

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

## Cálculo de bid prices para la gestión de ingresos bajo preferencias de clientes

ALUMNO: Juan Manuel Chaneton  
L.U. 577/03  
jchaneton@dc.uba.ar

DIRECTOR: Dr. Gustavo Vulcano  
gvulcano@stern.nyu.edu  
Stern School of Bussines, New York University, New York

Marzo de 2009

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



# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Generalidades Sobre el Revenue Management . . . . .	9
1.1.1. ¿Qué es <i>Revenue Management</i> ? . . . . .	9
1.1.2. Los orígenes del <i>Revenue Management</i> . . . . .	10
1.1.3. Un marco conceptual para <i>Revenue Management</i> . . . . .	11
1.1.4. Descripción de un sistema de <i>Revenue Management</i> . . . . .	14
1.2. Aspectos relativos al problema . . . . .	15
1.2.1. El control de capacidad . . . . .	15
1.2.2. Modelos de Demanda . . . . .	18
1.2.3. Optimización basada en simulación . . . . .	20
1.2.4. Sobre este trabajo . . . . .	22
<b>2. Modelo</b>	<b>25</b>
2.1. Definiciones generales . . . . .	25
2.2. Modelo de Demanda . . . . .	26
2.3. Generalidad del modelo de demanda . . . . .	29
<b>3. Método</b>	<b>33</b>
3.1. Las funciones de aceptación e ingreso . . . . .	33
3.2. Las derivadas de las funciones de aceptación e ingreso . . . . .	35
3.3. Cálculo de las derivadas . . . . .	42
3.4. Esquema de alto nivel del método . . . . .	48
3.5. Algunas consideraciones restantes . . . . .	49
<b>4. Experimentos numéricos</b>	<b>55</b>
4.1. Aplicación directa del método . . . . .	56
4.1.1. Ejemplo 1: Tipos de cliente fijos . . . . .	56
4.1.2. Ejemplo 2: MNL con segmentos . . . . .	59
4.1.3. Ejemplo 3: Compras con <i>Buyups</i> . . . . .	68
4.2. Experimentos con Reoptimización . . . . .	71
4.2.1. Ejemplo 1 con reoptimizaciones . . . . .	72

4.2.2. Ejemplo 2 con reoptimizaciones . . . . .	73
4.2.3. Ejemplo 3 con reoptimizaciones . . . . .	76
<b>5. Conclusiones</b>	<b>79</b>



## Resumen

En los últimos años ha habido un creciente interés en resolver los problemas de *Revenue Management* a nivel de red, y en particular bajo modelos de elección. En este contexto, se han propuesto diversos enfoques para tratar con el problema del control de capacidad, utilizando diferentes políticas. En particular, consideramos aquí la política de bid prices, que consiste en asignar un valor a cada uno de los recursos de la red y sólo aceptar la solicitud de compra de un producto si el precio del mismo es mayor que la suma de los valores de los recursos que utiliza. Para capturar el comportamiento de elección utilizamos un modelo muy general de demanda, siguiendo el estudiado por van Ryzin y Vulcano[vRV08a], en el que las decisiones de los clientes se modelan a través de una secuencia estocástica de tipos de clientes, cada uno con una lista de preferencias.

Propondremos un método de optimización basado en simulación (una extensión al propuesto por Topaloglu[Top08]), que utiliza la política de bid prices, y demanda y capacidad continuas, lo que permite la aceptación parcial de solicitudes de compra. El modelo admite un cálculo eficiente del gradiente muestral de la función de ingreso, que es utilizado para un algoritmo estocástico de maximización por “dirección más empinada”. Para evaluar la eficacia del algoritmo y, a la vez, evidenciar la generalidad del modelo de demanda, consideramos 3 sets de experimentos, correspondiente a modelos de elección ampliamente utilizados. En la mayoría de los casos el método produce mejoras significativas en el ingreso respecto de DLP(*Programa Lineal Determinístico*), un método tradicional de optimización de controles, basado en la política de bid prices, que trabaja con demandas independientes. También comparamos el ingreso generado con CDLP(*Programa Lineal Determinístico basado en Elección*), un método de optimización recientemente propuesto, para problemas de red, y que trabaja con modelos de elección, con resultados prometedores.

## Abstract

In the past few years there has been an increasing interest in solving Revenue Management problems at the network level, and especially under customer choice demand models. In this context, several approaches have been proposed to solve the capacity control problem, using different types of policies. We consider the bid prices policy for network problems, where every resource in the network has an assigned value, and a product request is accepted only if its price is greater than the sum of the values of the resources that the product uses. In order to account for customer choice behavior we consider a very general demand model, following the one studied by van Ryzin and Vulcano[vRV08a], where customer choice preferences are modeled an ordered list of alternatives. Minimal assumptions are made about the statistical properties of this demand sequence.

We propose a simulation-based method, extending the one proposed in Topaloglu[Top08], using the bid-price policy, and assuming a continuous demand and capacity setting, which allows for the partial acceptance of requests. The model admits an efficient calculation of the sample path gradient of the network revenue function, that is used to construct a stochastic steepest ascent algorithm. In order to test the algorithm and, at the same time, show the generality of the demand model, we consider three different experiment sets, each one corresponding to a widely-used choice

based model. In most of the cases the method produces significant revenue increases relative to DLP(*Deterministic Linear Program*), a traditional independent demand bid-price based method. We also compare the revenue performance against CDLP(*Choice based Deterministic Linear Program*), a well known choice based network method, with promising results.



# Capítulo 1

## Introducción

La disciplina de management cuantitativo conocida como *Revenue Management* se ocupa de establecer políticas de control sobre distintos recursos con el objetivo de maximizar los ingresos. Se puede utilizar en diferentes áreas de servicios, por ejemplo en las aerolíneas, hoteles, ferrocarriles, etc. En este trabajo, desarrollaremos un método de optimización del ingreso, basado en una política clásica de control de recursos, y que incorpora explícitamente el comportamiento selectivo de los clientes.

Este capítulo se encuentra organizado de la siguiente manera: comenzaremos con una introducción a los fundamentos del *Revenue Management*, incluyendo aspectos históricos generales, conceptos básicos, y áreas de aplicación, y luego, desarrollaremos aspectos más específicos, relacionados con el problema que vamos a resolver.

### 1.1. Generalidades Sobre el Revenue Management

#### 1.1.1. ¿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 aún más complejas, relacionadas con la demanda. Por ejemplo, ¿cómo puede una compañía segmentar a sus clientes ofreciéndoles diferentes términos contractuales y condiciones que exploten los diferentes comportamientos o disponibilidades al comprar sus productos? Si esta segmentación es posible, ¿qué precios debe asignarle a cada segmento? ¿cómo deben ajustarse estos precios para cada producto o servicio a través del tiempo en función de factores estacionales y la evolución de las ventas? etc.

La disciplina conocida como *Revenue Management* se encarga de este tipo de decisiones relacionadas con la demanda, y de los procesos y sistemas requeridos para implementarlas. Involucra el manejo de la interfaz de la compañía 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 antiguo. 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 sí 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 de la toma de decisiones.

#### **1.1.2. Los orígenes del *Revenue Management***

Los orígenes del *Revenue Management* están directamente relacionados con la Ley de Desregulación Aérea 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 reguladas. 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 tarifas 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, viajeros 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 ofreciendo 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 cuál era efectivamente la cantidad de asientos en este excedente, para poder ofrecer productos con tarifas más bajas. En segundo lugar, tenían que asegurarse de no perder a los viajeros 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 mucho 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 igualaba o mejoraba las 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 afectó 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.

### 1.1.3. 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 cuándo 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 tipos clientes a los que los mismos 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.

Las soluciones 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 comportamiento del 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 otras compañías, 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 y optimización*: cuando se realiza la estimación de la demanda de la compañía, se la utiliza para efectuar una optimización y obtener, como resultado, controles que se aplicarán en el mercado. Una vez aplicados, la reacción de la competencia se evalúa a través de una nueva estimación de la demanda, que representa el comienzo de un nuevo ciclo.

## **Relaciones entre las decisiones**

Si la gestión de demanda y 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 recursos 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. En el caso de las aerolíneas, aceptar el pedido por un determinado producto puede significar la negación de otro 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 sustitutos 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 predecir 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 extras en un 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 estratégicamente 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 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.

#### 1.1.4. Descripción de un sistema de *Revenue Management*

En la práctica, un sistema de *Revenue Management* puede ser separado conceptualmente en distintas capas. Éstas 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 otras medidas relevantes, como por ejemplo, las cancelaciones.
3. *Optimización*: Buscar el conjunto óptimo de controles. Estos controles serán utilizados hasta la próxima reoptimización.
4. *Control*: Aplicar sobre el inventario (por ejemplo, asientos en una aerolínea) el control optimizado.

El proceso de *Revenue Management* involucra ciclar a través de estos pasos. Con qué frecuencia se lleva a cabo cada paso depende de distintos factores, como por ejemplo el volumen de los datos, qué tan rápido cambian éstos, 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<sup>1</sup>.

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

---

<sup>1</sup>Extraído de Talluri y van Ryzin [TvR04a], con permiso de los autores.

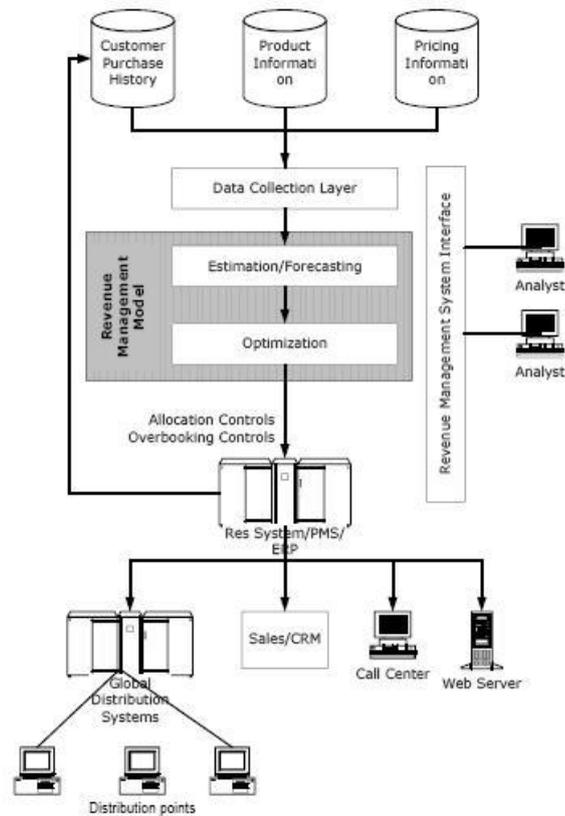


Figura 1.1: Descripción de un sistema de *Revenue Management*.

## 1.2. Aspectos relativos al problema

En esta sección continuaremos introduciendo conceptos relativos al *Revenue Management*, pero orientados al problema particular que nos concierne.

Como se ha mencionado, el *Revenue Management* tiene aplicaciones más allá del negocio de las aerolíneas. Sin embargo, a partir de ahora, y a lo largo del presente trabajo, vamos a asumir que el problema que se resuelve proviene del mundo aerocomercial, con lo que, tanto el modelo matemático como la terminología utilizada, estarán orientadas a dicho contexto. Cabe aclarar que la mayoría de los contenidos desarrollados y los resultados obtenidos se pueden adaptar sin mayor dificultad a otros posibles contextos donde sea aplicable el *Revenue Management*.

### 1.2.1. El control de capacidad

Una de las modalidades de optimización más utilizadas en *Revenue Management* es lo que se conoce como *control de capacidad*, que consiste en administrar los recursos que se poseen y venderlos de manera que se maximice el ingreso<sup>2</sup>. En el caso de las aerolíneas, los recursos son los asientos en un vuelo. Un producto es un pasaje para un itinerario, que puede comprender uno

<sup>2</sup>Una alternativa también ampliamente difundida es el control dinámico de precios

o más vuelos punto a punto (*legs*) conectados; por ejemplo, un vuelo de Buenos Aires a Nueva York puede hacer escala en Santiago de Chile. En general, y en particular para el problema que vamos a resolver, se considera que durante la etapa de optimización el conjunto de productos se mantiene fijo. Teniendo en cuenta que cada producto tiene un ingreso asociado, la tarifa, y “consume” una cantidad determinada de recursos (un asiento por *leg* que utiliza), el problema operativo es entonces decidir qué productos ofrecer (y en qué momento) para maximizar el ingreso. Dicha optimización puede realizarse, a grandes rasgos, en dos niveles: a nivel de *leg* y a nivel de *red*. A nivel de *leg*, la optimización se realiza teniendo sólo en cuenta un vuelo punto a punto por vez: se estima la demanda que habrá para las distintas clases del vuelo y se optimiza en función de ello, obteniendo una solución local a ese vuelo. Este enfoque fue el utilizado por los sistemas tradicionales de *Revenue Management*, principalmente debido la facilidad para recolectar datos, y a la estructura organizacional de la gestión, donde había analistas asignados a cada vuelo. En los últimos años se ha cambiado el enfoque, y se ha empezado a analizar el problema a nivel de red, es decir haciendo la optimización sobre *itinerarios*, que podrían tener vuelos en común. Esta generalización complica el problema, y lo vuelve un verdadero desafío para instancias de tamaño moderado. Son problemas de nivel de red los que se tratan en el presente trabajo.

Para llevar a cabo la optimización se suele tomar una *política* o tipo de control, es decir, una forma en que se administran los recursos. El objetivo es cuantificar o parametrizar la toma de decisiones, de manera que pueda ser sujeta a optimización matemática. A continuación una breve descripción de los tipos de controles más utilizados en *Revenue Management*<sup>3</sup>:

### Límites de reservas y niveles de protección anidados

En la práctica del *Revenue Management* para aerolíneas los productos para un vuelo o itinerario se suelen agrupar de manera jerárquica en *clases*. La forma más simple de hacerlo sería, por ejemplo, una clase por producto, jerarquizados según la tarifa. Los límites de reservas y niveles de protección son controles que limitan la capacidad disponible y la distribuyen entre las diferentes clases. En el caso de los límites de reserva, se asigna una cantidad máxima de asientos que se podrán vender (reservar) para cada clase. Los niveles de protección anidados son el complemento respecto de la capacidad total, donde las clases de mayor jerarquía tienen acceso a la capacidad de las clases de menor jerarquía. En lugar de definir límites para la cantidad máxima de ventas por clase, se definen niveles de protección como una cantidad mínima de asientos que tienen que ser reservados para las clases superiores. Por ejemplo, supongamos que un vuelo tiene 30 asientos de capacidad y existen 3 clases ordenadas por precio: la clase 1 las más costosa, la 3 la más económica. Un límite de reserva de 20 asientos para la clase 2 establece que a lo sumo se aceptarán 20 compras para esa clase; un nivel de protección de 20 para esa clase, en cambio, establece que se reservan 20 asientos para las clases 1 y 2, es decir que se pueden aceptar a lo sumo 10 (recordar que la capacidad es 30) de la clase 3. La Figura 1.2 da un ejemplo de la aplicación para ambos controles.

La razón para querer *anidar*, y no particionar, las clases es que si un cliente decide comprar

---

<sup>3</sup>Para una descripción más completa referirse a [TvR04a].

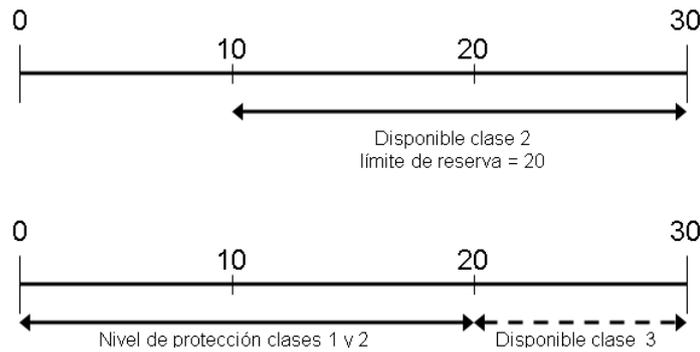


Figura 1.2: Ejemplos de límites de reserva y niveles de protección anidados: arriba, ejemplo de límites de reserva para un vuelo con 30 asientos y 3 clases; abajo, ejemplo de niveles de protección anidados para el mismo vuelo

un producto en una clase que ya alcanzó el límite permitido, tiene sentido poder utilizar un asiento asignado a una clase más económica para satisfacer el pedido. Tanto los límites de reserva, como los niveles de protección fueron concebidos para realizar optimizaciones a nivel de *leg*, aunque, como veremos a continuación, se los puede adaptar para que se pueda capturar el comportamiento de red.

### Controles virtuales anidados

El anidamiento virtual (*virtual nesting*) utiliza niveles de protección anidados a nivel de *leg* 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 la tarifa de un producto con el valor de la capacidad necesaria para satisfacer un pedido del mismo, al 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 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 se observa un pedido para un determinado producto, se identifica la clase virtual a la cual está asignado sobre cada tramo, y se verifica la disponibilidad de la clase virtual correspondiente sobre cada uno. La compra se acepta solamente si la clase se encuentra disponible en todos los tramos.

Esta política es muy utilizada en la práctica, y ha dado buenos resultados. Fue introducida por *American Airlines* en 1983, y aún hoy se encuentra operando en muchos sistemas de reservas de pasajes

### Bid Prices

La política de control de bid prices, que es la que utilizaremos en el presente trabajo, está basada en el ingreso asociado a los productos, y no en clases, como los controles antes mencionados. Consiste

en asignar un valor (*bid price*) a cada uno de los recursos de la red (los asientos en cada vuelo punto a punto) y utilizar esos valores como umbral para decidir si se acepta o no una venta. Ante la solicitud de compra para un producto, se compara el precio del mismo con la suma de los bid prices asociados a los recursos que utiliza, y si el ingreso obtenido es superior a la suma de los bid prices, se acepta la venta. Intuitivamente los bid prices representan el valor que se le da a una unidad de cada recurso, íntimamente relacionado con la ganancia que se espera obtener de ellos. La suma de los bid prices asociados a un producto se puede pensar entonces como el costo de oportunidad, según fue definido anteriormente, y la diferencia entre el precio del producto y el costo de oportunidad como el beneficio marginal neto de venderlo; a ésta última la llamaremos *ganancia marginal ajustada*.

### 1.2.2. Modelos de Demanda

Como hemos ya mencionado anteriormente, y en especial está fuertemente implícito en la sección anterior, la estimación de la demanda es un elemento crucial para que se pueda aplicar el *Revenue Management* con éxito. Los mecanismos que se utilizan para hacer una estimación adecuada de la demanda en un mercado son complejos, y hay mucho trabajo teórico-práctico hecho al respecto. Basta pensar, por ejemplo, que con capacidad finita y en la presencia de controles la demanda real no es un fenómeno observable (las compras lo son, pero si un producto no está disponible, la demanda para éste no se observa), con lo que es necesario utilizar mecanismos para estimar la demanda real (este proceso se denomina *destruncamiento*), que además, como dijimos, está sujeta a las decisiones de control que se tomen. Hacer una descripción, aunque sea breve, de este tipo de mecanismos está fuera del alcance de este trabajo. Sin embargo, es imprescindible presentar los diferentes modelos matemáticos para la demanda que más se utilizan en *Revenue Management*, y en particular en las aplicaciones para aerolíneas, pues tienen una influencia fundamental en las técnicas de optimización. Existen, a grandes rasgos, dos tipos de modelos de demanda que se han utilizado hasta hoy, y que detallamos a continuación.

#### Modelos de demanda independiente

Los modelos de demanda independiente son los que se utilizaron originariamente en *Revenue Management*, principalmente debido a su simplicidad. Se asume que la demanda para cada uno de los productos o clases es independiente, lo que simplifica mucho el trabajo de optimización, pero a la vez parece estar lejos de ser realista, ya que un cliente que quiere tomar un vuelo no busca un precio en particular sino, en general, el más económico disponible. Esto implica dependencia entre las clases, y en particular implica que un cliente podría comprar un producto de menor valor de lo que estaba dispuesto a pagar. Teniendo esto en mente es que, cuando se implementaron los controles de *Revenue Management*, se instauraron restricciones a las diferentes clases. Por ejemplo, si había un vuelo que tenía una tarifa alta y una tarifa económica, para comprar ésta última hacía falta hacerlo con mes de anticipación, o comprar un regreso con una semana de diferencia. De esta manera se buscaba que los viajeros de negocios, que se los suponía más dispuestos a pagar un mayor precio, no pudieran comprar la tarifa económica. Estas restricciones intentaban capturar una

segmentación del mercado, y volver a las demandas para ambas clases virtualmente independientes.

Otro tipo de supuesto que se hacía sobre la demanda era respecto del orden de llegada de los clientes para cada una de las clases. Tradicionalmente se suponía que los clientes llegaban en orden ascendente según la tarifa: los de la tarifa más económica primero, y los de la más alta últimos. Nuevamente, las restricciones arriba mencionadas, podían hacer que esto fuera suficientemente realista.

En los últimos años, el abandono progresivo de las restricciones por parte de las aerolíneas hizo necesario el desarrollo de modelos de demanda más sofisticados, que complican el problema, pero que se ajustan mejor a la realidad. Entre estos, se inscriben los modelos de elección.

## Modelos de elección

Tras relajar el supuesto de la independencia entre las demandas para las diferentes clases, lo que parece más razonable es modelar el hecho de que los clientes que consideran comprar un pasaje *eligen* entre las opciones que tienen disponibles, donde no comprar se considera una opción. Si el análisis es a nivel de un vuelo punto a punto, es razonable pensar que el cliente va a elegir la clase abierta de menor precio <sup>4</sup> (una clase se dice abierta si tiene asientos disponibles para vender). Sin embargo, es más realista pensar que un cliente que desea viajar a un destino va a analizar las diferentes opciones que hay entre los vuelos disponibles, de acuerdo a parámetros que le sean relevantes, como el día, el horario, el precio, etc. Este tipo de comportamiento es el que intentan capturar los modelos de elección.

Una buena forma de modelarlo es comenzar enumerando las diferentes *alternativas* que tiene para elegir un cliente (compuestas por los vuelos, y las tarifas disponibles en cada uno de ellos) y luego definir una relación de preferencia entre éstas, que dependa de los parámetros de cada una (decidir qué parámetros son relevantes es un trabajo estadístico importante, que se suele considerar ya efectuado en la etapa de optimización). Una práctica común para definir y cuantificar dicha relación es asignarle a cada alternativa una *utilidad*, lo que define automáticamente una relación de orden: a mayor utilidad mayor preferencia. Dicha utilidad puede tener un componente aleatorio, con el objetivo de capturar algún grado de incertidumbre siempre presente, lo que da lugar a los modelos de utilidad aleatoria, ampliamente utilizados en la actualidad. Existen diversos modelos de elección: comprar el más económico disponible, Multinomial Logit, compras con buyups, etc. Una de las principales ventajas del método propuesto en este trabajo es que es suficientemente general como para trabajar con diversos modelos de elección, los nombrados más arriba y otros, muy diferentes entre sí. En la sección 2.2 haremos una descripción más detallada de los que consideramos en este trabajo.

---

<sup>4</sup>Incluso últimamente, debido a la amplia gama de canales disponibles para las reservas, asumir esto podría no ser del todo realista. Una aerolínea puede ofrecer determinada clase en un vuelo, pero ésta no estar disponible en todos los canales (agencias, páginas web, etc.)

### 1.2.3. Optimización basada en simulación

Es muy común que en la ciencia de la toma de decisiones se tenga que trabajar en presencia de incertidumbre. Un aspecto crucial es decidir qué modelo utilizar para capturar la incertidumbre, de manera que se aproxime lo mejor posible a la realidad, sin que la complejidad del mismo lo vuelva intratable, tanto analítica como numéricamente. Existe una gran cantidad de enfoques que se han desarrollado a lo largo de los años, y está fuera del alcance de este trabajo hacer una exposición abarcativa al respecto (Kleywegt y Shapiro[KS00] realizan un recorrido por las técnicas más utilizadas), con lo que nos limitaremos a introducir los conceptos fundamentales de la metodología que utilizaremos en nuestro trabajo.

Para formular un problema de optimización estocástica, en general se parte de la definición de una función objetivo  $G(x, \omega)$ , donde  $x$  denota la variable, o variables, de decisión y  $\omega$  denota un parámetro, o parámetros que se desconocen al momento de llevar a cabo la decisión, junto con un conjunto de soluciones factibles  $X$  y un espacio muestral  $\Omega$ . Una de las formulaciones más comunes, que utilizaremos para resolver nuestro problema, es la que se conoce como *problema de programación estocástica*:

$$\max_{x \in X} \{g(x) \equiv \mathbb{E}[G(x, \omega)]\}$$

De más está decir que se trata de una formulación muy general, que se puede utilizar para modelar una gran variedad de problemas que aparecen en la práctica. También existen diversas metodologías para resolver los problemas formulados de esta manera, de las cuales nos van a interesar las que entran en la categoría de *métodos muestrales*. Los métodos muestrales se basan en hacer una aproximación del valor de  $g(x)$  tomando muestras de  $\omega$ . Si  $\{\omega^1, \dots, \omega^N\}$  es una muestra independiente de  $\omega$  de tamaño  $N$ , entonces podemos definir una aproximación de  $g$  de la siguiente manera:

$$\hat{g}_N(x) = \frac{\sum_{i=1}^N G(x, \omega^i)}{N}.$$

Debido a la ley de los grandes números, podemos afirmar que con probabilidad 1,  $g_N(x) \rightarrow g(x)$ . La ventaja de utilizar esta aproximación, es que  $g_N(x)$  es una función que no tiene componentes aleatorias, y se puede hacer un tratamiento determinístico de ella.

Un camino usual para optimizar  $g(x)$  es utilizar estimaciones similares a la anterior, pero para el gradiente de  $g$ ,  $\nabla g(x)$ , en lo que se conoce como *análisis de perturbación*. Consideremos por un momento que vale la siguiente igualdad:

$$\nabla g(x) \equiv \nabla \mathbb{E}[G(x, \omega)] = \mathbb{E}[\nabla G(x, \omega)] \quad (1.1)$$

Si este fuera el caso, entonces podríamos afirmar dos cosas:

$$\begin{aligned} \text{i) } \nabla g_N(x) &= \sum_{i=1}^N \frac{\nabla G(x, \omega^i)}{N} \longrightarrow \mathbb{E}[\nabla G(x, \omega)] = \nabla \mathbb{E}[G(x, \omega)] = \nabla g(x) \quad \text{y} \\ \text{ii) } \mathbb{E}[\nabla g_N(x)] &= \sum_{i=1}^N \frac{\mathbb{E}[\nabla G(x, \omega^i)]}{N} = \mathbb{E}[\nabla G(x, \omega)] = \nabla \mathbb{E}[G(x, \omega)] = \nabla g(x) \end{aligned}$$

donde i) se deduce de la ley de los grande números y la ecuación (1.1), y ii) de la ecuación (1.1) y la linealidad de la esperanza. Lo que estamos diciendo con i) y ii), es que  $\nabla g_N(x)$  es un estimador consistente e insesgado de  $\nabla g(x)$ . Esto podría ser de gran utilidad para realizar una optimización basada en el gradiente de  $g(x)$ . Sin embargo, primero hay que determinar bajo qué condiciones se puede usar (1.1). Respecto a esto último, se puede probar que para que (1.1) se cumpla en un punto  $x$  son suficientes dos condiciones (lema 6.3.1 en Glasserman[Gla94]):

1. La función  $G(\bullet, \omega)$  es diferenciable en  $x$  con probabilidad 1.
2. Existe una variable aleatoria positiva  $K(\omega)$ , tal que  $E[K(\omega)]$  es finita, y se cumple la igualdad:

$$|G(x_1, \omega) - G(x_2, \omega)| \leq K(\omega) \|x_1 - x_2\|$$

con probabilidad 1 para todo par de valores  $x_1, x_2$  en alguna bola alrededor de  $x$ . En otras palabras,  $G$  es Lipschitz continua con probabilidad 1 alrededor de  $x$ <sup>5</sup>.

En muchas aplicaciones prácticas la función objetivo puede no ser derivable en todo punto, para determinada realización de  $\omega$ , y más aún es posible que el óptimo se encuentre en un punto de no diferenciabilidad. Sin embargo, si la distribución de  $\omega$  es continua, es común que la probabilidad de que ocurran dichos casos sea 0.

Dicho esto, ya estamos en condiciones de describir el método estocástico que vamos a desarrollar en nuestro trabajo. Está dentro de la categoría de los métodos de *optimización basada en simulación*, y es conocido como *método de aproximación estocástica* o *método de gradientes estocásticos* (descrito por primera vez en Robbins and Monro[RM51]). La idea es desarrollar un procedimiento iterativo, que parta de una solución inicial  $x_0$ , y vaya generando soluciones, que converjan a un punto crítico de la función objetivo. Supongamos que  $X = \mathbf{R}^n$ , la fórmula que utilizaremos para calcular cada iteración es la siguiente:

$$x_{k+1} = x_k + \alpha_k \widehat{\nabla} g(x_k) \tag{1.2}$$

Donde  $\widehat{\nabla} g(x_k)$  representa una aproximación de  $\nabla g(x_k)$ , y  $\alpha_k$  el tamaño del  $k$ -ésimo paso. Lo que se está haciendo es tomar cada solución y efectuar un paso en la dirección de la estimación del gradiente (conocido como *dirección más empinada*). Supongamos que se cumplen las siguientes condiciones sobre  $g$ ,  $\widehat{\nabla} g$  y  $\alpha_k$ :

1.  $g(x) > 0 \quad \forall x \in \mathfrak{R}^n$
2.  $E[\widehat{\nabla} g(x_k) | \mathcal{F}^k] = \nabla g(x_k)$  para todo  $k = 1, 2, \dots$  (donde  $\mathcal{F} = \{x_0, \widehat{\nabla} g(x_0), \dots, \widehat{\nabla} g(x_{k-1})\}$  representa la *historia* del algoritmo hasta el momento).
3. Existe un escalar finito  $M$  para el que se cumple:  $|\widehat{\nabla} g(x_k)| \leq M$  para todo  $k = 1, 2, \dots$
4. Existe un escalar finito  $L$  que cumple  $\|\nabla g(\lambda) - \nabla g(\gamma)\| \leq L \|\lambda - \gamma\|$

---

<sup>5</sup>Una función  $f : \mathbf{R}^m \rightarrow \mathbf{R}$  se dice Lipschitz continua sí y sólo sí existe una constante  $M \geq 0$  tal que  $|f(x_1) - f(x_2)| \leq M \|x_1 - x_2\|$  para todo  $x_1, x_2 \in \mathbf{R}^m$

$$5. \sum_{k=1}^{\infty} \alpha_k = \infty$$

$$6. \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

Entonces, podemos garantizar que, con probabilidad 1:  $\lim_{k \rightarrow \infty} \nabla g(x_k) = 0$  (proposición 4.1 en Bertsekas and Tsitsiklis[BT96])). Es común que se tome  $\alpha_k \equiv \frac{A}{B+k}$ , con  $A \in \mathbf{R}_+$  y  $B \in \mathbf{R}_{\geq 0}$ , y es lo que haremos en nuestro trabajo. Es importante notar que no se pide que la estimación del gradiente sea muy buena, basta que sea un estimador insesgado: lo que haremos en nuestro caso es tomar  $\nabla g_N(x)$ , con diferentes valores de  $N$ .

La mayor ventaja de este método es su simplicidad de implementación, y el hecho de que provee convergencia teórica bajo condiciones bastante generales. Sin embargo, la eficacia práctica del método depende de varios factores: la definición del tamaño de paso (en nuestro caso, la elección de  $A$  y  $B$ ), la calidad de la solución inicial y la estimación de los gradientes (es conveniente, por ejemplo, tener buenas estimaciones en las primeras iteraciones para seguir pasos confiables en la dirección correcta). Otro de los problemas que tiene, en realidad una dolencia de todos los métodos muestrales de optimización, es que en general no existe un buen criterio de parada. Volveremos sobre estos temas, al describir nuestra implementación de esta metodología.

#### 1.2.4. Sobre este trabajo

Una vez introducidos los conceptos fundamentales del área, y los aspectos particulares más relevantes al problema que resolveremos, ya estamos en condiciones de enunciar el trabajo que realizaremos: vamos a estudiar un método de optimización estocástica (en particular, uno de gradientes estocásticos) para *Revenue Management*, asumiendo demandas y capacidades continuas, y utilizando la política de bid prices. Nuestro trabajo completa en la literatura la extensión de las dos políticas más usuales de control de capacidad, niveles de protección anidados y bid prices, al caso de comportamiento selectivo de clientes. Respecto a los niveles de protección, van Ryzin y Vulcano [vRV08b] desarrollan un algoritmo de gradientes estocásticos para mejorar el conjunto inicial de niveles de protección anidados, considerando el caso de demandas independientes. Posteriormente, en van Ryzin y Vulcano [vRV08a], los autores extienden el algoritmo para el caso de comportamiento de elección. Respecto a bid prices, Topaloglu[Top08] diseña un algoritmo de gradientes estocásticos para el modelo de demanda independiente. Nuestro trabajo completa la grilla, extendiendo el mismo a demandas con comportamiento de elección.

Además de desarrollar en detalle todos los aspectos relevantes al método, lo implementaremos en el lenguaje de programación C++, y realizaremos diversos experimentos para evaluar su efectividad. Dado que la principal innovación está relacionada con la incorporación de modelos de elección a la implementación de bid prices, los experimentos estarán orientados a evaluar el comportamiento del algoritmo bajo este tipo de modelos de demanda. Como ya hemos mencionado, los modelos de elección han cobrado mucho interés en los últimos años. Desde el punto de vista teórico, Talluri y van Ryzin[TvR04b] proveen 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[ZC05] la elección se considera a nivel de vuelos paralelos entre dos terminales. Para los problemas a nivel

de red, además del trabajo, ya mencionado, de van Ryzin y Vulcano[vRV08a], una aproximación alternativa, que ha dado buenos resultados en experimentos computacionales, es la desarrollada por Gallego et al.[GIPD04], que proponen un programa lineal determinístico que considera la elección de los clientes. Posteriormente, van Ryzin y Liu[LvR08] profundizan este enfoque, y Miranda Bront et. al[JMBV] lo generalizan. Los prometedoros resultados reportados en éste último serán utilizados como punto de comparación respecto de los ingresos generados por nuestra metodología.

Nuestros resultados mostrarán que el método provee importantes mejoras en el ingreso, provenientes de aprovechar el comportamiento selectivo de clientes. Para evaluar su eficacia en este aspecto, vamos a comparar los resultados que arroja nuestra metodología con los del Programa Lineal Determinístico, un método de optimización ampliamente utilizado en el área, basado en demandas independientes y determinísticas, que además utilizaremos para generar la solución inicial de las iteraciones de nuestro algoritmo. Desde el punto de vista teórico, sin embargo, vamos a comprobar que el algoritmo no garantiza optimalidad, aunque sí se puede garantizar la convergencia teórica, pero no lo demostraremos en este trabajo.

El resto de este trabajo se encuentra organizado de la siguiente manera: en el capítulo 2 se hace una descripción general del modelo matemático que se utiliza, y en particular del modelo de demanda, haciendo especial hincapié en la generalidad del mismo, y la posibilidad de adaptarlo a diferentes modelos de elección; en el capítulo 3 se hace una descripción exhaustiva de la metodología, y se discuten diversos aspectos relativos a la misma; en el capítulo 4 se describen los experimentos numéricos que se realizan y se muestran los resultados; y finalmente, en el capítulo 5 se presentan las conclusiones de este trabajo.



# Capítulo 2

## Modelo

En este capítulo, presentaremos las características del modelo matemático que utilizaremos para resolver el problema planteado. Comenzaremos describiendo las definiciones básicas de *Revenue Management* para problemas a nivel de red (sección 2.1), que son ampliamente utilizadas en la literatura del área. Luego definiremos el modelo de demanda continua que utilizaremos (sección 2.2), junto con una introducción a la metodología de decisión, basada en la política de bid prices, pero con modificaciones producto del uso de demanda continua. Por último, en la sección 2.3, discutiremos sobre la generalidad del modelo de demanda elegido, y su adaptabilidad a distintos modelos de elección. Respecto a esto último, daremos algunos ejemplos de los más utilizados en la literatura, y de la forma en que el nuestro puede reducirse a los mismos.

### 2.1. Definiciones generales

En un modelo de red para *Revenue Management* existen dos componentes fundamentales, ya presentados en este trabajo: los productos y los recursos. Los recursos son los asientos disponibles en los vuelos de una aerolínea, y los productos son los pasajes que la aerolínea vende, que corresponden a un itinerario (una secuencia de uno o más vuelos), y tienen un precio asociado, que representa el ingreso de la aerolínea por vender ese producto. La venta de un producto consume un asiento en cada uno de los vuelos que utiliza el itinerario. El conjunto de productos (con los recursos que éstos utilizan) define lo que se conoce como una *red*. La notación que vamos a utilizar es la siguiente: definiremos al conjunto de recursos como  $\{1, \dots, m\}$ , al conjunto de productos como  $\{1, \dots, n\}$ , con  $r_j$  el ingreso asociado al producto  $j$ , y a la *matriz de incidencia* de la red como  $A = [a_{ij}] \in \{0, 1\}^{m \times n}$ , con  $a_{ij} = 1$  si y sólo si el producto  $j$  utiliza el recurso  $i$ . De manera análoga definiremos el vector de incidencia del producto  $j$ ,  $A_j$ , como la  $j$ -ésima columna de  $A$ , y escribiremos  $i \in A_j$  cuando  $j$  utiliza  $i$ . Las capacidades de los vuelos están definidas por el vector  $x^\top = (x_1, \dots, x_m) \in \mathbf{R}^m$ . Asumiremos que las capacidades son continuas; así, por ejemplo, un vuelo  $i$  puede tener 52.74 asientos disponibles. En la nomenclatura usual de *Revenue Management* al vector  $x$  suele llamárselo *estado* de la red, ya que se trata del único parámetro del problema que

varía, sujeto a la política que se aplique. En símbolos, cuando se acepta una solicitud de  $\alpha$  unidades de un producto  $j$ , el estado de la red pasa a ser  $x - \alpha A_j$ . Por último, los bid prices asociados a cada recurso se notarán como  $\pi = (\pi_1, \dots, \pi_m) \in \mathbf{R}^m$ .

## 2.2. Modelo de Demanda

El modelo de demanda que vamos a utilizar en el presente trabajo es el desarrollado por Van Ryzin y Vulcano[vRV08a]. Es, como hemos apuntado, un modelo de demanda muy general, basado en el análisis de la cantidad de clientes que arriban, el orden de llegada y las preferencias de los mismos. Los clientes, o más precisamente los *tipos* de clientes, tienen asociada una lista que contiene los productos que el cliente considera, en orden de preferencia. Llamaremos entonces *sample path*, o tira muestral de clientes, a una secuencia de arribos de clientes, que define la realización de la demanda de la red. Sea  $T$  el largo, aleatorio, del sample path: vamos a suponer que  $T$  está acotado por una constante  $\tau$  con probabilidad 1, es decir  $P(T \leq \tau) = 1$ . Como dijimos, cada cliente  $t$  de la secuencia está representado por una lista de preferencia, a la que notaremos  $l_t = [l_{t1}, \dots, l_{tn_t}]$ , donde  $n_t$  denota el largo de la lista, y  $t = 1, \dots, T$ . Cada arribo tiene asociado además una cantidad de producto solicitada, o demanda del cliente,  $q_t$ , que suponemos como la realización de una variable aleatoria discreta  $Q$  con soporte  $\{1, \dots, \bar{q}\}$ .

Las listas de preferencia podrían provenir, por ejemplo, de una maximización de utilidad entre los diferentes productos de la red, y es importante notar que no todas tienen que contener a todos los productos, lo que modela el hecho de que no todos los clientes consideran el mismo conjunto de productos cuando desean viajar de un punto a otro.

Para ilustrar el funcionamiento del modelo de demanda, la figura 2.1 contiene un ejemplo de una red pequeña, tipos de cliente y un sample path ilustrativo. Se trata de una red con tres recursos y cinco productos: cuatro productos corresponden a un vuelo directo y uno corresponde a un itinerario con una parada. Existen 3 tipos de cliente que obedecen a posibles patrones de elección, como puede ser priorizar el precio, o el hecho de que el vuelo no tenga paradas.

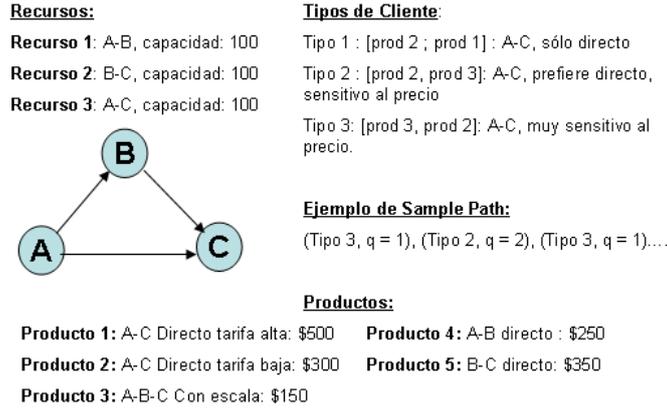


Figura 2.1: Un ejemplo de red, y de tipos de clientes asociados

Denotaremos  $b(t, j)$  a la posición del producto  $j$  en la lista del cliente  $t$ , es decir  $b(t, j) = k$  si y sólo si  $j = l_{tk}$ . Por completitud definimos  $b(t, j) = n_t + 1$  si  $j$  no está incluido en la lista  $l_t$ . Para simplificar la notación, vamos a escribir  $[k]$  al  $k$ -ésimo elemento de la lista, en símbolos:  $[k] = j$  si y sólo si  $b(t, j) = k$ . En el Sample Path del ejemplo de la figura 2.1,  $b(1, 2) = 2$ , porque el primer cliente es de tipo 3, y el segundo producto de esa lista es el 2.

Una vez definido el concepto de sample path, es necesario definir cómo será el *procesamiento* del mismo. Con procesamiento, nos referimos a utilizar el sample path para realizar una simulación, que tenga como salida el ingreso obtenido, de acuerdo a la política de control utilizada. Implícitamente estamos definiendo una función de ingreso que depende de 3 elementos: el Sample Path al que se aplica, que describe la llegada de los clientes, las capacidades de la red, y los bid prices, que definen las decisiones que se toman respecto de las solicitudes de los clientes. La metodología con la que un Sample Path es procesado es una adaptación a la política de bid prices clásica, ya presentada anteriormente. Existen dos diferencias fundamentales respecto de ésta, relativas a la continuidad del modelo: por un lado, la cantidad de asientos de las solicitudes de los clientes provienen de una distribución que es continua. Por otro, y tal vez más importante, durante el procesamiento de un Sample Path vamos a permitir que un cliente reciba una cantidad no entera de pasajes. Dicho de otra manera, si un cliente solicita un asiento en un vuelo, vamos a suponer que es factible venderle una fracción de asiento. Es claro que una relajación de este tipo no es realista, pero tiene como objetivo “suavizar” la función de ingreso para que sea continua. Cabe aclarar, sin embargo, que durante la etapa de experimentación, en la que mediremos la eficacia del método, las simulaciones que utilizaremos sólo permitirán la venta de cantidades enteras de asientos, como explicaremos más adelante.

La consecuencia inmediata, respecto de la política de bid prices, que conlleva la intención de suavizar la función de ingreso es que, en principio, no existe una asociación evidente entre la ganancia marginal ajustada, según la definimos en la sección 1.2.1, y la cantidad de producto que se debe vender. Originalmente la política se definía como: vender solamente si el precio del producto es mayor a la suma de los bid prices (es decir, si la ganancia marginal ajustada es positiva), pero

esta naturaleza binaria es lo que hace que la función de ingreso no sea continua en los bid prices. Definimos entonces, siguiendo la idea planteada en Topaloglu[Top08], una función  $\theta : \mathbf{R} \rightarrow (0, 1)$  que provee la conexión necesaria entre la ganancia marginal ajustada, que depende de los bid prices asociados al producto, y la cantidad de producto que se venderá. Vamos a suponer además que  $\theta$  es Lipschitz continua en  $\mathbf{R}$ , creciente, diferenciable y que  $\lim_{x \rightarrow -\infty} \theta(x) = 0$  y  $\lim_{x \rightarrow +\infty} \theta(x) = 1$ .

En resumen, la metodología es la siguiente: por cada llegada de un cliente, mientras haya capacidad disponible, se van a ir otorgando los productos en el orden que establece la lista de preferencia, y de acuerdo al valor de  $\theta$  evaluado en la ganancia marginal ajustada, multiplicado por la demanda residual del cliente (la demanda original  $q_t$  menos lo que consumieron los productos de mayor prioridad), que denotaremos  $q_t^j$ ; en símbolos,  $q_t^j \theta(r_j - \sum_{i \in A_j} \pi_j)$ . Siguiendo con el ejemplo de la figura 2.1, supongamos que los bid prices asociados a cada recurso son  $\pi_1$ ,  $\pi_2$  y  $\pi_3$  respectivamente. Entonces, para el primer cliente ( $t = 1$ ), de tipo 3, se va a otorgar  $q_1 \cdot \theta(\$150 - (\pi_1 + \pi_2))$  del producto 3 -el primero en la lista-<sup>1</sup>, y prosiguiendo con el siguiente producto de la lista, el producto 2, se calcula cuánto se va a entregar: el mínimo entre  $q_1^2 \theta(\$300 - \pi_3)$  -donde  $q_1^2 = q_1 - q_1 \cdot \theta(\$150 - (\pi_1 + \pi_2))$ - y lo que reste de los recursos que el segundo producto utiliza. De una manera análoga se prosigue con los siguientes arribos.

El procedimiento que acabamos de describir corresponde a la etapa de optimización de nuestro trabajo, que tiene como salida un vector de bid prices, del que se espera tenga un buen comportamiento respecto del ingreso generado. La etapa de experimentación tiene como objetivo evaluar qué tan bueno es el set de bid prices, a través de la simulación numérica. En pocas palabras, lo que se hace es generar sample paths aleatorios y determinar qué ingreso se obtiene utilizando los bid prices generados en la etapa de optimización. Un aspecto crucial a tener en cuenta es que, como dijimos anteriormente, no es razonable pensar que en un proceso real de ventas se pueda otorgar una cantidad continua de producto a los clientes. Por esta razón es que el procesamiento de los sample paths durante la experimentación es *discreto*: la idea es muy similar a la anterior, sólo que en vez de utilizar la función  $\theta(x)$ , se utiliza la política clásica. Es decir, se van otorgando unidades enteras de producto según el orden de preferencia, en la medida que la ganancia marginal ajustada del producto sea positiva, haya capacidad disponible y no se haya alcanzado la cantidad solicitada. En el ejemplo de la Figura 2.1, si asumimos que  $\pi_1 = \pi_2 = \pi_3 = 100$ , entonces el primer cliente del sample path no recibe nada del primer producto de la lista, porque  $\$150 < \pi_1 + \pi_2$ , pero recibe una unidad entera del producto 2, porque  $\$300 > \pi_3$ .

A este procesamiento de los sample paths lo llamaremos *procesamiento discreto*, y es importante tenerlo en cuenta, pues es el tipo de procesamiento que describe el comportamiento operacional que se utiliza en la realidad, y el que utilizaremos nosotros para evaluar la eficacia del set de bid prices generado por nuestro método.

---

<sup>1</sup>Notar que si se definiera a  $\theta(\$150 - (\pi_1 + \pi_2)) = \mathbf{I}\{\$150 > \pi_1 + \pi_2\}$ , con  $\mathbf{I}\{\}$  la función indicatriz, se trataría de la política original de bid prices

## 2.3. Generalidad del modelo de demanda

Ya hemos mencionado en más de una ocasión la generalidad del modelo de demanda con el que tratamos. El objetivo de esta sección es tratar de mostrar a qué nos referimos con esto, y dar algunos ejemplos de modelos muy utilizados en la práctica, a los que el de este trabajo se puede adaptar.

Antes que nada, nos queda pendiente de la sección anterior el definir con un poco más de formalidad la noción de sample path: un sample path es una secuencia  $\omega = \{(l_1, q_1), (l_2, q_2), \dots, (l_T, q_T)\}$ , que suponemos definida en un espacio de probabilidad  $(\Omega, \mathcal{F}, P)$ . La generalidad del modelo proviene del hecho de que no asumimos ninguna característica particular respecto de la distribución de los sample paths: la preferencia de los clientes puede cambiar con el tiempo, puede haber cualquier tipo de correlación entre los arribos y/o las preferencias, etc. En esencia, lo único que necesitamos para que el método funcione es un *oráculo* que pueda generar sample paths a partir de alguna distribución, no necesariamente definida en forma explícita.

Existen diversos modelos de elección utilizados en la práctica. Uno de los más sencillos e intuitivos es en el que los clientes compran el pasaje más barato disponible. Desde el punto de vista de la generación de sample paths, para modelar este comportamiento basta con poner en las listas de preferencia todos los productos en orden ascendente de precio. De esta manera, ante la solicitud de un cliente, se otorgará el primer producto que esté disponible <sup>2</sup>, es decir el más económico. La principal ventaja de un modelo así es su sencillez: problemas modelados de esta manera no conllevan grandes dificultades en su resolución, incluso de manera exacta. Sin embargo en general, y en especial en la optimización a nivel de red, es poco razonable pensar que existe un único tipo de cliente, y que la utilidad que le asigna a los diferentes productos depende estrictamente del precio de los mismos. Es de esperar que existan distintos tipos de clientes con diferentes intereses, que para un tipo de cliente existan otros factores que influyan además del precio y más aún que exista un nivel no despreciable de incertidumbre, lo que vuelve poco efectivo a cualquier modelo determinístico. Esto último es lo que da lugar a los modelos de elección aleatorios, muy difundidos en la actualidad. Vamos a presentar a continuación a dos de ellos, que utilizaremos para la etapa de experimentación: el *Multinomial Logit* y las compras con *buyups*.

### El *Multinomial Logit*

El Multinomial Logit (MNL) es un modelo de elección aleatorio de amplio uso en la literatura de marketing y economía (ver Ben-Akiva y Lerman [BAL94], y Train [Tra03]), que ha llamado la atención del *Revenue Management* como modelo de elección (ver Talluri y van Ryzin [TvR04a], Ratliff et al. [RRNY], y Vulcano et al. [VvRC08]). En MNL, a cada cliente  $t$  se le asigna una utilidad

---

<sup>2</sup>En este y los demás ejemplos de modelos de elección, se encuentra implícito que el procesamiento de los sample paths es discreto. Cuando hablamos de productos no disponibles nos referimos a que la suma de los bid prices del producto sea mayor o igual que su precio. Si este es el caso, cuando se procese el sample path no se otorgará ninguna unidad de dicho producto a ningún cliente, lo que es equivalente a la noción de no disponibilidad.

que tiene la siguiente forma:

$$U_t^j = \mu_t^j + \xi_t^j, \quad j = 0, 1, \dots, n,$$

donde  $\mu_t^j$  representa la utilidad media y  $\xi_t^j$   $j = 0, 1, \dots, n$ , representan ruidos aleatorios independientes. La utilidad media  $\mu_t^j$  se calcula generalmente como una combinación lineal de atributos observables:  $\mu_t^j = \beta^\top a_t^j$ , donde  $\beta$  es un vector de parámetros, que representa la importancia o “peso” que le dan los clientes a los diferentes atributos de un pasaje (el precio, horario, día de partida, etc.) y  $a_t^j$  son los valores que toman los atributos en el producto  $j$ . Por ejemplo, supongamos que  $\beta^\top = (-0.01, 3.2)$  es el vector de parámetros, donde la primera componente se refiere al precio, y la segunda al horario de salida del vuelo. Entonces, un vuelo descrito por atributos  $a_t^j = (300, 1)$ , donde la segunda coordenada es un indicador que toma el valor 1 si el vuelo es matutino y 0 si es vespertino, tiene una utilidad media asociada de  $\mu_t^j = -0.01 \times 300 + 3.2 \times 1 = 0.2$ . Del mismo modo, un vuelo vespertino con tarifa \$150 tiene una utilidad media  $\mu_t^j = -0.01 \times 3.2 \times 0 = -1.5$ .

Existen diversos aspectos a tener en cuenta a la hora de ajustar un MNL, para que pueda describir de manera aceptable un mercado real, como cuántos y cuáles atributos de un pasaje tener en cuenta, o qué técnica estadística utilizar para estimar los valores de  $\beta$ . Este tipo de consideraciones pertenecen a la etapa de estimación de un sistema de *Revenue Management*, y están fuera del alcance de nuestro trabajo (en Vulcuano et al.[VvRC08], por ejemplo, se plantea un método de estimación para un modelo MNL y se lo implementa con éxito en datos reales de una aerolínea). En la etapa de optimización se asume que el valor de  $\beta$  ya está calculado.

La componente de ruido de la utilidad aleatoria,  $\xi_t^j$  sigue una distribución Gumbel (o doble exponencial) de media 0, con función de distribución acumulada:

$$F_\xi(x) = P(\xi_t^j \leq x) = e^{-e^{-(\delta x + \gamma)}},$$

donde  $\gamma$  es la constante de Euler ( $\gamma \approx 0.5772$ ), y  $\delta$  es un parámetro de escala, que usualmente se fija en 1. Cada cliente  $t$  tiene asociado un conjunto de elección  $C_t$  que contiene todos los productos que el cliente considera (también se los denomina *alternativas*, en este contexto). Se asume que siempre existe la opción de no comprar ningún producto, y se representa a esta opción con el producto  $j = 0$  con media  $\mu_t^0$ . El cliente comprará el producto  $j$  si y sólo si, la realización de la variable  $U_t^j$  es mayor que la realización del resto de las utilidades. En símbolos:  $P_t(j) = P(U_t^j > U_t^k \quad \forall k \in C_t - \{j\} \cup \{0\})$ . Se puede probar además que la probabilidad de elección se puede escribir de la siguiente forma:

$$P_t(j) = \frac{e^{\mu_t^j}}{\sum_{i \in C_t} e^{\mu_t^i} + e^{\mu_t^0}},$$

Si pensamos ahora a  $e^{\mu_t^j}$  como el peso que le asigna el cliente  $t$  al producto  $j$ , entonces la probabilidad de elección  $P_t(j)$  no es otra cosa que el peso relativo de la alternativa, respecto del resto. En la nomenclatura usual de *Revenue Management* se suele llamar *attractiveness* (de *attractive*, atractivo en inglés) al peso  $e^{\mu_t^j}$  asociado al producto. Nosotros lo notaremos  $v_t^j := e^{\mu_t^j}$ .

Para adaptar nuestro modelo de demanda a un modelo MNL, basta generar sample paths de la siguiente manera: por cada arribo de un cliente  $t$  (la generación de arribos puede llevarse a cabo de

diferentes formas, por ejemplo, simulando un proceso de Poisson) se genera una realización de la utilidad de cada uno de los productos del conjunto de elección  $C_t$ , además de la utilidad (también aleatoria) de la no compra. Supongamos que  $\hat{U}_t^j$  es la realización de  $U_t^j$ , y supongamos que, ordenando de mayor a menor las realizaciones, obtenemos la siguiente lista:  $[\hat{U}_t^{j_1} \dots \hat{U}_t^{j_r} \hat{U}_t^0 \dots \hat{U}_t^{j_n}]$ . La lista de preferencia del cliente  $t$  se construye entonces tomando en orden todos los productos que tienen mayor utilidad que la no compra, es decir, en este caso  $l_t = [j_1, \dots, j_r]$  (en el caso en el que la no compra sea la de mayor utilidad, se omite el arribo). En realidad, si se tiene en cuenta que el procesamiento discreto de alguna manera “ignora” los elementos de la lista no disponibles (ver nota al pie nº2), cada lista tendrá una sublista de productos relevantes: los disponibles. Supongamos que existen 3 productos:  $\{1, 2, 3\}$ , y que el 1, por los valores de los bid prices asociados, no está disponible. Supongamos además que se simulan las utilidades y se obtiene la lista:  $[\hat{U}_t^3, \hat{U}_t^1, \hat{U}_t^2, \hat{U}_t^0]$ . Cuando el sample path se procese, el producto 1 se va a omitir, en esta y todas las listas que lo contengan. Como el orden de los productos depende únicamente de sus utilidades, que son independientes entre sí, el orden relativo de un par de productos es independiente del resto de la lista. Esto quiero decir que, a efectos prácticos, es como si el producto 1 no existiera, y la lista es equivalente a  $[\hat{U}_t^3, \hat{U}_t^2, \hat{U}_t^0]$ ; es decir, el cliente en realidad elige entre los productos disponibles: 3 y 2, en ese orden.

El hecho de que los bid prices se mantengan fijos durante el procesamiento de un sample path implica que el conjunto de productos ofrecidos se mantiene fijo. Esto es una desventaja de nuestro enfoque, con respecto a otros, pero existen formas de remediarlo, como veremos más adelante con la *reoptimización*.

### Compras con *buyups*

Otro de los conceptos más usuales hoy día en los modelos de elección está relacionado con el comportamiento de los clientes que deciden comprar un producto, pero dicho producto está cerrado. El término *buyup* (algo así como “comprar el superior”) se refiere al hecho de que el cliente puede considerar comprar un producto abierto de valor superior al que pretendía en un principio. Este concepto ha sido modelado y aplicado de diversas maneras, históricamente fue la primera aproximación que se utilizó para el problema de demandas no independientes (ver, por ejemplo, Belobaba[Bel89], Cooper et al.[CdMK06], ). Se observó que los controles aplicados a las diferentes clases podían tener una fuerte influencia en la demanda de las demás (tanto por clientes dispuestos a pagar caro que encuentran abierto un producto de menor precio, como clientes que acceden a pagar mayor precio del que esperaban), lo que de alguna manera entrecruza las tareas de estimación y optimización, que pasan a depender una de la otra. Esto complica el problema, pero a la vez da mayores oportunidades de mejora.

En general los modelos que utilizan *buyups* dividen la demanda para una clase en dos componentes. Por un lado existe una demanda inherente a la clase, producida por una atracción natural de los clientes a comprarla, y por otro, existe la demanda absorbida de las clases que están cerradas. Cuando un cliente quiere comprar un producto que no está disponible, se asume que hay una probabilidad de que compre en alguna de las clases superiores que están abiertas. Este tipo

de metodología puede aplicarse de diferentes maneras: puede asumirse que el cliente sólo podría comprar la clase abierta más barata, o cualquiera de las clases abiertas, o incluso algún producto en un vuelo alternativo (en este último caso el término que se suele utilizar es *cross flight recapture*)<sup>3</sup>. En cualquiera de los casos se puede suponer que existe un método para simular el arribo de clientes.

Vamos a dar un ejemplo de cómo procederíamos para construir un sample path, modelando el comportamiento de *buyups*: por cada arribo de cliente, establecemos como primer elemento de la lista de preferencia al producto que el cliente quería originalmente. Supongamos que existe una probabilidad  $p_k$  de que el cliente compre el producto  $k$  si el primer producto no está disponible, y una probabilidad  $1 - p_k$  de que el cliente no compre nada. En tal caso, se “tira una moneda” con probabilidad  $p_k$  (se simula una v.a. Bernoulli con  $p = p_k$ ) y en caso de éxito se agrega el producto  $k$  a la lista. Para ver por qué esto funciona, pensemos lo siguiente: supongamos que tenemos un tipo de cliente que tiene a  $j$  como producto principal y a  $k$  como alternativo, y que durante la simulación el set de bid prices que se utiliza hace que  $j$  no esté disponible. Cada arribo correspondiente a este tipo de cliente tendrá a  $j$  como producto inicial de su lista de preferencias, y como  $j$  no está disponible, el procesamiento evaluará el segundo elemento de la lista, que será  $k$  en una proporción  $p_k$  de los casos, lo que es equivalente a decir que el cliente intenta comprar  $k$  con una probabilidad  $p_k$ .

---

<sup>3</sup>Para calcular las probabilidades de que esto ocurra se puede usar a su vez un modelo de elección, como el MNL.

# Capítulo 3

## Método

El objetivo del presente capítulo es definir formalmente el problema que vamos a tratar, y el método para resolverlo. Comenzaremos definiendo con precisión la ya introducida *función de ingreso*, que será, en esperanza, la función objetivo que maximizaremos; para esto definiremos una función auxiliar de *aceptación* que indicará, dado un conjunto de bid prices, qué cantidad se venderá de cada producto en una lista de preferencia. Esto dará lugar a la definición del problema que tratamos como un problema optimización estocástica. Una vez hecho esto, comenzaremos a desarrollar la metodología de resolución, calculando las derivadas de las funciones de aceptación e ingreso. Presentaremos además el pseudocódigo de un algoritmo para calcular dichas derivadas, junto con el cálculo del orden de complejidad temporal del mismo. Para terminar, mostraremos un esquema con una descripción de alto nivel del funcionamiento del método, y discutiremos algunos aspectos que no fueron incluidos en las secciones anteriores, pero que son de relevancia para la definición y eficacia del mismo: la elección de la función  $\theta(y)$ , la concavidad o cuasiconcavidad de la función de ingreso y la convergencia teórica.

### 3.1. Las funciones de aceptación e ingreso

Comenzaremos con la definición formal de las nociones de procesamiento de un sample path y función de ingreso, que introdujimos en la sección 2.2. Antes de definir la función de ingreso vamos a definir la función de *aceptación*, que va a indicar, ante una solicitud de un cliente, qué cantidad se venderá de cada uno de los productos de la lista de preferencia. La cantidad vendida de un producto  $j$  dependerá de 3 factores: i) la capacidad disponible para dicho producto, es decir, el mínimo de las capacidades de los recursos que utiliza; ii) lo que reste de la demanda del cliente  $q_t$ , que no haya sido otorgada a productos de mayor prioridad que  $j$  y iii) la función de bid prices. Respecto de ésta última, como dijimos en el capítulo 2, lo que haremos será reescalar el valor de  $\theta(r_j - \sum_{i \in A_j} \pi_i)$  según la demanda restante, es decir tomar  $q_t^j \theta(r_j - \sum_{i \in A_j} \pi_i)$ , donde  $q_t^j$  representa la demanda restante para  $j$ , que no ha sido entregada a productos de mayor prioridad. Sea  $x(t) = (x_1(t), \dots, x_m(t))$  el vector de capacidades disponibles en el momento anterior a la llegada del cliente  $t$ ; y sea  $x_i^j(t)$  la capacidad disponible del recurso  $i$ , para el producto  $j$  (que es la capacidad

original  $x_i(t)$  menos lo que se haya vendido de los productos de mayor prioridad que utilicen  $i$ ). Denotaremos al vector que representa la función de aceptación para cada uno de los productos como  $u(x(t), \pi, l_t, q_t)$ . En símbolos:

$$u_j(x(t), \pi, l_t, q_t) = \begin{cases} \min \left\{ q_t^j \theta \left( r_j - \sum_{i \in A_j} \pi_i \right), x_i^j(t) : i \in A_j \right\} & \text{si } 1 \leq b(t, j) \leq n_t, \\ 0 & \text{si } b(t, j) = n_t + 1, \end{cases} \quad (3.1)$$

donde

$$\begin{aligned} q_t^j &= q_t^{[b(t,j)-1]} - u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \\ &= q_t - \sum_{k=1}^{b(t,j)-1} u_{[k]}(x(t), \pi, l_t, q_t), \\ x_i^j(t) &= x_i^{[b(t,j)-1]} - u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[b(t,j)-1]}\} \\ &= x_i(t) - \sum_{k=1}^{b(t,j)-1} [u_{[k]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[k]}\}], \end{aligned} \quad (3.2)$$

y definiendo

$$q_t^{[0]} = q_t, \quad x_i^{[0]}(t) = x_i(t), \quad u_{[0]}(x(t), \pi, l_t, q_t) = 0, \quad A_{[0]} = \emptyset,$$

Intuitivamente, el comportamiento de la función es el que ya hemos descrito, y la formulación es relativamente natural, si se considera la recursión, empezando por el producto de mayor prioridad. Es importante notar que como  $\theta(x) < 1$ , entonces  $q_t^j \theta(r_j - \sum_{i \in A_j} \pi_i) < q_t^j$ . Esto tiene dos consecuencias: primero, que no hace falta incluir la demanda restante  $q_t^j$  en el mínimo en 3.1, y segundo, puede observarse fácilmente que la demanda original  $q_t$  no va a consumirse completamente nunca.

Una vez definida  $u$ , ya estamos en condiciones de definir la función de ingreso. Denotaremos como  $R_t(x(t), \pi, \omega)$  al ingreso obtenido, procesando desde el cliente  $t$  en adelante, en función de la capacidad disponible para el cliente  $t$ ,  $x(t)$ , y los bid prices  $\pi$ . Las ecuaciones recursivas (hacia adelante) que definen a  $R_t(\dots)$  son las siguientes:

$$R_t(x(t), \pi, \omega) = r^\top \times u(x(t), \pi, l_t, q_t) + R_{t+1}(x(t+1), \pi, \omega) \quad (3.3)$$

$$x(t+1) = x(t) - A \times u(x(t), \pi, l_t, q_t); \quad (3.4)$$

para  $t = 1, \dots, T$ , y con:

$$x(T+1) = 0, \quad R_{T+1}(x, \pi, \omega) = 0 \text{ para todo } x, \pi, \omega, \quad \text{y } x(1) = x > 0.$$

Nuevamente, la interpretación es directa: el ingreso producido por los clientes  $t, t+1, \dots, T$  es el ingreso generado por el cliente  $t$  -la combinación lineal entre los precios de los productos de su

lista y las cantidades asignadas a cada uno de ellos- sumado al ingreso generado por los clientes  $t + 1, \dots, T$ , con la reducción en la capacidad que produce la venta de los productos de  $t$ .

A partir de  $R_t(\dots)$ , podemos definir la función de ingreso total para un sample path:

$$R(\pi, \omega) := R_1(x, \pi, \omega). \quad (3.5)$$

El objetivo que tiene nuestro método entonces es maximizar la función de ingreso total esperado,  $g(\pi) := E[R(\pi, \omega)]$ , sobre los bid prices :

$$\max_{\pi} g(\pi). \quad (3.6)$$

## 3.2. Las derivadas de las funciones de aceptación e ingreso

Para llevar a cabo la optimización, es necesario considerar la continuidad y diferenciabilidad de la función que vamos a maximizar. En primer lugar, es fácil ver que  $u_j(x(t), \pi, l_t, q_t)$  es continua sobre  $\pi$  : basta razonar inductivamente sobre  $j$ , y utilizando como hipótesis inductiva que  $q_t^k$  y  $x_i^k$  son continuas para todo  $k = 1, \dots, b(t, j) - 1$ , y para todo  $i$ , se puede decir que todas las posibles componentes del mínimo en (3.1) son continuas (recordar que pedimos que  $\theta$  lo sea) y que, por consiguiente  $u_j$  lo es, para todo  $j = 1, \dots, n_t$ . Utilizando un razonamiento análogo, pero haciendo inducción (hacia atrás) sobre  $t$ , se puede probar que  $R_t(x(t), \pi, \omega)$  es continua sobre  $\pi$  y sobre  $x(t)$ . Respecto de la diferenciabilidad de  $u$ , y como es de esperarse ante la presencia de un mínimo en la definición, no podemos decir lo mismo. Consideremos, por ejemplo, el siguiente caso: supongamos que  $n_t = 1$  para todas las solicitudes. Entonces los valores de  $\pi_i$  ( $i \in A_j$ ) tales que

$$u_j(x(t), \pi, l_t, q_t) = q_t \theta \left( r_j - \sum_{i \in A_j} \pi_i \right) = x_i^j(t), \text{ con } x_i^j(t) = \min\{x_h^j(t) : h \in A_j\}, \quad (3.7)$$

son potenciales lugares en los que la derivada de  $u_j(x(t), \pi, l_t, q_t)$  respecto de  $\pi_i$ , con  $i \in A_j$ , no existe. La Figura 3.1 ilustra un posible caso de no diferenciabilidad:

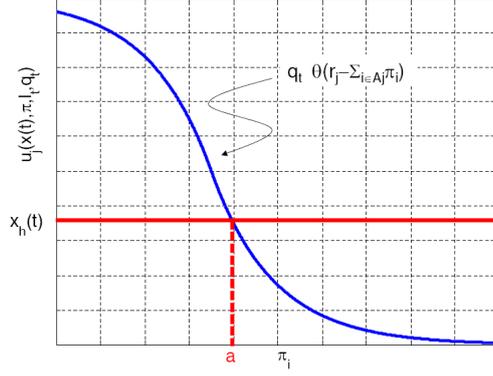


Figura 3.1: No diferenciabilidad de  $u_j(x(t), \pi, l_t, q_t)$  como función de  $\pi_i$ , para  $j = 1$ :  $\pi_i = b$  corresponde al caso  $u_1(x(t), \pi, l_t, q_t) = q_t \theta(r_1 - \sum_{i \in A_1} \pi_i) = x_i(t)$ , con  $x_i(t) = \min\{x_h(t) : h \in A_1\}$ .

Para resolver este problema vamos a hacer una modificación de carácter técnico a la función de ingreso: ante la llegada de cada cliente, vamos a perturbar aleatoriamente las capacidades de los vuelos, de manera de poder asegurar que, con probabilidad 1, nunca va a haber más de un mínimo en la definición 3.1, lo que vuelve a  $u_j(\dots)$  (y por consiguiente a  $R(\dots)$ ) diferenciable con probabilidad 1. La nueva definición de  $R_t(\dots)$ , incluyendo las perturbaciones, es la siguiente:

$$\begin{aligned} R_t(x(t), y, \omega) &= r^\top u(x(t), \pi, l_t, q_t) + R_{t+1}(x(t+1), \pi, \omega) \\ x(t+1) &= x(t) - \zeta_t - A u(x(t), \pi, l_t, q_t). \end{aligned}$$

$$\text{con } \zeta_t \sim U[-\epsilon, \epsilon], \text{ y } \epsilon > 0$$

Ya estamos en condiciones de definir el gradiente de la función de ingreso con respecto a los bid prices y a la capacidad:

$$\begin{aligned} \nabla_\pi R(\pi, \omega) &= \nabla_\pi R_1(x, \pi, \omega) = \left( \frac{\partial}{\partial \pi_1} R_1(x, \pi, \omega), \dots, \frac{\partial}{\partial \pi_m} R_1(x, \pi, \omega) \right). \\ \nabla_x R_1(x, \pi, \omega) &= \left( \frac{\partial}{\partial x_1} R_1(x, \pi, \omega), \dots, \frac{\partial}{\partial x_m} R_1(x, \pi, \omega) \right). \end{aligned}$$

Para calcular la derivada total de  $R_t(x(t), \pi, \omega)$  respecto de  $\pi_i$  procedemos de la siguiente manera: por definición de  $R_t(x(t), \pi, \omega)$ , en (3.4), y por regla de la cadena, tenemos que:

$$\begin{aligned} \frac{\partial}{\partial \pi_i} R_t(x(t), \pi, \omega) &= r^\top \times \frac{\partial}{\partial \pi_i} u(x(t), \pi, l_t, q_t) + \nabla_{x(t+1)} R_{t+1}(x(t+1), \pi, \omega) \times \frac{\partial}{\partial \pi_i} x(t+1) \\ &\quad + \frac{\partial}{\partial \pi_i} R_{t+1}(x(t+1), \pi, \omega) \end{aligned} \tag{3.8}$$

Ahora, utilizando la definición de  $x(t+1)$  en (3.4), tenemos que

$$\frac{\partial}{\partial \pi_i} x(t+1) = -A \times \frac{\partial}{\partial \pi_i} u(x(t), \pi, l_t, q_t) \tag{3.9}$$

Reemplazando esta última igualdad en (3.8)

$$\begin{aligned} \frac{\partial}{\partial \pi_i} R_t(x(t), \pi, \omega) &= r^\top \times \frac{\partial}{\partial \pi_i} u(x(t), \pi, l_t, q_t) + \nabla_{x(t+1)} R_{t+1}(x(t+1), \pi, \omega) \times (-A) \times \frac{\partial}{\partial \pi_i} u(x(t), \pi, l_t, q_t) \\ &\quad + \frac{\partial}{\partial \pi_i} R_{t+1}(x(t+1), \pi, \omega) \end{aligned} \quad (3.10)$$

y sacando factor común:

$$\begin{aligned} \frac{\partial}{\partial \pi_i} R_t(x(t), \pi, \omega) &= (r^\top - \nabla_{x(t+1)} R_{t+1}(x(t+1), \pi, \omega) \times A) \times \frac{\partial}{\partial \pi_i} u(x(t), \pi, l_t, q_t) + \\ &\quad + \frac{\partial}{\partial \pi_i} R_{t+1}(x(t+1), \pi, \omega) \quad \forall i, t \end{aligned} \quad (3.11)$$

con el caso base

$$\frac{\partial}{\partial \pi_i} R_{T+1}(x(T+1), \pi, \omega) = 0, \quad \forall i;$$

Análogamente, se deduce que las derivadas parciales de  $R_t(x(t), \pi, \omega)$  respecto de  $x_i(t)$  tienen la siguiente forma:

$$\begin{aligned} \frac{\partial}{\partial x_i(t)} R_t(x(t), \pi, \omega) &= (r^\top - \nabla_{x(t+1)} R_{t+1}(x(t+1), \pi, \omega) \times A) \times \frac{\partial}{\partial x_i(t)} u(x(t), \pi, l_t, q_t) + \\ &\quad + \frac{\partial}{\partial x_i(t+1)} R_{t+1}(x(t+1), \pi, \omega) \quad \forall i, t \end{aligned} \quad (3.12)$$

con

$$\frac{\partial}{\partial x_i(t+1)} R_{T+1}(x(T+1), \pi, \omega) = 0, \quad \forall i.$$

Podemos ver que ambas derivadas tienen una forma muy similar. Como interpretación se puede pensar que el primer término representa la variación en el ingreso obtenido con el cliente  $t$ : por cada producto  $j$  de la lista, se multiplica lo que varía la cantidad de  $j$  vendida -de acuerdo a incrementos en  $\pi_i$  o  $x_i(t)$  respectivamente-, por la diferencia entre el precio de  $j$  y el costo para la red de venderle una unidad de  $j$  al cliente  $t$  (y no guardarla para después). Dicho costo se calcula sumando los incrementos que se producen en el ingreso, del cliente  $t+1$  en adelante, al disponer de una unidad más de capacidad en todos los recursos que consume  $j$ . Para completar la derivada se suma al término descripto un segundo término con la variación en el ingreso a partir de  $t+1$ .

Para terminar de definir completamente la derivada del ingreso, resta calcular explícitamente las derivadas parciales de  $u_j(x(t), \pi, l_t, q_t)$  respecto de  $\pi_i$  y  $x_i(t)$ . Comenzando con las derivadas respecto de  $\pi_i$ , y a partir de la definición de  $u_j(x(t), \pi, l_t, q_t)$  en (3.1) y en (3.2):

$$\frac{\partial}{\partial \pi_i} u_j(x, \pi, l_t, q_t) = \begin{cases} \frac{\partial}{\partial \pi_i} q_t^j \theta \left( r_j - A_j^\top \pi \right) - q_t^j \dot{\theta} \left( r_j - A_j^\top \pi \right) \mathbb{1}\{i \in A_j\} \\ \quad \text{si la función } \theta \text{ es la única restricción para el producto } j, \text{ es decir, si} \\ \quad 0 < q_t^j \theta \left( r_j - A_j^\top \pi \right) < x_h^j, \forall h \in A_j \\ \frac{\partial}{\partial \pi_i} x_h^j \quad \text{si } x_h^j \text{ es únicamente restrictivo para } j, \text{ es decir, si simultáneamente} \\ \quad \text{i) } x_h^j < \min\{x_l^j : l \in A_j, h \neq l\} \\ \quad \text{ii) } 0 < x_h^j < q_t^j \theta \left( r_j - \sum_{i \in A_j} \pi_i \right) \\ 0 \quad \text{si } u_j(x, \pi, l_t, q_t) = 0 \end{cases} \quad (3.13)$$

Donde:

$$\begin{aligned} \frac{\partial}{\partial \pi_i} x_i^j &= \frac{\partial}{\partial \pi_i} x_i^{[b(t,j)-1]} - \frac{\partial}{\partial \pi_i} u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[b(t,j)-1]}\} \\ &= - \sum_{k=1}^{b(t,j)-1} \left[ \frac{\partial}{\partial \pi_i} u_{[k]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[k]}\} \right], \end{aligned} \quad (3.14)$$

$$\begin{aligned} \frac{\partial}{\partial \pi_i} q_t^j &= \frac{\partial}{\partial \pi_i} q_t^{[b(t,j)-1]} - \frac{\partial}{\partial \pi_i} u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \\ &= - \sum_{k=1}^{b(t,j)-1} \frac{\partial}{\partial \pi_i} u_{[k]}(x(t), \pi, l_t, q_t). \end{aligned} \quad (3.15)$$

Si no hay un único mínimo en la definición de (3.1) podría no existir la derivada<sup>1</sup>.

El primer caso en (3.13) corresponde a cuando  $q_t^j \theta(r_j - \sum_{i \in A_j} \pi_i)$  es el único mínimo en (3.1). Si este es el caso, entonces la cantidad de demanda aceptada para  $j$  varía de acuerdo a un incremento en el valor de  $\pi_i$ , por dos posibles causas: i) el recurso  $i$  es utilizado por el producto  $j$ : esto quiere decir habrá un decremento marginal en el valor de  $\theta(r_j - \sum_{i \in A_j} \pi_i)$ ; y ii) el recurso  $i$  es utilizado por un producto de mayor prioridad que  $j$ . Esto último produce un efecto que cae en la categoría conocida como *efectos de red*, refiriéndose al hecho de que una variación marginal en  $\pi_i$  produce una variación en la cantidad vendida a un producto de mayor preferencia que  $j$  (puede ser un incremento o un decremento), y esto a su vez repercute en la aceptación de  $j$ . Específicamente, en el primer caso de la definición en (3.13), un efecto de red puede hacer que la cantidad aceptada por un producto de mayor prioridad varíe, y eso haga que la demanda disponible para  $j$ ,  $q_t^j$ , cambie, haciendo que la función  $\theta$  sea reescalada.

Similarmente, el segundo caso en (3.13) se refiere a un efecto de red, pero ahora el único mínimo es un recurso  $h$ . Una variación en el valor de  $\pi_i$  podría afectar la cantidad vendida de un producto de mayor preferencia, que utilice el recurso  $h$ , y cambiar la disponibilidad que tenga el producto  $j$  de  $h$ . Esto modificaría, ya que  $h$  es el recurso restrictivo, la cantidad de  $j$  vendida.

El tercer caso ocurre cuando la función de aceptación es 0, es decir  $u_j(\dots) = 0$ , lo que quiere decir que algún producto  $k$  de mayor prioridad que  $j$  ha agotado algún recurso que  $j$  utiliza (recordar

<sup>1</sup>Excepto cuando  $x_i^j = x_h^j = 0$  para algún producto  $j$ , y recursos  $i$  y  $h$ . En tal caso  $u_j(\dots) = 0$ , y hay dos mínimos en la definición de (3.1), pero la derivada existe y es 0

que, como dijimos en las sección anterior, la demanda  $q_t$  no puede ser agotada por ningún recurso). Sea  $h$  el recurso agotado. Asumiendo que hay un único mínimo en la definición (3.1) en todos los productos, cualquier variación en la aceptación de los productos de mayor preferencia, que utilizan  $h$ , va a ser “absorbida” por  $k$ , con lo que nunca afectará al producto  $j$ . Esto quiere decir que la derivada de  $u_j(\dots)$  es 0, respecto de cualquier bid price.

Vamos a tratar de ilustrar un poco mejor el funcionamiento de los diferentes casos de  $\frac{\partial}{\partial \pi_i} u_j(\dots)$  a través de ejemplos visuales. Para ello vamos a interpretar a la demanda del cliente y los recursos como si se tratara de flúidos. De esta manera, la política de asignación de recursos consiste en ir consumiendo la demanda  $q_t$  junto con cada uno de los recursos utilizados por los productos, hasta que se agote alguno de ellos o se acaben los productos de la lista de preferencia. La función  $\theta$  se puede pensar como un tope para el consumo de cada producto. Si vemos, por ejemplo, la Figura 3.2 superior, tenemos representada una red, y la venta de los productos de un cliente. La red tiene 3 recursos (representados por tres barras verticales, la cuarta representa el consumo de la demanda) y 3 productos (representados por diferentes tonos de gris). El producto 1 utiliza los recursos 1 y 2, el producto 2 los recursos 2 y 3, y el producto 3 únicamente el recurso 3. Las 3 capacidades se suponen igual a 1, el cliente tiene la lista de preferencia: [1, 2, 3] y solicita  $q_t = 1$ . Se puede observar cómo el espacio disponible en cada barra se va consumiendo por los productos en orden (de abajo hacia arriba). Si un producto consume menos del mínimo del fluido restante en los recursos que usa (y de la demanda restante), es porque está coartado por  $\theta$ . Este es el caso de los 3 productos en las Figuras 3.2 superior e inferior.

Teniendo en cuenta esto, la Figura 3.2 inferior representa qué ocurre con la aceptación de los productos cuando se perturba el valor de  $\pi_1$ . Podemos ver cómo se vende menos del producto 1 que antes debido a que decrece el valor de  $\theta(r_j - (\pi_1 + \pi_2))$ , y que, como consecuencia, el producto 1 libera demanda  $q_t$  que es aprovechada (al reescalar el valor de  $\theta$ ) por los productos 2 y 3 .

Ahora supongamos que tenemos la misma red y cliente que en la Figura 3.2, pero con un valor diferente de  $\pi$ , y esta vez el cliente solicita  $q_t = 1.2$ . La Figura 3.3 superior ilustra esta situación. En este caso, el factor que domina el mínimo de  $u_j(\dots)$  para el producto 2 es la disponibilidad del recurso 2. En la parte inferior vemos lo que ocurre en este caso al incrementar marginalmente el valor de  $\pi_1$ : para el primer y segundo producto de la lista ocurre algo similar a lo que ocurría en el ejemplo anterior, se reduce la cantidad aceptada del primero y como consecuencia aumenta la del segundo. Sin embargo, podemos ver que la cantidad aceptada para el producto 3 no varía. La diferencia fundamental con respecto al caso de la Figura 3.2 radica en que en este caso lo que domina al mínimo del producto 2 es un recurso. Cuando el producto 1 libera este recurso, junto con demanda  $q_t$ , todo lo liberado de recurso 2 y de demanda  $q_t$  es absorbido por el producto 2, con lo que el producto 3 recibe exactamente la misma demanda residual que antes, y su aceptación no cambia (es decir,  $\frac{\partial}{\partial \pi_1} u_3(\dots) = 0$ ).

Es importante notar que en ambos ejemplos, y por razones obvias, el incremento que se hace en el valor de  $\pi_1$  no es infinitesimal, como debiera ser, si se está hablando de la derivada de la función de aceptación. En realidad, independientemente del tamaño del incremento, lo más importante es que la variación no sea lo suficientemente grande como para que cambie el mínimo en la definición

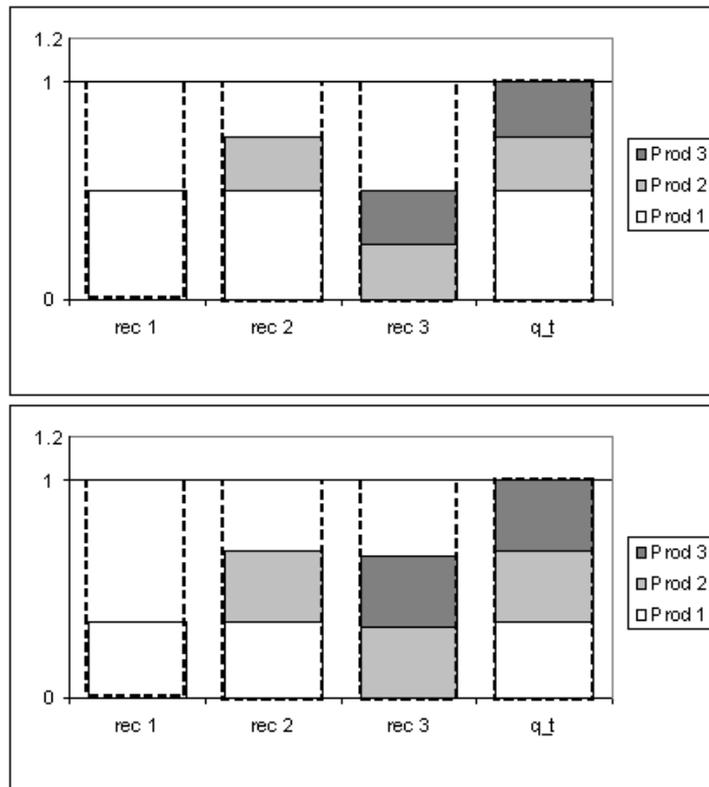


Figura 3.2: Arriba: Una red, y la aceptación de productos de un cliente, para un valor fijo de  $\pi$ . Abajo: la misma red pero incrementando ligeramente el valor de  $\pi_1$

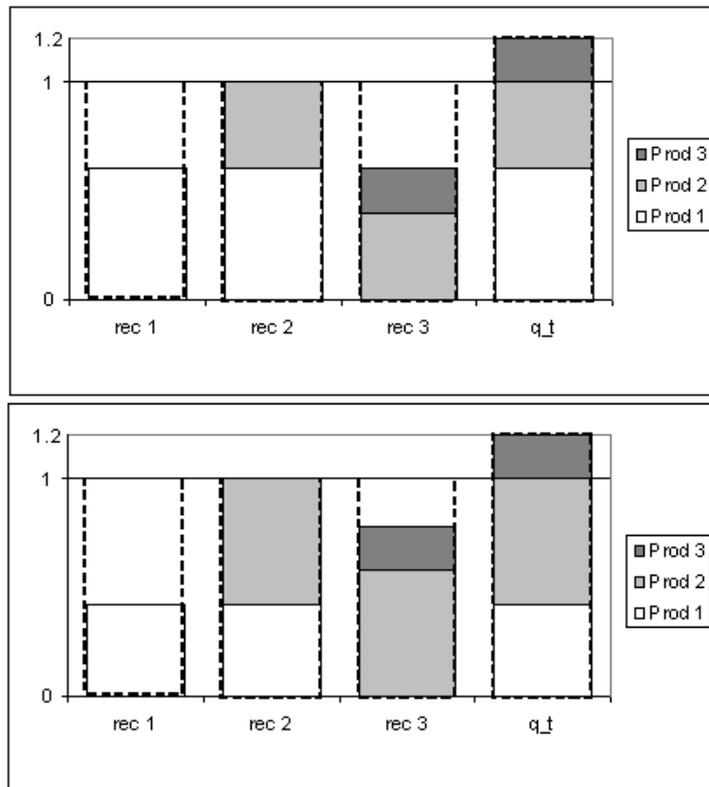


Figura 3.3: Arriba: La misma red que en la Figura 3.2, pero para otro valor de  $\pi$ , con el que la aceptación del producto 2 está coartada por el recurso 2. Abajo: la misma situación pero después de incrementar ligeramente  $\pi_1$

de  $u_j(\dots)$ . En ambos casos tuvimos cuidado en que esto no ocurra, con lo que se trata de ejemplos que ilustran correctamente el comportamiento de la derivada.

Para terminar con esta sección resta definir las derivadas parciales de la función de aceptación, respecto de  $x_i(t)^2$ . Haciendo razonamientos similares a los anteriores obtenemos la siguiente fórmula:

$$\frac{\partial}{\partial x_i} u_j(x, \pi, l_t, q_t) = \begin{cases} 1 & \text{si la capacidad del leg } i \text{ } x_i^j \text{ es el único mínimo para el producto } j, \text{ es decir, si} \\ & \text{i) } i \in A_j \\ & \text{ii) } x_i^j < x_h^j, \forall h \in A_j, h \neq i \\ & \text{iii) } 0 < x_i^j < q_t^j \theta(r_j - A_j^\top \pi) \\ \frac{\partial}{\partial x_i} x_h^j & \text{si } x_h^j \text{ es el único mínimo para } j, h \neq i, \text{ es decir, si} \\ & \text{i) } x_h^j < \min\{x_l^j : l \in A_j, h \neq l, h \neq i\} \\ & \text{ii) } 0 < x_h^j < q_t^j \theta(r_j - A_j^\top \pi) \\ \frac{\partial}{\partial x_i} q_t^j \theta(r_j - A_j^\top \pi) & \text{si la función } \theta \text{ es el único mínimo para } j, \text{ es decir, si} \\ & 0 < q_t^j \theta(r_j - A_j^\top \pi) < x_h^j, \forall h \in A_j \\ 0 & \text{si } u_j(x, \pi, l_t, q_t) = 0 \end{cases} \quad (3.16)$$

Con

$$\begin{aligned} \frac{\partial}{\partial x_i} x_h^j &= \frac{\partial}{\partial x_i} x_h^{[b(t,j)-1]} - \frac{\partial}{\partial x_i} u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[b(t,j)-1]}\} \\ &= - \sum_{k=1}^{b(t,j)-1} \frac{\partial}{\partial x_i} u_{[k]}(x(t), \pi, l_t, q_t) \mathbb{1}\{i \in A_{[k]}\}, \\ \frac{\partial}{\partial x_i} q_t^j &= \frac{\partial}{\partial x_i} q_t^{[b(t,j)-1]} - \frac{\partial}{\partial x_i} u_{[b(t,j)-1]}(x(t), \pi, l_t, q_t) \\ &= - \sum_{k=1}^{b(t,j)-1} \frac{\partial}{\partial x_i} u_{[k]}(x(t), \pi, l_t, q_t). \end{aligned}$$

La interpretación de la derivada es muy similar a la anterior. La mayor diferencia se encuentra en el primer caso, en el que vale 1. Lo que ocurre, es que el recurso  $i$  respecto del cual se deriva es el mismo que es restrictivo para el cliente  $j$ . Por esta razón, cualquier incremento marginal en  $x_i$  va a ser absorbido en su totalidad por el producto  $j$  (recordar que, como hay un único mínimo, hay suficiente disponibilidad del resto de los recursos que usa  $j$ ), y de ahí que el incremento de la aceptación sea igual al incremento en la disponibilidad del recurso (lo que equivale a decir que la derivada es 1). Los demás casos son similares a los casos de la derivada respecto de  $\pi_i$ : el segundo y el tercero se refieren a las otras 2 posibilidades de mínimo en la definición de  $u(\dots)$ , y representan esencialmente efectos de red, y el cuarto es equivalente al tercer caso en (3.13).

### 3.3. Cálculo de las derivadas

En esta sección vamos a definir un algoritmo para calcular las derivadas parciales de la función de ingreso. Teniendo en cuenta lo dicho en la sección 1.2.3, el núcleo de un método de gradientes

<sup>2</sup>Por simplicidad notacional, escribimos a esta variable directamente como  $x_i$

estocásticos es, justamente, el cálculo de los gradientes. Como pudimos apreciar a través de fórmulas y ejemplos en la sección anterior, dicho cálculo no es nada trivial en nuestro caso. Además es necesario que sea eficiente ya que, en el método muestral que presentamos, va a ser realizado una gran cantidad de veces.

El algoritmo toma como entrada una red, un sample path, y un vector de bid prices  $\pi$ , y calcula el gradiente de la función de ingreso, evaluado en  $\pi$ :  $\nabla_{\pi} R(\pi, \omega)$ . Esencialmente lo que hace es recorrer el sample path en dos pasadas, una hacia adelante y otra hacia atrás. En la pasada hacia adelante, se hace una simulación de ventas (asumiendo que tanto capacidad como demanda son continuas) y para cada cliente y producto en la lista de preferencia se guardan 2 cosas: i) cuál es la *binding condition* del producto, es decir cuál es el mínimo en la definición de  $u(\dots)$  (se guardan 3 opciones posibles, una por cada caso de la definición de  $\frac{\partial}{\partial \pi_i} u$  en (3.13) ); y ii) cuál es el valor de  $u_j(\dots)$  para cada producto. Una vez hecho esto se procede con la pasada hacia atrás en la que, por cada cliente, se van calculando las derivadas de  $u(\dots)$  y con éstas las derivadas de  $R(\dots)$ . El pseudocódigo del método que realiza este trabajo es el siguiente:

---

```

1 // Variables:
2 // resources(j) es un conjunto que contiene los recursos utilizados por el producto j
3 // rev[j] es el precio del producto j
4 // x[i] es la capacidad del recurso i.
5 // pi[i] es el bid price del recurso i.
6 // BC(j,t) define la "Binding Condition" (cu' al es el unico minimo) del producto j, cliente t.
7 //   Puede tomar 3 valores: "THETA" (etiqueta, cuando se trata de le funcion theta) ,
8 //   el indice del unico recurso restrictivo (entero con el numero de recurso)
9 //   o "NOLEFT" (etiqueta, cuando uno de los recursos del producto se agoto)
10 // u[t][j] guarda el valor de u para el producto j del cliente t.
11 // j[t,k] guarda el k-esimo elemento de la lista del cliente t
12 // OUTPUT: grad_R_pi[i] contendra el gradiente de R respecto de pi
13
14 /**PASADA HACIA ADELANTE:**/
15 para (t := 1 hasta T){
16     q_j := q[t]
17
18     para (k := 1 hasta l_t){
19         j := j[t,k] // El producto j es el k-esimo product de la lista del cliente t.
20         min_res := 0 // Indice del recurso restrictivo
21         min_res_quant := +inf // cantidad restante del recurso restrictivo
22         SUM_pi = 0
23         para cada (i en resources(j)){
24             si(x[i] > 0 y i no fue perturbado)
25                 perturbar(x[i])
26             si (x[i] < min_res_quant){
27                 min_res := i
28                 min_res_quant := x[i]
29                 SUM_pi += pi[i]
30             }
31         }
32         tetha_j := q_j*Theta(rev[j] - SUM_pi)
33
34         case:{ // Hay tres casos posibles en la definicion del minimo de u_j
35
36             [Condition 1: tetha_j < min_res_quant]

```

```

37         BC(j,t) := THETA
38         u_j := theta_j
39         [Condition 2: 0 < min_res_quant < tetha_j]
40         BC(j,t) := min_res
41         u_j := min_res_quant
42         [Condition 3: min_res_quant == 0]
43         BC(j,t) := NOLEFT
44         u_j := 0
45     }// fin case
46
47     u[t][j] := u_j
48     para cada (i en resources(j)){
49         x[i] -= u_j
50     }
51
52     q_j -= u_j
53 }// End para k
54
55 }// End fin pasada hacia atras
56
57 /** PASADA HACIA ATRAS */:
58 // Variables:
59 // grad_u_pi[j][i] contendra los gradientes de u_j con respecto to pi_i
60 // grad_u_x[j][i] contendra los gradientes de u_j con respecto to x_i(t)
61 // temp_grad_R_x[] guardara valores temporales para el calculo
62 // del gradiente de R con respecto a x(t)
63 para (t := T hasta 1){
64     // Used_resources contendra los recursos utilizados por
65     // todos los productos en la lista de preferencia del cliente t.
66     used_resources := empty_set
67     q^j = q[t]
68     para (k := 1 hasta l_t){
69         j := j[t,k]
70         SUM_pi := 0;
71         SUM_grad_R_x_t := 0;
72         para cada ( i en resources(j)){
73             SUM_pi += pi[i]
74             SUM_grad_R_x_t += grad_R_x[i]
75             used_resources.add(i);
76         }
77
78         case:{
79             [Condition 1: BC(j,t) == THETA]
80
81             para cada (i en used_resources){
82                 //variables temporales
83                 deriv_q^j_pi_i := 0
84                 deriv_q^j_x_i := 0
85                 // Recorrer todos los productos de mayor preferencia
86                 // para calcular 'deriv_q^j_pi_i' y 'deriv_q^j_x_i'
87                 para (k2 := 1 to k - 1){
88                     j2 := j[t,k2]
89                     deriv_q^j_pi_i -= grad_u_pi[j2][i]
90                     deriv_q^j_x_i -= grad_u_x[j2][i]
91                 }
92                 //calcular la derivada de u_j con respecto a x_i y pi_i
93                 arg := rev[j] - SUM_pi

```

```

94         grad_u_pi[j][i] = deriv_q^j_pi_i*Theta(arg) -
95             q^j*DerivTheta(arg)*[if(i en recursos(j))then 1 else 0]
96     grad_u_x[j][i] = deriv_q^j_x_i*Theta(arg)
97     grad_R_pi[i] += (rev[j] - SUM_grad_R_x_t) * grad_u_pi[j][i]
98     temp_grad_R_x[i] += (rev[j] - SUM_grad_R_x_t) * grad_u_x[j][i]
99 } // fin para cada i
100 [Condition 2: BC(j,t) == i] // i es el unico recurso restrictivo.
101     grad_u_x[j][i] = 1
102     para cada (i2 en used_resources){
103         para (k2 := 1 hasta k - 1){
104             j2 := j[t,k2]
105             si (i esta en recursos(j2)){
106                 grad_u_pi[j][i2] -= grad_u_pi[j2][i2]
107                 grad_u_x[j][i2] -= grad_u_x[j2][i2]
108             }
109         }
110         grad_R_pi[i2] += (rev[j] - SUM_grad_R_x_t) * grad_u_pi[j][i2]
111         temp_grad_R_x[i2] += (rev[j] - SUM_grad_R_x_t) * grad_u_x[j][i2]
112     } // fin para cada i2
113 [Condition 3: BC(j,t) == NOLEFT] // Representa el caso en que u_j(..) = 0 porque
114     // No hacer nada // se agoto un recurso que utiliza j.
115
116     q^j -= u(t,j)
117 } // fin para k
118
119 // Ahora actualizar el valor de grad_R_x
120 para cada (i en used_resources){
121     grad_R_x[i] += temp_grad_R_x[i]
122 }
123 } // fin pasada hacia atras

```

Como se ve, la pasada hacia adelante tiene una interpretación bastante directa. La pasada hacia atrás resulta un poco menos intuitiva, pero no deja de ser una implementación más o menos directa de las fórmulas presentadas en la sección anterior. Lo que se hace es ir retrocediendo en los clientes (el retroceso se debe a que las ecuaciones de  $R(\dots)$  y su derivada son recursivas hacia adelante) y por cada uno ir calculando el primer término de  $\frac{\partial}{\partial \pi_i} R_t(\dots)$  y  $\frac{\partial}{\partial x_i(t)} R_t(\dots)$ , definido en (3.11) y (3.12), y sumárselo a lo acumulado hasta el momento (representado por el segundo término). Como lo que se necesita es el gradiente completo, hay que hacer dicho cálculo para todo recurso  $i$ . Sin embargo, en realidad basta con recorrer solamente los recursos *relevantes* para el cliente  $t$  y producto  $j$ : aquellos recursos que son utilizados por el producto  $j$  o por alguno de los productos de mayor preferencia que éste. Para esto se utiliza el conjunto `used_resources`: se recorren sólo los recursos definidos en él (línea 81 del pseudocódigo), y para el resto la derivada de  $u(\dots)$  es simplemente 0, con lo que se anula el primer término de (3.11) y (3.12) y no hay nada que hacer. Este hecho es importante para la reducción en la complejidad del método, como veremos después.

Para calcular el primer término de (3.11) y (3.12) se recorren los productos en orden ascendente, y se van calculando y guardando las derivadas de  $u_j(\dots)$  respecto de  $x(t)$  y de  $\pi$  para cada  $j$ , utilizando las de mayor preferencia, ya calculadas anteriormente. Dependiendo cuál sea la *binding condition* para cada producto (calculada en la pasada hacia adelante) se utilizan los diferentes casos de las definiciones (3.13) y (3.16) (ver `case` en la línea 78). Una vez calculadas  $\frac{\partial}{\partial \pi_i} u_j(\dots)$  y  $\frac{\partial}{\partial x_i(t)} u_j(\dots)$ ,

se multiplican dichos valores por la diferencia entre el precio de  $j$  y la sumas:  $\sum_{i \in A_j} \frac{\partial}{\partial \pi_i} R_{t+1}(\dots)$  y  $\sum_{i \in A_j} \frac{\partial}{\partial x_i(t+1)} R_{t+1}(\dots)$  respectivamente; luego se suma el resultado al valor de la derivada de  $R$  acumulado hasta el momento, según indican (3.11) y (3.12), y se prosigue con el siguiente producto -líneas 97 y 98, y 110 y 111-.

### ***Cálculo de complejidad***

Para calcular la complejidad del algoritmo, vamos a asumir dos hechos importantes. Por un lado, asumiremos que para cualquier sample path existe una cota para el tamaño de las listas de preferencia: a este valor lo llamaremos  $L$ . Es importante notar que asumir esto es razonable para una aplicación real de venta de pasajes; es esperable que cuando un cliente decide volar de un punto a otro sólo considere un número acotado, y más aún *reducido*, de opciones para llevarlo a cabo. Por otro lado, vamos a asumir que la cantidad de recursos que puede llegar a utilizar un producto está acotada por un número chico  $M$ . Nuevamente, pensando en el negocio de las aerolíneas, la cantidad de paradas para un itinerario real no va a sobrepasar las 3 o 4, lo que da un valor de  $M$  de 4 o 5.

Dicho esto, pasamos a calcular detalladamente el orden de complejidad del algoritmo. Dado que son independientes, calcularemos en forma separada la complejidad de la pasada hacia adelante y la pasada hacia atrás:

Pasada hacia adelante:

---

- Ciclo general: líneas 15 a 55. Se ejecuta  $T$  veces.

[

- Se realiza el ciclo que recorre los productos de la lista (líneas 18 a 53.): Se ejecuta a lo sumo  $L$  veces.

[

- Se realizan dos ciclos que recorren los recursos del producto  $j$ .

Ambos tienen complejidad  $O(M)$ .

- El resto de las operaciones tiene complejidad  $O(1)$ .

]: $O(L \times M)$

]: $O(T \times (L \times M))$

---

Complejidad total de la pasada hacia adelante:  $O(T \times L \times M)$

Pasada hacia atrás:

---

- Ciclo general: lineas 63 a 123. Se ejecuta T veces.

[

- Se realiza el ciclo que recorre los productos de la lista: lineas 68 a 117. Se ejecuta a lo sumo L veces.
- [

  - Ciclo que suma sobre los recursos del producto j (lineas 72 a 76):  $O(M)$
  - Case: El primer y segundo caso tienen la misma complejidad. y el tercero es  $O(1)$ , vamos a asumir que en peor caso se ejecutan el segundo o el tercero en todas las ocasiones:
  - [

    - Se recorren los recursos utilizados hasta el momento, guardados en usedresources (lineas 81 a 99 y 102 a 112). La cota superior para la cantidad es  $O(L \times M)$
    - [

      - Se recorre nuevamente la lista de productos desde el primero hasta el actual j (lineas 87 a 91 y 103 a 109):  $O(L)$
      - Se realizan calculos, todos  $O(1)$

]:  $O((L \times M) \times L)$

]:  $O(L \times M \times L)$

]:  $O(L \times (L \times M \times L))$

- Se realiza un ciclo sobre los recursos en used`resources:  $O(L \times M)$

]:  $O(T \times (L \times L \times M \times L))$

---

Complejidad total de la pasada hacia atrás:  $O(T \times L^3 \times M)$

La complejidad del algoritmo es la mayor de las dos complejidades, es decir:  $O(T \times L^3 \times M)$ . En este punto, queda claro cuál es la ventaja de reconocer la existencia de las cotas  $L$  y  $M$ . Si bien el algoritmo es cúbico sobre  $L$ , como dijimos antes, se espera que este valor sea relativamente chico, con lo que, en el orden teórico y en la práctica, se trata de un algoritmo lineal en el tamaño del sample path.

### 3.4. Esquema de alto nivel del método

Presentamos a continuación un esquema a alto nivel del funcionamiento del método de gradientes estocásticos que desarrollamos, ilustrado en la Figura 3.4:

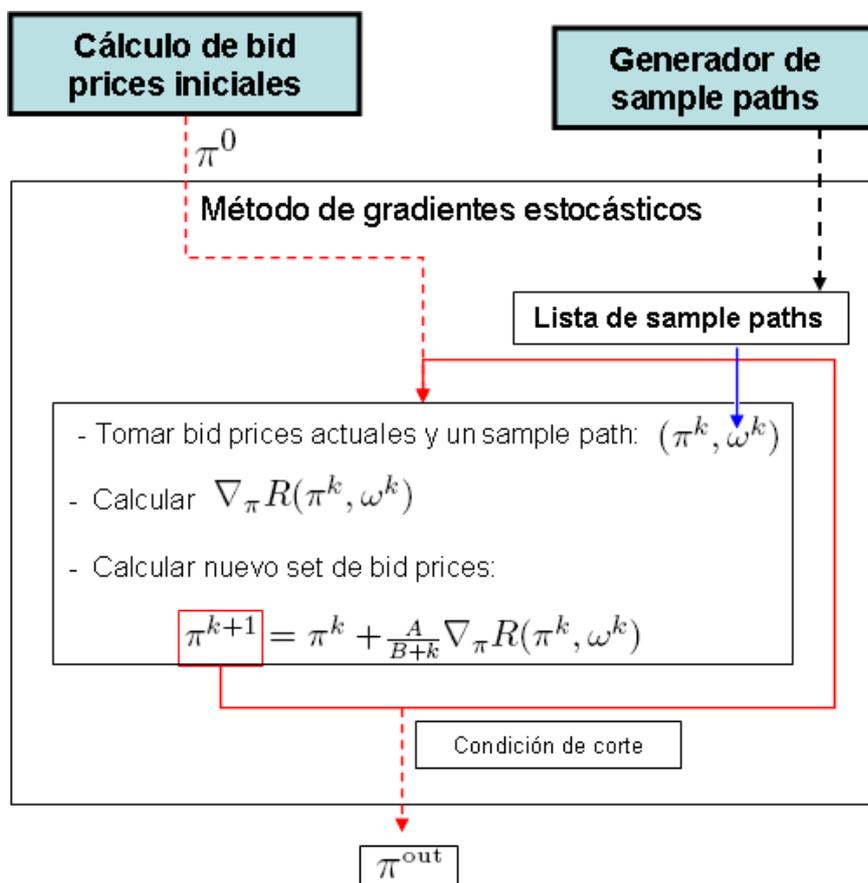


Figura 3.4: Esquema general del método de gradientes estocásticos

Entre las diferentes componentes del método que podemos identificar en la Figura, el núcleo es una adaptación de la metodología ya descrita en la sección 1.2.3, y el generador de sample paths representa un simulador que implementa un modelo de demanda, por ejemplo alguno de los descritos en la sección 2.2. Sobre el cálculo de bid prices iniciales sólo hemos dicho hasta ahora que tenía una influencia importante sobre la eficacia del método. Lo que haremos es implementar el DLP, un método que genera bid prices, ampliamente utilizado en el *Revenue Management* real, y utilizaremos estos valores como punto de partida para nuestro método, con la esperanza de que éste los mejore (más adelante describimos con exactitud la formulación de DLP). Por último, para la condición de corte, que decide en qué momento finaliza la iteración, utilizaremos cantidad de iteraciones y la norma del gradiente muestral actual (que idealmente debería converger a 0).

### 3.5. Algunas consideraciones restantes

#### *La elección de la función $\theta$*

En la sección 2.2, cuando introdujimos los conceptos básicos del procesamiento de sample paths, definimos a la función  $\theta(y)$ , y las propiedades que debería cumplir. Una vez definido el método, en el presente capítulo resta definir la función, o familia de funciones, que vamos a utilizar para  $\theta$ . Para ello, vamos a hacer una pequeña generalización del enfoque utilizado por Topaloglu[Top08], y vamos a definir a  $\theta$  de la siguiente manera:

$$\theta(y) = \begin{cases} 1 - a e^{-\frac{1-a}{b}(y-\gamma)} & \text{if } y \geq \gamma \\ (1-a) e^{\frac{a}{b}(y-\gamma)} & \text{if } y < \gamma, \end{cases}$$

donde  $a \in (0, 1)$  es un parámetro de forma,  $b > 0$  es un parámetro de escala y  $\gamma$  uno de centralidad, todos a calibrar, según la calidad de los resultados. La Figura 3.5 ilustra algunos casos de  $\theta(y)$

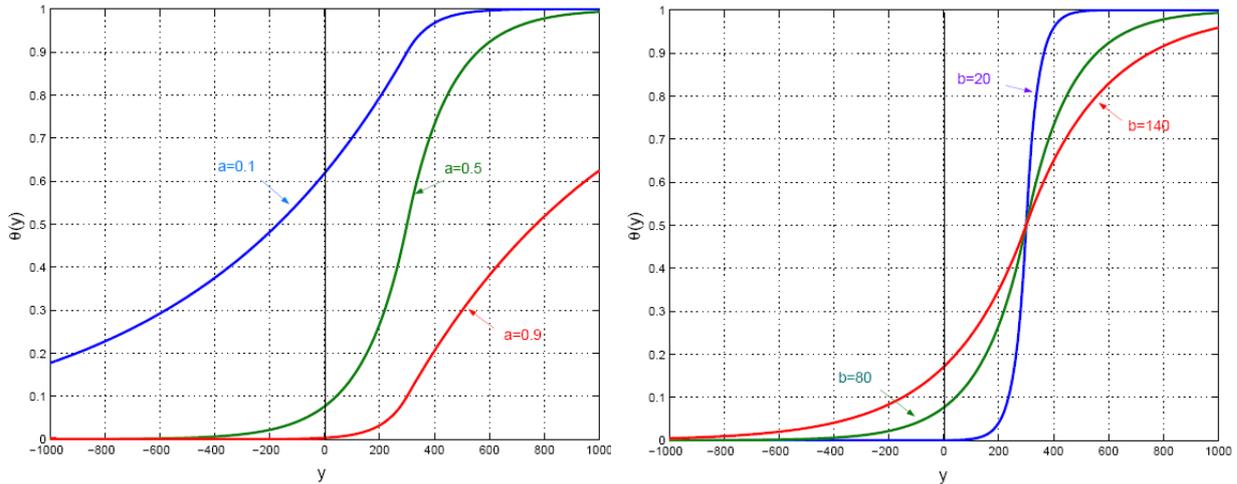


Figura 3.5: Diferentes version de la función  $\theta(y)$ , según variaciones en  $a$  y  $b$ , con  $\gamma = 300$ . Izquierda: impacto de los cambios en el parámetro  $a$ . Derecha: impacto de los cambios en el parámetro  $b$ .

La razón para elegir esta definición de  $\theta(y)$  y no otra, no está lejos de ser arbitraria. Simplemente consideramos que los 3 parámetros de los que depende proveen suficiente generalidad como para englobar los posibles comportamientos de otras definiciones de  $\theta$ , que cumplan con las características que le pedimos. Notar por ejemplo que reduciendo el valor de  $b$ ,  $\theta$  va tendiendo a una función escalonada, lo que acerca el comportamiento entre el procesamiento discreto y continuo. Intuitivamente, esto puede ser beneficioso desde el punto de vista de la optimización, ya que de alguna manera reduce la distancia entre la relajación y el problema real. Sin embargo, valores chicos de  $b$  hacen crecer arbitrariamente la derivada de  $\theta$ , lo que trae problemas numéricos, y experimentalmente no hemos encontrado mejoras sustanciales. De hecho hemos comprobado, también experimentalmente, que el algoritmo tiene una eficacia relativamente independiente a la elección de  $a$ ,  $b$  y  $\gamma$ .

### Concavidad de $R(\pi, \omega)$

Un resultado negativo, que no podíamos dejar de mencionar, es respecto de la concavidad de la función de ingreso. Dado que estamos resolviendo un problema de maximización, idealmente deseáramos que la función de ingreso sea cóncava sobre  $\pi$ , para cualquier realización de  $\omega$ . Lamentablemente este no es el caso; ni siquiera podemos garantizar la cuasi-concavidad de  $R(\pi, \omega)$ <sup>3</sup>. La Figura 3.6 representa un caso en el que la función de ingreso no es cuasicóncava. El ejemplo es muy similar a los de la sección 3.2, sólo que en este caso se trata de 3 clientes  $t = 1, 2, 3$ , con listas de preferencias [1], [2] y [3] respectivamente, y en todos los casos piden cantidad  $q_t = 1$ . Los precios de los productos son: (\$100, \$100, \$300).

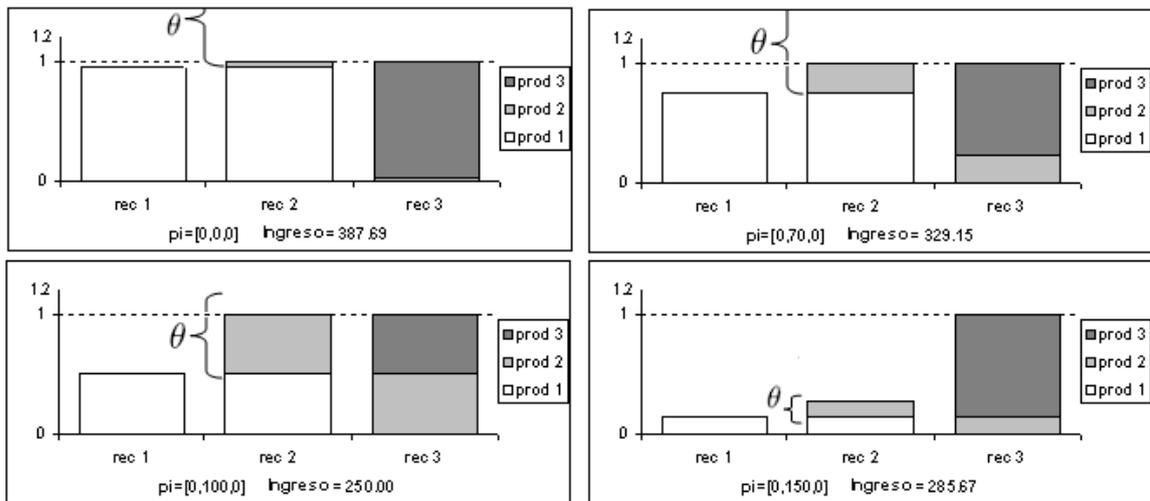


Figura 3.6: Ejemplo de no cuasiconcavidad: de izquierda a derecha, y de arriba hacia abajo, variación en el ingreso obtenido, en función del crecimiento de  $\pi_2$

La no cuasiconcavidad se observa cuando se va incrementando el valor de  $\pi_2$ , fijando el resto de los bid prices, y se ve que la función decrece hasta un punto, y luego vuelve a crecer. La razón por la que decrece en los primeros pasos, es que el aumento del bid price  $\pi_2$  reduce la aceptación del producto 1 (que está sujeta al valor de  $\theta$ ), pero esto es compensado por una mayor aceptación del producto 2 (cuya restricción era el recurso 2, liberado ahora por el producto 1), y esto a su vez hace que se acepte menos producto 3. Como el producto 3 es más caro que los productos 1 y 2 (que salen lo mismo, con lo que el intercambio de uno por otro no cambia el ingreso), esto reduce la ganancia producida. En determinado momento, sin embargo, el aumento de  $\pi_2$  hace que cambie la condición restrictiva para el producto 2: antes era el recurso 2, y en la cuarta Figura pasa a ser  $\theta$  (en todas las figuras se indica con un  $\theta$  y una llave cuánto da el valor de  $q_t^2 \theta (r_j - \sum_{i \in A_j} \pi_i)$  para el

<sup>3</sup>Una función  $f : X \subseteq \mathbf{R}^n \rightarrow \mathbf{R}$  definida sobre un conjunto convexo  $X$  se dice cuasi-cóncava sobre  $X$  si y sólo si para todo  $x_1, x_2 \in X$  y  $\lambda \in [0, 1]$   $f(\lambda x_1 + (1 - \lambda)x_2) \geq \min(f(x_1), f(x_2))$

producto  $j = 2$ ). A partir de entonces, todo aumento en el valor de  $\pi_2$  produce una reducción en la aceptación de los productos 1 y 2 (ambos restringidos por  $\theta$  ahora) que es transferida al producto 3, y como éste es más caro que la suma de los otros dos, el ingreso crece.

Por si queda alguna duda de que se trata de un caso en el que  $R$  no es cuasicóncava, la Figura 3.7 contiene un gráfico de  $R$  para el mismo ejemplo, en función de  $\pi_2$ :

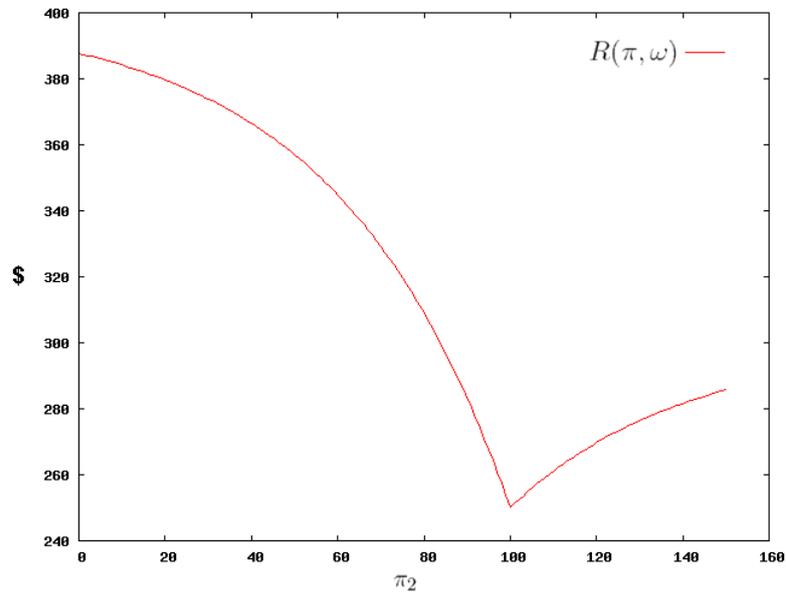


Figura 3.7:

La conclusión inmediata respecto de este ejemplo, es que no podemos asegurar que el método que proponemos converja a un máximo de la función de ingreso. El gradiente nulo puede corresponder a un punto estacionario, o incluso a un mínimo local; sólo podemos esperar tener suerte, y que esto no ocurra.

### ***Bid prices iniciales: El DLP***

El *Programa Lineal Determinístico* (DLP, por sus iniciales en inglés) es uno de los métodos más utilizados en la práctica del *Revenue Management* para problemas de red, más que nada debido a su formulación simple, y a su facilidad de resolución. Fue introducido por Simpson[Sim89], y ha sido estudiado y extendido desde entonces. Sus principales desventajas son el hecho de que se basa en que la demanda para cada producto se considera independiente (cosa que, como hemos dicho ya repetidas veces, en general no es realista) y que resuelve el problema determinísticamente: toma como entrada la media de las demandas de los productos como si fuera la demanda real, sin tener en cuenta ninguna componente aleatoria. Más allá de estas falencias, se han mostrado buenas propiedades teóricas del DLP (como por ejemplo, optimalidad asintótica bajo ciertas condiciones Cooper[Coo02]) y buena performance práctica, sobre todo al usar su solución dual como valor de

bid prices.

Siguiendo la notación utilizada en este trabajo, la formulación del DLP es la siguiente:

$$\begin{aligned}
 & \max \quad r^\top \times z \\
 & \text{s.a.} \\
 & \sum_{j=1}^n z_j \mathbf{I}\{i \in A_j\} \leq x_i \quad \text{para } i = 1 \dots m \\
 & 0 \leq z_j \leq E[D_j] \quad \text{para } j = 1 \dots n
 \end{aligned} \tag{3.17}$$

donde  $z_1, \dots, z_n$  son las variables de decisión, que indican qué cantidad de asientos se vende de cada producto  $j$ . Las desigualdades en el segundo renglón aseguran que no se violen las capacidades de los vuelos, y las cotas que la cantidad que se vende de un producto no supere a la demanda pronosticada (que se estima con la media).

Lo que haremos nosotros es tomar la solución dual  $\pi = (\pi_1, \dots, \pi_m)$  y utilizarla como set de bid prices iniciales para el algoritmo de gradientes estocásticos. Una vez hecha la optimización, se utiliza esta solución inicial como punto de comparación con el nuevo set de bid prices, para ver qué mejora provee el algoritmo.

### ***Bid prices negativos***

Cuando en la sección 1.2.3 introdujimos el concepto de optimización estocástica y formulamos el problema general de programación estocástica, hablamos de la *región de soluciones factibles* sobre la cual estaba definido el problema. Más adelante, en la misma sección, hicimos la descripción de la metodología que utilizaríamos en nuestro trabajo para resolver el problema que nos concierne, que luego fue enunciada con mayor detalle en el presente capítulo. Sin embargo, existe una salvedad en la formulación del método, que no hemos mencionado hasta ahora, pero que consideremos importante dejar clara. Recordemos que la definición de la iteración general del algoritmo de gradientes estocásticos, según la enunciamos, es:

$$x_{k+1} = x_k + \alpha_k \widehat{\nabla} g(x_k)$$

La pregunta que podría surgir es qué ocurriría si, al efectuar uno de los pasos,  $x_{k+1}$  quedara fuera de la región factible (que notamos como  $X$ ). No se trata de una cuestión menor, pero existe una razón para que hayamos podido pasarla por alto, y es el hecho de que, en nuestro caso,  $X = \mathbf{R}^m$ ; es decir, nunca vamos a tener ese problema. Cuando la región de factibilidad está estrictamente incluida en  $\mathbf{R}^m$  lo que se hace es agregar en el algoritmo una proyección ortogonal sobre la región de factibilidad. En caso de quedar  $x_{k+1}$  dentro de  $X$ , esta proyección es simplemente la identidad.

Ahora, si pensamos nuevamente en el hecho de que cualquier set de bid prices se considera válido (es decir  $X = \mathbf{R}^m$ ), esto parece contradecir la interpretación que hicimos de los mismos,

como el valor que se le asigna a una unidad de capacidad en un vuelo de la red. Lo cierto es que tiene sentido considerar bid prices negativos, ya que existen configuraciones en los que el valor óptimo contiene necesariamente un bid price negativo, como por ejemplo, el de la Figura 3.8.

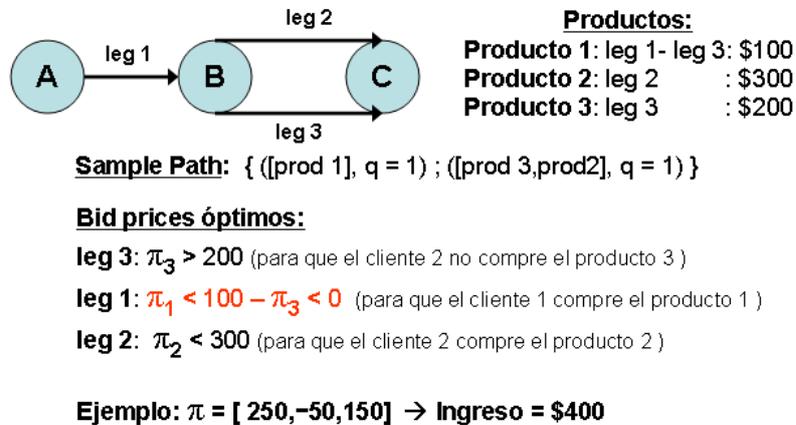


Figura 3.8: Ejemplo de optimalidad de bid prices negativos

El ejemplo es sencillo, pero ilustrativo. Se trata de una red con 3 vuelos, uno de los cuáles tiene un parada, 3 productos y 2 clientes. La idea es tratar de que ambos clientes compren un pasaje, pero forzar al segundo cliente a que compre en el vuelo más costoso. En caso de lograrlo, es claro que el ingreso obtenido será óptimo, ya que se venderá el pasaje más costoso posible a ambos clientes. Para llevarlo a cabo, como se puede apreciar en la descripción de los valores de los bid prices, el bid price asociado al leg 1 tiene que ser necesariamente negativo, para permitir la venta de un pasaje con parada al primer cliente, y a la vez evitar que el segundo cliente compre un pasaje en el vuelo económico. De esta manera, tenemos un ejemplo de optimalidad de bid prices negativos que, si bien es meramente ilustrativo, representa una situación factible de ventas en la realidad. De hecho, para más de un ejemplo, el método de gradientes estocásticos arroja coordenadas negativas para  $\pi$ .

## *Sobre la convergencia*

Uno de los aspectos más importantes de la metodología, sobre el que hablamos cuando la introducimos en la sección 1.2.3, y que hemos eludido hasta el momento, es respecto de la convergencia teórica, y práctica, del método. En dicha sección, página 21, enunciamos una serie de condiciones para asegurarla. La demostración formal de convergencia está basada en los argumentos utilizados en la prueba de convergencia en Topaloglu[Top08], y es una adaptación de la prueba similar hecha en van Ryzin y Vulcano[vRV08a].

## Capítulo 4

# Experimentos numéricos

En este capítulo mostraremos la descripción y los resultados de los experimentos que llevamos a cabo para evaluar el comportamiento del método de gradientes estocásticos. Los experimentos están divididos en 3 clases, de acuerdo a su origen. Dos corresponden a ejemplos extraídos de la literatura, y el tercer tipo fue diseñado específicamente para este trabajo. En todos los casos la metodología es la misma: comenzaremos definiendo el modelo de demanda del ejemplo, siempre basado en un modelo de elección, y en el que los tipos de clientes siguen procesos de Poisson independientes entre sí. Para simularlos, dividiremos el horizonte de compra en una cantidad grande  $T$  de períodos chicos (como para que sea razonable asumir que la probabilidad de más de un arribo por período sea despreciable) y calcularemos en cada período la probabilidad  $\lambda_t$  (extraída de los parámetros del ejemplo) de que haya un arribo de un cliente de determinado tipo. La generación de las listas de preferencia será diferente en cada serie de experimentos, de manera que se adapte a cada tipo de modelo, según ya hemos descrito en la sección 2.3, y vamos a suponer que en todos los casos los clientes solicitan una unidad de producto (es decir,  $q_t = 1$  en todos los casos).

Una vez definida la adaptación del modelo de demanda, implementaremos el generador de sample paths aleatorios que le corresponda, que se utilizará para generar una lista de sample paths, tanto para la optimización como para la estimación de la performance del método. Luego, resolveremos el problema con DLP<sup>1</sup>, y obtendremos el set de bid prices iniciales para el método de gradientes estocásticos. Se correrá a continuación la optimización, que dará como resultado un nuevo set de bid prices y finalmente se calcularán estadísticas del comportamiento del algoritmo.

Específicamente vamos a calcular dos indicadores. Por un lado, la media del ingreso para tres conjuntos de bid prices: el generado por DLP, el optimizado, y  $\pi = 0$ ; este último representa la política de vender en el orden de llegada, es decir no utilizar *Revenue Management*, y lo denotaremos FCFS (*First-Come First-Served*). En todos los casos, el procesamiento que se usa para calcular el ingreso generado por un set de bid prices, para un sample path dado, es el procesamiento *discreto* (es decir, sólo se venden cantidades enteras de productos). Además de las medias de las ganancias,

---

<sup>1</sup>Para calcular la media de la demanda de un producto  $j$ , se recorren todos los sample paths de la lista y se promedia la cantidad de veces que aparece  $j$  como primer elemento -esto es equivalente a asumir que la demanda para cada producto es independiente-

mostraremos la diferencia porcentual entre éstas, junto con un intervalo de confianza del 95 %.

En la segunda parte del capítulo, tomaremos algunos de los ejemplos previos y realizaremos experimentos de *reoptimización*, que consistirán en ir repitiendo la optimización a lo largo del horizonte de compra, para obtener diferentes sets de bid prices, que se puedan adaptar a la variabilidad de la demanda.

La totalidad de la implementación (tanto el método de gradientes estocásticos, el método DLP, los modelos de demanda, los experimentos, etc) fue realizada en el lenguaje de programación *C++*, utilizando el compilador de *Microsoft Visual Studio 2005 pro*. Los experimentos realizados fueron corridos en una computadora de escritorio Pentium IV de 2.4 GHz con 512 MB de memoria RAM, corriendo bajo Windows XP <sup>2</sup>.

## 4.1. Aplicación directa del método

### 4.1.1. Ejemplo 1: Tipos de cliente fijos

El presente ejemplo fue extraído de van Ryzin y Vulcano[vRV08a](ejemplo 5 en la sección de experimentos numéricos). Se trata de un ejemplo pequeño, que sirve como primera aproximación al tipo de problemas que puede resolver el método que desarrollamos. La Figura 4.1.1 muestra la red del experimento:

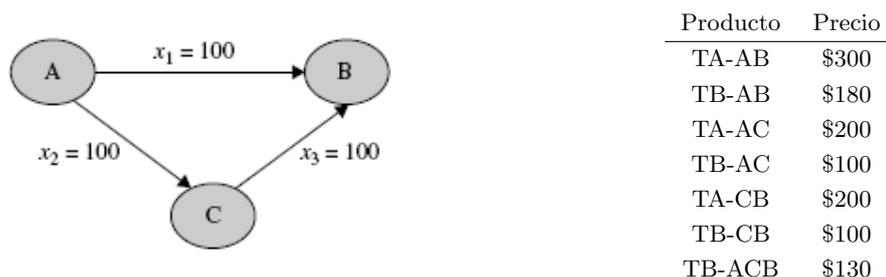


Figura 4.1: La red del experimento 1: Izquierda: el grafo de la red. Derecha: los productos de la red (TA se refiere a *tarifa alta*, y TB a *tarifa baja*)

El modelo de demanda que vamos a utilizar se desprende directamente del que definimos en el capítulo 2. Vamos a suponer que tenemos una serie de tipos de cliente definidos, y que la cantidad de arribos de cada tipo de cliente  $k$  sigue una distribución normal de media y varianza  $\mu_k$ , discretizada y truncada entre 0 y  $2\mu_k$ . Vamos a asumir además que existen 3 períodos en el horizonte de ventas (numerados del 1 al 3, con 1 el más alejado en el tiempo y el 3 el más cercano a la partida de los vuelos), y que los tipos de clientes están clasificados por el período en el que arriban. El cuadro 4.1 muestra la descripción de los tipos de cliente.

La razón por la que la columna de  $\mu_k$  está dividida en 2, es que el experimento se realiza bajo 2 escenarios. El primero corresponde a una situación en la cual el vuelo AB está congestionado, y quisiéramos redireccionar parte de ese tráfico al vuelo con parada en C. Notar que lo que congestiona

<sup>2</sup>Excepto los experimentos de reoptimización que se corrieron utilizando una computadora portátil AMD Turion X2 Ultra Dual-Core de 2.10 Ghz y 3GB de memoria RAM, corriendo bajo Windows Vista Home Premium

Lista de preferencia	Descripción	Período	$\mu_k$	
			esc. 1	esc. 2
[TA-AB]	Sólo directo	3	30	10
[TB-AB, TB-ACB]	Prefiere levemente directo	1	60	10
[TB-ACB, TB-AB]	Sensitivo al precio	1	20	30
[TA-AC]	Tarifa Alta, AC	3	10	30
[TA-CB]	Tarifa Alta, CB	3	10	10
[TB-AC, TA-AC]	AC buy-up	2	10	30
[TB-CB, TA-CB]	CB buy-up	2	10	10
[TB-AC]	AC sensitivo al precio	1	10	60
[TB-CB]	CB sensitivo al precio	1	10	10
[TB-AB, TA-AB]	AB buy-up	2	30	20

Cuadro 4.1: Tipos de cliente del experimento 1

a AB es el hecho de que el primero, segundo y último tipo de cliente quieren comprar un pasaje en AB, y sus demandas medias suman 120 (la capacidad es 100). Sin embargo, el segundo está dispuesto a cambiarse al vuelo con paradas, y es lo que queremos aprovechar. El cuadro 4.2 muestra los resultados de este escenario. DLP se refiere al set de bid prices calculados con DLP y SG al calculado con nuestro método de gradientes estocásticos. LF (*load factor*) representa el porcentaje de ocupación promedio de un vuelo determinado, o el porcentaje de ocupación global (la suma total de las capacidades dividido la suma total de asientos ocupados) en el caso del LF general.

	Método	ingreso medio	LF leg AB	LF leg AC	LF leg CB	LF general
FCFS	[0,0,0]	29896	0.99	0.50	0.50	0.66
DLP	[180,0,0]	32942	0.60	0.99	0.99	0.86
SG	[192.3,-30.5,-31.1]	32942	0.60	0.99	0.99	0.86
		Comparación	Gap	95 % CI		
		SG vs FCFS	10.18 %	(9.49 %,10.87 %)		
		SG vs DLP	0 %	(0 %,0 %)		

Cuadro 4.2: Resultados del escenario 1

El resultado del experimento muestra que DLP representa una mejora sustancial sobre no hacer ningún tipo de control (representado por FCFS), y que nuestro método no logra mejorar a DLP. La mejora de DLP proviene de lo que mencionamos anteriormente: establecer como \$180 el bid price del leg AB hace que se redirija la demanda del segundo tipo de cliente hacia el vuelo con conexión (evidencia de esto es la diferencia entre los load factors de los legs AC y CB con respecto a FCFS) y a la vez se guarden asientos de AB para los clientes de tipo 1 que arriban en el último período. Sin embargo, esto no quiere decir que ésta sea la política óptima; existen más oportunidades de mejora, teniendo en cuenta que la redirección de los clientes de tipo 2 al vuelo con parada produce un aumento considerable de la carga en los legs AC y CB, congestionándolos, y generando la necesidad de realizar controles para no perder oportunidades de ingreso (especialmente teniendo en cuenta que los clientes de tipo 2 están dispuestos a pagar sólo lo mínimo que podemos

obtener por una unidad de leg AC y CB). Este es el comportamiento que, por asumir que las demandas son independientes, no logra capturar DLP, y tampoco logra mejorar nuestro algoritmo, aunque esto cambiará con la incorporación de reoptimizaciones, como veremos más adelante.

El segundo escenario representa una situación en la que el vuelo AB no está congestionando, pero los vuelos AC y CB sí lo están. Los resultados se muestran en el cuadro 4.3.

Método	$\pi$	ingreso medio	LF leg AB	LF leg AC	LF leg CB	LF general
FCFS	[0,0,0]	23295	0.39	1	0.60	0.66
DLP	[0,100,0]	28191	0.39	0.89	0.60	0.63
SG	[-22.4,94.2,112.3]	28781	0.69	0.99	0.20	0.63

Comparación	Gap	95 % CI
SG vs FCFS	23.55 %	(23.21 %,23.88 %)
SG vs DLP	2.09 %	(1.64 %,2.53 %)

Cuadro 4.3: Resultados del escenario 2

En este caso, la mejora de DLP proviene de restringir la compra de la tarifa baja de AC, que tiene una demanda importante en el primer período, y así forzar la compra de la tarifa alta, o dejar lugar para clientes posteriores. Es interesante notar que, pese a la gran diferencia en el ingreso generado, los load factors generales de DLP y FCFS son similares, lo que indica que ambas estrategias venden una cantidad similar de pasajes, pero DLP lo hace más “inteligentemente”. Lo que hace SG es restringir además las compras de tarifa baja de CB por la misma razón que DLP, y porque además, sumado a la restricción en AC, esto también anula el producto ACB, lo que fuerza compras an AB (tercer tipo de cliente). En realidad, para obtener mejores resultados, sería conveniente dejar el valor del bid price de AC en 100 (en tal caso el Gap respecto de DLP asciende al 5.99%), pero no hay que olvidar que la optimización es una heurística y no se garantiza que provea un óptimo local.

Para ambos escenarios, la optimización se realizó con mil iteraciones (la condición de corte fue las iteraciones, en ambos casos), y para el tamaño de paso se tomo  $A = 10$  y  $B = 10$ . Las constantes para la función  $\theta$  fueron  $a = 0.5$ ,  $b = 20$  y  $\gamma = 0$ .

### 4.1.2. Ejemplo 2: MNL con segmentos

Este experimento, o serie de experimentos, fue extraído de Miranda Bront[JMBV], y representa el ejemplo con el que vamos a evaluar la eficacia del método, en el contexto de un modelo MNL. En la sección 2.1 introdujimos el modelo general MNL, e hicimos una descripción de cómo adaptaríamos el modelo de demanda que utilizamos, para que sea equivalente a éste. Los experimentos que vamos a realizar se basan en MNL, pero incorporan el concepto de *segmentos*. La noción de segmentación de un mercado es fundamental en las ciencias económicas y de marketing, y en este caso va a ser modelada con el hecho de que los clientes puedan tener diferentes conjuntos de elección. Cada segmento se asocia a un conjunto determinado de alternativas, que son las que los miembros del segmento consideran. En Miranda Bront et al.[JMBV] se realiza la optimización decidiendo qué productos se van a ofrecer en cada instante, con lo que el conjunto que un cliente  $t$  considera, es en realidad la intersección entre el conjunto de consideración del segmento al que pertenece y el conjunto de productos ofrecidos en ese momento. En nuestro caso, como ya hemos explicado, la política de bid prices hace que el conjunto ofrecido sea constante.

Vamos a suponer que tenemos  $L$  segmentos, numerados de 1 a  $L$ , y que, por cada arribo de cliente, la probabilidad de que sea del segmento  $l$  es  $p_l$ , con  $\sum_{l=1}^L p_l = 1$ , lo que significa que los arribos del segmento  $l$  siguen un proceso de Poisson de parámetro  $\lambda p_l$ . Para decidir si hay un arribo, y de qué segmento, se va simulando en cada período una variable aleatoria discreta con rango  $\{1, \dots, L, L + 1\}$  y probabilidades  $\{\lambda_1, \dots, \lambda_L, \lambda_{L+1}\}$ , donde  $\lambda_l = \frac{\lambda}{T} p_l$  para  $l = 1 \dots L$  y  $\lambda_{L+1} = 1 - \sum_{l=1}^L \lambda_l$ . Si en efecto hay un arribo (es decir, la realización de la v.a. está entre 1 y  $L$ ), se toman los productos del segmento al que pertenece el cliente y se construye la lista de preferencia, según ya lo describimos en la sección 2.1.

Los experimentos de esta sección están divididos en 3 instancias, de complejidad creciente. La primera corresponde a un grupo de 3 vuelos paralelos (el primero a la mañana, el segundo a la tarde y el tercero a la noche) entre dos terminales, con dos productos por vuelo. A esta instancia la llamaremos *Vuelos Paralelos*, y es ilustrada en la Figura 4.1.2.

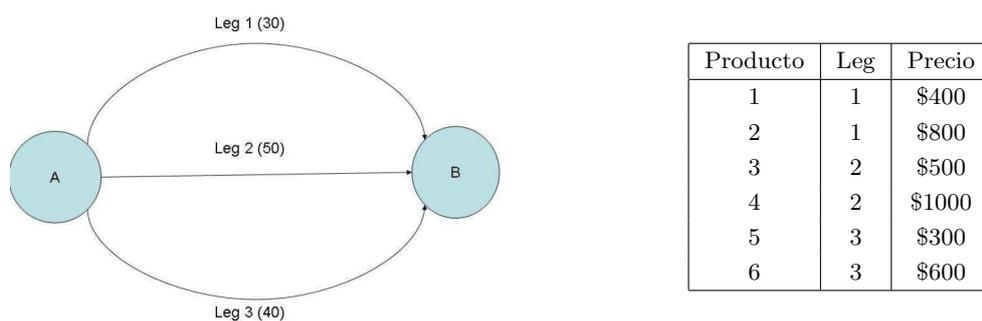


Figura 4.2: La red de Vuelos Paralelos del experimento 2: Izquierda: el grafo de la red. Derecha: los productos de la red

Los parámetros del modelo de demanda de este ejemplo se muestran en el cuadro 4.4. Existen 4 segmentos, cada uno con su conjunto de elección  $C_l$ , el conjunto de pesos para cada producto  $(v_l)_j = e^{\mu_j^l}$ , la probabilidad  $\lambda_l$  de arribo en un período, y la descripción. Vamos a suponer que se

consideran  $T = 300$  períodos en el horizonte de reserva, lo que da una media de 150 arribos totales.

Segmento	$C_l$	$v_l$	$\lambda_l$	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 mañana.

Cuadro 4.4: Definición de segmentos para la instancia de Vuelos Paralelos.

Si prestamos atención al cuadro 4.4 vamos a notar que no está presente la información de la probabilidad de no compra de los segmentos; en los vectores  $v_l$  sólo Figuran los pesos de los productos de  $C_l$ . Esto se debe a que se realizarán diferentes experimentos variando los valores de  $v_0$  para cada segmento (siguiendo la metodología de Miranda Bront[JMBV]). El segundo parámetro que se variará en los experimentos es la capacidad de los vuelos: se definirá  $\alpha \in [0, 1]$ , y en cada experimento las capacidades reales que se tomarán son:  $\alpha 30$ ,  $\alpha 50$  y  $\alpha 40$ , respectivamente. Lo que se quiere lograr con esto, es ver cómo varía el comportamiento del algoritmo al cambiar la carga de la red (a menor capacidad, con demanda constante, se aumenta la carga, y viceversa). Vamos a tomar 3 conjuntos de valores de  $v_0$ , y 5 valores posibles para  $\alpha$ . El cuadro 4.5 muestra los resultados.

En términos generales, los resultados muestran que, para los casos en que la red tiene una carga alta (los primeros), DLP se comporta pobremente (incluso peor que FCFS), y que SG mejora en gran medida este comportamiento. A medida que la carga de la red disminuye, la oportunidad de hacer *Revenue Management* se reduce, y a partir del sexto experimento, SG deja de mejorar los ingresos generados por DLP (que también disminuyen, en relación a FCFS, indicando que el exceso de capacidad hace que aceptar en orden de llegada sea (casi) óptimo). En realidad, si observamos la tabla inferior del cuadro 4.5, vemos que la razón para la pobre performance de DLP en las primeras 6 experiencias es que realiza un control excesivo sobre el primer leg y esto hace que no haya ninguna compra de productos 1 y 2. En los experimentos posteriores esta restricción se reduce y como consecuencia mejora la performance de dicho algoritmo.

Para tener una idea de qué tan buenos son los resultados de esta instancia, podemos ver el cuadro 4.6, en el que se comparan los ingresos generados por SG y los reportados en Miranda[JMBV] para el método CDLP (estudiado por Gallego et al.[GIPD04] y van Ryzin y Liu[LvR08]), con el que se han obtenido buenos resultados experimentales, y resulta un buen punto de comparación. Las dos primeras filas de la tabla tienen los valores de ingreso, y la tercera la diferencia porcentual entre ambos. Como podemos ver, SG se comporta mejor para los primeros casos, en los que hay mayor carga de red, y comparablemente a DLP para los últimos.

Para terminar, respecto de los parámetros de la optimización, cabe aclarar que tomamos  $A = B = 10$  para el tamaño de paso,  $a = 0.5$ ,  $b = 20$  y  $\gamma = 0$  para  $\theta$  y 3000 iteraciones. En todos los casos el algoritmo finalizó por iteraciones, y los tiempos oscilaron entre los 5 y 15 segundos.

exp	$\alpha$	$v_0$	DLP	SG	FCFS	SG vs DLP		SG vs FCFS	
1	0.6	(1 5 5 1)	\$ 43265	\$ 55057	\$ 43766	27.24 %	(27.1%;27.3 %)	25.63 %	(25.3%;25.8 %)
2		(1 10 5 1)	\$ 43312	\$ 55158	\$ 44381	27.35 %	(27.2%;27.4 %)	24.15 %	(23.9%;24.3 %)
3		(5 20 10 5)	\$ 38908	\$ 52226	\$ 44262	34.27 %	(34.1%;34.4 %)	18.10 %	(17.9%;18.2 %)
4	0.8	(1 5 5 1)	\$ 51513	\$ 69517	\$ 58295	35.09 %	(34.9%;35.2 %)	19.31 %	(19.1%;19.4 %)
5		(1 10 5 1)	\$ 51841	\$ 69565	\$ 59057	34.12 %	(33.9%;34.2 %)	17.75 %	(17.5%;17.9 %)
6		(5 20 10 5)	\$ 59607	\$ 59607	\$ 55823	0 %	(0%;0 %)	6.72 %	(6.60%;6.82 %)
7	1	(1 5 5 1)	\$ 75981	\$ 75981	\$ 71557	0 %	(0%;0 %)	6.23 %	(6.13%;6.32 %)
8		(1 10 5 1)	\$ 74952	\$ 74952	\$ 70774	0 %	(0%;0 %)	5.87 %	(5.77%;5.96 %)
9		(5 20 10 5)	\$ 62850	\$ 62850	\$ 60022	0 %	(0%;0 %)	4.76 %	(4.62%;4.89 %)
10	1.2	(1 5 5 1)	\$ 79440	\$ 79440	\$ 76156	0 %	(0%;0 %)	4.35 %	(4.25%;4.44 %)
11		(1 10 5 1)	\$ 76823	\$ 76823	\$ 73699	0 %	(0%;0 %)	4.17 %	(4.06%;4.27 %)
12		(5 20 10 5)	\$ 63456	\$ 63456	\$ 61509	0 %	(0%;0 %)	3.04 %	(2.90%;3.16 %)
13	1.4	(1 5 5 1)	\$ 79530	\$ 79530	\$ 76669	0 %	(0%;0 %)	3.71 %	(3.61%;3.81 %)
14		(1 10 5 1)	\$ 77326	\$ 77326	\$ 74635	0 %	(0%;0 %)	3.55 %	(3.44%;3.65 %)
15		(5 20 10 5)	\$ 63132	\$ 63132	\$ 62240	0 %	(0%;0 %)	1.45 %	(1.32%;1.56 %)

exp	DLP LF				SG LF				FCFS LF			
	leg			global	leg			global	leg			global
1	0	1	0.92	0.72	1	0.99	0.74	0.91	1	1	1	1
2	0	1	0.92	0.72	1	0.99	0.74	0.91	1	1	1	1
3	0	0.99	0.99	0.74	0.99	0.97	0.99	0.98	1	0.99	0.98	0.99
4	0	0.99	1	0.74	0.99	0.98	1	0.99	1	0.99	0.99	0.99
5	0	0.99	1	0.74	0.99	0.97	0.99	0.98	1	0.99	0.99	0.99
6	0.97	0.95	0.82	0.91	0.97	0.95	0.82	0.91	1	0.96	0.81	0.92
7	0.97	0.96	0.97	0.97	0.97	0.96	0.97	0.97	1	0.98	0.95	0.97
8	0.97	0.95	0.89	0.93	0.97	0.95	0.89	0.93	1	0.97	0.87	0.94
9	0.87	0.8	0.64	0.76	0.87	0.8	0.64	0.76	0.99	0.81	0.6	0.78
10	0.88	0.82	0.85	0.84	0.88	0.82	0.85	0.84	1	0.85	0.77	0.86
11	0.87	0.8	0.72	0.79	0.87	0.8	0.72	0.79	0.99	0.82	0.66	0.81
12	0.75	0.66	0.53	0.64	0.75	0.66	0.53	0.64	0.99	0.64	0.48	0.67
13	0.77	0.69	0.73	0.72	0.77	0.69	0.73	0.72	0.99	0.68	0.63	0.74
14	0.77	0.68	0.61	0.68	0.77	0.68	0.61	0.68	0.99	0.66	0.54	0.71
15	0.65	0.56	0.45	0.54	0.65	0.56	0.45	0.54	0.97	0.51	0.39	0.59

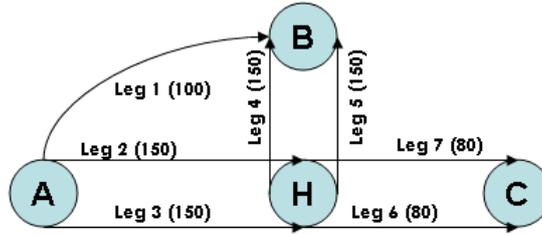
Cuadro 4.5: Resultados de la instancia de Vuelos Paralelos. Arriba, ingresos y mejoras porcentuales del método respecto de DLP y FCFS. Abajo, Load factors globales y por leg.

	exp 1	exp 2	exp 3	exp 4	exp 5	exp 6	exp 7	exp 8	exp 9	exp 10	exp 11	exp 12	exp 13	exp 14	exp 15
CDLP	54,177	54,051	50,058	68,105	67,896	59,073	75,726	73,788	62,702	79,666	77,348	62,491	80,362	78,091	62,553
SG	55057	55158	52226	69517	69565	59607	75981	74952	62850	79440	76823	63456	79530	77326	63132
SG vs CDLP	1.62 %	2.05 %	4.33 %	2.07 %	2.46 %	0.90 %	0.34 %	1.58 %	0.24 %	-0.28 %	-0.68 %	1.54 %	-1.04 %	-0.98 %	0.93 %

Cuadro 4.6: resultados comparativos de la instancia Vuelos Paralelos respecto de CDLP

La segunda instancia de este experimento se realiza sobre un red pequeña, con 4 terminales, 7 vuelos y 22 productos, y la llamaremos la instancia de *Red Pequeña*. La red y los productos se describen en la Figura 4.3. En este caso, como en la instancia anterior, hay 2 productos por cada vuelo directo, pero además se agregan 8 productos para vuelos con una parada, incrementando los efectos de red en el proceso de ventas.

Los segmentos de clientes que se consideran se encuentran descriptos en el cuadro 4.7. Para todos los experimentos se toma un horizonte con  $T = 1000$  periodos de tiempo, con un promedio de



Producto	Leg(s)	Clase	Tarifa	Producto	Leg(s)	Clase	Tarifa
1	1	A	1000	12	1	B	500
2	2	A	400	13	2	B	200
3	3	A	400	14	3	B	200
4	4	A	300	15	4	B	150
5	5	A	300	16	5	B	150
6	6	A	500	17	6	B	250
7	7	A	500	18	7	B	250
8	{2, 4}	A	600	19	{2, 4}	B	300
9	{3, 5}	A	600	20	{3, 5}	B	300
10	{2, 6}	A	700	21	{2, 6}	B	350
11	{3, 7}	A	700	22	{3, 7}	B	350

Figura 4.3: Definición de la red y los productos para la instancia Red Pequeña.

910 arribos de clientes. Los valores del vector  $v_0$  para cada instancia están descriptos resumidamente por un par ordenado de números. Por ejemplo, si se indica  $v_0 = (1, 5)$  para un experimento, se debe interpretar como que los segmentos de índice impar (1, 3, 5, 7 y 9) tienen  $v_0 = 1$ , y los de índice par (2, 4, 6, 8 y 10) tienen  $v_0 = 5$ .

Segmento	O-D	$C_l$	$v_l$	$\lambda_l$	Descripción
1	A → B	{1,8,9,12,19,20}	(10,8,8,6,4,4)	0.08	Insensibles al precio, pref. por la mañana (leg 1)
2	A → B	{1,8,9,12,19,20}	(1,2,2,8,10,10)	0.2	Sensibles al precio.
3	A → H	{2,3,13,14}	(10,10,5,5)	0.05	Insensibles al precio.
4	A → H	{2,3,13,14}	(2,2,10,10)	0.2	Sensibles al precio.
5	H → B	{4,5,15,16}	(10,10,5,5)	0.1	Insensibles al precio.
6	H → B	{4,5,15,16}	(2,2,10,8)	0.15	Sensibles al precio, leve pref. por la mañana(leg4)
7	H → C	{6,7,17,18}	(10,8,5,5)	0.02	Insensibles al precio, leve pref. por la mañana(leg6)
8	H → C	{6,7,17,18}	(2,2,10,8)	0.05	Sensibles al precio.
9	A → C	{10,11,21,22}	(10,8,5,5)	0.02	Insensibles al precio, leve pref. por la mañana(legs 2 y 6)
10	A → C	{10,11,21,22}	(2,2,10,10)	0.04	Sensibles al precio.

Cuadro 4.7: Definición de segmentos para la instancia Red Pequeña.

Los parámetros del método que se tomaron para los experimentos de esta instancia fueron  $a = 0.5$ ,  $b = 20$  y  $\gamma = 0$  para  $\theta$ ,  $A = B = 10$  para el tamaño de paso y 2000 iteraciones (en todos los casos, el corte fue por iteraciones, y los tiempos de procesamiento de SG variaron entre 50 y 170 segundos). Los resultados se encuentran descriptos en el cuadro 4.8.

El resultado de las experiencias muestra que, en la mayoría de los casos, SG logra mejorar significativamente el valor de DLP (recordemos que en un contexto real de venta de pasajes, mejoras del 2% o 3% se consideran significativas). Como era de esperarse, las oportunidades de mejora basadas

exp	$\alpha$	$v_0$	DLP	SG	FCFS	SG vs DLP		SG vs FCFS	
1	0.6	(1 5)	207367	212918	145343	2.68 %	(2.57 %;2.77 %)	46.49 %	(46.30 %;46.6 %)
2		(5 10)	189234	190307	142844	0.57 %	(0.53 %;0.60 %)	33.23 %	(33.02 %;33.4 %)
3		(10 20)	163070	163070	139954	0 %	(0 %;0 %)	16.52 %	(16.35 %;16.6 %)
4	0.8	(1 5)	235031	244754	186625	4.14 %	(4.09 %;4.17 %)	31.15 %	(31.05 %;31.2 %)
5		(5 10)	210432	216441	183531	2.86 %	(2.77 %;2.93 %)	17.93 %	(17.7 %;18.0 %)
6		(10 20)	177569	186426	174933	4.99 %	(4.91 %;5.06 %)	6.57 %	(6.45 %;6.68 %)
7	1	(1 5)	264027	255654	227213	-3.17 %	(-3.26 %;-3.07 %)	12.52 %	(12.4 %;12.6 %)
8		(5 10)	228158	233049	216791	2.14 %	(2.06 %;2.21 %)	7.50 %	(7.38 %;7.61 %)
9		(10 20)	183385	191430	191430	4.39 %	(4.31 %;4.46 %)	0 %	(0 %;0 %)
10	1.2	(1 5)	279875	269279	254665	-3.79 %	(-3.84 %;-3.72 %)	5.74 %	(5.64 %;5.83 %)
11		(5 10)	226361	240671	230565	6.32 %	(6.23 %;6.40 %)	4.38 %	(4.29 %;4.47 %)
12		(10 20)	194808	194808	194808	0 %	(0 %;0 %)	0 %	(0 %;0 %)
13	1.4	(1 5)	267531	287054	265065	7.30 %	(7.21 %;7.37 %)	8.30 %	(8.21 %;8.37 %)
14		(5 10)	234950	234950	234950	0 %	(0 %;0 %)	0 %	(0 %;0 %)
15		(10 20)	194595	194595	194595	0 %	(0 %;0 %)	0 %	(0 %;0 %)

exp	DLP LF								SG LF								FCFS LF
	leg LF				global LF				leg LF				global LF				global LF
1	1.00	1.00	1.00	1.00	1.00	0.89	0.80	0.97	1.00	0.96	0.93	0.90	0.93	0.86	0.92	0.97	
2	0.78	1.00	1.00	1.00	0.99	0.79	0.89	0.94	0.77	1.00	1.00	1.00	0.81	0.72	0.93	0.96	
3	1.00	0.93	0.89	0.85	0.86	0.85	0.73	0.88	1.00	0.93	0.89	0.85	0.86	0.85	0.73	0.88	
4	1.00	1.00	1.00	0.99	0.99	0.85	0.74	0.96	1.00	1.00	1.00	0.99	0.99	0.69	0.61	0.93	
5	1.00	0.97	0.81	0.85	1.00	0.79	0.69	0.89	1.00	0.87	0.95	0.86	0.85	0.61	0.78	0.86	
6	0.98	0.69	0.64	0.90	0.85	0.65	0.55	0.76	0.98	0.98	0.96	0.89	0.84	0.64	0.55	0.87	
7	1.00	0.95	0.93	1.00	1.00	0.74	0.64	0.92	1.00	0.97	1.00	0.91	1.00	0.48	0.68	0.90	
8	1.00	0.75	0.70	0.91	0.85	0.65	0.54	0.79	1.00	0.99	0.98	0.89	0.84	0.64	0.54	0.87	
9	0.60	0.79	0.76	0.95	0.92	0.52	0.44	0.76	0.62	0.99	0.99	0.93	0.90	0.51	0.43	0.83	
10	1.00	0.81	0.76	0.90	0.84	0.63	0.53	0.80	0.98	1.00	0.77	1.00	0.65	0.52	0.63	0.82	
11	0.59	0.83	0.80	0.96	0.93	0.54	0.45	0.78	0.93	0.85	0.81	0.89	0.45	0.54	0.45	0.72	
12	0.50	0.88	0.85	0.81	0.76	0.44	0.37	0.71	0.50	0.88	0.85	0.81	0.76	0.44	0.37	0.71	
13	0.56	0.88	0.84	0.95	0.91	0.54	0.45	0.78	0.97	0.51	1.00	0.60	0.57	0.31	0.61	0.66	
14	0.50	0.93	0.89	0.84	0.79	0.46	0.39	0.74	0.50	0.93	0.89	0.84	0.79	0.46	0.39	0.74	
15	0.42	0.76	0.73	0.69	0.66	0.37	0.31	0.61	0.42	0.76	0.73	0.69	0.66	0.37	0.31	0.61	

Cuadro 4.8: Resultados de la instancia Red Pequeña: arriba, resultados comparativos respecto del ingreso generado; abajo, load factor a nivel de leg y global (por razones de presentación, se omiten los load factors a nivel de leg para  $\pi = 0$ )

en *Revenue Management* (ya sea a través de DLP o SG) son mucho mayores para los ejemplos en los que la carga de la red es alta. Por ejemplo, en las 3 primeras experiencias, la diferencia entre SG y FCFS va decreciendo sustancialmente; esto se debe a que los pesos de la no compra  $v_0$  se van incrementando, reduciendo la frecuencia de las compras en los arribos. Por otra parte, en la tabla de load factors, tenemos otra evidencia de las oportunidades que existen al utilizar *Revenue Management*, y en particular teniendo en cuenta el comportamiento de elección: en muchos casos se observa una mejora en el ingreso, junto con una reducción en el load factor entre un método y otro, lo que indica que se obtienen mayores ingresos, pese a vender una menor cantidad de asientos. En un contexto en el que tenemos una alta incertidumbre, y más aún, tenemos clientes que prefieren comprar barato, pero consideran la posibilidad de comprar opciones más costosas, en muchas ocasiones tiene sentido realizar controles más agresivos sobre la capacidad, con la esperanza de forzar a los clientes a comprar alternativas más costosas, y asumiendo el riesgo de que decidan no comprar

nada. Resolver el problema es justamente decidir en qué medida conviene tomar este riesgo, según los parámetros particulares del problema, para maximizar el ingreso a largo plazo.

Para terminar con las estadísticas de esta instancia, el cuadro 4.9 muestra la tabla comparativa de ingresos, con respecto a los generados por la implementación de CDLP de Miranda Bront[JMBV]. Podemos ver que, para esta instancia, SG se comporta mejor que CDLP en el 60 % de los casos,

exp	CDLP	SG	% SG vs CDLP	exp	CDLP	SG	% SG vs CDLP
1	207,890	212918	2.42 %	9	190,283	191430	0.60 %
2	194,393	190307	-2.10 %	10	282,842	269279	-4.80 %
3	164,089	163070	-0.62 %	11	238,299	240671	1.00 %
4	261,264	244754	-6.32 %	12	192,511	194808	1.19 %
5	215,884	216441	0.26 %	13	285,417	287054	0.57 %
6	184,182	186426	1.22 %	14	238,843	234950	-1.63 %
7	277,738	255654	-7.95 %	15	192,541	194595	1.07 %
8	230,342	233049	1.18 %				

Cuadro 4.9: Tabla comparativa de la instancia Red Pequeña, respecto de CDLP.

pero que, sin embargo, en los casos en los que CDLP supera a SG, la diferencia entre ambos llega a ser pronunciada.

La tercera y última instancia del experimento 2 es a la que llamaremos *Hub and Spoke*. Se trata de un tipo clásico de red en la que una terminal, a la que suele conocerse como *hub* o *centro de distribución*, es un paso intermedio para los itinerarios con conexión con otras terminales. Es común que en la literatura de *Revenue Management* se trabaje con redes de este tipo, debido a que muchas aerolíneas implementan este esquema (por ejemplo, Delta Airlines en Atlanta, American Airlines en Dallas Fort Worth, entre otros). La red que vamos a utilizar, ilustrada en la Figura 4.4, tiene 5 terminales, cada una correspondiente a un aeropuerto real en Estados Unidos: Los Ángeles (LAX), Atlanta (ATL), Boston (BOS), Savannah (SAV) y Miami (MIA) (ATL sería el *hub* en este caso). Los productos se describen en el cuadro 4.10: se consideran los 40 posibles itinerarios (20 ida y vuelta) entre los 5 aeropuertos, y cada itinerario tiene 4 clases de tarifa: Y, M, B, Q, haciendo un total de 80 productos, numerados del 1 al 80, con ATLBOS-Y el primero, BOSATL-Y el quinto y SAVMIA-Q el último.

Los segmentos que se definen, detallados en el cuadro 4.11, corresponden a 2 tipos por cada itinerario de la red. Los segmentos H (*High*) prefieren las tarifas económicas, pero consideran a todas, y los segmentos L (*Low*) sólo consideran las tarifas bajas, B y Q. Esto da un total de 40 segmentos.

Para generar los sample paths, se tomaron  $T = 2000$  períodos, con un promedio de 1732 arribos por sample path. Los parámetros del método que se utilizaron para los experimentos de esta instancia fueron  $a = 0.5$ ,  $b = 20$  y  $\gamma = 0$  para  $\theta$ ,  $A = B = 10$  para el tamaño de paso y 3000 iteraciones. En todos los casos el algoritmo terminó por iteraciones, y los tiempos oscilaron entre los 216 y 560 segundos. Los resultados se encuentran descriptos en el cuadro 4.12.

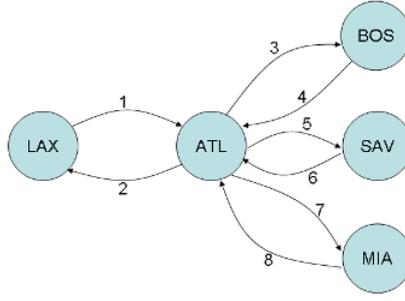


Figura 4.4: Red de la instancia Hub and Spoke (todas las capacidades son de 180 asientos)

Origen - Destino	Legs	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 4.10: Definición de productos para la red *Hub and Spoke*.

Segmento	$C_i$	$v_i$	$\lambda_i$	Segmento	$C_i$	$v_i$	$\lambda_i$
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 4.11: Definición de segmentos para la red *Hub and Spoke*.

exp	$\alpha$	$v_0$	ingreso DLP	ingreso SG	ingreso FCFS	SG vs DLP		SG vs FCFS	
1	0.6	(1 5)	126326	141441	70920.4	11.96 %	(11.8 %;12.1 %)	99.44 %	(99.1 %;99.7 %)
2		(5 10)	112558	112978	71592.4	0.37 %	(0.24 %;0.50 %)	57.81 %	(57.5 %;58.0 %)
3		(10 20)	96553	97570.2	73276.6	1.05 %	(0.92 %;1.18 %)	33.15 %	(32.9 %;33.3 %)
4	0.8	(1 5)	144112	149443	94430.3	3.70 %	(3.58 %;3.81 %)	58.26 %	(58.0 %;58.4 %)
5		(5 10)	124718	128636	95393.8	3.14 %	(3.03 %;3.24 %)	34.85 %	(34.6 %;35.0 %)
6		(10 20)	109146	111544	97022.6	2.20 %	(2.13 %;2.26 %)	14.97 %	(14.8 %;15.0 %)
7	1	(1 5)	154625	157980	117878	2.17 %	(2.06 %;2.27 %)	34.02 %	(33.8 %;34.1 %)
8		(5 10)	136888	138535	118606	1.20 %	(1.16 %;1.24 %)	16.80 %	(16.6 %;16.9 %)
9		(10 20)	121926	118253	118253	-3.01 %	(-3.09 %;-2.93 %)	0 %	(0 %;0 %)
10	1.2	(1 5)	165867	168229	140886	1.42 %	(1.3 %;1.49 %)	19.41 %	(19.2 %;19.5 %)
11		(5 10)	152375	145326	140292	-4.63 %	(-4.67 %;-4.57 %)	3.59 %	(3.54 %;3.62 %)
12		(10 20)	128255	128255	128255	0 %	(0 %;0 %)	0 %	(0 %;0 %)
13	1.4	(1 5)	178204	178754	162205	0.31 %	(0.29 %;0.31 %)	10.20 %	(10.1 %;10.2 %)
14		(5 10)	156884	156264	156264	-0.39 %	(-0.42 %;-0.36 %)	0 %	(0 %;0 %)
15		(10 20)	128354	128354	128354	0 %	(0 %;0 %)	0 %	(0 %;0 %)

exp	DLP LF									SG LF								FCFS LF	
	leg LF					global LF				leg LF				global LF				global LF	
1	1.00	1.00	0.98	0.98	0.99	0.99	1.00	1.00	0.99	0.80	0.80	0.74	0.74	0.77	0.77	0.68	0.68	0.75	1.00
2	0.99	0.99	0.98	0.98	0.91	0.91	1.00	1.00	0.97	0.62	0.62	0.87	0.87	0.84	0.59	0.86	0.86	0.77	1.00
3	0.99	0.99	1.00	1.00	0.95	0.95	0.90	0.90	0.96	0.88	0.75	0.82	0.65	0.63	0.76	0.78	0.94	0.77	1.00
4	0.98	0.98	1.00	1.00	0.79	0.79	1.00	1.00	0.94	0.61	0.60	0.85	0.85	0.59	0.84	0.84	0.83	0.75	1.00
5	0.84	0.97	1.00	1.00	0.82	1.00	0.98	0.90	0.94	0.76	0.76	0.85	0.85	0.64	0.73	0.93	0.93	0.81	1.00
6	0.71	0.82	0.95	0.94	0.98	0.97	0.91	0.91	0.90	0.94	0.94	0.93	0.93	0.97	0.97	0.99	0.99	0.96	0.99
7	0.83	0.95	1.00	1.00	0.86	1.00	0.98	0.90	0.94	0.46	0.47	0.88	0.85	0.73	0.88	1.00	0.93	0.78	1.00
8	0.76	0.84	0.97	0.97	1.00	1.00	0.93	0.93	0.92	0.84	0.84	0.92	0.92	1.00	1.00	0.88	0.88	0.91	1.00
9	0.90	0.90	0.96	0.96	0.94	0.94	0.90	0.91	0.93	1.00	1.00	0.98	0.98	0.96	0.96	0.93	0.93	0.97	0.97
10	0.80	0.85	0.98	0.98	1.00	1.00	0.94	0.94	0.94	0.38	0.71	0.97	0.96	0.97	0.93	0.98	0.91	0.85	1.00
11	0.98	0.97	0.97	0.97	0.88	0.94	0.96	0.96	0.95	1.00	1.00	0.98	0.96	0.96	0.99	0.97	0.97	0.98	0.98
12	0.93	0.93	0.88	0.88	0.85	0.85	0.82	0.82	0.87	0.93	0.93	0.88	0.88	0.85	0.85	0.82	0.82	0.87	0.87
13	0.99	0.99	0.99	0.99	0.93	0.96	0.97	0.97	0.97	0.99	0.99	0.99	0.99	0.94	0.96	0.97	0.95	0.97	0.98
14	0.95	0.95	0.95	0.96	0.92	0.92	0.84	0.84	0.92	0.98	0.98	0.95	0.95	0.92	0.92	0.88	0.88	0.93	0.93
15	0.80	0.80	0.76	0.76	0.73	0.73	0.70	0.70	0.75	0.80	0.80	0.76	0.76	0.73	0.73	0.70	0.70	0.75	0.75

Cuadro 4.12: Resultados de la instancia *Hub and Spoke*: arriba, resultados comparativos respecto del ingreso generado; abajo, load factor a nivel de leg y global (por razones de presentación, se omiten los load factors a nivel de leg para  $\pi = 0$ )

El comportamiento del método, en relación a DLP, es similar a la instancia anterior. Nuevamente, los mejores resultados se obtienen cuando hay mayor carga, y oportunidad de aprovecharse de la naturaleza selectiva de los clientes. Es interesante notar además que los controles aplicados por SG son en general mucho más agresivos que los aplicados por DLP; evidencia de ello es el hecho de que los load factors (generales y a nivel de leg) producidos por el primero son sustancialmente menores que los producidos por el segundo. Esto se debe a que los bid prices generados por SG aprovechan el comportamiento selectivo de los clientes y tratan de restringirles las compras a los segmentos “L”, y forzar a los clientes de los segmentos “H” a que compren los productos caros, lo que produce una merma en la cantidad total de pasajes vendidos, pero un incremento en el ingreso, ya que se trata de ventas más rentables (observar, en el cuadro 4.10 de productos de la instancia, que la diferencia entre el precio de las clases altas y las bajas es muy significativa).

Sin embargo, pese a mejorar la performance de DLP en la mayoría de los casos, en esta instancia SG da peores resultados que CDLP, como lo muestra el cuadro 4.13.

<b>exp</b>	<b>CDLP</b>	<b>SG</b>	<b>% SG vs CDLP</b>	<b>exp</b>	<b>CDLP</b>	<b>SG</b>	<b>% SG vs CDLP</b>
<b>1</b>	156,557	141441	-9.66 %	<b>9</b>	125,811	118253	-6.01 %
<b>2</b>	126,425	112978	-10.64 %	<b>10</b>	190,000	168229	-11.46 %
<b>3</b>	106,688	97570.2	-8.55 %	<b>11</b>	157,877	145326	-7.95 %
<b>4</b>	170,301	149443	-12.25 %	<b>12</b>	128,336	128255	-0.06 %
<b>5</b>	140,857	128636	-8.68 %	<b>13</b>	196,639	178754	-9.10 %
<b>6</b>	117,621	111544	-5.17 %	<b>14</b>	160,350	156264	-2.55 %
<b>7</b>	181,673	157980	-13.04 %	<b>15</b>	128,336	128354	0.01 %
<b>8</b>	151,907	138535	-8.80 %				

Cuadro 4.13: Tabla comparativa de la instancia *Hub and Spoke*, respecto de CDLP.

### 4.1.3. Ejemplo 3: Compras con *Buyups*

Para estos experimentos, formulamos un ejemplo pequeño, con una red de 3 terminales, 3 legs y 10 productos repartidos en 4 posibles itinerarios. La red y los productos son ilustrados en la Figura 4.5. El modelo de demanda que utilizaremos asume que los clientes arriban con intenciones de comprar un pasaje en determinado itinerario (la demanda media para cada itinerario se describe en la última columna de la tabla), e intentan obtener la tarifa más económica; pero si ésta no está disponible, con cierta probabilidad intentan comprar en la tarifa inmediata superior, y si a su vez tampoco está disponible, intentan comprar en la siguiente tarifa (en caso de que exista), también con determinada probabilidad. Por ejemplo, supongamos que la probabilidad de buyup entre el producto 3 y 2 es de 0.5, y entre el producto 2 y 1 es de 0.3. Entonces, si arriba un cliente del itinerario A-D y la tarifa baja (producto 3, \$200) está cerrada, con probabilidad 0.5 intentará comprar la inmediata siguiente (producto 2, \$400). Y si ésta a su vez también está cerrada, con probabilidad 0.3 intentará comprar la tarifa alta (producto 1, \$800). Es decir que, suponiendo que los productos 2 y 3 no están disponibles, la probabilidad real de que un cliente que quiere viajar de A a D compre el producto 1 es de  $0.3 \times 0.5 = 0.15$ . Las probabilidades de buyup sólo se consideran entre dos clases adyacentes dentro de un itinerario; vamos a tomar como probabilidad de que haya buyup de una clase a la siguiente al precio de la segunda dividido el precio de la primera. Por ejemplo, la probabilidad de buyup entre el producto 3 y el producto 2 es de  $200/400 = 0.5$

Supondremos que las demandas para cada uno de los itinerarios siguen procesos de Poisson independientes de parámetro  $\mu_k$ , con  $k = 1, 2, 3, 4$  los diferentes itinerarios. Para generar los sample paths dividimos el horizonte en  $T = 1000$  períodos, y simulamos arribos para cada itinerario en cada período con probabilidad  $\lambda_k/T$ .

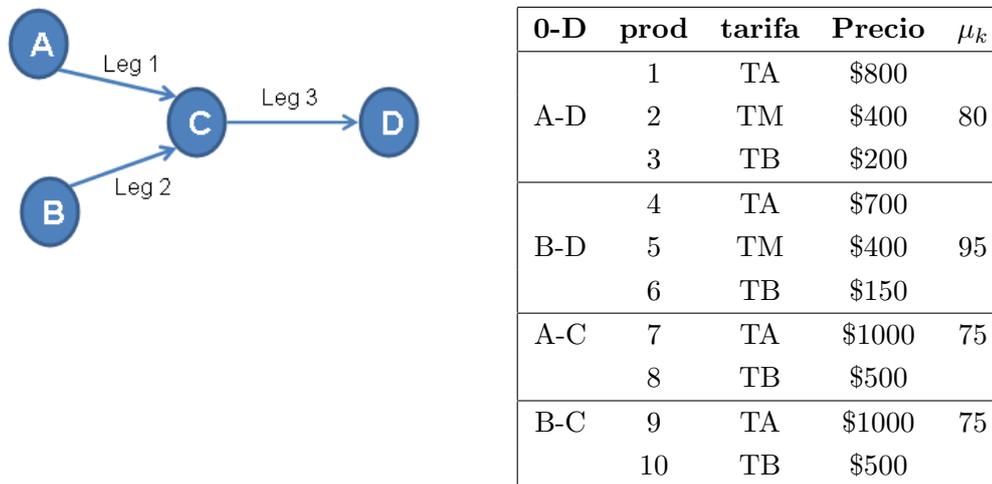


Figura 4.5: Definición de la red y los productos (todas las capacidades son de 100 asientos).

Los experimentos que realizamos con esta instancia tienen como objetivo mostrar cómo varía la mejora producida por el método de gradientes estocásticos, a medida que se va aumentando la presencia de buyups en la demanda, lo que, se supone, da mayor posibilidad para el aprovechamiento

del comportamiento de elección de los clientes. Para llevarlo a cabo vamos a definir una constante  $C$  que va a multiplicar a todas las probabilidades de buyup del ejemplo. Si fijamos  $C = 0$ , por ejemplo, estamos en presencia de un modelo de demanda independiente, ya que las probabilidades de buyup son 0 en todos los casos; si en cambio fijamos  $C = 1$  la configuración de las probabilidades es la inicial, según la describimos más arriba. El otro parámetro que variaremos, similar al de los experimentos de la sección 4.1.2, será el valor  $\alpha$ , que multiplicará a todas las capacidades. Tomaremos  $\alpha \in \{1; 0.8\}$  y  $C \in \{0; 0.2; 0.4; 0.6; 0.8; 1\}$ . Los resultados se describen en los cuadros 4.14 y 4.15.

exp	C	ingreso DLP	ingreso SG	ingreso FCFS	SG vs DLP		SG vs FCFS	
1	0	73610	73610	54027.7	0%	(0%;0%)	36.2449%	(35.9%;36.5%)
2	0.2	76242.6	74544	54047	-2.23%	(-2.35%;-2.09%)	37.9244%	(37.5%;38.2%)
3	0.4	76760.5	77239.1	54028.5	0.62%	(0.43%;0.81%)	42.9601%	(42.6%;43.2%)
4	0.6	76192.1	81042.1	54039.7	6.37%	(6.19%;6.54%)	49.9677%	(49.7%;50.2%)
5	0.8	75392.4	84872.6	54035.9	12.57%	(12.4%;12.7%)	57.0671%	(56.8%;57.2%)
6	1	74650.6	104929	54036.1	40.56%	(40.0%;41.1%)	94.1839%	(93.4%;94.9%)

exp	DLP leg LF			DLP global LF	SG leg LF			SG global LF	FCFS leg LF			FCFS global LF
1	0.92	0.92	0.00	0.61	0.92	0.92	0.00	0.61	1.00	1.00	0.99	1.00
2	0.97	0.97	0.18	0.71	0.93	0.93	0.02	0.62	1.00	1.00	0.99	1.00
3	0.99	0.99	0.33	0.77	0.95	0.95	0.08	0.66	1.00	1.00	0.99	1.00
4	1.00	1.00	0.46	0.82	0.97	0.97	0.17	0.70	1.00	1.00	0.99	1.00
5	1.00	1.00	0.57	0.86	0.99	0.99	0.29	0.76	1.00	1.00	0.99	1.00
6	1.00	1.00	0.67	0.89	0.72	0.72	0.50	0.65	1.00	1.00	0.99	1.00

Cuadro 4.14: Resultados de los experimentos de buyups para  $\alpha = 0.8$ : arriba, resultados comparativos respecto del ingreso generado; abajo, load factor a nivel de leg y global

exp	C	ingreso DLP	ingreso SG	ingreso FCFS	SG vs DLP		SG vs FCFS	
7	0	75004.5	75004.5	67458.1	0%	(0%;0%)	11.1867%	(10.8%;11.5%)
8	0.2	81066.9	81066.9	67482.8	0%	(0%;0%)	20.1298%	(19.7%;20.5%)
9	0.4	86441.9	86441.9	67472	0%	(0%;0%)	28.1151%	(27.7%;28.4%)
10	0.6	90418.7	90418.7	67449	0%	(0%;0%)	34.0549%	(33.7%;34.3%)
11	0.8	92498.8	93939.8	67500.6	1.56%	(1.32%;1.79%)	39.1687%	(38.7%;39.5%)
12	1	92820.6	103318	67457.4	11.31%	(11.1%;11.5%)	53.1608%	(52.7%;53.5%)

exp	DLP leg LF			DLP global LF	SG leg LF			SG global LF	FCFS leg LF			FCFS global LF
7	0.75	0.75	0.00	0.50	0.75	0.75	0.00	0.50	1.00	1.00	1.00	1.00
8	0.83	0.82	0.15	0.60	0.83	0.82	0.15	0.60	1.00	1.00	1.00	1.00
9	0.90	0.89	0.30	0.70	0.90	0.89	0.30	0.70	1.00	1.00	1.00	1.00
10	0.96	0.94	0.44	0.78	0.96	0.94	0.44	0.78	1.00	1.00	1.00	1.00
11	0.99	0.97	0.56	0.84	0.88	0.88	0.26	0.67	1.00	1.00	0.99	1.00
12	1.00	0.99	0.67	0.89	0.93	0.94	0.40	0.76	1.00	1.00	1.00	1.00

Cuadro 4.15: Resultados de los experimentos de buyups para  $\alpha = 1$ : arriba, resultados comparativos respecto del ingreso generado; abajo, load factor a nivel de leg y global

Antes de analizar los resultados, es importante hacer una aclaración sobre el comportamiento de DLP. Si bien se puede observar que el valor de ingreso generado por dicho algoritmo varía con los experimentos, el valor de los bid prices generados por éste se mantiene *constante*. Esto se debe a que la estimación de las demandas medias para DLP (que se asumen independientes) se hace

teniendo en cuenta la principal preferencia del cliente (el primer elemento de la lista), es decir que se ignora la presencia de potenciales buyups. La razón para que el ingreso varíe con el aumento de los buyups es que los bid prices generados por DLP cierran algunos productos, y en la simulación existe la probabilidad que los clientes compren otro, que DLP deja abierto, aumentando el ingreso y la carga de la red. Lo que hay que tener en mente es que DLP no tiene en cuenta los buyups a la hora de calcular los bid prices, pero SG sí, y eso debería reflejarse en la diferencia entre los ingresos generados.

Los resultados de las experiencias muestran que, a medida que aumenta el valor del coeficiente de las probabilidades, mayor es la diferencia entre SG y DLP. En ambas series, no se reportan mejoras en las primeras experiencias, pero a partir de determinado valor de  $C$  ( $C = 0.4$  en el primer caso, y  $C = 0.8$  en el segundo), se comienzan a observar mejoras, que llegan a ser muy pronunciadas en los últimos experimentos. Es interesante notar cómo en ambos casos la carga del tercer leg va aumentando a medida que se incrementa el valor de  $C$ : por ejemplo, en el primer experimento de  $\alpha = 1$  (experimento 7), para los bid prices generados por DLP la carga del leg 3 es 0, y la de los otros dos es del 75 % (75 asientos ocupados cada uno, que es justamente la media de las demandas de los vuelos directos), lo que indica que DLP cierra los vuelos con parada, para aprovechar los clientes de los vuelos directos, cuyas clases bajas son bastante más costosas que las de los vuelos con conexión. A medida que aumenta  $C$ , los controles aplicados por DLP comienzan a producir buyups de los clientes de los vuelos con parada, debido al incremento en las probabilidades de buyup, aumentando el ingreso y la carga de la red, y a partir de cierto momento SG comienza a aprovecharse de este hecho, y producir ingresos muy superiores a los de DLP.

## 4.2. Experimentos con Reoptimización

Como ya mencionamos en algún momento, una de las principales desventajas de nuestro método es que el set de bid prices que se calcula se aplica durante todo el horizonte de compras. En un contexto en el que se tiene una estimación confiable de la demanda, y no existe un mayor grado de incertidumbre, esta desventaja no representa una gran limitación. Sin embargo, para internalizar la dinámica de la demanda en nuestro método, se particiona el horizonte de compra en etapas, y se utiliza el método que implementamos para realizar una optimización en cada etapa, que tiene en cuenta la demanda previamente realizada. De esta manera se pueden utilizar nuevas predicciones de demanda que se hagan con el correr del tiempo (de hecho, ésta es la manera en que se aplican las políticas de *Revenue Management* en la realidad).

La metodología que utilizaremos para implementar las reoptimizaciones en los experimentos es la siguiente: vamos a dividir el horizonte de compra en  $K$  etapas de igual longitud, cada una con (aproximadamente)  $\frac{T}{K}$  períodos chicos, y vamos a generar listas de sample paths asociados. Los sample paths asociados a una etapa se generan simulando compras partiendo de la etapa en cuestión, y para todas las posteriores, hasta el final del horizonte. Esto es necesario porque la optimización que se realiza en cada etapa, que utiliza los sample paths generados para ésta, tiene que considerar la demanda que se espera obtener en adelante. Todas estas listas de sample paths son las que se utilizarán para la *optimización*; paralelamente se generará otra lista, con sample paths de todo el horizonte, que se utilizará para simular las ventas que se producen, aplicando las reoptimizaciones. Para clarificar el funcionamiento, mostramos a continuación un pseudocódigo de alto nivel de la metodología.

---

```
- Para k desde 1 hasta K (cantidad de etapas)
  - opt_lists[k] = generar lista de sample paths
                    con arribos de las etapas k,k+1,...,K
fin Para
- sim_list = lista de sample paths para todo el horizonte
- Calcular los bid prices iniciales con DLP utilizando sim_list
///Optimizacion / Simulacion///
- Para cada sample path 'SP' en sim_list
  - revenue_SP = 0
  - pi_inic = bid prices de DLP.
  - Para k desde 1 hasta K
    - pi_optimo_etapa_k = Correr SG utilizando opt_lists[k]
                        y como pi inicial a pi_inic
    - revenue_SP += simular utilizando pi_optimo_etapa_k
                    los arribos pertenecientes a la etapa k de SP
    - pi_inic = pi_optimo_etapa_k
  fin Para
fin Para cada
- Promediar los valores revenue_SP obtenidos para cada
  sample path y reportar las estadísticas pertinentes
```

---

Cabe aclarar que implícitamente estamos asumiendo que, dado un arribo en un sample path, es posible identificar a qué etapa del horizonte pertenece, lo que es razonable, sobre todo si se almacena esta información en el momento en que se genera dicho sample path. Otro comentario

que se puede hacer respecto del pseudocódigo es que se trata de un algoritmo computacionalmente muy intenso, aunque en realidad esto se debe a que lo que se hace es *evaluar* por simulación la performance de SG con reoptimizaciones. En una aplicación real las reoptimizaciones se realizarán para recalculer el valor de los bid prices, en la medida que se obtengan nuevas estimaciones de la demanda, lo que usualmente no ocurre con una frecuencia mayor que una vez por día. En otras palabras, en un contexto real de *Revenue Management*, aplicar SG con reoptimizaciones significa simplemente correrlo más de una vez durante el horizonte de compras.

A continuación, tomaremos los ejemplos de la sección 4.1, haremos experimentos utilizando reoptimizaciones, y compararemos los resultados con DLP, SG, y un resultado reportado en la literatura para el primer y segundo caso. Una aclaración importante es que el DLP que tomamos es el mismo que el de la sección anterior, es decir DLP *sin reoptimizar*. En realidad, para hacer una comparación más razonable, se podría tomar la metodología de reoptimización que utilizamos y en vez de reoptimizar con SG, debería con DLP. Sin embargo, lo que queremos observar es más que nada la diferencia entre utilizar SG con y sin reoptimización, razón por la cual decidimos omitir DLP reoptimizado en este trabajo.

#### 4.2.1. Ejemplo 1 con reoptimizaciones

La primera serie de experimentos de reoptimización corresponde al ejemplo 1, de la sección 4.1.1. Lo que hicimos fue tomar los dos escenarios de dicho ejemplo, e implementar sendos experimentos de reoptimización, basados en el pseudocódigo de más arriba. Partimos el horizonte en  $K = 5$  etapas, de 180 períodos cada una, y construimos la lista de sample paths de optimización para cada etapa (`opt_lists[k]` en el pseudocódigo), y la lista de sample paths para la simulación (`sim_list`). Tanto las listas de optimización como la de simulación tuvieron 500 sample paths<sup>3</sup>. En todos los casos que se ejecutó SG los parámetros de éste fueron los mismos que se utilizaron en el ejemplo 1. El cuadro 4.16 contiene los resultados de la reoptimización del primer escenario de dicho ejemplo, junto con las estadísticas ya presentadas en el cuadro 4.2.

Método	$\pi$	ingreso medio	LF leg AB	LF leg AC	LF leg CB	LF general
FCFS	[0,0,0]	29896	0.99	0.50	0.50	0.66
DLP	[180,0,0]	32942	0.60	0.99	0.99	0.86
SG	[192.3,-30.5,-31.1]	32942	0.60	0.99	0.99	0.86
<b>rSG</b>	-	<b>35885</b>	<b>0.78</b>	<b>0.91</b>	<b>0.91</b>	<b>0.87</b>
VN-SG	-	38381	-	-	-	0.79

Comparación	Gap	95% CI
SG vs FCFS	10.2%	(9.5%,10.8%)
SG vs DLP	0%	(0%,0%)
rSG vs DLP	16.88%	(16.5%,17.2%)

Cuadro 4.16: Resultados del escenario 1, incluyendo los resultados de la reoptimización (rSG representa a SG con reoptimizaciones y VN-SG - *Virtual Nesting Stochastic Gradients*- al valor reportado en van Ryzin y Vulcano[vRV08a])

<sup>3</sup>Esto quiere decir que las aplicaciones de SG se realizaron con un máximo de 500 iteraciones, un número menor al utilizado en el ejemplo 1, pero decidimos hacerlo de esta manera por la complejidad temporal del experimento, y porque además fue suficiente para obtener buenos resultados experimentales

Podemos observar que la mejora obtenida con la incorporación de las reoptimizaciones es sustancial: la diferencia porcentual entre el método de gradientes estocásticos y DLP pasa de ser del 0% a casi el 17%. Esto es algo que esperábamos que ocurra para este ejemplo, pues estamos tratando con demanda no homogénea en el tiempo, lo que da mayor oportunidad para utilizar diferentes sets de bid prices en diferentes momentos. En este caso, podemos ver que, respecto de SG sin reoptimizar, la carga sobre AB aumenta, indicando un incremento en la cantidad de ventas del vuelo directo, y las cargas sobre AC y CB disminuyen, evidenciando la aplicación de controles para manejar la congestión que existe sobre dichos legs, producida a su vez por los controles aplicados sobre AB (hecho sobre el que ya hablamos cuando presentamos los resultados de este escenario en la sección 4.1.1).

Para el segundo escenario, corrimos los experimentos de manera equivalente que para el primero y los resultados fueron los que se muestran en el cuadro 4.17.

Método	$\pi$	ingreso medio	LF leg AB	LF leg AC	LF leg CB	LF general
FCFS	[0,0,0]	23295	0.39	1	0.60	0.66
DLP	[0,100,0]	28191	0.39	0.89	0.60	0.63
SG	[-22.4,94.2,112.3]	28781	0.69	0.99	0.20	0.63
<b>rSG</b>	-	<b>33051</b>	<b>0.70</b>	<b>0.94</b>	<b>0.20</b>	<b>0.61</b>
VN-SG	-	31237	-	-	-	0.64

Comparación	Gap	95% CI
SG vs FCFS	23.5 %	(23.2 %,23.9 %)
SG vs DLP	2.1 %	(1.64 %,2.53 %)
rSG vs DLP	16.88 %	(16.6 %,17.2 %)

Cuadro 4.17: Resultados del escenario 2, incluyendo reoptimización

Nuevamente, observamos que el algoritmo es mucho más efectivo si se agregan reoptimizaciones; el incremento respecto de DLP pasa del 2% a casi 17%. Los load factors de SG con y sin reoptimizaciones son, en cambio, bastante similares. La diferencia entre uno y otro radica en que las reoptimizaciones permiten que se aprovechen los clientes de mayor poder adquisitivo que arriban en los períodos más cercanos en el tiempo.

#### 4.2.2. Ejemplo 2 con reoptimizaciones

El siguiente grupo de experimentos de reoptimización corresponde a las 3 instancias de la sección 4.1.2 de este capítulo. El procedimiento es el mismo que para el ejemplo anterior: para cada una de las instancias se partió el horizonte en 5 etapas (cada una con alrededor de  $\frac{T}{5}$  periodos chicos, con  $T$  la cantidad total de períodos de la instancia) y se contruyeron las listas de sample paths de optimización y la de simulación, ésta última de tamaño 50 para las instancias de Red Pequeña y Hub and Spoke, y 100 para Vuelos Paralelos. Las listas de optimización que se contruyeron para cada etapa fueron de tamaño 500, 200 y 300 respectivamente y los parámetros de SG fueron los

mismos que para los experimentos de la sección 4.1.2. Las Figuras 4.6, 4.7 y 4.8 contienen los valores porcentuales de ingreso de cada método, en relación al valor de DLP, para las 10 primeras experiencias de cada instancia.

De los resultados, podemos observar que en la mayoría de los casos rSG representa una mejora, muchas veces significativa, sobre SG, e incluso llega a superar a CDLP en más de una experiencia. Como conclusión vale decir que queda en evidencia la ventaja que resulta de aplicar las reoptimizaciones, incluso para ambientes en los que no existe un grado amplio de variabilidad de la demanda, como es el caso de estos experimentos.

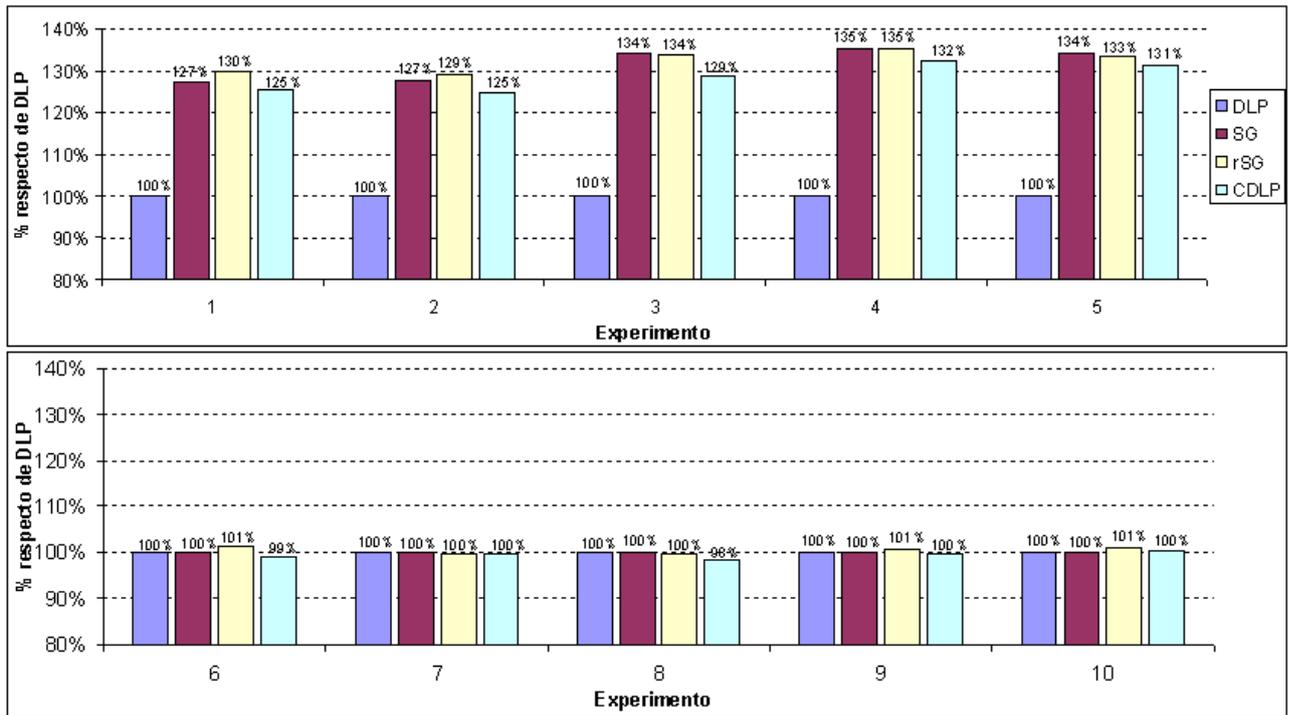


Figura 4.6: Resultados de experimentos para la instancia Vuelos Paralelos: variación porcentual, respecto de DLP, del ingreso generado por los diferentes métodos, para los experimentos 1 a 10 de la sección 4.1.2, instancia Vuelos Paralelos

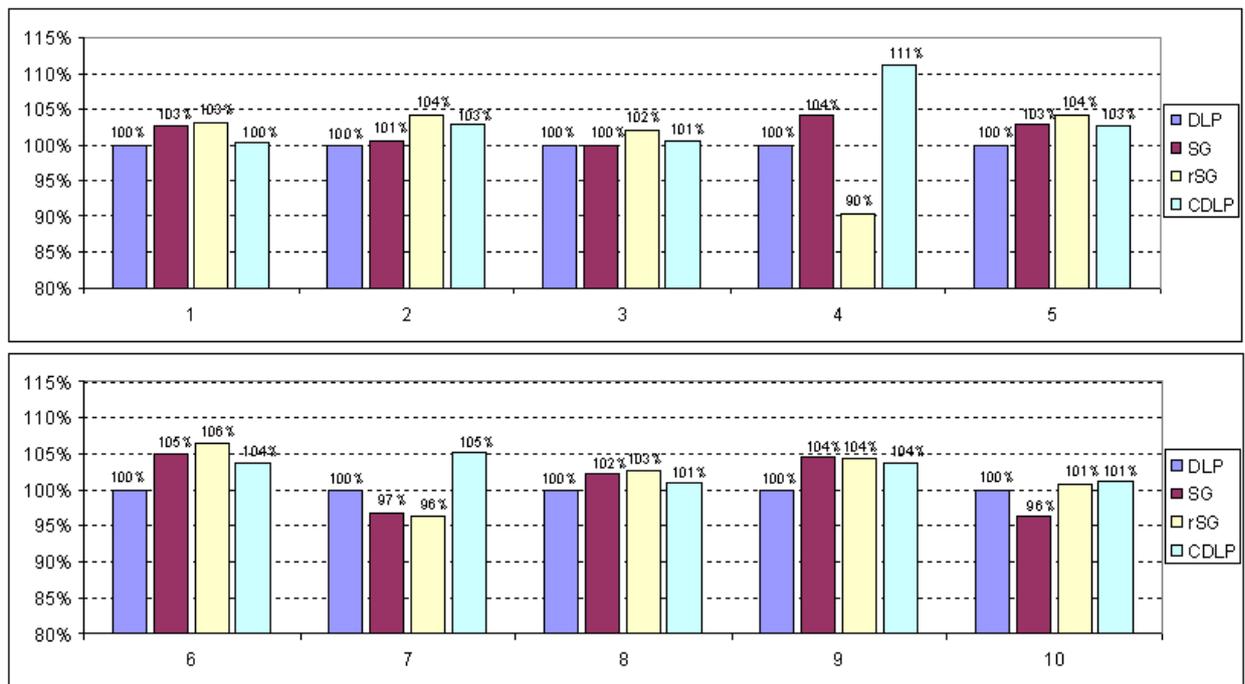


Figura 4.7: Resultados de experimentos para la instancia Red Pequeña: variación porcentual, respecto de DLP, del ingreso generado por los diferentes métodos, para los experimentos 1 a 10 de la sección 4.1.2, instancia Red Pequeña

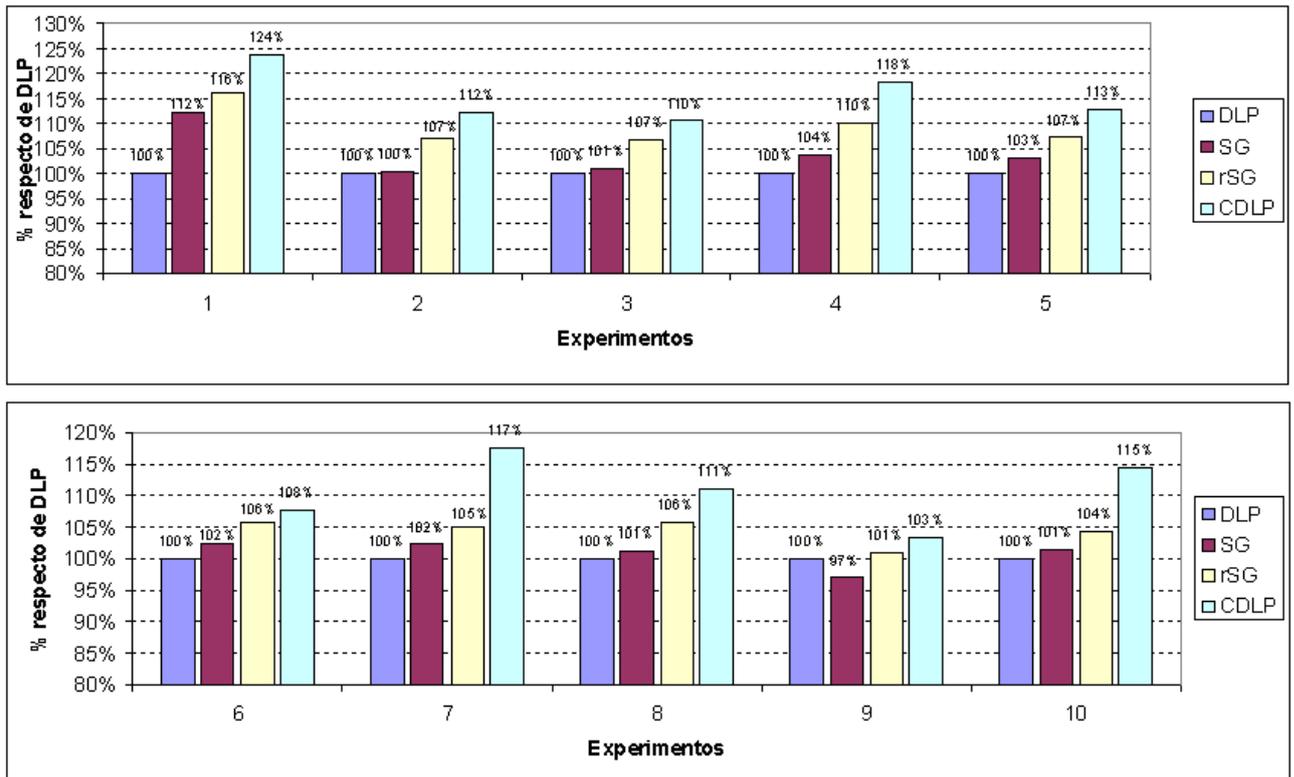


Figura 4.8: Resultados de experimentos para la instancia *Hub and Spoke*: variación porcentual, respecto de DLP, del ingreso generado por los diferentes métodos, para los experimentos 1 a 10 de la sección 4.1.2, instancia *Hub and Spoke*

### 4.2.3. Ejemplo 3 con reoptimizaciones

El último set de experimentos de reoptimización que realizaremos corresponde a los experimentos del ejemplo 3, sección 4.1.3. El procedimiento es análogo a los dos ejemplos anteriores. Para cada una de las instancias partimos el horizonte en 5 etapas, cada una con 200 períodos, y generamos sample paths para optimizar, 1000 en cada caso; para la simulación generamos 50 sample paths. Los parámetros del método fueron los mismos que para los experimentos originales, y los resultados se encuentran en la Figura 4.9.

De los resultados podemos concluir que en este caso las oportunidades de mejora con la reoptimización son menos significativas que en los ejemplos anteriores, ya que en la mayoría de las instancias rSG tiene un comportamiento similar al de SG.

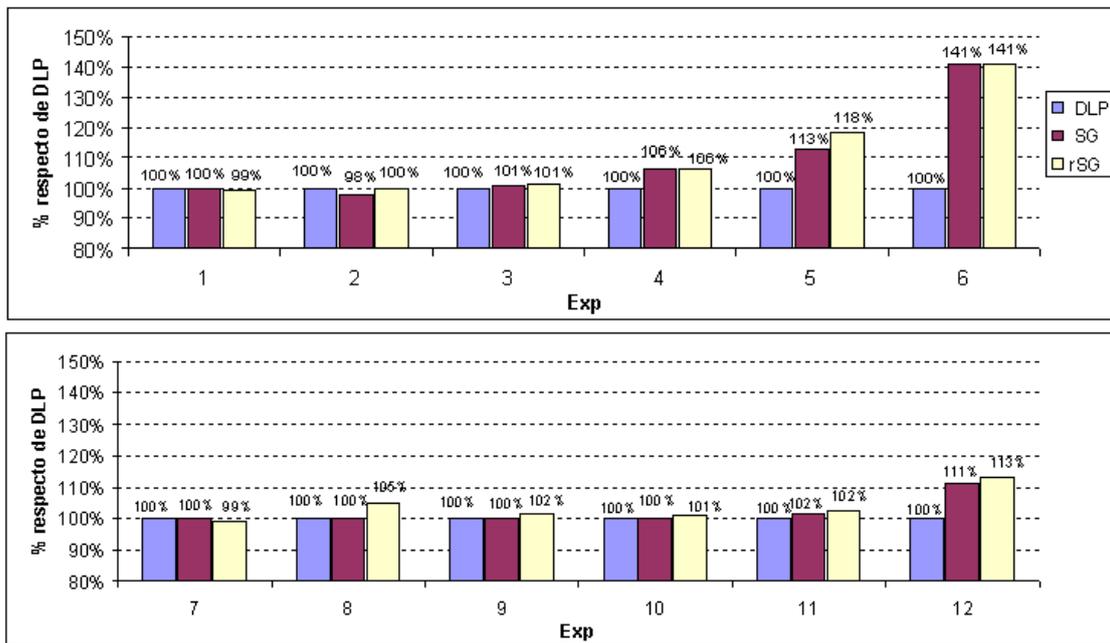


Figura 4.9: Resultados de experimentos de reoptimización para el ejemplo 3: variación porcentual, respecto de DLP, del ingreso generado por los diferentes métodos, para los experimentos 1 a 12 de la sección 4.1.3



# Capítulo 5

## Conclusiones

En el presente trabajo definimos un método basado en simulación para resolver el problema de control de capacidad en *Revenue Management* a nivel de red, que utiliza la política de bid prices, y que extiende el enfoque seguido por Topaloglu[Top08]. Utilizamos un modelo de demanda muy general, basado en secuencias estocásticas de clientes, donde cada cliente tiene una lista con sus opciones de compra, ordenadas por el grado de preferencia. Asumimos condiciones relativamente generales respecto de la distribución de la que provienen dichas listas, permitiendo cualquier tipo de correlación entre los clientes, o sus preferencias. Esto nos permite adaptar este modelo de demanda a diferentes modelos de elección utilizados en la práctica.

La definición del método se centra en la formulación de la función de ingreso y su gradiente muestral, y la principal característica es que se trabaja con demandas y capacidades continuas, lo que es necesario para que exista dicho gradiente. Para calcularlo, formulamos un algoritmo de complejidad lineal sobre la cantidad de arribos de clientes, y lo utilizamos como estimación muestral del gradiente de la esperanza del ingreso, que representa la función objetivo a maximizar, dentro de un algoritmo de optimización estocástica de “dirección más empinada”. Desde el punto de vista teórico garantizamos la convergencia del método, que puede probarse con argumentos similares a los expuestos en Topaloglu[Top08] y van Ryzin y Vulcano[vRV08a]. Como resultado negativo, mostramos un ejemplo en el que la función de ingreso no es cuasi-cóncava, de lo que se puede desprender que el algoritmo, que converge teóricamente a un punto crítico de la esperanza del ingreso, podría terminar en un punto “silla”, o incluso en un mínimo local.

Pese a estas desventajas teóricas, los experimentos que realizamos muestran que la metodología arroja buenos resultados prácticos, en problemas muy diferentes entre sí. Además de evaluar la eficacia práctica del método, en términos del ingreso generado, el objetivo de los experimentos es mostrar la generalidad del modelo de demanda en que se basa, a través de la implementación de diferentes modelos de elección que se han utilizado en la práctica: tipos de clientes fijos, modelos MNL y compras con buyups. Para evaluar la performance implementamos DLP: un método clásico del *Revenue Management*, muy aplicado en la práctica, que utilizamos como punto de partida para la optimización, y como valor comparativo para el ingreso generado. Los resultados mostraron que el algoritmo mejora la generación de ingreso de DLP en la gran mayoría de los casos, especialmente en

aquellos donde la oportunidad de realizar *Revenue Management* es pronunciada, es decir, cuando hay una carga alta en la red. En dichos casos, la mejora producida por el algoritmo, respecto de DLP y de no aplicar controles (FCFS), fue muy significativa.

Sin embargo, es necesario tener en cuenta que, dado que DLP es un método heurístico, que también utiliza demanda continua, pero que asume que las demandas son independientes, no debería ser muy sorprendente que, en ejemplos donde se modela la elección de los clientes, sea superado por un método que contempla explícitamente dichas elecciones. Por esta razón es que incluimos en los experimentos resultados generados por métodos basados en modelos de elección, y sobre los que se han reportado buenos resultados en la práctica: CDLP y VN-SG. Si bien la comparación con éstos es menos favorable que respecto a DLP, los resultados muestran que nuestro método los supera en muchos casos, especialmente con la aplicación de reoptimizaciones. En realidad, para hacer un estudio más exhaustivo al respecto, se deberían implementar diferentes métodos que capturen comportamiento de elección, y evaluar la performance en diversos ejemplos, incluyendo ejemplos con redes de mayor tamaño, que se asemejen a las que existen en la realidad, cosa que, por razones de extensión y complejidad, no hemos hecho en este trabajo.

Finalmente, una ventaja significativa de nuestro enfoque, compartida por VN-SG (y no así por CDLP) es que es incremental, en el sentido de que puede incorporarse a los sistemas actuales empleados por las aerolíneas, lo que, junto a la buena performance observada, lo convierte en un método con un importante potencial para su implementación comercial.

# Bibliografía

- [BAL94] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Applications to Travel Demand*. The MIT Press, Cambridge, MA, sixth edition, 1994.
- [Bel89] P.P. Belobaba. Application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37:183–197, 1989.
- [BT96] Bertsekas and Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [CdMK06] W. Cooper, T. Homem de Mello, and A. Kleywegt. Models of the spiral-down effect in revenue management. *Operations Research*, 54:968–987, 2006.
- [Coo02] W. Cooper. Asymptotic behavior for an allocation policy for revenue management. *Operations Research*, 50:720–727, 2002.
- [GIPD04] 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.
- [Gla94] P. Glasserman. Perturbation analysis of production networks. In D.D. Yao, editor, *Stochastic Modeling and Analysis of Manufacturing Systems*, chapter 6, pages 233–280. Springer, New York, NY, 1994.
- [JMBV] I. Mendez-Diaz J. Miranda Bront and G. Vulcano. A column generation algorithm for choice-based network revenue management. Forthcoming in *Operations Research*.
- [KS00] Anton Kleywegt and Alexander Shapiro. Stochastic optimization. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta Georgia, 2000.
- [LvR08] Q. Liu and G.J. van Ryzin. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Opreations Management*, 10:288–310, 2008.
- [RM51] H. Robbins and S. Monro. On a stochastic approximation method. In *Annals of Mathematical Statistics*. 1951.

- [RRNY] R. Ratliff, B. Rao, C. Narayan, and K. Yellepeddi. A multi-flight recapture heuristic for estimating unconstrained demand from airline bookings. *Journal of Revenue and Pricing Management*.
- [Sim89] R. W. Simpson. Using network flow techniques to find shadow prices for market and seat inventory control. Technical report, Flight Transportation Laboratory, MIT, Cambridge, MA, 1989.
- [Top08] H. Topaloglu. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS Journal on Computing*, 20:596–610, 2008.
- [Tra03] K. Train. *Discrete choice methods with simulation*. Cambridge University Press, New York, NY, 2003.
- [TvR04a] K. Talluri and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, New York, NY, 2004.
- [TvR04b] 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.
- [vRV08a] G.J. van Ryzin and G. Vulcano. Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, 10:448–467, 2008.
- [vRV08b] G.J. van Ryzin and G. Vulcano. Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research*, 56:865–880, 2008.
- [VvRC08] G. Vulcano, G. van Ryzin, and W. Chahr. Choice-based revenue management: An empirical study of estimation and optimization. Working paper, Stern School of Business, New York University, New York, 2008.
- [ZC05] D. Zhang and W. L. Cooper. Revenue management for parallel flights with customer-choice behavior. *Operations Research*, 53:415–431, 2005.

# Agradecimientos

Este trabajo se desarrolló en el marco del Proyecto FONCyT Nro. PICT 2006-01070 (Raices), y ha sido financiado parcialmente por una beca de la Research Division de Sabre Airline Group, Southlake, Texas. U.S.A, y por la Escuela de Negocios de la Universidad Torcuato di Tella.

## Gracias a:

### ▪ Por la Tesis:

- A Gustavo, por el apoyo, la paciencia, y las gestiones asociadas a este trabajo, y por todo el tiempo y esfuerzo que le dedicó a lo largo de este último año.
- A los jurados, Paula Zabala y Pablo Jacovkis.
- A Juanjo Miranda Bront, por el material, y por la continua buena onda y gran predisposición que lo caracterizan.
- A mi amiga arti, por el oído y la compañía en este tramo final.
- A la Universidad Di Tella y el Sabre Holdings Research Group, por el financiamiento.

### ▪ Por la Carrera:

- A la fundación Carlos y María Sol Chaneton, por la financiación, la compañía a la distancia, y el continuo interés a lo largo de todos estos años.
- A mi abuela Susana, por la confianza, el entusiasmo, y el alojamiento.
- Al cuerpo de docentes y no docentes del Departamento, y de la Facultad.
- Al eterno y siempre victorioso grupo de TP: Lea, Cesaro y el Marine.
- A los muchachos de la generación: Guille, Cisquito, El Pocho De la Gente, Lea, Cesaro, el Marine, Arti, Nigel, Gaboto, Gennaro, Sol, Daniel, Lucio, David, Martín, Herni.

### ▪ Por qué no:

- Al resto de la flía: Ama, Fico y Manolo; Lucho, Tata y los primos.
- Al glorioso equipo Tu amiga te tiene ganas, y al no tan glorioso pero digno Kaiserlautern.
- A los neuquinos de cerca y de lejos: Pedro, el Gordo, Hugo, el Negro, el Turco, el Joqui y el Yampi.
- A los muchachos de la lista negra, y en especial a los que (todavía) hacen algún esfuerzo por salir de ella.

- A la casa del Pocho, testigo de nuestra adolescencia tardía durante todos estos años.
- Al Siena que ya no está y se ganó nuestro corazón y respeto, y al Siena que vino a reemplazarlo.