj}y_j}{\sum_{j\in C_l} v_{lj}y_j + v_{l0}}\right\} - \sigma \qquad (2.11)
\end{aligned}$$

La dificultad en la resolución de este problema depende de las hipótesis establecidas sobre el *choice model* elegido para caracterizar el comportamiento de los clientes. Por ejemplo, van Ryzin y Liu [10] muestran que utilizando el MNL con segmentos disjuntos ésto puede hacerse eficientemente. En nuestro caso, como veremos más adelante, al quitar esta restricción no se puede asegurar el mismo resultado.

2.6. Heurística de descomposición

Una vez resuelto el CDLP, todavía queda una cuestión por resolver: ¿cómo puede ser utilizada la salida del CDLP en la práctica? Una posible respuesta es utilizar directamente la solución primal, pero esta opción presenta algunos inconvenientes.

En primer lugar, el orden en el cual deben ofrecerse los conjuntos S correspondientes a las variables $t(S)$ presentes en la solución óptima del CDLP (2.8) no se encuentra especificado. La salida del CDLP solamente especifica la cantidad de tiempo que un *offer set* debe ser ofrecido, presentando cierta ambigüedad sobre cómo hacerlo. Igualmente, uno podría intentar diferentes heurísticas con el fin de determinar el orden en el que serán ofrecidos. Por ejemplo, se podrían ordenar lexicográficamente considerando las respectivas variables del CDLP, o también determinar el orden en forma aleatoria. En segundo lugar, sin embargo, cabe destacar que la mayor dificultad radica en que la solución es estática y no se ajusta a los cambios en la demanda y al tiempo y capacidades remanentes. Uno podría intentar resolver el CDLP frecuentemente, pero de todos modos entre cada optimización sigue estando presente el problema de la ambigüedad en la solución.

Para evitar estas dificultados, van Ryzin y Liu [10] proponen un método de descomposición que utiliza información de la solución del CDLP. La idea principal es descomponer el problema de programación dinámica a nivel de red (2.1) en un conjunto de problemas similares a nivel de tramo, donde cada uno de ellos es unidimensional y mas fácil de resolver en forma exacta.

Con el fin de obtener una política operacional, la aerolínea primero debe seguir un procedimiento que se encuentra dividido en dos etapas. La primera de ellas es un proceso que se realiza en forma *off-line*, y consiste en determinar (al menos en forma aproximada) el valor esperado de la capacidad remanente. Este valor depende del número de asientos disponibles en los tramos de la red, y del tiempo remanente en el horizonte de reservas. La segunda etapa consiste en un proceso que debe ser realizado en tiempo real. En cada período, la aerolínea debe calcular dinámicamente el siguiente *offer set*.

A continuación describimos la heurística de descomposición propuesta por van Ryzin y Liu [10], y cómo adaptarla cuando el comportamiento selectivo de los clientes se modela mediante el MNL.

2.6.1. Etapa 1: Determinando el valor de la capacidad

Sean $\pi^* = (\pi_1^*, \ldots, \pi_n^*)$ los valores duales óptimos de (2.3), correspondientes a las m restricciones de capacidad. Llamamos $\hat{V}_t^i(x_i)$ a la aproximación dinámica del valor de la capacidad remanente (dependiente del tiempo) sobre un tramo i. Interpretando $\pi_k^* x_k$ como las aproximaciones estáticas (independientes del tiempo) del valor de la capacidad remanente sobre los restantes tramos, podemos aproximar el valor de la función de ingreso (2.1) como

$$V_t(x) \approx \hat{V}_t^i(x_i) + \sum_{k \neq i} \pi_k^* x_k \tag{2.12}$$

Teniendo en cuenta esta aproximación, buscamos un estimador para el costo de oportunidad de un producto j, $V_t(x) - V_t(x - A_j)$ en la función (2.1), dado que la aproximación dinámica se realiza sobre el tramo i y las capacidades y el tiempo remanente son x y t, respectivamente. Llamamos $\Gamma_i^j(t, x_i)$ a este estimador, y en base a la aproximación (2.12) queda definido como

$$V_t(x) - V_t(x - A_j) \approx \Gamma_i^j(t, x_i) = \begin{cases} \Delta \hat{V}_t^i(x_i) + \sum_{k \in A_j, k \neq i} \pi_k^* & \text{si } i \in A_j \\ \sum_{k \in A_j} \pi_k^* & \text{si } i \notin A_j \end{cases}$$

donde $\Delta \hat{V}_t^i(x_i) = \hat{V}_t^i(x_i) - \hat{V}_t^i(x_i - 1)$.

Tomamos este estimador y lo reemplazamos en la formulación exacta (2.1). La expresión resultante depende solamente del instante de tiempo t y la capacidad del tramo i considerado en la aproximación. Esto es justamente lo que anteriormente llamamos $\hat{V}_t^i(x_i)$, y queda definido mediante la siguiente expresión

$$\hat{V}_t^i(x_i) = \max_{S\subseteq N}\left\{\sum_{j\in S}\lambda P_j(S)\big(r_j - \Gamma_i^j(t+1,x_i)\big)\right\} + \hat{V}_{t+1}^i(x_i) \tag{2.13}$$

con condiciones de borde

$$\begin{aligned} \hat{V}_{T+1}^i(x_i) &= 0, \qquad\quad \forall x_i \geq 0 \\ \hat{V}_t^i(0) &= 0, \quad \forall t = 1,\ldots,T+1. \end{aligned}$$

El cálculo de cada una de estas tablas $\hat{V}_t^i(c_i)$, $i = 1,\ldots,m$, consiste en resolver $T\times c_i$ problemas de optimización del tipo (2.13). Típicamente, la aerolínea calcula estas tablas mediante un proceso *batch*. La complejidad y la dificultad en la resolución del problema (2.13) dependerá, al igual que en los casos vistos anteriormente, del *choice model* elegido. Es importante notar también que el problema (2.13) tiene la misma estructura del subproblema de generación de columnas (2.11): simplemente reemplazamos $(rj - A_j^\top\pi)$ por $(rj - \Gamma_i^j(t+1,x_i))$.

La aproximación (2.12) podemos aplicarla para cada uno de los tramos considerados en la red. Cada una de las funciones a nivel de tramo, $\hat{V}_t^i(x_i)$, corresponde a una aproximación dinámica de la función de ingreso a nivel de red en el instante t, considerando que hay x_i unidades de capacidad disponibles. Luego, podemos obtener una aproximación al valor real de la función de ingreso original (2.1) promediando las aproximaciones (2.12), es decir,

$$V_t(x) \approx \frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_t^i(x_i) + \sum_{k\neq i}\pi_k^* x_k\right). \tag{2.14}$$

En esta aproximación de la función $V_t(x)$ se basa la segunda etapa del proceso: calcular dinámicamente el *offer set* que será ofrecido si se produce un arribo en el instante t, considerando que las capacidades remanentes de los distintos tramos es x.

2.6.2. Etapa 2: Calculando los *offer sets* dinámicamente

Una vez calculadas las aproximaciones a nivel de tramo, queremos utilizar esta información para determinar qué subconjunto de productos $S \subseteq N$ es conveniente ofrecer en cada instante de tiempo t, dadas las capacidades remanentes x. Cómo calcular el *offer set* S se puede deducir a partir de la aproximación (2.14). Parte de esta deducción se muestra a continuación.

Comenzamos por reemplazar en la aproximación del valor real de la función de ingreso los $\hat{V}_t^i(x_i)$ por sus respectivas definiciones. Luego de manipular las sumatorias, haciendo un poco de álgebra, llegamos a que el valor de la función $V_t(x)$ puede aproximarse como el valor óptimo de un problema de maximización, sumado a un término constante. La ecuación (2.15) muestra este resultado parcial.

$$
\begin{aligned}
V_t(x) &\approx \frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_t^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)\\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\max_{S^i\subseteq N}\left\{\sum_{j\in S^i}\lambda P_j(S^i)(r_j-\Gamma_i^j(t+1,x_i))\right\}+\hat{V}_{t+1}^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)\\
&= \frac{1}{m}\sum_{i=1}^{m}\max_{S^i\subseteq N}\left\{\sum_{j\in S^i}\lambda P_j(S^i)(r_j-\Gamma_i^j(t+1,x_i))\right\}+\frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_{t+1}^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)\\
&\geq \max_{S\subseteq N}\left\{\frac{1}{m}\sum_{i=1}^{m}\sum_{j\in S}\lambda P_j(S)(r_j-\Gamma_i^j(t+1,x_i))\right\}+\frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_{t+1}^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)\\
&= \max_{S\subseteq N}\left\{\sum_{j\in S}\lambda P_j(S)\frac{1}{m}\sum_{i=1}^{m}(r_j-\Gamma_i^j(t+1,x_i))\right\}+\underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_{t+1}^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)}_{\approx\, V_{t+1}(x)} \qquad (2.15)
\end{aligned}
$$

Considerando nuevamente la aproximación (2.14), podemos ver que el término constante aproxima a $V_{t+1}(x)$. Respecto al problema de maximización, todavía queda un pequeño análisis por realizar. En primer lugar, definimos $\Delta\hat{V}_{t+1}(x) = (\Delta\hat{V}_{t+1}^1(x_1),\ldots,\Delta\hat{V}_{t+1}^m(x_m))$ como el vector de estimadores dinámicos del costo de oportunidad. Teniendo en cuenta el problema de maximización, podemos ver que

$$
\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m}\Big(r_j-\Gamma_i^j(t+1,x_i)\Big) &= \frac{1}{m}\Big(m\ r_j-\Delta\hat{V}_{t+1}(x)A_j-(m-1)\pi^{*\mathsf{T}}A_j\Big)\\
&= r_j-\frac{1}{m}\Delta\hat{V}_{t+1}(x)A_j-\frac{(m-1)}{m}\pi^{*\mathsf{T}}A_j.
\end{aligned}
$$

De esta forma obtenemos una aproximación del costo de oportunidad real, es decir,

$$
V_{t+1}(x)-V_{t+1}(x-A_j) \approx \frac{1}{m}\Delta\hat{V}_{t+1}(x)A_j+\frac{(m-1)}{m}\pi^{*\mathsf{T}}A_j.
$$

Esta aproximación es levemente distinta a la propuesta por van Ryzin y Liu [10], dado que explícitamente expresa que los valores marginales de las capacidades son una combinación convexa entre los estimadores estáticos y los dinámicos. Para los casos en los que m toma valores muy grandes (es decir, para redes con un gran número de tramos), $V_{t+1}(x)-V_{t+1}(x-A_j)\approx\pi^{*\mathsf{T}}A_j$. Luego, podemos considerar como alternativa una aproximación heurística, $\Delta\bar{V}_{t+1}(x)$, de la forma

$$
V_{t+1}(x)-V_{t+1}(x-A_j) \approx \Delta\bar{V}_{t+1}(x)=\beta\Delta\hat{V}_{t+1}(x)A_j+(1-\beta)\pi^{*\mathsf{T}}A_j, \quad 0\leq\beta\leq 1. \qquad (2.16)
$$

Reemplazando esta expresión en (2.15), llegamos a que la aproximación de la función real del valor de la capacidad es

$$
\begin{aligned}
V_t(x) &\approx \frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_{t+1}^i(x_i)+\sum_{k\neq i}\pi_k^* x_k\right)\\
&\approx \max_{S\subseteq N}\left\{\sum_{j\in S}\lambda P_j(S)(r_j-\Delta\bar{V}_{t+1}(x)A_j)\right\}+V_{t+1}(x).
\end{aligned}
$$

Esta desarrollo nos permite concluir que, durante el horizonte de reservas, la aerolínea debe elegir dinámicamente un conjunto S en cada instante t resolviendo

$$\max_{S \subseteq N} \left\{ \sum_{\substack{j \in S, \\ j \text{ disponible}}} \lambda P_j(S)(r_j - \Delta \bar{V}_{t+1}(x) A_j) \right\} \tag{2.17}$$

donde los productos candidatos son aquellos que están disponibles, en el sentido de que satisfacen que $x_i > 0$ para todo $i \in A_j$.

Nuevamente, podemos ver que (2.17) tiene la misma estructura que el subproblema de generación de columnas (2.11). El problema (2.17) tiene especial importancia, dado que representa el problema de optimización que la aerolínea debe resolver en tiempo real para determinar cuál es el *offer set* para que el cliente haga su elección (entre ellas, la de no comprar). La eficiencia en su resolución dependerá del *choice model* elegido. En el caso considerado por van Ryzin y Liu [10], el MNL con segmentos disjuntos, la resolución puede realizarse utilizando un algoritmo polinomial. En nuestro caso, al considerar el MNL con segmentos superpuestos, no podemos afirmar esto último. El siguiente capítulo, entre otras cosas, analiza en detalle qué sucede con las distintas etapas del modelo (la resolución del CDLP, el subproblema de generación de columnas, la heurística de descomposición, etc.) al calcular las probabilidades basándonos en el MNL con segmentos superpuestos.

2.7. Asunción de demanda independiente

Para poder evaluar el beneficio de incorporar explícitamente el comportamiento selectivo de los clientes, consideraremos también el comportamiento el modelo de demanda independiente (ver Sección 1.6) para poder hacer una comparación cualitativa. Este método consiste en resolver primero el problema de programación lineal determinístico (DLP), utilizado tradicionalmente en la literatura de *Airline Revenue Management*. La formulación del DLP es

$$\begin{aligned} \max \quad & rz \\ \text{s.t} \quad & Az \leq x \\ & 0 \leq z \leq E[D]. \end{aligned} \tag{2.18}$$

El vector z corresponde a las variables de decisión, y representan la cantidad de asientos (fraccionarios) que se ofrecerán de cada producto, y $E[D] \in \mathbb{R}^n$ es el vector de demandas esperadas D_j de cada producto j. Bajo las hipótesis del modelo de demanda independiente, asumimos que todos los productos se ofrecen simultáneamente, con lo cual la esperanza de la demanda del producto j es $\lambda \times P_j(N) \times T$. Luego utilizamos las variables duales correspondientes a las restricciones de capacidad en un esquema de descomposición por programación dinámica como se describe en Talluri y van Ryzin [8, Capítulo 3], que es equivalente al algoritmo de descomposición visto en la sección anterior, pero reemplazando $P_j(S)$ por $P_j(N)$, para todo j y S.

Capítulo 3

CDLP bajo MNL con Segmentos Superpuestos

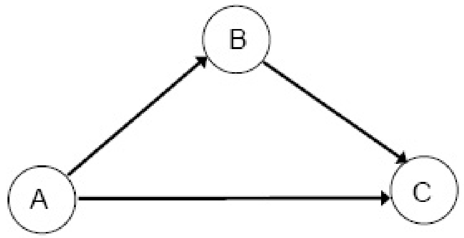
El Capítulo 2 explica en detalle el CDLP (Gallego et al. [4]) y la heurística de descomposición (van Ryzin y Liu [10]). Además, estas aproximaciones se encuentran especializadas en el caso en que la elección de productos por parte de los clientes se realiza en base al modelo MNL. Sin embargo, la dificultad en la resolución de los distintos problemas que aparecen en ambas aproximaciones puede variar dependiendo de las hipótesis que se le impongan al MNL.

Como vimos anteriormente, nosotros consideramos como *choice model* el MNL con segmentos superpuestos. Esta generalización del caso estudiado por van Ryzin y Liu [10] influye significativamente en distintos aspectos del modelo. El objetivo de este capítulo es analizar el impacto de quitar esta restricción sobre los segmentos del MNL.

El capítulo se encuentra organizado de la siguiente forma. Para ilustrar el modelo en el caso del MNL con segmentos superpuestos, en la Seccion 3.1 mostramos una red pequeña a modo de ejemplo. La Sección 3.2 analiza en detalle el subproblema de generación de columnas definido en el capítulo anterior, incluyendo propiedades y un análisis de complejidad del mismo. La siguiente sección muestra dos métodos para resolverlo, uno heurístico y otro exacto. El algoritmo de generación de columnas que resuelve el CDLP utiliza ambos métodos, según se explica en la Sección 3.4. La Sección 3.5 analiza cómo resolver las dos etapas de la heurística de descomposición en el caso particular del MNL con segmentos superpuestos. Finalmente, en la Sección 3.6 se propone un método alternativo para el cálculo de los *offer set* que intenta evitar la carga computacional de la primera etapa de la heurística de descomposición.

3.1. Ejemplo

Consideramos una red pequeña con tres ciudades (A, B y C) y tres tramos (A $\rightarrow$ B, A $\rightarrow$ C y B $\rightarrow$ C). Cada tramo ofrece tarifas altas (high) y bajas (low) (H y L respectivamente). Hay un total de ocho productos. La Figura 3.1 muestra la definición de la red, y lista las definiciones de los correspondientes productos. Las capacidades de los tramos son $x = (10, 5, 5)$, y el horizonte de reservas consiste en $T = 30$ períodos.



Red para el Ejemplo

Producto	Origen-Destino	Clase	Tarifa
1	A $\rightarrow$ C	H	1200
2	A $\rightarrow$ B $\rightarrow$ C	H	800
3	A $\rightarrow$ B	H	500
4	B $\rightarrow$ C	H	500
5	A $\rightarrow$ C	L	800
6	A $\rightarrow$ B $\rightarrow$ C	L	500
7	A $\rightarrow$ B	L	300
8	B $\rightarrow$ C	L	300

Figura 3.1: Red pequeña para ilustrar el modelo.

Dividimos a los potenciales clientes en cinco segmentos distintos. La definición de los mismos puede verse en el Cuadro 3.1. La primera columna de la tabla identifica los distintos segmentos; la segunda columna describe la probabilidad de tener un arribo para cada segmento en particular; la columna 3 muestra el *consideration set* correspondiente a cada segmento; la cuarta columna especifica los valores de preferencia para cada producto considerado en el segmento (incluída la preferencia por la no-compra en la última coordenada); finalmente, la columna 5 incluye una breve descripción del segmento. Por cuestiones de simplicidad, estamos asumiendo que $\lambda = 1$ (es decir, en todos los instantes de tiempo $t = 1, \ldots, 30$ se produce el arribo de un cliente).

Segmento	p_l	Consideration Set	Vector de Preferencia	Breve Descripción
1	0.15	$\{1, 5\}$	(5,8,2)	Sensibles al precio, vuelo directo (A $\rightarrow$ C)
2	0.15	$\{1, 2\}$	(10,6,5)	Insensibles al precio (A $\rightarrow$ C)
3	0.20	$\{5, 6\}$	(8,5,2)	Sensibles al precio (A $\rightarrow$ C)
4	0.25	$\{3, 7\}$	(4,8,2)	Sensibles al precio (A $\rightarrow$ B)
5	0.25	$\{4, 8\}$	(6,8,2)	Sensibles al precio (B $\rightarrow$ C)

Cuadro 3.1: Definición de los segmentos para el Ejemplo.

Los primeros tres segmentos consideran clientes que desean volar desde A hasta C, pero con diferentes características. El primer segmento considera a aquellos que son sensibles al precio y desean hacer un vuelo sin escalas. El segundo segmento consiste en clientes que no son sensibles al precio (por ejemplo, aquellos que viajan por negocios), mientras que aquellos que son sensibles al precio y prefieren hacer un vuelo sin escalas pertenecen al segmento 3. El segmento 4 incluye a clientes que vuelan desde A hasta B, y finalmente el quinto segmento considera a los clientes sensibles al precio que viajan desde B a C. Notar que entre los segmentos 1 y 2, así como también entre el 1 y el 3, hay superposición de productos.

Este ejemplo posee $2^8 - 1 = 255$ *offer sets* posibles. Dado que hay cuatro restricciones en la formulación del CDLP (tres de ellas correspondientes a las restricciones de capacidad sobre los tramos, más una relacionada con el tiempo disponible), a lo sumo hay cuatro *offer sets* cuyas variables de tiempo asociadas tienen valores positivos. En este caso, una solución óptima considera a los conjuntos $S_1 = \{1, 2, 3, 4, 6\}$, $S_2 = \{2, 3\}$ y $S_3 = \{1, 2, 3\}$, donde $t(S_1) = 11{,}0074$, $t(S_2) = 2$ y $t(S_3) = 16{,}9926$. Ésto nos está diciendo que los productos 2 y 3 se ofrecen durante todo el horizonte de reservas, el producto 1 se ofrece durante 28 períodos, los productos 4 y 6 se ofrecen en 11,0074 períodos, y los productos 7 y 8 (aquellos que tienen las tarifas más bajas) no se ofrecen nunca. La dificultad que presentan este tipo de soluciones radica en que para los productos 1, 4 y 6 no sabemos específicamente en que períodos debemos ofrecerlos. La heurística de descomposición vista en (2.6) intenta quitar esta ambigüedad que presentan las soluciones primales del CDLP.

3.2. Subproblema de generación de columnas

Debido al número exponencial de variables que posee el CDLP (2.8), utilizamos la técnica de generación de columnas para resolverlo. Como se menciona en la Sección 2.5, comenzamos con un conjunto limitado y manejable de variables $\mathcal{N}$. Resolvemos el CDLP restringiéndonos a este conjunto y analizamos los costos reducidos para ver si existe algun conjunto $S \notin \mathcal{N}$ con costos reducidos positivos. Este análisis, dado que las probabilidades se calculan de acuerdo al modelo MNL, determina que el subproblema de generación de columnas (2.11) es un problema de programación hiperbólica 0-1.

La eficiencia con la que podamos resolver el problema (2.11) tiene un papel fundamental en la resolución del CDLP. La dificultad del problema (2.11) está directamente relacionada con las hipótesis establecidas sobre los distintos *consideration sets*. Por ejemplo, van Ryzin y Liu [10] muestran que en el caso de que los segmentos sean disjuntos, el problema (2.11) puede ser resuelto con un algoritmo polinomial. Sin embargo, al quitar esta restricción, no podemos asegurar lo mismo.

En esta sección nos ocuparemos de analizar distintas características del subproblema de generación de columnas cuando el *choice model* es el MNL con segmentos superpuestos. Primero veremos dos propiedades relacionadas con el valor de determinadas variables en la solución óptima y luego pasaremos al análisis de complejidad.

3.2.1. Propiedades del subproblema de generación de columnas

Con el fin de obtener una descripción más precisa del subproblema de generación de columnas, en esta sección veremos dos resultados que nos permitirán saber de antemano para ciertas variables y_j, cuál es el valor que tendrán en la solución óptima, en función de las características de ciertos parámetros del problema. Recordemos que la formulación del problema (2.11) es

$$\max_{y \in \{0,1\}^n} \left\{ \sum_{l=1}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} y_j}{\sum_{j \in C_l} v_{lj} y_j + v_{l0}} \right\}. \tag{3.1}$$

El primer resultado está relacionado con el costo de oportunidad de un producto. Si un determinado producto k cumple que $(r_k - A_k^\mathsf{T}\pi)$ es menor o igual que cero, entonces $y_k = 0$ en la solución óptima. Esta propiedad nos está diciendo que si el ingreso instantáneo, r_k, no supera la suma del ingreso estimado generado por una unidad de capacidad correspondiente a los tramos que utiliza el producto j, entonces no es conveniente considerar ese producto en el conjunto $S \subseteq N$ que maximice el problema (2.11).

Proposición 1 *Sea $k \in N$ tal que $(r_k - A_k^\mathsf{T}\pi) \leq 0$ y sea y^* el óptimo del problema (2.11). Entonces, $y_k^* = 0$.*

Dem.: La demostración se basa en comparar que sucede en cada caso. Intuitivamente, lo que queremos ver es que si el costo de oportunidad del producto k es menor o igual que cero, entonces los numeradores de los sumandos correspondientes a segmentos a los cuales el producto k pertenece se ven decrementados, mientras que con los denominadores sucede exactamente lo contrario. Dado que $(r_k - A_k^\mathsf{T}\pi) \leq 0$, entonces para todo l tal que $k \in C_l$ vale que $(r_k - A_k^\mathsf{T}\pi)v_{lk} \leq 0$, ya que $v_{lk} > 0$ si k pertenece al segmento l. Luego, analizando los numeradores de los sumandos correspondientes a los segmentos que contienen al producto k, tenemos que

$$\sum_{j\in C_l, j\neq k} (r_j - A_j^\mathsf{T}\pi)v_{lj}y_j + (r_k - A_k^\mathsf{T}\pi)v_{lk} \quad \leq \quad \sum_{j\in C_l, j\neq k} (r_j - A_j^\mathsf{T}\pi)v_{lj}y_j \quad \forall l \ / \ k \in C_l.$$

En el caso de los denominadores, sucede lo contrario. Como $v_{lk} > 0$, podemos ver que los denominadores de estos segmentos cumplen que al considerar el producto k, éstos son estrictamente mayores que al no considerarlo. Es decir,

$$\sum_{j\in C_l, j\neq k} v_{lj}y_j + v_{lk} + v_{l0} \quad > \quad \sum_{j\in C_l, j\neq k} v_{lj}y_j + v_{l0} \quad \forall l \ / \ k \in C_l.$$

Considerando el cociente en cada uno de los casos, podemos ver que los segmentos l tales que $k \in C_l$ tienen un valor estrictamente menor asignando $y_k = 1$ que tomando $y_k = 0$.

$$\frac{\sum_{j\in C_l, j\neq k}(r_j - A_j^\mathsf{T}\pi)v_{lj}y_j + (r_k - A_k^\mathsf{T}\pi)v_{lk}}{\sum_{j\in C_l, j\neq k} v_{lj}y_j + v_{lk} + v_{l0}} \quad < \quad \frac{\sum_{j\in C_l, j\neq k}(r_j - A_j^\mathsf{T}\pi)v_{lj}y_j}{\sum_{j\in C_l, j\neq k} v_{lj}y_j + v_{l0}} \quad \forall l \ / \ k \in C_l$$

Esta desigualdad se cumple independientemente del valor que tengan asignado el resto de las variables. Finalmente, sumando sobre los segmentos $l = 1, \ldots, L$, el valor de la función objetivo del problema (2.11) es estrictamente menor si y_k toma en el valor uno en lugar de cero. $\square$

La segunda propiedad tambien está relacionada con el valor del costo de oportunidad, pero no es tan general como la primera. Si un determinado producto j pertenece a un único segmento k, y además de que el costo de oportunidad sea positivo es el máximo entre los productos del segmento, entonces el producto j estará presente en el conjunto S que maximice el problema (2.11).

Proposición 2 *Sean $j \in N$, $1 \leq k, k' \leq L$ tales que $j \in C_k$, $j \notin C_{k'}$ si $k' \neq k$. Supongamos además que $(r_j - A_j^\mathsf{T}\pi) > 0$ y $(r_j - A_j^\mathsf{T}\pi) \geq (r_i - A_i^\mathsf{T}\pi)$ $\forall i \in C_k$. Si y^* es el óptimo del problema (2.11), entonces $y_j^* = 1$.*

Dem.: La demostración la hacemos por el absurdo. Sea y^* el óptimo de (2.11), y supongamos que $y_j^* = 0$. Definimos $\bar{y}$ solución factible, donde $\bar{y}_i = y_i^*\ \forall i \neq j$, e $\bar{y}_j = 1$. Dado que y^* es óptimo, sabemos que el valor de la función objetivo en y^* es mayor o igual que el resultado de considerar cualquier otra solución, en particular $\bar{y}$. Es decir,

$$\sum_{l=1}^{L} \lambda_l \frac{\sum_{i\in C_l}(r_i - A_i^\intercal \pi)v_{li}y_i^*}{\sum_{i\in C_l} v_{lj}y_j^* + v_{l0}} \quad \geq \quad \sum_{l=1}^{L} \lambda_l \frac{\sum_{i\in C_l}(r_i - A_i^\intercal \pi)v_{li}\bar{y}_i}{\sum_{i\in C_l} v_{li}\bar{y}_i + v_{l0}}.$$

Como el producto j solamente se encuentra presente en el conjunto C_k, y además por definición $y_i^* = \bar{y}_i\ \forall i \neq j$, los sumandos correspondientes a los segmentos $k' \neq k$ se cancelan. La desigualdad anterior entonces queda como

$$\frac{\sum_{i\in C_k}(r_i - A_i^\intercal \pi)v_{ki}y_i^*}{\sum_{i\in C_k} v_{ki}y_i^* + v_{k0}} \quad \geq \quad \frac{\sum_{i\in C_k}(r_i - A_i^\intercal \pi)v_{ki}\bar{y}_i}{\sum_{i\in C_k} v_{ki}\bar{y}_i + v_{k0}}.$$

En el caso en particular del segmento k, por definición sabemos que $y_j^* = 0$ e $\bar{y}_j = 1$, sumado a que para los restantes productos $i \neq j$, $\bar{y}_i = y_i^*$. Reemplazamos las variables $\bar{y}_j$ e y_j^* por su valor en el sumando correspondiente, y $\bar{y}_i$ por y_i^* en el sumando del lado derecho de la desigualdad.

$$\frac{\sum_{i\in C_k, i\neq j}(r_i - A_i^\intercal \pi)v_{ki}y_i^*}{\sum_{i\in C_k, i\neq j} v_{ki}y_i^* + v_{k0}} \quad \geq \quad \frac{(r_j - A_j^\intercal \pi)v_{kj} + \sum_{i\in C_k, i\neq j}(r_i - A_i^\intercal \pi)v_{ki}y_i^*}{v_{kj} + \sum_{i\in C_k, i\neq j} v_{ki}y_i^* + v_{k0}} \tag{3.2}$$

Definimos $A = \sum_{i\in C_k, i\neq j}(r_i - A_i^\intercal \pi)v_{ki}y_i^*$ y $B = \sum_{i\in C_k, i\neq j} v_{ki}y_i^* + v_{k0}$ y reemplazamos estas expresiones en (3.2). Como se muestra a continuación, despejando A y B llegamos a la expresión (3.3).

$$\begin{aligned}
\frac{A}{B} \quad &\geq \quad \frac{(r_j - A_j^\intercal \pi)v_{kj} + A}{v_{kj} + B} \iff \\
v_{kj}A + AB \quad &\geq \quad (r_j - A_j^\intercal \pi)v_{kj}B + AB \iff \\
A \quad &\geq \quad (r_j - A_j^\intercal \pi)B \iff \\
\sum_{i\in C_k, i\neq j}(r_i - A_i^\intercal \pi)v_{ki}y_i^* \quad &\geq \quad (r_j - A_j^\intercal \pi)\Bigg(\sum_{i\in C_k, i\neq j} v_{ki}y_i^* + v_{k0}\Bigg) \qquad (3.3)
\end{aligned}$$

Sin embargo, sabemos por hipótesis que $(r_j - A_j^\intercal \pi)$ es positivo y máximo entre los costos de oportunidad de los productos del segmento k. Además por definición $v_{k0} > 0$. Luego, la desigualdad (3.3) es un absurdo. Este provino de suponer que $y_j^* = 0$. $\square$

Estas propiedades resultan interesantes por dos razones. La primera de ellas es que podemos, en función de los parámetros del problema, identificar aquellos productos para los cuales conocemos cuál será su valor en la solución óptima del problema (2.11). Por otro lado, tenemos una mejor descripción del subproblema de generación de columnas. En particular, asumimos de aquí en adelante que los productos que aparecen en este problema tienen costos de oportunidad estrictamente positivos. Ésto último nos será útil para realizar el análisis de complejidad del mismo.

3.2.2. Complejidad del subproblema de generación de columnas

El problema (2.11) es un caso particular de programación hyperbólica 0-1. La formulación general para este tipo de problemas es

$$\max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \frac{a_{l0} + \sum_{j=1}^{n} a_{lj} y_j}{b_{l0} + \sum_{j=1}^{n} b_{lj} y_j} \tag{3.4}$$

donde $a_{lj}, b_{lj} \in \mathbb{R}$, $j = 0, \ldots, n$, $l = 1, \ldots, L$. Este problema es conocido por ser NP-Hard, incluso cuando todos los coeficientes son positivos (Prokopyev et al. [6]). Sin embargo, las variables del problema (2.11) están relacionadas con menor grado de libertad de manera tal que el problema (3.4) no puede ser reducido a el mismo, aún para valores de $L = 2$, $n \geq 2$.

A diferencia del problema estudiado por van Ryzin y Liu [10], donde los *consideration sets* son disjuntos, nuestro problema no es separable en las variables y_j dado que cada producto j puede pertenecer simultáneamente a dos o más *consideration sets*. Luego, y_j aparecería en múltiples denomidadores para los distintos segmentos l a los que pertence. Bajo la hipótesis de segmentos disjuntos, al hacer la separación en función de las variables y_j, el problema (2.11) puede ser resuelto con un algoritmo polinomial (van Ryzin y Liu [10]). Veamos que éste no es el caso cuando los segmentos no son disjuntos.

El siguiente teorema establece que el problema *recubrimiento mínimo de aristas*, conocido por ser NP-Hard (Garey y Johnson [5, problema GT1]), puede ser reducido a (2.11), con lo cual nuestro problema también es NP-Hard.

Teorema 1 *El problema de programación hyperbólica 0-1*

$$\max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \frac{\sum_{j\in C_l} w_j v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} \tag{3.5}$$

donde w_j, $v_{lj} > 0$, $v_{l0} > 0$, $l = 1, \ldots, L$, $j \in C_l$, $C_l \subseteq N$, *es NP-Hard.*

Dem.: Dado un grafo conexo $G = (V, E)$, con vértices $V = \{1, \ldots, v\}$, $v \geq 2$, aristas $E \subseteq \{(i, j) \in V \times V, i < j\}$, $|E| = m$, un recubrimiento mínimo de aristas para G es un subconjunto $V' \subseteq V$ de cardinal mínimo tal que todo arco en E es incidente en al menos un nodo en V'.

Sea I una instancia de recubrimiento mínimo de aristas. Construiremos una instancia J del problema (3.5) correspondiente a I. Consideremos $v+1$ variables de decisión binarias, $y_1, \ldots, y_{v+1}$, y $v + m$ sumandos en la función objetivo. Los primeros v sumandos son

$$\frac{2y_{v+1} + y_1}{y_{v+1} + y_1 + 1/v^2} + \frac{2y_{v+1} + y_2}{y_{v+1} + y_2 + 1/v^2} + \cdots + \frac{2y_{v+1} + y_v}{y_{v+1} + y_n + 1/v^2}$$

y los últimos m sumandos son de la forma

$$\frac{y_i + y_j}{y_i + y_j + 1/v^2}, \text{ para todo } (i, j) \in E$$

Notar que podemos obtener la formulación (3.5) tomando $n = v + 1$, $w_{v+1} = 2$, $w_j = 1$, $1 \leq j \leq v$, y definiendo $L = v + m$, $C_i = \{v + 1, i\}$, $i = 1, \ldots, v$, y $C_{v+j} = \{i, j\}$, $j = 1, \ldots, m$, donde las aristas están etiquetados según el orden lexicográfico. Para $l = 1, \ldots, L$, también definimos $\lambda_l = 1$, $v_{l0} = 1/v^2$, y $v_{lj} = \mathbb{1}\{j \in V \cap C_l\}$.

Esta formulación cumple con las siguientes propiedades:

1. Una solución óptima de J tiene que verificar que $y_{v+1} = 1$, dado que para todo $i \in V$,

$$\frac{2 + y_i}{1 + y_i + 1/v^2} > \frac{y_i}{y_i + 1/v^2}, \forall v \geq 1$$

2. Para cada eje $(i_0, j_0) \in E$, en la solución óptima de J no podemos tener $y_{i_0} = y_{j_0} = 0$. Para justificar esto, supongamos que S es una solución óptima de J, con $y_{i_0} = y_{j_0} = 0$. Definimos S' igual que S, pero con la diferencia de que $y_{i_0} = 1$. Los términos involucrados en la expresión de la función objetivo serían

$$\frac{2y_{v+1} + y_{i_0}}{y_{v+1} + y_{i_0} + 1/v^2} \quad \text{y} \quad \frac{y_{i_0} + y_j}{y_{i_0} + y_j + 1/v^2} \text{ para todo } j \text{ tal que } (i_0, j) \in E.$$

   En el primer caso, tomando $y_{i_0} = 1$, estamos decrementando el valor de la función objetivo en la solución S' respecto del valor de la función objetivo en S en menos que $1/2$ si $v \geq 1$

$$\frac{3v^2}{2v^2 + 1} - \frac{2v^2}{v^2 + 1} = -\frac{v^4 - v^2}{2v^4 + 3v^2 + 1} > -\frac{1}{2}$$

   Sin embargo, en el segundo caso estamos incrementando el valor de la solución S' en al menos $1/2$ (si $v \geq 1$) por cada arista en E tal que es incidente a i_0

$$\frac{1}{1 + 1/v^2} \geq \frac{1}{2}$$

   Luego, asignando $y_{i_0} = 1$, tenemos un incremento en el valor de la función objetivo, con lo cual el valor de S' es mayor que el de S. Absurdo. Entonces, todo eje en E es incidente a al menos un vértice en V en la solución óptima.

3. Nos queda probar que la solución óptima de J tiene el menor número posible de variables asignadas en uno. Nuevamente, sean S y S' dos soluciones óptimas de J, pero con la diferencia de que S' tiene menos variables asignadas en uno que S. Definimos $Value_1(S)$ y $Value_2(S)$ como el valor de la función objetivo del problema (3.5) correspondiente a los primeros v y los últimos $|E|$ términos respectivamente, considerando la asignación de las variables y en S.

   Comparando $Value_1(S)$ con $Value_1(S')$ obtenemos

$$\begin{aligned} Value_1(S) - Value_1(S') &= |S|\frac{3}{2 + 1/v^2} + (n - |S|)\frac{2}{1 + 1/v^2} - |S'|\frac{3}{2 + 1/v^2} - (n - |S'|)\frac{2}{1 + 1/v^2} \\ &= (|S'| - |S|)\frac{v^4 - v^2}{2v^4 + 3v^2 + 1} \\ &< -\frac{1}{4}, \quad \forall v \geq 2 \end{aligned}$$

Considerando los últimos $|E|$ términos, la diferencia entre el valor del sumando $(k, j) \in S$ y el valor del $(k, j) \in S'$ es a lo sumo

$$\frac{2}{2 + 1/v^2} - \frac{1}{1 + 1/v^2} = \frac{2v^2}{2v^2 + 1} - \frac{v^2}{v^2 + 1} = \frac{v^2}{2v^4 + 3v^2 + 1}$$

Dado que el número de aristas verifica que $|E| \leq n(n-1)/2$, entonces

$$\begin{aligned} Value_2(S) - Value_2(S') &\leq \frac{v(v-1)}{2} \frac{v^2}{2v^4 + 3v^2 + 1} \\ &= \frac{v^4 - v^3}{4v^4 + 6v^2 + 2} \\ &< \frac{1}{4} \end{aligned}$$

Luego, $Value_1(S') + Value_2(S') > Value_1(S) + Value_2(S)$, que contradice el hecho de que S es una solución óptima de la instancia J.

Finalmente, hemos encontrado una transformación polinomial de una instancia I de recubrimiento mínimo de aristas a una instancia J del problema (3.5) tal que una solución de J implica una solución de I. □

El siguiente ejemplo muestra como realizar la transformación polinomial de una instancia del problema de recubrimiento mínimo de aristas al problema (2.11).

Ejemplo: Consideremos el grafo $G = (V, E)$ de la Figura 3.2. En este caso, $v = 4$ y $m = 4$. En la instancia correspondiente del problema de programación hiperbólica 0-1 definida en (3.5) hay 5 variables y 8 sumandos definidos de la siguiente forma

- Los primeros $v = 4$ sumandos son

$$\frac{2y_5 + y_1}{y_5 + y_1 + 1/16} + \frac{2y_5 + y_2}{y_5 + y_2 + 1/16} + \frac{2y_5 + y_3}{y_5 + y_3 + 1/16} + \frac{2y_5 + y_4}{y_5 + y_4 + 1/16}$$

- Los últimos $m = 4$ sumandos son

$$\frac{y_1 + y_2}{y_1 + y_2 + 1/16} + \frac{y_1 + y_3}{y_1 + y_3 + 1/16} + \frac{y_2 + y_3}{y_2 + y_3 + 1/16} + \frac{y_3 + y_4}{y_3 + y_4 + 1/16}$$

Una posible instancia de recubrimiento mínimo de aristas está dada por los vértices 2 y 3. Ésta no es la única solución. El subconjunto de vertices $\{1, 3\}$ también es un óptimo alternativo.

Maximizando la suma total respecto de las variables $y_1, \ldots, y_5 \in \{0, 1\}^n$ obtenemos 3 como valor de la función objetivo. La solución que muestra la Figura 3.2 corresponde a la solución $y_5 = y_2 = y_3 = 1$, e $y_1 = y_4 = 0$. □

Ahora podemos enunciar el siguiente corolario.

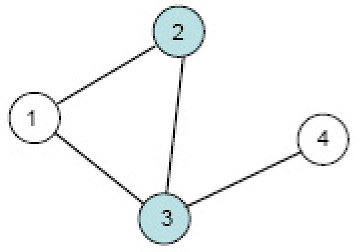


Figura 3.2: Instancia de recubrimiento mínimo de aristas para el grafo $G = (V, E)$.

Corolario 1 *El problema de decisión asociado al problema de optimización (3.5), es decir, "Dada una constante $K > 0$, ¿existe una asignación de variables $y \in \{0,1\}^n$ tales que*

$$\sum_{l=1}^{L} \frac{\sum_{j\in C_l} w_j v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} > K?'' \tag{3.6}$$

es NP-Completo.

Dem.: El problema de decisión (3.6) es NP, ya que dada una instancia de SI, claramente lleva tiempo polinomial verificar que la desigualdad se satisface. Su pertenencia a la clase NP-Hard vale por el Teorema 1. □

El Teorema 1 determina que el problema (3.5) pertenece a la clase NP-Hard. De la demostración podemos ver que, para poder hacer la reducción polinomial de recubrimiento mínimo de aristas, los parámetros importantes son los w_j, los v_{l0} y, fundamentalmente, la intersección entre los conjuntos C_l. El siguiente resultado establece que fijando en una constante el valor de los w_j, el problema (3.5) tiene una solución trivial.

Proposición 3 *Si $w_j = c > 0$, $j = 1, \ldots, n$ y $\lambda_l > 0$, $l = 1, \ldots, L$, entonces el problema (3.5) tiene un óptimo global en $y = (1, \ldots, 1)$.*

Dem.: Consideramos el problema (3.5). Dado que $w_j = c > 0\ \forall\ j$, podemos ver que

$$\begin{aligned} \max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \lambda_l \frac{\sum_{j\in C_l} w_j v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} &= c \max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \lambda_l \frac{\sum_{j\in C_l} v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} \\ c \max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \lambda_l \frac{\sum_{j\in C_l} v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} &\leq c \sum_{l=1}^{L} \lambda_l \max_{y\in\{0,1\}^n} \frac{\sum_{j\in C_l} v_{lj} y_j}{\sum_{j\in C_l} v_{lj} y_j + v_{l0}} \end{aligned}$$

Tomamos $y^* = (1, \ldots, 1)$, una posible solución al problema. Ambos lados de la desigualdad tienen el mismo valor al ser evaluados en y^*. Además, puede verse fácilmente que y^* es una solución óptima del lado derecho, con lo cual también lo es en el problema (3.5). □

Siguiendo con este razonamiento, si $w_j = c \leq 0$, $j = 1, \ldots, n$, entonces por la Proposición 1 podemos deducir que las variables y_j valen todas cero en el óptimo. Luego, $y = (0, \ldots, 0)$ maximiza el problema (3.5).

Desde el punto de vista del análisis de complejiad, estas observaciones nos están mostrando que la característica de los parámetros w_j modifica la clase de complejidad a la que pertence el problema (3.5). El Cuadro 3.2 resume a qué clase de complejidad pertenece el problema considerando las distintas hipótesis establecidas sobre los parámetros de entrada.

	w_j constante	w_j variable
Segmentos Disjuntos	P	P
Segmentos Superpuestos	P	NP-Hard

Cuadro 3.2: Complejidad del problema (3.5) en función de los parámetros de entrada.

Notar que la equivalencia entre la formulación (3.5) y nuestro subproblema de generación de columnas (2.11) radica en el hecho de tomar $w_j = (r_j - A_j^\intercal \pi)$, y en que por la Proposición 1, $y_j = 0$ si $w_j \leq 0$.

3.3. Resolución del subproblema de generación de columnas

Durante la ejecución del algoritmo de generación de columnas, necesitamos resolver de forma eficiente una instancia del problema (2.11) en cada iteración para verificar si nos encontramos en una solución óptima. Dado que el subproblema de generación de columnas es NP-Hard, consideramos dos alternativas distintas para resolverlo: una heurística golosa, y una linealización del problema de programación hipérbolica. Esta última nos permite obtener una solución exacta al problema (2.11), aunque a un costo computacionalmente mayor que el de la heurística.

3.3.1. Formulación lineal entera mixta

El problema de programación fraccionaria (2.11) puede ser reformulado como uno de programación lineal entera mixta (MIP) equivalente (Wu [13]). Definiendo las variables

$$x_l = \frac{1}{\sum_{i \in C_l} v_{li} y_i + v_{l0}} \tag{3.7}$$

el subproblema de generación de columnas (2.11) puede reescribirse como

$$
\begin{array}{rrcl}
\text{máx} \displaystyle\sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^{\mathsf{T}} \pi) v_{lj} y_j x_l & & & \\
\text{s.t.:} & x_l v_{l0} + \displaystyle\sum_{i \in C_l} v_{li} y_i x_l & = & 1, \forall l \\
& y_j \in \{0,1\}, & & x_l \geq 0
\end{array} \tag{3.8}
$$

Los términos no lineales $y_i x_l$ pueden ser linealizados (Wu, [13]), obteniendo una formulación lineal entera mixta. La linealización se basa en el siguiente teorema

Teorema 2 (Wu) *Un término polinomial $z = xy$, donde x es una variable continua, e y es una variable 0-1, puede ser representado por las siguientes desigualdades: (1) $x - z \leq K - Ky$; (2) $z \leq x$; (3) $z \leq Ky$; (4) $z \geq 0$, donde K es un número grande mayor que x.*

En el contexto de nuestro problema, definiendo las variables $z_{li} = y_i x_l$, y aplicando este resultado, obtenemos la siguiente formulación lineal entera mixta

$$
\begin{array}{rrcl}
\text{máx} \displaystyle\sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^{\mathsf{T}} \pi) v_{lj} z_{lj} & & & \\
\text{s.t.:} & x_l v_{l0} + \displaystyle\sum_{i \in C_l} v_{li} z_{li} & = & 1, \forall l \\
& x_l - z_{li} & \leq & K - K y_i \ , \forall l, \, i \in C_l \\
& z_{li} & \leq & x_l \ , \forall l, \, i \in C_l \\
& z_{li} & \leq & K y_i \ , \forall l, \, i \in C_l \\
& y_j \in \{0,1\}, \ x_l \geq 0, & & z_{li} \geq 0
\end{array}
$$

En este nuevo modelo es necesario determinar el valor de la constante K. Teniendo en cuenta las hipótesis del teorema de Wu, es necesario que $K > x_l$, $l = 1, \ldots, L$. Para ésto hay dos opciones: la primera de ellas es determinar un valor genérico de K de manera tal que

$$
K > \max_{l} \{x_l\}
$$

En la segunda, para cada segmento buscamos constantes $K_1, \ldots, K_L$ de forma tal que se satisfaga la condición $K_l > x_l$, $l = 1, \ldots, L$, y utilizarlas en las respectivas desigualdades. En función del valor de K, teniendo en cuenta que los valores $v_{li} \geq 0$ y $v_{l0} > 0$, $l = 1, \ldots, L$, definimos

$$
\hat{x}_l = \frac{1}{v_{l0}}
$$

El valor de $\hat{x}_l$ representa una cota superior para x_l. Volviendo a las posibles definiciones de K, en el primer caso podemos tomar $K = \text{máx}_l \{\hat{x}_l\}$ y en el segundo $K_l = \hat{x}_l$.

El modelo resultante tiene una variable $y_j \in \{0,1\}$ por cada producto $j \in N$, una variable continua x_l por cada segmento, y una variable z_{li} si el producto i pertenece al segmento l. Luego, la cantidad total de variables es $n + L + \sum_{l=1}^{L} |C_l|$. En función de la cantidad de desigualdades, podemos ver que el primer conjunto de restricciones define una por cada segmento. Cada uno de los restantes conjuntos de desigualdades determina una restricción por cada variable z_{li} definida. Por lo tanto, la cantidad total de restricciones es $L + 3\sum_{l=1}^{L} |C_l|$, sin contar las correspondientes a cotas inferiores de las variables z_{li} y x_l.

La principal ventaja de esta reformulación es que se puede utilizar cualquier paquete de software de optimización para obtener una solución óptima al problema (2.11). De hecho, en los casos utilizados como pruebas de la linealización, las rutinas estandar ofrecidas por el paquete de optimización CPLEX 8.1 resuelven el problema en un tiempo razonable.

3.3.2. Heurística Golosa

Si bien los tiempos obtenidos en la resolución del problema (2.11) utilizando la linealización son buenos, implementamos una heurística golosa con el fin de obtener soluciones de buena calidad en tiempos aún menores que los de la linealización. La heurística comienza con un conjunto vacío S y le va agregando en forma progresiva productos que maximicen el incremento parcial a la solución actual.[1] El pseudocódigo de la heurística es el siguiente

1. Para aquellos productos j tales que $r_j - A_j^\mathsf{T}\pi \leq 0$, asignar $y_j = 0$.

2. Sea $S' \subseteq N$ el conjunto de productos que aún no tienen un valor asignado en y_j.

3. **Repetir**

   a) Calcular $j^* = \arg_{j \in S'} \left\{ \sum_{l=1}^{L} \lambda_l \frac{\sum_{i \in C_l \cap (S \cup \{j\})} (r_i - A_i^\mathsf{T}\pi) v_{li}}{\sum_{i \in C_l \cap (S \cup \{j\})} v_{li} + v_{l0}} \right\}$

   b) Si Value$(S \cup \{j^*\}) >$ Value(S), entonces $S = S \cup \{j^*\}$ y $S' = S' - \{j^*\}$.

   mientras S se modifique.

4. Para todo $j \in S$, asignar $y_j = 1$. Si $j \notin S$, asignar $y_j = 0$.

En caso de considerar el conjunto de índices asignados S como un parámetro de entrada, esta heurística puede ser utilizada para refinar una solución encontrada previamente.

Para los casos de prueba considerados, pudimos observar que esta heurística golosa encuentra rápidamente soluciones de muy buena calidad. De hecho, en la mayoría de los casos encuentra una solución óptima. Sin embargo, a continuación mostramos un ejemplo donde la heurística es subóptima.

[1] Esta heurística es utilizada en Prokopyev et al. [7] como fase construciva de un GRASP para resolver problemas generales de programación hiperbólica.

Ejemplo: Consideremos la siguiente instancia del problema (3.5), con tres productos y tres segmentos. Definimos los *consideration sets* $C_1 = \{1, 2, 3\}$, $C_2 = \{2\}$ y $C_3 = \{3\}$. Tomamos $\lambda_l = 1$, $v_{l0} = 1$ y $v_{lj} = 1$, $l = 1, \ldots, L$, $j \in C_l$. Definimos $w_1 = 100$ y $w_2 = w_3 = 19$. El problema de programación hiperbólica es

$$\frac{100y_1 + 19y_2 + 19y_3}{y_1 + y_2 + y_3 + 1} + \frac{19y_2}{y_2 + 1} + \frac{19y_3}{y_3 + 1}$$

La solución encontrada por la heurística es $y_1 = 1$, $y_2 = y_3 = 0$ y el valor de la función objetivo es 50. Sin embargo, tomando $y_1 = y_2 = y_3 = 1$ la función objetivo vale 53,5. $\square$

En el peor caso, la complejidad de la heurística es $O(n^2 L)$, donde en la práctica L es en general un número significativamente más chico que n. Considerando este hecho, y analizando el comportamiento de la heurística, es posible realizar algunas mejoras en la implementación con el fin de acelerar el tiempo de ejecución de la misma.

Implementación

En cada iteración del algoritmo, la heurística asigna el valor 1 a la variable que genera mayor incremento en la función objetivo. Para ello, es necesario determinar en cada paso el incremento parcial respecto de la solución encontrada hasta el momento. En principio, se podría evaluar para cada producto en cada iteración la función objetivo y determinar cuál es el producto que será elegido. Si bien esta solución es válida, el costo computacional que genera es relativamente alto, además de no explotar algunas características particulares del algoritmo y del problema.

Supongamos que y^* es la solución hallada hasta la iteración i y S_i el conjunto de productos ya asignados. El valor de la función objetivo es:

$$z^* = \sum_{l=1}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} y_j^*}{\sum_{j \in C_l} v_{lj} y_j^* + v_{l0}}$$

Sea k un producto tal que $k \notin S_i$. Consideremos la solución factible $\hat{y}$ tal que $\hat{y}_j = y_j^* \ \forall \ j \neq k$ e $\hat{y}_k = 1$. Sea $\hat{z}$ el valor de la función objetivo evaluada en $\hat{y}$. El incremento parcial de incluir al producto k en la iteración $i + 1$ es:

$$\hat{z} - z^* = \sum_{l=1}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} \hat{y}_j}{\sum_{j \in C_l} v_{lj} \hat{y}_j + v_{l0}} - \sum_{l=1}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} y_j^*}{\sum_{j \in C_l} v_{lj} y_j^* + v_{l0}}$$

Dado que $\hat{y}_j = y_j^* \ \forall \ j \neq k$, los sumandos correspondientes a aquellos *consideration sets* C_l tales que $k \notin C_l$ se cancelan. Luego, la fórmula para determinar el incremento parcial de agregar la variable k a la solución está determinada por

$$\sum_{l=1, k \in C_l}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} \hat{y}_j}{\sum_{j \in C_l} v_{lj} \hat{y}_j + v_{l0}} - \sum_{l=1, k \in C_l}^{L} \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\intercal \pi) v_{lj} y_j^*}{\sum_{j \in C_l} v_{lj} y_j^* + v_{l0}}$$

Este desarrollo nos permite deducir que para determinar el incremento parcial de agregar la variable y_k a la solución actual sólo es necesario considerar los cocientes correpondientes a C_l's tales que $k \in C_l$. Dado que un conjunto puede contener más de un producto, el valor de estos sumandos evaluados en y_j^* es el mismo para todos los conjuntos, pudiendo ser calculados una única vez al comienzo de la iteración. De esta forma, solamente es necesario evaluarlos en la solución $\bar{y}$, y el cálculo del máximo puede realizarse a medida que se actualizan estos valores.

Dado que por las características del problema, en general, los productos no van a pertenecer a todos los *consideration sets*, es razonable suponer que este análisis produzca una mejora en el tiempo de ejecución de la heurística. De hecho, pudimos corroborar nuestra hipótesis en las instancias de prueba consideradas para evaluar el comportamiento de la heurística.

3.4. Algoritmo de generación de columnas

Como se mencionó anteriormente, resolvemos el CDLP (2.8) utilizando la técnica de generación de columnas. Si bien la estructura general de este tipo de algoritmos es bastante estandar, es necesario establecer en qué forma se realizarán determinados pasos del algoritmo en función del problema en particular que estamos resolviendo.

Una primera decisión a tomar es la selección del conjunto inicial de columnas $\mathcal{N}$. En nuestro caso, el conjunto inicial $\mathcal{N}$ tendrá una única columna. Ésta será generada a partir de un conjunto $S_0 \subseteq N$, formado por un producto de cada segmento (en particular, el primero según la numeración establecida). Considerando nuevamente el Ejemplo de la Sección 3.1, el conjunto inicial $\mathcal{N}$ contiene solamente a la columna correspondiente al conjunto $S_0 = \{1, 3, 4, 5\}$.

La segunda decisión está relacionada con la resolución del subproblema de generación de columnas. Si bien ambos algoritmos (la heurística golosa y la linealización) resuelven el problema eficientemente, debemos adoptar una estrategia para utilizar los mismos. La heurística golosa requiere menos tiempo que la linealización para resolver el problema, pero como contraparte no tenemos garantía acerca de si la solución es óptima o no. Este hecho es sumamente importante en el criterio de parada del algoritmo, que se basa, además de un número máximo de iteraciones, en analizar los costos reducidos de las variables. Por esta razón, la estrategia elegida se basa en utilizar primero la heurística golosa para resolver el problema (2.11). Si como resultado obtenemos una variable con costo reducido positivo, entonces la agregamos a $\mathcal{N}$ y volvemos a iterar. En caso contrario, se resuelve la linealización con el fin de verificar si efectivamente estamos en una solución óptima, o si existe una variable con costo reducido positivo que la heurística golosa no pudo encontrar y debe ser agregada a $\mathcal{N}$. A continuación mostramos el pseudocódigo del algoritmo de generación de columnas.

1. Generar el conjunto restringido de columnas $\mathcal{N}$.

2. **Repetir**

   a) Resolver el CDLP (2.8) restringido al conjunto $\mathcal{N}$.

   b) Obtener las variables duales π y σ.

c) Resolver el problema (2.11) utilizando la heurística golosa. Si el resultado de la heurística es un conjunto $S^* \subseteq N$ con costo reducido positivo, agregar S^* a $\mathcal{N}$ y volver a 2. En caso contrario, ir al paso siguiente.

d) Linealizar y resolver el problema (2.11). Si existe un conjunto $S^* \subseteq N$ con costo reducido positivo, agregar S^* a $\mathcal{N}$ y volver a 2. En caso contrario, ir a 3.

mientras no existan columnas con costo reducido positivo.

3. **Fin**

En los casos de prueba considerados, este algoritmo converge a una solución óptima en forma eficiente y en un número razonable de iteraciones. Además, en la gran mayoría de los casos, la heurística golosa es el algoritmo que encuentra la columna a ser agregada a $\mathcal{N}$, reduciendo así los tiempos de ejecución del algoritmo.

Sin embargo, la efectividad del método general depende de la heurística de descomposición, que se basa fuertemente en las variables duales correspondientes a las restricciones de capacidad del CDLP (2.8). En el capítulo siguiente veremos como se comportan ambos métodos mediante experimentos computacionales.

3.5. Aplicación de la Heurística de Descomposición para el MNL

Una vez resuelto el CDLP (2.8), necesitamos la información de las variables duales para utilizarlas en la heurística de descomposición. El propósito de esta heurística es obtener una política de control de venta de pasajes que pueda ser implementada por la aerolínea.

El cálculo de la descomposición se encuentra dividido en dos etapas. La primera de ellas corresponde a un proceso que puede ser realizado en forma *off line*, y consiste en determinar el ingreso esperado en función de las capacidades y el tiempo remanente. Para ello, se descompone el problema a nivel de tramo como vimos en la Sección 2.6. Esta descomposición lleva a la formulación de m problemas de programación dinámica, correspondientes a la ecuación (2.13).

La segunda etapa consiste en un proceso que debe ser realizado en tiempo real, y consiste en determinar cuál es el conjunto óptimo $S \subseteq N$ a ofrecer considerando que estamos en el instante t y tenemos capacidades x. Esto se obtiene resolviendo el problema (2.17) cada vez que tenemos un arribo de un cliente.

Es importante notar que los problemas (2.13) y (2.17), bajo las hipótesis del MNL con segmentos superpuestos, poseen la misma estructura que el subproblema de generación de columnas (3.5). Simplemente tenemos que tomar $w_j = r_j - \Gamma_i^j(t+1, x_i)$ en el primer caso, y $w_j = r_j - \Delta\bar{V}_{t+1}(x)A_j$ en el segundo. Luego, ambos problemas son NP-Hard.

En la primera etapa, debido a la gran cantidad de problemas del tipo (2.13) que es necesario resolver, utilizamos la heurística golosa. Esta decisión se basa también en los buenos resultados que hemos obtenido por parte de este algoritmo: soluciones de buena calidad (en general, soluciones óptimas) en menor tiempo que el requerido por la linealización.

Sin embargo, para el cálculo de los *offer sets* dinámicamente, el algoritmo utilizado varía dependiendo de la instancia. El problema (2.17) es particularmente importante dado que la aerolínea debe resolverlo en tiempo real, considerando que ofrece productos a clientes pertenecientes a distintos segmentos que hacen su elección de acuerdo al modelo MNL.

Siguiendo con el ejemplo visto en la Sección 3.1, el Cuadro 3.3 muestra cuál es el resultado de calcular dinámicamente los *offer sets* mediante la resolución del problema (2.17), tomando $t = 27$ y variando las capacidades remanentes de los tramos. Las variables duales obtenidas al resolver el CDLP son (0, 1134.55, 500).

Capacidad remanente	*Offer set*	Capacidad remanente	*Offer set*
(3,2,2)	$\{1,3,4,5\}$	(2,2,1)	$\{1,3,4,5\}$
(2,3,2)	$\{1,3,4,5\}$	(2,1,2)	$\{1,2,3,4,6\}$
(2,2,3)	$\{1,2,3,4,5\}$	(1,2,2)	$\{1,3,4,5\}$
(3,2,1)	$\{1,3,4,5\}$	(2,1,1)	$\{1,3,4\}$
(3,1,2)	$\{1,2,3,4,6\}$	(1,2,1)	$\{1,3,4,5\}$
(2,3,1)	$\{1,3,4,5\}$	(1,1,2)	$\{1,2,3,4,6\}$
(1,3,2)	$\{1,3,4,5\}$	(2,1,0)	$\{1,3\}$
(2,1,3)	$\{1,2,3,4,6\}$	(2,0,1)	$\{2,3,4\}$
(1,2,3)	$\{1,3,4,5\}$	(1,2,0)	$\{1,3,5\}$
(3,1,1)	$\{1,3,4,6\}$	(0,2,1)	$\{1,4,5\}$
(1,3,1)	$\{1,3,4,5\}$	(1,0,2)	$\{2,3,4,6\}$
(1,1,3)	$\{1,2,3,4,6\}$	(0,1,2)	$\{1,4\}$
(3,1,0)	$\{1,3\}$	(1,1,1)	$\{1,3,4\}$
(3,0,1)	$\{2,3,4,6\}$	(1,1,0)	$\{1,3\}$
(1,3,0)	$\{1,3,5\}$	(1,0,1)	$\{2,3,4\}$
(0,3,1)	$\{1,4,5\}$	(0,1,1)	$\{1,4\}$
(1,0,3)	$\{2,3,4,6\}$	(1,0,0)	$\{3\}$
(0,1,3)	$\{1,4\}$	(0,1,0)	$\{1\}$
(2,2,2)	$\{1,3,4,5\}$	(0,0,1)	$\{4\}$

Cuadro 3.3: *Offer sets* dinámicos para el Ejemplo de la Sección 3.1 cuando $t = 27$.

Esta tabla nos permite seguir el comportamiento de la política de descomposición en un instante en particular. Por ejemplo, cuando empezamos del estado (1,1,2) en lugar de empezar desde el (1,2,2), se cierra el producto 5 pero se abren los productos 2 y 6. Esto corresponde a cerrar la tarifa baja sobre el tramo 2, y forzar a los clientes a considerar el vuelo con conexión A $\rightarrow$ B $\rightarrow$ C. La tarifa alta sobre el tramo 2, el producto 1, continúa disponible a pesar de verse reducida su capacidad en una unidad. Contrariamente, podemos ver que los estados (3,2,2) y (2,2,2) poseen el mismo *offer set*. Esto significa que disminuir en una unidad la capacidad para el tramo 1 cuando aún quedan diez períodos en el horizonte de reservas no amerita cerrar o abrir nuevos productos.

De hecho, tres de los cuatro productos en el conjunto, corresponden a tarifas altas. Esto se debe a que, como en este ejemplo $\lambda = 1$ (es decir, en todos los períodos hay un arribo), la política supone que todavía es posible vender productos que generen mayor ingreso en lugar de aquellos correspondientes a tarifas bajas. Notar también que dada la cantidad de tiempo remanente en el horizonte de reservas, cuando sólo queda a lo sumo un asiento libre sobre los distintos tramos, sólo los productos con tarifas altas permanecen abiertos.

3.6. Reoptimización

Si bien la primera etapa de la heurística de descomposición puede ser calculada mediante procesos batch, los tiempos requeridos pueden ser considerablemente altos. Por esta razón, y basándonos en los tiempos obtenidos por el algoritmo de generación de columnas, proponemos un método alternativo para establecer una política implementable en la práctica que produzca ingresos razonables a un menor costo computacional.

La idea principal de este método es proveer estimadores estáticos del valor marginal de las capacidades de los diferentes tramos, pero actualizando estos valores frecuentemente de forma tal que la estimación se ajuste al estado de la red. Estas estimaciones se corresponden con las variables duales π_i del CDLP. Luego, resolvemos el problema (2.17) tomando $\beta = 0$ en (2.16). Con esta idea, de alguna forma estamos capturando la aproximación dinámica de la heurística de descomposición, pero a menor costo computacional dado que no es necesiario calcular previamente ninguna tabla del estilo de las $\hat{V}_t^i(x_i)$.

La calidad de las estimaciones estáticas depende en parte de la frecuencia con la que se resuelva el CDLP con el fin de obtener una actualización del vector π. En nuestra implementación, el método depende de un parámetro que indica el porcentaje de tiempo que debe esperarse para obtener las nuevas variables duales. A este parámetro lo llamamos *freq*, donde *freq*$\in [0,1]$, e interpretamos el producto *freq*$\times T$ como la cantidad de instantes que deben pasar para reoptimizar el CDLP.

Es importante destacar que esta actualización se realiza en tiempo real, y los buenos tiempos obtenidos en la resolución del CDLP juegan un papel crítico en este método. El siguiente pseudocódigo muestra la rutina que, en función del instante t y las capacidades remanentes x, calcula las nuevas variables duales π y el nuevo *offer set* S que será ofrecido hasta la nueva actualización.

1. **Input:** t, x.

2. **Si** t es múltiplo de *freq*$\times T$

   a) Reoptimizar el CDLP considerando las nuevas capacidades x y actualizando $T = T - t$.

   b) Obtener las nuevas variables duales π.

   c) Resolver el problema de optimización

$$\max_{y\in\{0,1\}^n} \sum_{l=1}^{L} \frac{\sum_{j\in C_l}(r_j - A_j^{\mathsf{T}}\pi)v_{lj}y_j}{\sum_{j\in C_l} v_{lj}y_j + v_{l0}}$$

d) Sea y^* la solución óptima. Si $y_j^* = 1$, agregar j a S.

e) Ofrecer el conjunto de productos S hasta la próxima actualización.

3. **Fin.**

Alcanzar el límite máximo no es la única instancia en la cual se recalcula el *offer set*. Si durante uno de los intervalos se agota la capacidad de alguno de los tramos, el conjunto que se está ofreciendo podría no corresponderse con la realidad de la red en ese instante. Por ejemplo, alguno de los productos definidos sobre los tramos que ya no se encuentran disponibles podría estar presente en el *offer set*, pero dado que la capacidad de uno de los tramos que utiliza está agotada, este producto no debería ofrecerse. Por esta razón, cada vez que algún tramo deja de estar disponible, se recalcula el *offer set*.

En las pruebas realizadas, consideramos 0.01 y 0.1 como valores posibles para *freq*. Es de esperar que a medida que se reoptimiza con mayor frecuencia, los ingresos generados por este método sean mayores. Para ilustrar el comportamiento de este método, volvemos al ejemplo de la red pequeña propuesto en la Sección 3.1, Considerando que $T = 30$, tomar *freq* $= 0.1$ trae como resultado la actualización de las variables duales en los instantes $t = 0, 3, 6, 9$, etc., y asignar *freq*$= 0.01$ provoca que se reoptimice en todos los instantes del horizonte de reservas.

Capítulo 4

Resultados Computacionales

En este capítulo evaluamos y comparamos el comportamiento de las políticas vistas en los capítulos anteriores mediante una simulación computacional de situaciones reales de arribos de clientes. Para realizar esta comparación, tenemos en cuenta dos medidas fundamentales: los ingresos ingresos económicos generados, y el costo computacional requerido por cada una de ellas.

El capítulo está organizado de la siguiente manera. En la Sección 4.1 explicamos cómo se realiza la simulación numérica. Luego, en la Sección 4.2 mostramos el comportamiento de las políticas consideradas en tres ejemplos distintos. El primero de ellos consta de una red con tres vuelos paralelos; el segundo se basa en un red pequeña con siete tramos y 22 productos; finalmente el tercer ejemplo consiste en una red de tamaño mediano, con ocho tramos y 80 productos.

4.1. Proceso de simulación

Para simular el comportamiento selectivo de los clientes, utilizamos una simulación Monte Carlo. El proceso de arribo de clientes comienza en el instante $t = 0$, y concluye al finalizar el horizonte de reservas ($t = T$). En cada instante, el proceso se encuentra dividido en dos etapas. Primero, establecemos si efectivamente se produjo el arribo de un cliente, y en caso afirmativo, a qué segmento pertenece. Para eso, consideramos $L+1$ segmentos de clientes, con probabilidades $\lambda_1, \dots, \lambda_L, \lambda_{L+1}$, con $\lambda_{L+1} = 1-\lambda = \mathbb{P}(\text{no compra})$. Generamos una v.a. $u \in$ Unif[0,1], y utilizamos la función de distribución acumulada sobre los λ_l para determinar a qué segmento pertenece el arribo (denominado Método de la Función Inversa). En caso de que el arribo sea del segmento $L + 1$, quiere decir que en ese período no se produjo un arribo. Luego, sabiendo que el arribo es de un cliente del segmento l, $l = 1, \dots, L$, generamos otra v.a. z Unif[0,1] y seguimos el mismo razonamiento que antes, pero en este caso con la función de distribución acumulada de las probabilidades $P_{lj}(S^*)$, para aquellos productos $j \in C_l \cap S^*$, con el fin de verificar si se ha vendido algún producto de S^*.

El siguiente pseudocódigo describe el proceso que genera un *stream* de arribos de clientes. Es importante destacar que la diferencia entre las distintas políticas está dada por el cálculo *offer set* en cada instante de tiempo.

1. **Para** $t = 1, \ldots, T$
   a) Generar u v.a. Unif[0,1].
   b) Determinar si hubo un arribo utilizando la función de distribución acumulada de los λ_l. En caso afirmativo, sea l_0 el segmento e ir al siguiente paso. En caso contrario, volver a 1.
   c) Calcular el *offer set* S^*.
   d) Generar z v.a. Unif[0.1].
   e) Determinar si se produjo una venta utilizando la función de distribución acumulada $P_{l_0 j}(S^*)$, $j \in C_l \cap S^*$. En caso afirmativo, actualizar los ingresos generados y las capacidades.
2. **Fin.**

Es importante destacar que este proceso corresponde a un único *stream* de arribos de clientes. En las simulaciones, generamos una cantidad determinada de *streams* (en particular, 2,000 en los dos primeros ejemplos, y 1,000 en el tercero). El comportamiento de las políticas se evalúa en función de la media del ingreso generado en cada *stream*, para la que se calcula un intervalo de confianza del 95 %. Este intervalo determina un rango estadístico para los valores que esperamos de la media. De la misma forma, el costo computacional de los procesos que deben ser realizados en tiempo real (por ejemplo, el cálculo de los *offer sets*) también se evalúa en términos de la media de los tiempos obtenidos en los *streams* de simulación.

4.2. Ejemplos numéricos

En esta sección evaluamos el comportamiento de los distintos métodos mediante simulaciones con el fin de analizar como responden en la práctica al asumir que los clientes eligen según el modelo MNL con segmentos superpuestos. Para cada método en particular, analizamos dos aspectos: la calidad de las soluciones obtenidas, y el tiempo de cómputo requerido.

Los experimentos fueron realizados en un servidor SUN UltraSparc III (1 Ghz de CPU, 2GB de memoria RAM, sistema operativo SunOS 5.9). Los algoritmos fueron implementados en el lenguaje C++, y compilados usando la versión 3.4.6 del compilador *gcc*. Para la resolución de los distintos problemas lineales y lineales enteros, se utilizan las rutinas de optimización provistas por el paquete CPLEX versión 8.1.

Los métodos considerados son:

1. *DCOMP*: Éste es el método que se basa en la heurística de descomposición propuesta en van Ryzin y Liu [10], extendido para el caso del MNL con segmentos superpuestos (ver secciones 2.6 y 3.5). Primero resolvemos el CDLP usando el algoritmo de generación de columnas (Sección 3.4). Luego, calculamos las descomposiciones (2.13) para cada uno de los tramos $i = 1, \ldots, m$, considerando todos los instantes del horizonte de reservas, y para todos las

capacidades $x_i = 1, \ldots, c_i$. El valor marginal de la capacidad se calcula tomando $\beta = 1$ en (2.16). Al procesar los *streams* de demanda de los clientes, si se produce el arribo de un cliente resolvemos el problema (2.17) para determinar el *offer set* S^*.

2. *DCOMP-0.5*: Análogo a DCOMP, pero tomando $\beta = 0.5$ al en la ecuación (2.16).

3. *CDLP*: Éste método utiliza la solución primal del CDLP. Sabiendo que el orden en el cual se deben ofrecer los distintos conjuntos presentes en la solución óptima del CDLP (2.8) es ambiguo, la secuencia de *offer sets* queda determinada por el orden lexicográfico de los índices de las variables correspondientes. Dado que las variables $t(S)$ pueden tomar valores fraccionarios, los redondeamos al entero más cercano.

4. *RCDLP*: Ésta es una versión aleatoria del método anterior. Es decir, el orden de los índices de las variables primales $t(S)$ de (2.8) se determina mediante una permutación aleatoria.

5. *ROPT-freq*: Corresponde al método visto en la Sección 3.6. Consideramos los valores 0.01 y 0.1 para el parámetro *freq*.

6. *INDEP*: Corresponde al modelo de demanda independiente visto en la Sección 2.7.

En las tablas de resultados se reporta una cota superior (UB) para el ingreso óptimo esperado (dado por el valor de la función objetivo de (2.8)) y una medida de congestión denominada *load factor* (LF). Ésta representa un indicador de la utilización de la capacidad a nivel de red, y se calcula como el promedio de los *load factors* de los tramos, es decir,

$$\text{Load factor para el tramo } i = \frac{\text{promedio de asientos vendidos sobre el tramo } i}{c_i}$$

Notar que el *load factor* depende directamente de la política de control.

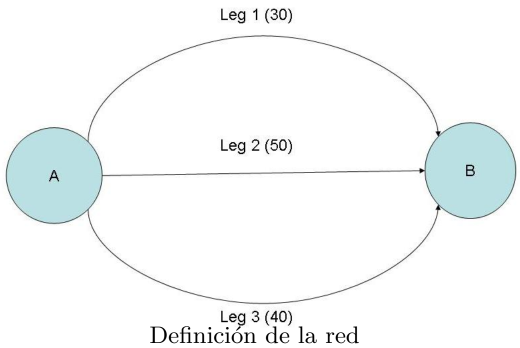
En cada uno de los ejemplos, escalamos las capacidades mediante un parámetro α para testear distintos *load factors* sobre la red. El caso $\alpha = 1$ corresponde a la definición original de la instancia. También alteramos las preferencias en el comportamiento selectivo modificando el vector de preferencias de no-compra $v_0 = (v_{10}, \ldots, v_{L0})$. Por último, como mencionamos anteriormente, por cada escenario generamos 2000 *streams* de demanda en los dos primeros ejemplos, y 500 en el tercero.

4.2.1. Vuelos Paralelos

La red correspondiente a este ejemplo se muestra en la Figura 4.1. Esta instancia se basa en una red con tres vuelos paralelos (tienen el mismo origen y destino), con dos productos definidos sobre cada uno (clase alta (H), y baja (L)). En total, tenemos seis productos.

Asumimos que los clientes se encuentran divididos en cuatro segmentos: aquellos clientes que eligen solamente entre productos de clase alta; los que solamente consideran los productos clase baja; los que consideran todos los productos, pero con mayor preferencia a los de la clase H; y finalmente los clientes que también consideran todos los productos, pero con mayor preferencia por los de clase baja. Este ejemplo es una generalización del modelo estudiado por Zhang and Cooper [14], también considerado por van Ryzin y Liu [10].

Las siguientes tablas muestran la definición de los productos y los segmentos. El vuelo 1 corresponde a un vuelo que se realiza durante la mañana, el vuelo 2 a uno de tarde, y finalmente el vuelo 3 es un vuelo nocturno. Las capacidades inciales son $c = (30, 50, 40)$. El horizonte de reservas consiste en $T = 300$ períodos, con un promedio de 150 arribos por *stream* de demanda.



Definición de la red

Producto	Tramo	Clase	Tarifa
1	1	L	400
2	1	H	800
3	2	L	500
4	2	H	1000
5	3	L	300
6	3	H	600

Definición de productos

Figura 4.1: Ejemplo de Vuelos Paralelos

Segmento	C_l	v_l	λ_l	Descripción
1	{2,4,6}	(5,10,1)	0.1	Insensibles al precio, pref. por la tarde.
2	{1,3,5}	(5,1,10)	0.15	Sensibles al precio, pref. por la noche.
3	{1,2,3,4,5,6}	(10,8,6,4,3,1)	0.2	Pref. por lo mañana, sensibles al precio.
4	{1,2,3,4,5,6}	(8,10,4,6,1,3)	0.05	Insensibles al precio, pref. por la manñana.

Cuadro 4.1: Definición de segmentos para la instancia de Vuelos Paralelos.

El Cuadro 4.2 resume los resultados obtenidos de las simulaciones por las distintas políticas. Para cada una de ellas, se reporta la media de los ingresos generados y el *load factor* obtenido en cada *stream* de demanda para los distintos escenarios considerados. El Cuadro 4.3 muestra los *gaps* en el ingreso generado (diferencia en porcentaje en el ingreso) por los otros métodos respecto de la política DCOMP, incluyendo además los intervalos de confianza para los mismos. Por ejemplo, la primera fila de las tablas corresponde al caso $c_{0,6} = 0.6 \times (30, 50, 40) = (18, 30, 24)$ y $v_0 = (1, 5, 5, 1)$, éste último correspondiente a los valores de no compra v_{l0} de cada uno de los cuatro segmentos. Dado que esta instancia es muy pequeña, no reportamos mediciones de tiempo.

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		RCDLP		ROPT-0.01		ROPT-0.1		INDEP	
			Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF
0.6	(1,5,5,1)	56,884	55,948	98.60	55,872	97.82	54,177	94.60	53,783	94.42	51,632	95.83	49,468	93.48	49,794	97.41
	(1,10,5,1)	56,848	55,882	98.36	55,698	97.29	54,051	94.45	53,619	94.08	51,946	93.81	49,994	91.01	49,655	96.77
	(5,20,10,5)	53,819	51,326	94.59	51,338	94.23	50,058	92.94	50,476	93.44	48,602	97.35	46,274	95.05	46,246	92.53
0.8	(1,5,5,1)	71,936	69,533	95.78	69,163	94.79	68,105	94.69	68,641	95.11	66,336	94.57	64,490	92.75	60,346	94.14
	(1,10,5,1)	71,794	69,129	95.24	68,863	94.50	67,896	94.58	68,491	95.18	66,237	93.82	64,522	91.91	59,552	91.40
	(5,20,10,5)	61,868	60,147	90.48	60,222	90.49	59,073	90.82	59,289	90.83	56,724	91.98	55,842	91.16	53,044	81.94
1.0	(1,5,5,1)	79,155	76,954	95.65	77,096	95.91	75,726	94.96	75,996	94.89	77,106	95.36	76,136	94.36	66,224	85.38
	(1,10,5,1)	76,866	75,639	90.88	75,655	90.97	73,788	90.26	74,100	90.19	75,399	91.26	74,635	90.73	64,831	81.76
	(5,20,10,5)	63,255	62,775	78.09	62,792	78.15	62,702	78.41	62,541	78.41	62,040	79.94	61,827	79.96	56,203	72.44
1.2	(1,5,5,1)	80,371	79,817	84.28	79,818	84.24	79,666	84.33	79,698	84.32	79,834	84.58	79,774	84.53	68,970	76.84
	(1,10,5,1)	78,045	77,520	79.06	77,526	79.07	77,348	79.26	77,332	79.22	77,529	79.49	77,476	79.45	67,570	73.71
	(5,20,10,5)	63,296	63,111	67.52	63,113	67.52	62,491	68.88	62,677	68.87	62,422	70.20	62,293	70.20	58,543	65.88
1.4	(1,5,5,1)	81,066	80,408	73.08	80,376	72.83	80,362	72.75	80,362	72.75	80,439	73.27	80,421	73.22	71,418	70.60
	(1,10,5,1)	78,816	78,123	68.56	78,097	68.32	78,091	68.24	78,091	68.24	78,136	68.80	78,120	68.76	69,949	67.87
	(5,20,10,5)	63,337	63,211	60.54	63,212	60.54	62,553	62.04	62,775	62.00	62,822	62.82	62,734	62.85	60,732	60.82

Cuadro 4.2: Resultados de las simulaciones para la instancia Vuelos Paralelos.

La primera observación que podemos hacer de ambas tablas es que DCOMP genera un ingreso esperado mayor que CDLP en todos los casos. De hecho, los intervalos de confianza para los *gaps* excluyen al cero en 10 de los 15 casos, indicando que en estos escenarios DCOMP generará en promedio un ingreso mayor que CDLP. Estas diferencias se incrementan en los casos donde el *load factor* es alto ($\alpha = 0.6$, 0.8 y 1.0), generando beneficios del orden del 1.5 % - 3.5 %.

Al igual que los resultados reportados por van Ryzin y Liu [10], podemos ver que la performance de las políticas CDLP y RCDLP es muy similar, y no es posible determinar si una de ellas domina a la otra.

La misma afirmación podemos hacer respecto de DCOMP y DCOMP-0.5. En este ejemplo en particular, ambas políticas generan ingresos comparables y no se puede observar una notoria superioridad de una respecto de la otra. Si consideramos además los ingresos generados por la política ROPT, podemos ver que el componente dinámico que se obtiene al tomar β =0.5 en (2.16) es suficiente como para mejorar en muchos los resultados producidos al tomar una aproximación estática del costo de oportunidad dentro de un esquema de reoptimización.

α	v_0	% DCOMP-0.5	% CDLP	% RCDLP	% ROPT-0.01	% ROPT-0.1	% INDEP
0.6	(1,5,5,1)	0.13 ± 0.18	3.16 ± 0.31	3.86 ± 0.31	7.71 ± 0.24	11.58 ± 0.28	10.99 ± 0.23
	(1,10,5,1)	0.32 ± 0.20	3.27 ± 0.32	4.04 ± 0.32	7.04 ± 0.27	10.53 ± 0.29	11.14 ± 0.23
	(5,20,10,5)	-0.02 ± 0.39	2.47 ± 0.41	1.65 ± 0.41	5.30 ± 0.36	9.84 ± 0.39	9.89 ± 0.37
0.8	(1,5,5,1)	0.53 ± 0.33	2.05 ± 0.33	1.28 ± 0.35	4.59 ± 0.30	7.25 ± 0.32	13.21 ± 0.33
	(1,10,5,1)	0.38 ± 0.35	1.78 ± 0.34	0.92 ± 0.36	4.18 ± 0.32	6.66 ± 0.33	13.85 ± 0.34
	(5,20,10,5)	-0.12 ± 0.47	1.78 ± 0.44	1.42 ± 0.44	5.69 ± 0.42	7.15 ± 0.41	11.80 ± 0.46
1.0	(1,5,5,1)	-0.18 ± 0.34	1.59 ± 0.33	1.24 ± 0.33	-0.19 ± 0.34	1.06 ± 0.34	13.94 ± 0.38
	(1,10,5,1)	-0.02 ± 0.42	2.44 ± 0.38	2.03 ± 0.39	0.31 ± 0.23	1.32 ± 0.40	14.28 ± 0.41
	(5,20,10,5)	-0.53 ± 0.56	0.11 ± 0.54	0.37 ± 0.54	1.17 ± 0.54	1.51 ± 0.54	10.46 ± 0.52
1.2	(1,5,5,1)	0.00 ± 0.46	0.18 ± 0.45	0.14 ± 0.45	-0.02 ± 0.29	0.05 ± 0.45	13.58 ± 0.43
	(1,10,5,1)	0.00 ± 0.48	0.22 ± 0.47	0.24 ± 0.47	-0.01 ± 0.48	0.05 ± 0.47	12.83 ± 0.45
	(5,20,10,5)	0.00 ± 0.57	0.98 ± 0.56	0.68 ± 0.56	1.09 ± 0.57	1.29 ± 0.63	7.23 ± 0.53
1.4	(1,5,5,1)	0.03 ± 0.48	0.05 ± 0.47	0.05 ± 0.47	-0.03 ± 0.49	-0.01 ± 0.49	11.18 ± 0.45
	(1,10,5,1)	0.03 ± 0.50	0.04 ± 0.50	0.04 ± 0.50	-0.01 ± 0.50	0.00 ± 0.50	10.46 ± 0.46
	(5,20,10,5)	0.00 ± 0.56	1.04 ± 0.56	0.68 ± 0.55	0.61 ± 0.56	0.75 ± 0.56	3.92 ± 0.54

Cuadro 4.3: Intervalos de conf. del 95 % para los ingresos generados respecto a DCOMP en la instancia Vuelos Paralelos.

En función de la política ROPT, podemos ver que tomando 0.01 para el parámetro *freq* el método mejora levemente los resultados obtenidos por DCOMP para los casos donde el *load factor* es bajo, en particular para las combinaciones $\alpha = 1.0$ y $v_0 = (1, 5, 5, 1)$, y $\alpha = 1.2$, 1.4 y $v_0 = (1, 5, 5, 1)$ ó $v_0 = (1, 10, 5, 1)$. Considerando los posibles valores evaluados para el parámetro *freq*, podemos ver que tomar *freq* $= 0.01$ genera mayor ingreso en promedio que en el caso de *freq* $= 0.1$. Esto indica que actualizar los valores de las variables duales frecuentemente mejora la performance de la política. En particular, esta diferencia es notoria en los escenarios donde el *load factor* es alto ($\alpha =$ 0.6, 0.8 y 1.0). Esto se debe, principalmente, a que en estos casos la capacidad es una restricción importante, y mantener actualizados los valores de las variables duales ajusta mejor la política a los cambios en las capacidades y el tiempo remanente.

Este ejemplo muestra que casi todas las políticas generan ingresos significativamente mayores que INDEP. En particular, DCOMP logra mejoras en los ingresos de hasta un 14 % mayores que INDEP, evidenciando la ventaja de incorporar explícitamente el comportamiento selectivo de los clientes a la hora de modelar.

En función del comportamiento del algoritmo de generación de columnas (GENCOL), la cantidad de iteraciones es sólo una pequeña fracción del número total de variables del CDLP. En los escenarios considerados, el número máximo de veces que itera el algoritmo es ocho, y se da cuando $\alpha = 0.8$, y $v_0 = (1, 10, 5, 1)$ ó $v_0 = (5, 20, 10, 5)$. El número total de iteraciones realizadas por el algoritmo GENCOL considerando todos los escenarios es 72. Sin embargo, solamente fue necesario recurrir a la linealización para encontrar una nueva columna tres veces (4.16 %), indicando que la heurística golosa provee soluciones de buena calidad en el 95.84 % de los casos.

4.2.2. Red Pequeña

El ejemplo se basa en la misma red considerada en van Ryzin y Liu [10]. Este consite en cuatro ciudades, con siete vuelos que las conectan. Al igual que en el ejemplo anterior, sobre cada vuelo se definen dos clases de tarifas. Esto genera un total de 22 productos. La Figura 4.2 muestra la definición de la red, con las capacidades iniciales de cada tramo. Lo interesante de este tipo de redes es que, además de haber tráfico local entre la ciudad H y las restantes ciudades, H también funciona como *hub* (punto de conexión) para pasajeros que desean viajar, por ejemplo, desde A hasta C. El Cuadro 4.4 muestra la definición de los productos de la red.

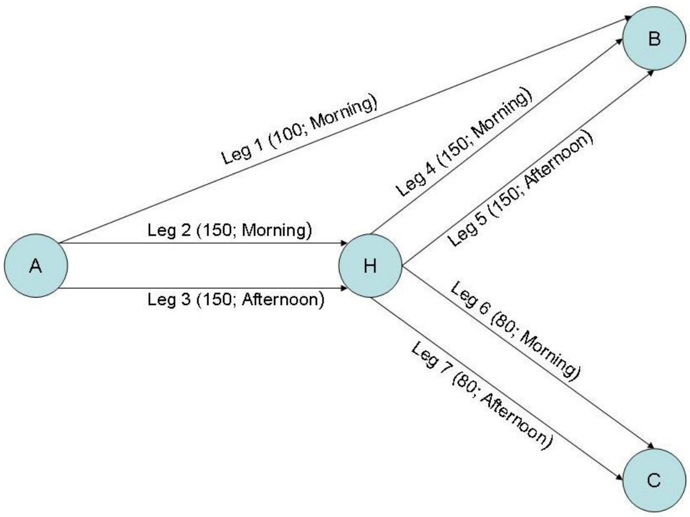


Figura 4.2: Red para la instancia Red Pequeña.

Producto	Tramo	Clase	Tarifa	Producto	Tramo	Clase	Tarifa
1	1	H	1000	12	1	L	500
2	2	H	400	13	2	L	200
3	3	H	400	14	3	L	200
4	4	H	300	15	4	L	150
5	5	H	300	16	5	L	150
6	6	H	500	17	6	L	250
7	7	H	500	18	7	L	250
8	$\{2,4\}$	H	600	19	$\{2,4\}$	L	300
9	$\{3,5\}$	H	600	20	$\{3,5\}$	L	300
10	$\{2,6\}$	H	700	21	$\{2,6\}$	L	350
11	$\{3,7\}$	H	700	22	$\{3,7\}$	L	350

Cuadro 4.4: Definición de productos para la instancia Red Pequeña.

En esta instancia, consideramos diez segmentos de clientes. Cada segmento está definido en función de las preferencias de origen-destino, tarifa, y horario por parte de los mismos. El Cuadro 4.5 describe las características de los diez *consideration sets*. El horizonte de reservas consiste de $T = 1000$ instantes de tiempo, con un promedio de 910 arribos por cada *stream* de demanda.

Segmento	O-D	C_l	v_l	λ_l	Descripción
1	A $\rightarrow$ B	{1,8,9,12,19,20}	(10,8,8,6,4,4)	0.08	Insensibles al precio, pref. por la mañana.
2	A $\rightarrow$ B	{1,8,9,12,19,20}	(1,2,2,8,10,10)	0.2	Sensibles al precio.
3	A $\rightarrow$ H	{2,3,13,14}	(10,10,5,5)	0.05	Insensibles al precio.
4	A $\rightarrow$ H	{2,3,13,14}	(2,2,10,10)	0.2	Sensibles al precio.
5	H $\rightarrow$ B	{4,5,15,16}	(10,10,5,5)	0.1	Insensibles al precio.
6	H $\rightarrow$ B	{4,5,15,16}	(2,2,10,8)	0.15	Sensibles al precio, leve pref. por la mañana.
7	H $\rightarrow$ C	{6,7,17,18}	(10,8,5,5)	0.02	Insensibles al precio, leve pref. por la mañana.
8	H $\rightarrow$ C	{6,7,17,18}	(2,2,10,8)	0.05	Sensibles al precio.
9	A $\rightarrow$ C	{10,11,21,22}	(10,8,5,5)	0.02	Insensibles al precio, leve pref. por la mañana.
10	A $\rightarrow$ C	{10,11,21,22}	(2,2,10,10)	0.04	Sensibles al precio.

Cuadro 4.5: Definición de segmentos para la instancia Red Pequeña.

El Cuadro 4.6 muestra los ingresos esperados y los *load factors* obtenidos por cada una de las políticas, y el Cuadro 4.7 los *gaps* de las otras políticas respecto a DCOMP. En estas tablas, la notación correspondiente al vector de preferencias de no-compra, v_0, la modificamos levemente. Ahora, v_0 tiene solo dos componentes[1]. La primera de ellas corresponde al valor de preferencia de no-compra de los segmentos que consideran clientes que no son sensibles al precio, y la segunda, a aquellos segmentos cuyos clientes sí lo son.

[1]Por ejemplo, si tomamos la notación original $v_0 = (1,5,1,5,\ldots,1,5)$, ahora lo representamos como $v_0 = (1,5)$.

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		RCDLP		ROPT-0.01		ROPT-0.1		INDEP	
			Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF
0.6	(1,5)	215,793	197,038	88.90	196,920	88.06	207,890	91.27	208,476	91.70	200,444	93.37	195,291	92.09	172,362	97.71
	(5,10)	200,515	194,146	93.35	191,443	92.09	194,393	91.90	193,658	91.82	192,896	95.07	189,182	94.42	163,905	96.76
	(10,20)	170,137	167,866	92.68	167,902	92.79	164,089	91.45	164,296	91.39	166,919	93.07	165,516	92.47	151,801	92.48
0.8	(1,5)	266,934	262,823	86.79	263,023	86.37	261,264	85.62	260,820	85.68	252,013	86.90	249,221	86.21	204,572	94.60
	(5,10)	223,173	220,891	90.48	221,012	90.51	215,884	89.38	217,105	89.40	217,073	90.75	214,192	90.15	191,066	90.22
	(10,20)	188,574	186,219	85.29	185,969	85.27	184,182	84.86	184,289	84.92	186,325	85.61	185,841	85.26	172,246	84.09
1.0	(1,5)	281,967	279,506	81.34	279,536	81.36	277,738	80.80	277,473	80.78	278,344	81.04	275,016	80.53	226,002	87.71
	(5,10)	235,284	233,929	84.41	233,891	84.27	230,342	83.86	231,350	83.89	233,138	84.09	232,376	83.68	209,701	83.64
	(10,20)	192,038	191,646	76.10	191,623	76.05	190,283	76.34	190,393	76.34	191,727	75.89	191,627	75.70	188,058	76.73
1.2	(1,5)	284,772	284,736	71.85	284,747	71.85	282,842	71.55	282,996	71.51	283,280	72.47	281,926	72.46	243,930	82.48
	(5,10)	238,562	238,539	72.38	238,502	72.26	238,299	72.03	238,299	72.03	238,548	72.35	238,523	72.33	225,691	77.65
	(10,20)	192,373	192,530	65.86	192,524	65.87	192,511	65.88	192,511	65.88	192,532	65.87	192,526	65.87	192,416	65.80
1.4	(1,5)	287,076	286,743	62.14	286,629	62.16	285,417	61.96	285,598	61.95	286,160	62.24	285,783	62.21	259,039	76.96
	(5,10)	238,562	238,843	61.80	238,843	61.80	238,843	61.80	238,843	61.80	238,843	61.80	238,843	61.80	231,937	68.82
	(10,20)	192,373	192,541	56.48	192,541	56.48	192,541	56.48	192,541	56.48	192,541	56.48	192,541	56.48	192,468	56.42

Cuadro 4.6: Resultados de las simulaciones para la instancia Red Pequeña.

Nuevamente, podemos observar que DCOMP mejora entre un 0.5 % y 2.26 % a CDLP en al menos la mitad de los casos, y ambas son comparables en los restantes. La comparación entre CDLP y RCDLP, al igual que entre DCOMP y DCOMP-0.5, es similar a lo mencionado para la instancia de Vuelos Paralelos. Ningúno de los métodos es consistentemente mejor que su respectiva variante.

Un resultado significativamente distinto a los obtenidos hasta ahora puede verse en las primeras filas de ambas tablas, y corresponde a una performance considerablemente peor por parte de DCOMP en comparación con los restantes métodos. En van Ryzin y Liu [10] reportan un caso análogo en un escenario similar al nuestro, y adjudican este comportamiento al hecho de que la solución del CDLP es degenerada, con lo cual existen múltiples soluciones factibles duales. De hecho, modificando el conjunto inicial de columnas, obtienen una nueva solución dual π, y con esta información DCOMP mejora su performance. Sin embargo, en nuestro caso la solución del CDLP no es degenerada, con lo cual sabemos que existe una única solución factible dual. Consideramos que es necesario realizar más experimentación para identificar en que tipo de escenarios DCOMP se ve afectado por el valor obtenido de las variables duales del CDLP.

α	v_0	% DCOMP-0.5	% CDLP	% RCDLP	% ROPT-0.01	% ROPT-0.1	% INDEP
0.6	(1,5)	0.05 ± 0.16	-5.50 ± 0.18	-5.80 ± 0.18	-1.72 ± 0.17	0.88 ± 0.17	12.52 ± 0.15
	(5,10)	1.39 ± 0.17	-0.12 ± 0.18	0.25 ± 0.18	0.64 ± 0.16	2.55 ± 0.17	15.57 ± 0.15
	(10,20)	-0.02 ± 0.18	2.25 ± 0.19	2.12 ± 0.18	0.56 ± 0.17	1.39 ± 0.17	9.57 ± 0.16
0.8	(1,5)	-0.07 ± 0.13	0.59 ± 0.15	0.76 ± 0.15	4.11 ± 0.14	5.17 ± 0.15	22.16 ± 0.13
	(5,10)	-0.05 ± 0.15	2.26 ± 0.15	1.71 ± 0.16	1.72 ± 0.15	3.03 ± 0.16	13.50 ± 0.16
	(10,20)	0.13 ± 0.20	1.09 ± 0.19	1.03 ± 0.19	-0.05 ± 0.20	0.20 ± 0.21	7.50 ± 0.20
1.0	(1,5)	-0.01 ± 0.14	0.63 ± 0.15	0.72 ± 0.16	0.41 ± 0.15	1.60 ± 0.15	19.14 ± 0.15
	(5,10)	0.01 ± 0.16	1.53 ± 0.16	1.10 ± 0.17	0.33 ± 0.19	0.66 ± 0.17	10.35 ± 0.17
	(10,20)	0.01 ± 0.22	0.71 ± 0.22	0.65 ± 0.22	-0.04 ± 0.30	0.00 ± 0.22	1.87 ± 0.22
1.2	(1,5)	0.00 ± 0.16	0.66 ± 0.16	0.61 ± 0.17	0.51 ± 0.17	0.98 ± 0.16	14.33 ± 0.16
	(5,10)	0.01 ± 0.19	0.10 ± 0.18	0.10 ± 0.18	0.00 ± 0.19	0.00 ± 0.19	5.38 ± 0.18
	(10,20)	0.00 ± 0.22	0.00 ± 0.23	0.00 ± 0.23	0.00 ± 0.22	0.00 ± 0.22	0.05 ± 0.21
1.4	(1,5)	0.03 ± 0.17	0.46 ± 0.17	0.39 ± 0.16	0.20 ± 0.17	0.33 ± 0.17	9.66 ± 0.16
	(5,10)	0.00 ± 0.19	0.00 ± 0.19	0.00 ± 0.19	0.00 ± 0.19	0.00 ± 0.19	2.89 ± 0.19
	(10,20)	0.00 ± 0.22	0.00 ± 0.22	0.00 ± 0.22	0.00 ± 0.22	0.00 ± 0.22	0.03 ± 0.22

Cuadro 4.7: Intervalos de conf. del 95 % para los ingresos generados respecto a DCOMP en la instancia Red Pequeña.

El valor utilizado para el parámetro *freq* en la política ROPT sigue siendo determinante en los casos donde el *load factor* es bajo ($\alpha = 0.6$, 0.8 y 1.2), produciendo mayores ingresos al tomar $freq = 0.01$. Además, podemos observar que esta diferencia se hace más grande a medida que los valores de preferencia para la no-compra disminuyen. Por otro lado, si bien DCOMP se comporta mejor que ROPT-0.01 en 12 de los 15 casos, del Cuadro 4.6 podemos notar que ROPT-0.01 genera mayor ingreso esperado que CDLP en 8 casos, en particular para valores de $\alpha = 0.8$, 1.0 y 1.2. Desde este punto de vista, la política ROPT (en especial, ROPT-0.01), se comporta muy bien cuando hay altas posibilidades de no compra (es decir, $v_0 = (10, 20)$). Esto puede estar relacionado con el hecho de que el valor marginal de la capacidad no se modifica de un período a otro debido a la ausencia de transacciones. Luego, las variables duales son una buena aproximación para el valor marginal de la capacidad.

En este ejemplo en particular, podemos observar que todas las políticas se comportan de forma similar cuando $\alpha = 1.2$ y 1.4. Esto se debe a que en estos casos la longitud del horizonte de reservas deja de imponer una restricción, en el sentido de que las capacidades de los tramos exceden en gran medida las cantidades esperadas de asientos vendidos. Luego, ofrecer todos (o casi todos) los productos es una buena estrategia de venta.

Por último, en esta instancia también podemos ver que los ingresos generados por los métodos basados en el modelado de la elección de los clientes son significativamente mayores en comparación con los obtenidos por INDEP.

α	v_0	COLGEN	V_t^i	Solución de (2.17)	ROPT-0.01	ROPT-0.1	INDEP
0.6	(1,5)	0.05	171.81	0.01	6.64	0.75	0.30
	(5,10)	0.11	246.14	0.02	11.00	1.12	0.31
	(10,20)	0.07	282.52	0.02	6.08	0.64	0.31
0.8	(1,5)	0.07	285.43	0.03	7.71	0.83	0.42
	(5,10)	0.15	372.04	0.09	10.41	1.03	0.42
	(10,20)	0.05	406.98	0.02	4.19	0.41	0.43
1.0	(1,5)	0.09	409.9	0.06	7.43	0.78	0.54
	(5,10)	0.08	498.79	0.04	4.52	0.44	0.53
	(10,20)	0.05	518.30	0.02	2.74	0.27	0.54
1.2	(1,5)	0.09	506.72	0.06	7.09	0.74	0.64
	(5,10)	0.04	616.86	0.02	2.74	0.27	0.64
	(10,20)	0.04	636.43	0.01	2.29	0.23	0.67
1.4	(1,5)	0.07	599.47	0.05	6.53	0.67	0.75
	(5,10)	0.04	729.83	0.02	2.62	0.26	0.78
	(10,20)	0.03	737.17	0.01	2.29	0.22	0.79

Cuadro 4.8: Tiempos computacionales (en segundos) para la instancia Red Pequeña

El Cuadro 4.8 muestra los tiempos computacionales (en segundos) requeridos por los distintos métodos para los escenarios estudiados. La columna COLGEN corresponde al tiempo de optimización del CDLP, incluyendo la resolución del subproblema de generación de columnas. La cuarta columna es el tiempo de llenado por parte de la política DCOMP de las tablas $\hat{V}_t^i(x_i)$, para todos los tramos $i = 1, \ldots, m$, mientras que la siguiente corresponde al tiempo requerido para resolución del problema (2.17) mediante la técnica de linealización. Las columnas 6 y 7 muestran el tiempo utilizado por la política ROPT, para valores 0.01 y 0.1 de *freq* respectivamente. Esta medición corresponde al promedio del tiempo acumulado de reoptimización del CDLP a lo largo del horizonte de reservas. Finalmente, la última columna corresponde al tiempo de resolución del DP (2.18) del modelo de demanda independiente.

La primera observación respecto a esta tabla es que todos los métodos considerados son aplicables desde el punto de vista computacional. Claramente, la construcción de las tablas $V_t^i(x_i)$ es el proceso que mayor tiempo de cómputo requiere, pero es importante recordar que éste corresponde a una tarea que puede ser realizada en forma *off line* mediante procesos *batch*. Además, éste calculo puede ser ejecutado en forma paralela generando un subproceso para cada tabla.

Eventualmente, si el tiempo de llenado de las tablas se vuelve restrictivo en redes más grandes, la aerolínea puede utilizar las políticas CDLP o ROPT, y aún obtener buenas soluciones.

Los resultados nos muestran también que tanto el cálculo dinámico de los *offer sets* como la resolución del CDLP es muy eficiente, requiriendo solamente una pequeña fracción de segundo. En particular, para el algoritmo de generación de columnas, esto se debe a que la cantidad de iteraciones del mismo es muy chica comparada con el número total de variables del CDLP. La Figura 4.3 muestra el número de iteraciones del algoritmo GENCOL para los distintos factores de escala considerados.

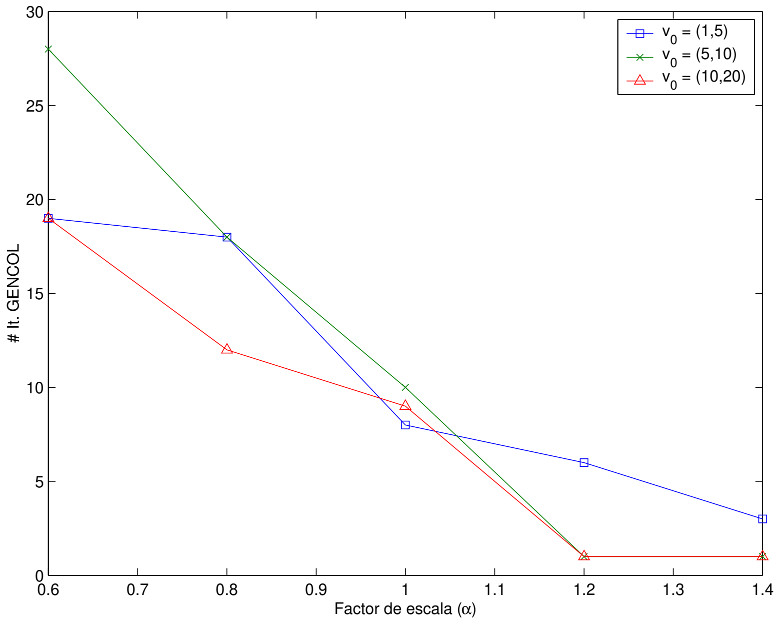


Figura 4.3: Cant. de iter. de GENCOL en función del α y v_0 para la instancia Red Pequeña.

Este ejemplo tiene definidos 22 productos, produciendo un total de $2^{22} - 1 = 4{,}194{,}303$ variables en el CDLP. Sin embargo, la cantidad máxima de veces que itera el algoritmo en los casos testeados es 28, un número significativamente menor que la cantidad de variables. Sumado a esto, sólo fue necesario utilizar la linealización para identificar una columna entrante en una única iteración, que representa menos del 1 % de la cantidad total de iteraciones realizadas por el algoritmo GENCOL considerando todos los escenarios.

Otro dato interesante es lo que sucede cuando los vectores de preferencia de no-compra son $v_0 = (5, 10)$ ó $v_0 = (10, 20)$. Para escenarios con *load factor* bajo, en particular $\alpha = 1.2$, 1.4, el algoritmo de generación de columnas sólo itera una vez. En estos casos, la solución óptima tiene una única variable estrictamente positiva, y corresponde a tomar $t(S) = N$, es decir, el conjunto de todos los productos. Esto se debe a que las capacidades no imponen ninguna restricción sobre el problema; el único factor crítico es la longitud del horizonte de reservas (T).

Por último, otra característica que se desprende del gráfico es que la cantidad de iteraciones para cada v_0 tiende a decrecer a medida que α se incrementa. Este decrecimiento en la cantidad de iteraciones puede estar relacionado con el hecho de que, a medida que las capacidades crecen, dejan de ser una retricción fuerte del problema, llevando a que el algoritmo GENCOL necesite menos pasos para encontrar la solución óptima del CDLP.

4.2.3. Red ***Hub and Spoke***

La red de este ejemplo es la que muestra la Figura 4.4, y se basa en uno de los casos considerados en van Ryzin y Vulcano [11] y Williamson [12]. Los nombres de la figura corresponden a los códigos de aeropuerto de distintas ciudades, que son: Los Ángeles (LAX), Atlanta (ATL), Boston (BOS), Savannah (SAV) y Miami (MIA).

La red contiene cinco aeropuertos y ocho tramos. Hay diez itinerarios ida y vuelta. Cada itinerario esta dividido en cuatro clases de tarifas (Y, M, B y Q). Cada producto es una combinación entre un itinerario de ida y una clase de tarifa. Por ejemplo, el producto 1 corresponde al origen-destino ATLBOS para la clase Y, con tarifa $r_1 = 310$; el producto 2 corresponde al itinerario ATLBOS para la clase M, cuya tarifa es $r_2 = 290$; el producto 5 corresponde al itinerario BOSATL para la clase Y, cuya tarifa es $r_5 = 310$, etc. El número total de productos es 80. El Cuadro 4.9 muestra la definición de los mismos.

Al igual que en el ejemplo anterior, esta instancia también tiene un punto de conexión (ATL), pero el nivel de congestión es mayor dado que hay más combinaciones posibles de itinerario y tarifa.

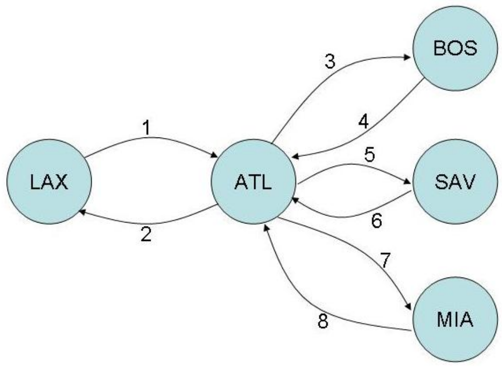


Figura 4.4: Red para la instancia *Hub and Spoke*.

Origen - Destino	Tramos	Tarifa			
		Y	M	B	Q
ATLBOS/BOSATL	3/4	310	290	95	69
ATLLAX/LAXATL	2/1	455	391	142	122
ATLMIA/MIAATL	7/8	280	209	94	59
ATLSAV/SAVATL	5/6	159	140	64	49
BOSLAX/LAXBOS	{4,2}/{1,3}	575	380	159	139
BOSMIA/MIABOS	{4,7}/{8,3}	403	314	124	89
BOSSAV/SAVBOS	{4,5}/{6,3}	319	250	109	69
LAXMIA/MIALAX	{1,7}/{8,2}	477	239	139	119
LAXSAV/SAVLAX	{1,5}/{6,2}	502	450	154	134
MIASAV/SAVMIA	{8,5}/{6,7}	226	168	84	59

Cuadro 4.9: Definición de productos para la red *Hub and Spoke*.

Segmento	C_l	v_l	λ_l	Segmento	C_l	v_l	λ_l
ATL/BOS H	{1,2,3,4}	{6,7,9,10}	0.015	BOS/MIA H	{41,42,43,44}	{6,7,10,10}	0.008
ATL/BOS L	{3,4}	{8,10}	0.035	BOS/MIA L	{43,44}	{8,10}	0.03
BOS/ATL H	{5,6,7,8}	{6,7,9,10}	0.015	MIA/BOS H	{45,46,47,48}	{6,7,10,10}	0.008
BOS/ATL L	{7,8}	{8,10}	0.035	MIA/BOS L	{47,48}	{8,10}	0.03
ATL/LAX H	{9,10,11,12}	{5,6,9,10}	0.01	BOS/SAV H	{49,50,51,52}	{5,6,9,10}	0.01
ATL/LAX L	{11,12}	{10,10}	0.04	BOS/SAV L	{51,52}	{8,10}	0.035
LAX/ATL H	{13,14,15,16}	{5,6,9,10}	0.01	SAV/BOS H	{53,54,55,56}	{5,6,9,10}	0.01
LAX/ATL L	{15,16}	{10,10}	0.04	SAV/BOS L	{55,56}	{8,10}	0.035
ATL/MIA H	{17,18,19,20}	{5,5,10,10}	0.012	LAX/MIA H	{57,58,59,60}	{5,6,10,10}	0.012
ATL/MIA L	{19,20}	{8,10}	0.035	LAX/MIA L	{59,60}	{9,10}	0.028
MIA/ATL H	{21,22,23,24}	{5,5,10,10}	0.012	MIA/LAX H	{61,62,63,64}	{5,6,10,10}	0.012
MIA/ATL L	{23,24}	{8,10}	0.035	MIA/LAX L	{63,64}	{9,10}	0.028
ATL/SAV H	{25,26,27,28}	{4,5,8,9}	0.01	LAX/SAV H	{65,66,67,68}	{6,7,10,10}	0.016
ATL/SAV L	{27,28}	{7,10}	0.03	LAX/SAV L	{67,68}	{10,10}	0.03
SAV/ATL H	{29,30,31,32}	{4,5,8,9}	0.01	SAV/LAX H	{69,70,71,72}	{6,7,10,10}	0.016
SAV/ATL L	{31,32}	{7,10}	0.03	SAV/LAX L	{71,72}	{10,10}	0.03
BOS/LAX H	{33,34,35,36}	{5,5,7,10}	0.01	MIA/SAV H	{73,74,75,76}	{6,7,8,10}	0.01
BOS/LAX L	{35,36}	{9,10}	0.032	MIA/SAV L	{75,76}	{9,10}	0.025
LAX/BOS H	{37,38,39,40}	{5,5,7,10}	0.01	MIA/SAV H	{77,78,79,80}	{6,7,8,10}	0.01
LAX/BOS L	{39,40}	{9,10}	0.032	MIA/SAV L	{79,80}	{9,10}	0.025

Cuadro 4.10: Definición de segmentos para la red *Hub and Spoke*.

Esta instancia tiene definidos 40 segmentos de clientes. Por cada posible origen - destino, dividimos a los clientes en dos grupos: uno de compradores que consideran todos los productos, pero con mayor preferencia los de tarifas bajas, y otro que solo considera vuelos de tarifas bajas (B y Q). Para ejemplificar esta situación, tengamos en cuenta el mercado ATL/BOS. Los segmentos definidos son:

- ATL/BOS H: segmento de clientes que consideran todos los productos disponibles de ATL a BOS (Y, M, B y Q), con valores de preferencia mayores para los productos de tarifas más bajas.
- ATL/BOS L: segmento de clientes que consideran sólo los productos B y Q de ATL a BOS, con mayor valor de preferencia por el producto Q.

El Cuadro 4.10 muestra la definición de todos los segmentos considerados. El horizonte de reservas consiste en T =2000 instantes de tiempo, con un promedio de 1732 arribos por cada *stream* de demanda.

Al igual que en los ejemplos anteriores, reportamos los ingresos esperados y los *load factors* obtenidos por cada una de las políticas (Cuadro 4.11), y los *gaps* de las restantes políticas respecto a DCOMP (Cuadro 4.12). La notación utilizada para los valores de preferencia de no-compra de cada escenario, v_0, es la misma que la utilizada en el ejemplo Red Pequeña.

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		RCDLP		INDEP	
			Media	%LF	Media	%LF	Media	%LF	Media	%LF	Media	%LF
0.6	(1,5)	163,897	160,624	97.10	160,206	95.03	156,557	95.70	156,410	95.72	110,471	98.64
	(5,10)	132,674	130,971	97.68	130,875	97.29	126,425	95.45	126,822	95.38	104,330	98.30
	(10,20)	111,897	110,314	97.61	110,209	96.93	106,688	95.53	106,879	95.45	96,661	97.85
0.8	(1,5)	177,384	175,598	97.70	173,520	93.66	170,301	96.05	170,562	96.13	130,841	98.72
	(5,10)	146,338	144,597	97.44	144,377	96.99	140,857	95.90	140,671	95.93	123,399	98.37
	(10,20)	122,464	121,062	96.24	120,985	96.14	117,621	96.03	117,654	96.07	114,012	97.53
1.0	(1,5)	187,270	185,384	96.43	184,785	95.99	181,673	95.57	181,751	95.60	149,246	98.63
	(5,10)	156,243	154,718	94.52	154,508	94.21	151,907	95.03	151,832	95.02	140,161	98.03
	(10,20)	128,386	127,343	91.65	127,255	91.88	125,811	92.27	125,883	92.27	126,091	92.09
1.2	(1,5)	195,269	193,511	94.88	192,953	94.89	190,000	93.71	190,248	93.70	165,880	98.29
	(5,10)	160,206	159,386	87.28	159,354	87.37	157,877	87.36	157,922	87.35	154,210	95.57
	(10,20)	128,448	128,336	78.36	128,336	78.36	128,336	78.36	128,336	78.36	128,361	78.38
1.4	(1,5)	197,113	196,886	86.70	196,860	86.77	196,639	86.79	196,639	86.79	179,983	96.54
	(5,10)	160,453	160,350	76.28	160,352	76.28	160,350	76.28	160,350	76.28	159,435	85.54
	(10,20)	128,448	128,336	68.22	128,336	68.22	128,336	68.22	128,336	68.22	128,363	68.24

Cuadro 4.11: Resultados de las simulaciones para la instancia *Hub and Spoke*.

Nuevamente, los resultados obtenidos por las políticas DCOMP y DCOMP-0.5 son muy similares, pero en este ejemplo la primera de ellas genera resultados levemente mejores en todos los casos. En 14 de los 15 escenarios, esta diferencia no supera el 0.32 %. En el único caso donde el *gap* entre las políticas es significativo se da para $\alpha = 0.8$ y $v_0 = (1,5)$, donde los ingresos generados por

DCOMP son un 1.18 % mayores. Este resultado también se ve reflejado en la diferencia de los *load factors* obtenidos por cada una de ellas. El *load factor* producido por DCOMP es aproximadamente un 4 % mayor que el producido por DCOMP-0.5, indicando que esta diferencia en los ingresos generados puede estar relacionada con el promedio de asientos vendidos por cada una de ellas. Es importante destacar que el hecho de que una política produzca un *load factor* mayor que el de otra no necesariamente implica ingresos mayores. Esta relación depende de las decisiones tomadas en cada instante de tiempo por cada método.

Por otro lado, la comparación entre las políticas CDLP y RCDLP sigue siendo consistente con lo visto en las instancias anteriores: ninguna domina a la otra.

α	v_0	% DCOMP-0.5	% CDLP	% RCDLP	% INDEP
0.6	(1,5)	0.26 ± 0.42	2.53 ± 0.38	2.62 ± 0.37	31.22 ± 0.33
	(5,10)	0.07 ± 0.41	3.47 ± 0.40	3.16 ± 0.39	20.34 ± 0.35
	(10,20)	0.09 ± 0.41	3.28 ± 0.40	3.11 ± 0.41	12.37 ± 0.37
0.8	(1,5)	1.18 ± 0.38	3.01 ± 0.36	2.86 ± 0.35	25.48 ± 0.31
	(5,10)	0.15 ± 0.37	2.58 ± 0.36	2.71 ± 0.36	14.66 ± 0.31
	(10,20)	0.06 ± 0.37	2.84 ± 0.36	2.81 ± 0.36	5.82 ± 0.33
1.0	(1,5)	0.32 ± 0.33	2.00 ± 0.32	1.96 ± 0.32	19.49 ± 0.27
	(5,10)	0.13 ± 0.34	1.81 ± 0.32	1.86 ± 0.32	9.40 ± 0.29
	(10,20)	0.06 ± 0.36	1.20 ± 0.35	1.14 ± 0.35	0.98 ± 0.35
1.2	(1,5)	0.28 ± 0.30	1.81 ± 0.29	1.68 ± 0.29	14.27 ± 0.25
	(5,10)	0.02 ± 0.32	0.94 ± 0.31	0.91 ± 0.31	3.24 ± 0.28
	(10,20)	0.0 ± 0.39	0.0 ± 0.39	0.0 ± 0.39	-0.01 ± 0.40
1.4	(1,5)	0.00 ± 0.31	0.11 ± 0.31	0.11 ± 0.31	8.57 ± 0.27
	(5,10)	0.00 ± 0.35	0.00 ± 0.35	0.00 ± 0.35	0.57 ± 0.32
	(10,20)	0.00 ± 0.39	0.00 ± 0.39	0.00 ± 0.39	-0.02 ± 0.40

Cuadro 4.12: Intervalos de conf. del 95 % para los ingresos generados respecto a DCOMP en la instancia *Hub and Spoke*.

El mensaje principal de ambas tablas es que tener en cuenta el comportamiento selectivo de los clientes incrementa considerablemente los ingresos, en particular para *load factors* altos. Las mejoras por parte de DCOMP respecto al modelo de demanda independiente, INDEP, superan el 10 %. A medida que hay más capacidad disponible en los tramos, y éstas dejan de ser restrictivas, los ingresos se equiparan. Nuevamente, estos resultados son análogos a los obtenidos en los ejemplos anteriores.

También podemos notar diferencias considerables en el *load factor* producido por las políticas basadas en elección e INDEP. En algunos escenarios, INDEP llega a generar *load factors* hasta un 10 % mayores que las demás políticas, pero el ingreso es bastante menor. Esta observación también se da en el ejemplo anterior, aunque la tendencia no es tan marcada. Ésto está indicando que INDEP vende más asientos que las políticas basadas en elección, pero que al mismo tiempo, vende “peor”. En otras palabras, considerar el comportamiento selectivo de los clientes cambia el conjunto de productos ofrecidos, y por lo tanto el conjunto de pasajeros a los que se les realiza una venta. Luego, en términos de ingresos generados, DCOMP y las políticas basadas en elección están tomando mejores decisiones a la hora de vender.

La política DCOMP también mejora considerablemente a CDLP y RCDLP en 10 de los 15 escenarios, donde los *gaps* respecto de los ingresos generados son superiores al 1 %. La diferencia se debe a que estos escenarios corresponden a situaciones donde las decisiones acerca de vender o no un determinado producto se vuelven críticas, y DCOMP responde mejor dado que puede ajustarse a las variaciones en el tiempo y las capacidades remanentes. En los cinco casos restantes las capacidades se incrementan y todas las políticas tienden a comportarse de la misma forma. Ésto se debe a que, al igual que en las instancias anteriores, el factor restrictivo es la longitud del horizonte de reservas en lugar de las capacidades de los vuelos.

α	v_0	COLGEN (seg.)	V_t^i (min.)	Solución de (2.17) (seg.)	INDEP (seg.)
0.6	(1,5)	73.31	168.28	0.005	3.62
	(5,10)	0.70	205.11	0.006	3.66
	(10,20)	1.53	228.25	0.007	3.73
0.8	(1,5)	851.36	245.9	0.006	4.93
	(5,10)	5.75	295.40	0.007	4.99
	(10,20)	10.60	327.94	0.008	5.10
1.0	(1,5)	3184.22	341.30	0.006	6.31
	(5,10)	3.01	393.24	0.008	6.40
	(10,20)	0.25	422.22	0.008	6.39
1.2	(1,5)	1188.89	442.15	0.007	7.67
	(5,10)	10.55	490.50	0.007	7.67
	(10,20)	0.13	516.08	0.008	7.68
1.4	(1,5)	107.52	548.21	0.007	8.94
	(5,10)	0.85	576.21	0.007	8.96
	(10,20)	0.13	584.09	0.007	8.85

Cuadro 4.13: Tiempos computacionales para la instancia *Hub and Spoke*.

Los tiempos computacionales los mostramos en el Cuadro 4.13. De esta tabla se desprende que todos los métodos son computacionalmente factibles. La política INDEP requiere menos de diez segundos para resolver el DLP (2.18). Nuevamente, el proceso que más tiempo consume es el cálculo de las tablas $V_t^i(x_i)$. Dependiendo del escenario, el tiempo requerido oscila entre dos horas y media y las diez horas. Ésto es razonable dado que el horizonte de reservas cuenta con 2,000 instantes, y las capacidades de tramos varían entre 120 y 280 asientos cada uno, dependiendo del factor de escala α. La cantidad de problemas del tipo (2.13) a resolver en cada caso es considerablemente alta.

A diferencia de los ejemplos anteriores, en este caso en particular utilizamos la heurística golosa para la resolución del problema (2.17) (el que debe resolver la compañía en cada instante del horizonte de reservas para determinar dinámicamente cuál es el *offer set* a mostrar). Tomamos esta decisión debido a que este ejemplo posee definidos ochenta productos, un número significativo comparado con los casos anteriores. La quinta columna de la tabla muestra que el tiempo promedio requerido por la heurística es menor a una décima de segundo. A su vez, teniendo en cuenta los ingresos obtenidos de las simulaciones, podemos ver que utilizar este algoritmo aproximado no afecta la performance de la política. Luego, utilizar la heurística es una buena alternativa tanto desde el punto de vista computacional como de la calidad de los ingresos generados por el método.

Un resultado distinto a los obtenidos en los ejemplos anteriores es el tiempo que requiere el algoritmo GENCOL. En primer lugar, es importante destacar que desde el punto de vista práctico la resolución del CDLP (2.8) sigue siendo factible dado que los tiempos de resolución son más que razonables. En 10 de los 15 escenarios se obtiene una solución óptima en menos de 11 segundos cuando este ejemplo tiene $2^{80}-1$ variables $t(S)$. Sin embargo, en los 5 casos restantes los tiempos se incrementan considerablemente. En particular, éstos corresponden a los escenarios donde el vector de preferencias de no-compra v_0 es $(1, 5)$.

Para analizar este comportamiento consideramos la Figura 4.5, que muestra el número de iteraciones del algoritmo GENCOL para cada v_0 en función del factor de escala α. Nuevamente vemos que hay una tendencia decreciente en el número de iteraciones a medida que α crece. También se da en este caso que para $\alpha = 1.4$, todos los métodos convergen en una iteración, y esto se debe a que las soluciones están compuestas por un único *offer set* que, si bien varía dependiendo de v_0, considera casi todos los productos.

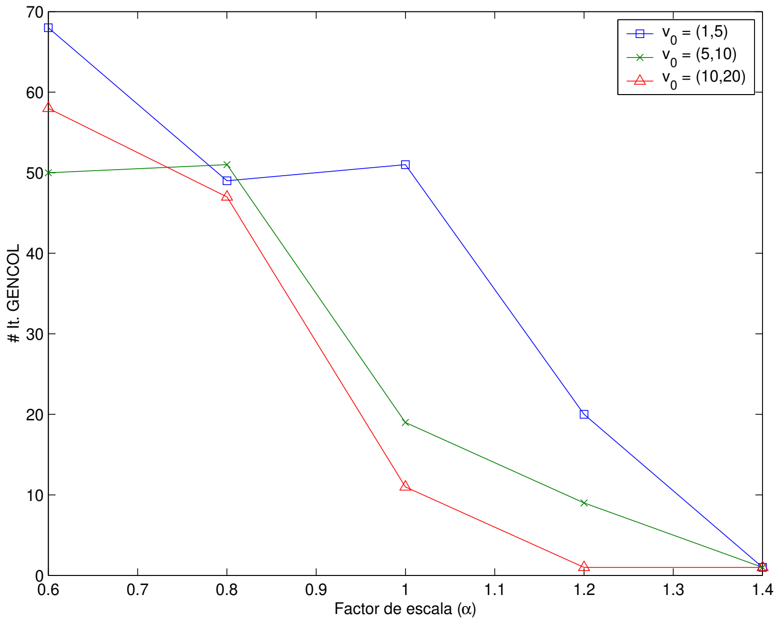


Figura 4.5: Cant. de iter. de GENCOL en función del α y v_0 para la instancia *Hub and Spoke*.

No obstante, la diferencia en los tiempos no se debe únicamente a la cantidad de iteraciones (dado que los tiempos del algoritmo GENCOL no decrecen a medida que α se incrementa), sino que ahora también influye el tiempo de resolución del subproblema de generación de columnas. En esta instancia, al haber un gran número de productos, el algoritmo GENCOL utiliza con mayor frecuencia la linealización para identificar una columna entrante. Sobre un total de 437 iteraciones (considerando todos los escenarios), es necesario recurrir a la linealización en 54, un 12.36 % de las veces. Pero 48 de las 54 veces corresponden a casos donde $v_0 = (1, 5)$: 24 para $\alpha = 0.6$; 11 para $\alpha = 0.8$; 10 para $\alpha = 1.0$; y 3 para $\alpha = 1.2$. Las otras 6 llamadas a la linealización se dan cuando $\alpha = 0.8$ y $v_0 = (5, 10)$.

Este análisis nos permite hacer dos observaciones. La primera de ellas es que el algoritmo GENCOL, en general, requiere mayor cantidad de tiempo para resolver el CDLP (2.8) cuando los valores de preferencia de no-compra son más chicos. Incluso cuando $\alpha = 1.4$, para $v_0 = (1, 5)$, el algoritmo itera una única vez, pero el tiempo computacional de verificar la condición de optimalidad produce que el algoritmo demore aproximadamente dos minutos.

La segunda observación es que los tiempos de resolución del subproblema de generación de columnas utilizando la linealización pueden ser muy variables. Podemos ver un ejemplo de esta situación al contrastar la cantidad de veces que se recurre a la linealización y el tiempo requerido por el algoritmo GENCOL en los casos $\alpha = 0.8$, $v_0 = (1, 5)$ y $v_0 = (5, 10)$. La proporción entre el tiempo requerido por cada método es mucho mayor que la de la cantidad de iteraciones, y no guarda relación alguna. Aún más: en todos los escenarios se recurre a la linealización para verificar la condición de optimalidad de la solución encontrada hasta el momento, y varios casos el tiempo total del algoritmo no supera el segundo. Ésto muestra la variación de tiempos mencionada anteriormente.

De todos modos, es necesario experimentar sobre redes más grandes para poder corroborar este comportamiento del algoritmo de generación de columnas.

Capítulo 5

Conclusiones

En este trabajo consideramos el modelo determinístico de programación lineal basado en elección (CDLP) de Gallego et al. [4], y lo generalizamos para el caso en el que los compradores pertenecen a segmentos superpuestos y eligen de acuerdo al modelo *multinomial logit*. Para poder resolver el CDLP para redes de tamaño real, desarrollamos un algoritmo de generación de columnas. Probamos que el subproblema de generación de columnas resultante, un problema de programación hiperbólica 0-1, es NP-Completo, y utilizamos una heurística golosa para resolverlo. Esta heurística funciona muy bien en la práctica, tanto desde el punto de vista computacional como en la calidad de las soluciones obtenidas. También presentamos una formulación lineal entera mixta para resolver el subproblema de generación de columnas en forma exacta, para aquellos casos donde la heurística golosa no puede identificar una columna entrante. De acuerdo a nuestra experiencia, en la mayoría de los casos la heurística golosa es suficiente para esta tarea.

También extendemos la heurística de descomposición propuesta por van Ryzin y Liu [10] para el caso del MNL con segmentos superpuestos. Comparamos su comportamiento con distintos métodos en una serie exhaustiva de tests computacionales. Los resultados que obtuvimos están alineados con los reportados por van Ryzin y Liu [10]. La heurística de descomposición mejora considerablemente los resultados obtenidos por el resto de los métodos en la gran mayoría de los casos, sobre todo cuando el *load factor* es alto. Sin embargo, las otras políticas también muestran una buena performance, produciendo resultados de alta calidad, mejorando en muchos casos al comportamiento del modelo basado en la asunción de demanda independiente por más de un 10 % (sobre todo para *load factors* altos). Ésto confirma que incluir el comportamiento selectivo de los clientes en el modelado y la optimización de los controles tiene un alto impacto desde la perspectiva del Revenue Management.

Nuestros experimentos también muestran que todas las políticas son factibles en la práctica, computacionalmente hablando. Los tiempos computacionales reportados muestran que la construcción de las tablas requeridas por la heurística de descompsición es el proceso más costoso. Sin embargo, este cálculo puede ser realizado en forma *off line* y puede ser fácilmente paralelizado para acelerar los tiempos de cómputo. De todos modos, si los tiempos requeridos por la heurística de descomposición se vuelven restrictivos, alguno de los otros métodos basados en modelos de elección puede ser utilizado y obtener igualmente soluciones de buena calidad. Considerando un balance entre el costo computacional y los ingresos esperados generados, CDLP parece ser una buena alternativa a la heurística de descomposición. En relación al algoritmo de generación de columnas, los tiempos obtenidos muestran que puede ser resuelto en tiempos razonables para instancias de tamaño mediano, y por lo tanto también puede ser aplicado en la práctica.

Todas las políticas pueden ser implementadas en la práctica debido a los tiempos computacionales que requieren. Sin embargo, todavía es posible realizar algunas mejoras.

Para el algoritmo de generación de columnas, la linealización comienza a utilizarse con mayor frecuencia en la tercera instancia, la red *Hub and Spoke*. Si bien los tiempos obtenidos en la resolución de subproblema de generación de columnas mediante esta técnica son razonables, sería interesante ir un paso más allá y buscar propiedades particulares del problema para utilizarlas en un algoritmo más inteligente. También resultaría útil evaluar si distintas estrategias de selección de columnas mejoran la performance del algoritmo (por ejemplo, identificar posibles columnas que no vayan a ser utilizadas y removerlas, o probar de agregar más de una columna por iteración con costos reducidos positivos, etc.).

Desde el punto de vista de los experimentos realizados, sería interesante evaluar las distintas políticas para redes de gran escala. Un paso imprescindible para poder llevar a cabo estas pruebas sería paralelizar el cálculo de las tablas de programación dinámica. Esta tarea no presenta mayores dificultades, dado que el problema es fácilmente separable en m subprocesos distintos, uno para cada tramo de la red. Una idea similar podría aplicarse para las simulaciones, procesando de forma simultánea distintos *streams* de arribos.

También es necesario realizar mayor experimentación para poder explicar el resultado obtenido en la instancia Red Pequeña, para el escenario donde $\alpha = 0.6$ y $v_0 = (1, 5)$. En van Ryzin y Liu [10] reportan que este tipo de resultados, en un caso con algunas semejanzas al nuestro, se debe a la existencia de múltiples soluciones duales óptimas ya que la solución del CDLP es degenerada. En nuestro caso, la solución óptima del CDLP no cumple con esta propiedad, y no pudimos encontrar la razón que justifica este comportamiento diferenciado. Por eso consideramos que es necesario realizar pruebas sobre más instancias, con el fin de identificar y fundamentar correctamente en que situaciones la heurística de descomposición presenta estas anomalías y también estudiar a fondo los efectos degenerativos reportados en van Ryzin y Liu [10].

Bibliografía

[1] R Bellman. *Applied Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[2] P.P. Belobaba and C. Hopperstad. Boeing/mit simulation study: Pods results update. In *1999 AGIFORS Reservations and Yield Management Study Group Symposium Proceedings*, London, UK, 1999.

[3] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Applications to Travel Demand*. The MIT Press, Cambridge, MA, sixth edition, 1994.

[4] G. Gallego, G. Iyengar, R. Phillips, and A. Dubey. Managing flexible products on a network. Technical Report CORC TR-2004-01, Department of Industrial Engineering and Operations Research, Columbia University, 2004.

[5] M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, CA, 1979.

[6] O. Prokopyev, H. Huang, and P. Pardalos. On complexity of unconstrained hyperbolic 0–1 programming problems. *Operations Research Letters*, 33:312–318, 2005.

[7] O. Prokopyev, C. Meneses, C. A. S. Oliveira, and P. M. Pardalos. On multiple-ratio hyperbolic 0-1 programming problems. *Pacific Journal of Optimization*, 1(2):327–345, 2005.

[8] K. Talluri and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, New York, NY, 2004.

[9] K. T. Talluri and G. J. van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50:15–33, 2004.

[10] G.J. van Ryzin and Q. Liu. On the choice-based linear programming model for network revenue management. Working paper, Graduate School of Business, Columbia University, New York, 2004.

[11] G.J. van Ryzin and G. Vulcano. Computing virtual nesting controls for network revenue management under customer choice behavior. Working paper, Graduate School of Business, Columbia University, New York, 2004.

[12] E. Williamson. *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts.* PhD thesis, Flight Transportation Laboratory, MIT, Cambridge, MA, 1992.

[13] T. Wu. A note on a global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 101:220–223, 1997.

[14] D. Zhang and W. L. Cooper. Revenue management for parallel flights with customer-choice behavior. *Operations Research*, 53:415–431, 2005.