



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

## Revenue Management bajo comportamiento selectivo de clientes en la industria hotelera

ALUMNO: Germán David Kruszewski Martel  
L.U. 274/03  
germank@gmail.com

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

CO-DIRECTORA: Dra. Paula Zabala  
pzabala@dc.uba.ar  
Departamento de Computación, FCEyN, UBA, Buenos Aires

Julio de 2011

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> . . . . .	11
1.1.3. Un marco conceptual para <i>Revenue Management</i> . . . . .	12
1.1.4. Descripción de un sistema de <i>Revenue Management</i> . . . . .	15
1.1.5. El control de capacidad . . . . .	16
1.1.5.1. Límites de reservas y niveles de protección anidados	17
1.1.6. Modelos de Demanda . . . . .	17
1.1.6.1. Modelos de demanda independiente . . . . .	18
1.1.6.2. Modelos de elección . . . . .	19
1.1.7. Control de capacidad basado en un modelo de demanda inde- pendiente: EMSR-b y MEMUR . . . . .	19
1.1.8. Control de capacidad basado en un modelo de elección: CDLP	20
1.1.9. <i>Network Revenue Management</i> . . . . .	20
1.2. <i>Revenue Management</i> en la industria hotelera . . . . .	20
1.2.1. Sobre este trabajo . . . . .	21
<b>2. Estimación</b>	<b>23</b>
2.1. Descripción del dataset . . . . .	23
2.2. Estimación de máxima verosimilitud . . . . .	24
2.3. Descripción del modelo . . . . .	25
2.4. El método <i>expectation-maximization</i> . . . . .	27
2.5. Procedimiento . . . . .	29
2.5.1. Definición de Producto . . . . .	30
2.5.2. Definición de Período . . . . .	31
2.5.3. Segmentación de los datos . . . . .	34
2.6. Resultados de la estimación . . . . .	35
2.7. Descomposición de las preferencias . . . . .	38
<b>3. Optimización</b>	<b>41</b>
3.1. Consideraciones preliminares . . . . .	41
3.2. Descripción del modelo . . . . .	42
3.2.1. Formulación basada en programación dinámica . . . . .	43
3.3. Instanciación del modelo . . . . .	43
3.4. Descripción de los algoritmos utilizados . . . . .	45
3.4.1. FCFS . . . . .	45

3.4.2.	CDLP . . . . .	46
3.4.2.1.	Construcción de la política . . . . .	47
3.4.2.2.	Generación de columnas . . . . .	48
3.4.3.	Heurística: agregación de estados del problema de programación dinámica . . . . .	51
3.4.3.1.	Selección eficiente de los <i>offer sets</i> . . . . .	54
3.4.4.	MEMUR . . . . .	55
3.5.	Análisis de CDLP . . . . .	56
3.5.1.	Un problema de CDLP: la política estática . . . . .	57
3.5.2.	Segundo problema de CDLP: subutilización de recursos . . . . .	58
3.5.3.	Explicación del caso real . . . . .	60
3.5.4.	Reoptimización de CDLP . . . . .	62
3.5.5.	Comparación con otros trabajos del área . . . . .	63
3.6.	Experimentos computacionales . . . . .	64
3.6.1.	Resultados . . . . .	67
3.6.1.1.	Tiempos de ejecución . . . . .	75
<b>4.</b>	<b>Conclusiones</b>	<b>77</b>
<b>A.</b>	<b>Resultados de la Estimación</b>	<b>79</b>
A.1.	Hotel 1 . . . . .	79
A.2.	Hotel 2 . . . . .	80
A.3.	Hotel 3 . . . . .	81
A.4.	Hotel 4 . . . . .	83
A.5.	Hotel 5 . . . . .	84
<b>B.</b>	<b>Resultados de la Optimización</b>	<b>85</b>
B.1.	Hotel 1 . . . . .	85
B.2.	Hotel 2 . . . . .	88
B.3.	Hotel 3 . . . . .	88
B.4.	Hotel 4 . . . . .	92
B.5.	Hotel 5 . . . . .	94

---

## Resumen

En los últimos años, se ha observado un creciente interés en utilizar modelos de elección para los problemas de *Revenue Management*. Sin embargo, se han presentado muy pocos resultados empíricos de su aplicación en el marco de una industria real. En este trabajo, utilizamos datos pertenecientes a cinco cadenas hoteleras publicados por Bodea et al. [BFG09] para calcular un modelo de demanda basado en elección. Luego, a esta caracterización la internalizamos en políticas de control de capacidad y medimos su impacto en la generación de ingreso. Para estimar el modelo de elección, utilizamos el método *expectation-maximization* propuesto por Vulcano et al. [VvRR08].

Resolvemos el problema de control de capacidad utilizando CDLP (un método muy conocido cuyos detalles se tratan en [LvR08]), MEMUR (propuesto por Ferguson et al. en [FGMN10]) y una heurística nueva, planteada por nosotros. Los resultados arrojaron un rendimiento inesperadamente bajo para CDLP, lo que nos llevó a indagar las causas de este comportamiento. De esta manera, llegamos a caracterizar dos problemas generales que pueden afectar el rendimiento de una política de control de capacidad obtenida por CDLP. Luego, buscamos resolver estos problemas mediante reoptimizaciones periódicas del problema, mejorando sustancialmente los resultados. Por su parte, los resultados de nuestra heurística se mostraron similares a los de CDLP con reoptimizaciones, aunque dominándolo. Por último, los resultados de MEMUR fueron más bien eclécticos, no permitiéndonos arribar a una conclusión general sobre este método.

## Abstract

In the last few years, there has been an increasing interest on choice based models of demand for *Revenue Management* problems. However, there has been little empirical research on their application to a real industry. In this work, we use a dataset published by Bodea et al. [BFG09] with information of five major hotel chains to build a choice based demand model. Then, we internalize this characterization in capacity control policies and measure their impact on the revenue generation. To build the demand model, we make use of the *expectation-maximization* method proposed by Vulcano et al. [VvRR08].

We solve the capacity control problem using CDLP (a well-known method discussed in [LvR08]), MEMUR (proposed by Ferguson et al. in [FGMN10]) and a new heuristic, developed by us. Since our results showed an unexpectedly low income when using the CDLP policy, we decided to investigate the causes of this behavior.

---

This allowed us to identify two general problems that can affect the performance of a policy obtained by CDLP. Then, we tried to solve these problems by enforcing periodical reoptimizations of the model, which resulted in a noticeable revenue improvement. Regarding our heuristic, it has produced results comparable to the reoptimized CDLP, but sometimes even dominating it. On the other hand, we have found MEMUR results to be eclectic, and for that reason we haven't been able to arrive to a general conclusion regarding that method.

*RESUMEN*

---



# 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, aplicaremos a un caso real de la industria hotelera métodos de optimización del ingreso que incorporan 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 comple-

jas, relacionadas con la demanda. Por ejemplo, ¿cómo puede una compañía segmentar a sus clientes ofreciéndoles diferentes términos contractuales y condiciones que exploren 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, el *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 en el 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 a operar 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 (típicamente, viajeros de placer), además de ofrecer un número fijo y limitado de este tipo de asientos por vuelo. Esta nueva estrategia trajo buenos resultados, pero todavía era posible

mejorarlo.

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

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

El efecto de esta nueva estrategia fue dramático. *PeopleExpress* fue una de las compañías a las que más afecto esta estrategia de ventas: pasó de obtener sus máximas ganancias en 1984, a perder 160 millones de dólares estadounidenses a fines de 1986. En solamente dos años, la compañía se declaró en bancarota. 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 podemos destacar los diferentes productos que ofrece, los diferentes tipos clientes con sus respectivas 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 los hoteles, un pedido por una habitación agotada se puede aceptar ofreciendo una habitación mejor por el mismo precio, pero esto implicaría dejar de vender esta última por su precio

original, más alto.

- *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 status 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* de clase turista 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. Los hoteles, al igual que las aerolíneas, proveen un buen ejemplo de esta característica. En un mismo hotel podemos encontrar huéspedes de alto poder adquisitivo presentes en fechas fijas por viajes de negocios, y turistas de cronogramas flexibles en busca de opciones más económicas.
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 hotelera como las aerolíneas son un claro ejemplo de esta condición, pues la capacidad es limitada y el costo de operación es prácticamente fijo.
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. Las grandes cadenas hoteleras disponen de sistemas de ventas automatizados que garantizan el cumplimiento de esta condición. Asimismo las aerolíneas disponen de servicios de venta automatizados desde hace más de 30 años.
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*.

Estas condiciones caracterizan aquellos negocios donde es posible aplicar *Revenue Management*, con la posibilidad de obtener buenos resultados. Vimos que las aerolíneas cumplen con todas ellas, convirtiéndola en una de las industrias donde más se aplica este tipo de técnicas. Por su parte, la industria hotelera comparte en buena medida estas características, razón por la cual, en ella también el *Revenue Management* se encuentra muy difundido.

### 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.

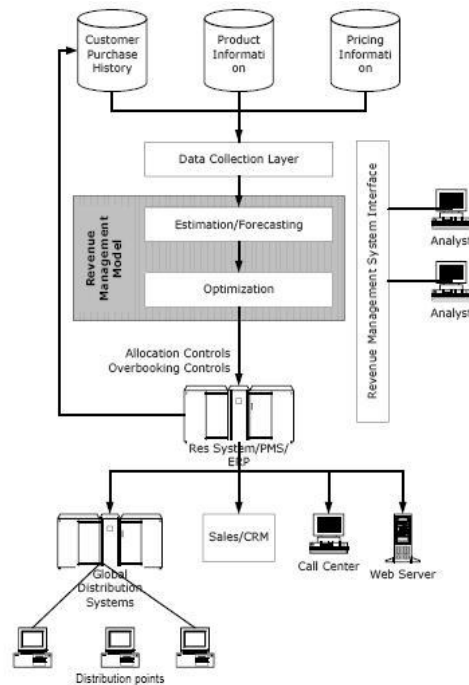


Figura 1.1: Descripción de un sistema de *Revenue Management*.

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>.

### 1.1.5. 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>. El problema operativo consiste en decidir qué productos ofrecer en función de la capacidad remanente y del tiempo restante hasta la fecha de consumo.

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

<sup>1</sup>Extraído de Talluri y van Ryzin [TvR04], con permiso de los autores.

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



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*. Para una descripción más completa referirse a [TvR04].

### 1.1.5.1. Límites de reservas y niveles de protección anidados

Los límites de reservas y niveles de protección son controles que limitan la capacidad disponible y la distribuyen entre las diferentes clases de tarifa. Para esto, se supone que las unidades de capacidad son homogéneas. En el caso de los límites de reserva, se asigna una cantidad máxima de recursos que se podrán vender (reservar) para cada clase. Los niveles de protección anidados son el complemento respecto de la capacidad total. En lugar de definir límites para la cantidad máxima de ventas por clase, se definen niveles de protección como una cantidad mínima de recursos que tienen que ser reservadas para las clases superiores. Por ejemplo, supongamos que se disponen de 30 unidades 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 para la clase 2 establece que a lo sumo se aceptarán 20 compras para esa clase. Esto equivale a un nivel de protección de 10 para la clase 1, por el cual se bloquean 10 unidades para dicha clase. La Figura 1.2 da un ejemplo de la aplicación para ambos controles.



Figura 1.2: Ejemplos de límites de reserva y niveles de protección anidados con 30 unidades de capacidad y 3 clases.

### 1.1.6. 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. Existen, a grandes rasgos, dos tipos de modelos de demanda que se utilizan hasta hoy, y que detallamos a continuación.

#### 1.1.6.1. Modelos de demanda independiente

Los modelos de demanda independiente son los que se utilizaron originariamente en *Revenue Management* (y se utilizan aún hoy mayoritariamente en la práctica), principalmente debido a su simplicidad. En ellos, se asume que la demanda para cada uno de los productos o clases es independiente, lo que simplifica mucho el trabajo de optimización. Sin embargo, esta presunción parece estar lejos de ser realista, ya que un cliente que quiere comprar un producto no busca un precio en particular sino, en general, el más económico disponible. Esto implica dependencia entre los productos, y en particular, 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, en el caso de la industria de transporte aéreo de pasajeros, 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 sensitivos al precio, sólo dispuestos a pagar la tarifa más económica primero, y los insensitivos al precio, virtualmente dispuestos a pagar algo más, últimos. Nuevamente, las restricciones arriba mencionadas, podían hacer que esto fuera suficientemente realista.

En los últimos años, la flexibilización de estas restricciones –fundamentalmente debido al nuevo auge de las aerolíneas de bajo costo– 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.

### 1.1.6.2. Modelos de elección

Tras relajar el supuesto de la independencia entre las demandas para las diferentes clases, se comienza a considerar dentro del modelo el comportamiento selectivo de los clientes. Para esto, los *modelos de elección* condicionan la probabilidad de compra de un determinado producto a las opciones disponibles. Dichas opciones van a estar dadas por los productos que puede comprar y también por la opción de no comprar nada. Por ejemplo, si un pasajero desea viajar a un destino, seguramente va a evaluar qué pasajes hay disponibles y va a ponderar sus necesidades (llegar en determinada fecha, minimizar el precio, etc.) para seleccionar el más conveniente. 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 y luego definir una relación de preferencia entre éstas, que dependa de los valores de los parámetros de cada una. 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. En la sección 2.3 haremos una descripción más detallada del modelo utilizado en este trabajo.

### 1.1.7. Control de capacidad basado en un modelo de demanda independiente: EMSR-b y MEMUR

Una de las heurísticas más extendidas en los sistemas de *Revenue Management* es EMSR-b (*Expected Marginal Seat Revenue-Version b*, ver [Bel89]). Dicha heurística está definida para un escenario en que un único recurso se comparte entre múltiples clases anidadas y permite obtener los niveles de protección para cada una de estas clases conociendo la distribución de las demandas independientes para cada clase.

*MEMUR* es el algoritmo que desarrolla Ferguson et al.[FGMN10] como una generalización de *EMSR-b* a un escenario de múltiples recursos. A grandes rasgos, este método asocia cada recurso con una clase de tarifa y, basado en una relación transitiva para definir un orden parcial entre las clases, establece los niveles de protección para las clases superiores en forma análoga a EMSR-b.

### 1.1.8. Control de capacidad basado en un modelo de elección: CDLP

En oposición a esto, existen los métodos de *control de capacidad* basados en elección, los cuales no dependen de una distribución de la demanda por cada producto o clase, sino que utilizan una función de probabilidad de elección de un cierto producto dadas las alternativas disponibles. En este tipo de métodos se inscribe CDLP, que se detalla en profundidad en [LvR08]. Este algoritmo reemplaza las magnitudes aleatorias por sus valores medios esperados y, mediante la resolución de un problema de programación lineal, obtiene la cantidad de tiempo en que se debe ofrecer cada posible subconjunto de productos. El orden en que esto se debe hacer no viene dado por el algoritmo y, por lo tanto, se debe utilizar alguna heurística para finalmente construir la política de control.

### 1.1.9. *Network Revenue Management*

El *Network Revenue Management* se asocia a situaciones en las cuales un cliente compra un producto por el cual consume un paquete de recursos sujeto a distintos términos y condiciones. El caso más típico es el de las aerolíneas, en donde, por ejemplo, un pasajero que quiere ir de Buenos Aires a Roma, puede elegir un vuelo directo (donde consume un único recurso que es un asiento en dicho vuelo) o bien, tomar un vuelo con escala en Madrid (donde consume dos recursos: uno en el vuelo de Buenos Aires a Madrid, y otro en el vuelo de Madrid a Roma). Cuando existen productos que consumen más de un recurso en simultáneo, se generan interdependencias entre los recursos pues si uno de los recursos se agota el producto entero ya no estará disponible. Esto requiere que el *control de capacidad* se haga en forma coordinada para todos los recursos.

## 1.2. *Revenue Management* en la industria hotelera

Vimos que el *Revenue Management* se desarrolló y cobró un gran impulso de la mano de la industria de transporte aéreo de pasajeros. Sin embargo, mencionamos condiciones por las cuales podía adaptarse a una variedad de industrias, entre las cuales, se encuentra la industria hotelera. Existen muchas diferencias entre ambas industrias, como por ejemplo, mientras que en las aerolíneas el producto es entregado en forma puntual y simultánea a todos los clientes, en los hoteles el producto se entrega a medida que llegan los clientes. Y de hecho, es muy común que muchas reservas se efectúen en el mismo día de *check-in*. Por lo tanto, detectar clientes que reservaron, pero que no van a hacer uso de dicha reserva es importante para la

administración del hotel, a fin de satisfacer a estos clientes de última hora. En este trabajo no vamos a considerar estas particularidades, para lo cual vamos a suponer que no hay cancelaciones ni clientes que no reclaman su reserva.

Asimismo, mientras que la capacidad de un vuelo se utiliza parcial o totalmente durante un viaje y luego se vacía, los hoteles y se desocupan en una forma que se extiende en el tiempo. Sin embargo, para simplificar nuestro trabajo, vamos a considerar que todas las reservas se hacen por una única noche.

Por último, en el caso de las aerolíneas, un recurso consiste en un asiento disponible en un vuelo punto a punto. Es razonable, si bien subóptimo, resolver el problema de control de capacidad a nivel de cada uno de estos recursos por separado, y de hecho, como ya vimos, es lo que se ha llevado en primer término a la práctica. Sin embargo, en el caso de la industria hotelera, donde cada tipo de recurso viene dado por el tipo de habitación, se hace más urgente modelar la dependencia entre los distintos tipos de recursos. Esto se debe a que por un lado, la elección misma del cliente va a predicar casi directamente sobre el tipo de recurso que quiere consumir (i.e. qué tipo de habitación quiere). Por otro lado, un esquema de múltiples recursos nos va a permitir reemplazar un recurso por otro en caso de agotamiento del primero.

### 1.2.1. Sobre este trabajo

Llegados a este punto, estamos en condiciones de explicar en qué consiste el presente trabajo. Nos vamos a ocupar de implementar los dos módulos que forman parte de un sistema de *Revenue Management* que se aprecian la figura 1.1: la estimación de los parámetros del modelo y la posterior optimización de las políticas de control. Para eso, necesitamos como entrada información histórica de reservas y, en particular, dado que vamos a utilizar modelos de elección, necesitamos conocer las alternativas disponibles al momento de efectuarse una reserva. Utilizamos para esto el dataset publicado por Bodea et al.[BFG09] en el que se detallan reservas hechas en 5 grandes hoteles con fechas de *check-in* en el período 12/03/2007 a 15/04/2007. En base al mismo, construimos un modelo de demanda basado en un *modelo de elección*. Específicamente, utilizamos el método *expectation-maximization* descrito por Vulcano et al.[VvRR08]. Dicho modelo se explica con detalle en la Sección 2.3.

Una vez completada esta etapa, internalizamos el modelo de demanda en diversos algoritmos para resolver el problema de *control de capacidad*. Utilizamos a este fin *MEMUR*, *CDLP* y una nueva heurística elaborada por nosotros. Luego probamos el rendimiento obtenido con cada uno de estos métodos mediante simulación.

Nuestros resultados mostraron que sorprendentemente *CDLP* ofrecía rendimien-

tos inferiores a no efectuar ningún control de capacidad. Estudiamos en profundidad la razón por la que ocurría este fenómeno e implementamos reoptimizaciones periódicas para mejorar su performance. Con esta mejora, el método comenzó a ofrecer incrementos apreciables en los ingresos respecto a no hacer ningún control de capacidad. Asimismo, los resultados de nuestra heurística se mostraron similares a los de *CDLP* con reoptimizaciones, aunque dominándolo. Por último, los resultados de *MEMUR* fueron más bien eclécticos, no permitiéndonos arribar a una conclusión general sobre este método.

El resto de este trabajo se encuentra organizado del siguiente modo: en el capítulo 2 abordamos el problema de la estimación de los parámetros, mientras que en el capítulo 3 nos dedicamos a optimizar las políticas de control y a estudiar sus resultados. Por último en el capítulo 4 elaboramos algunas conclusiones del presente trabajo.

## Capítulo 2

# Estimación

En este capítulo abordaremos el primer módulo de un sistema de *Revenue Management*: la estimación de los parámetros del modelo de demanda. Comenzaremos describiendo la base de datos utilizada como fuente de información básica (sección 2.1). Luego, mencionaremos los conceptos básicos de estimación de máxima verosimilitud (sección 2.2) para pasar a detallar el modelo utilizado para describir la demanda (sección 2.3). Luego, explicaremos el algoritmo que aplicamos para inferir los parámetros de dicho modelo (sección 2.4). Llegados a este punto, en la sección 2.5 explicamos de qué manera internalizamos en el modelo los datos disponibles y en la sección 2.6 mostramos los resultados obtenidos. Por último, en la sección 2.7, abordamos una breve discusión acerca de cómo extrapolar algunos datos adicionales de las estimaciones obtenidas.

### 2.1. Descripción del dataset

El dataset utilizado fue compilado por Bodea et al. ([BFG09]). En él se incluyen datos de reservas hechas en 5 hoteles para días de check-in que van del 12/03/2007 al 15/04/2007. Para cada una de estas posibles fechas de check-in, se comenzaron a registrar reservas desde un mínimo de 4 semanas de antelación.

Para cada reserva se detalla:

- Fecha de la reserva
- Fecha de check-in
- Fecha de check-out / Largo de la estadía

- Canal por el cual se efectuó la reserva (sistema del hotel, sitios web, agencias de viaje, etc.)
- Tipo de habitación reservada
- Tarifa abonada por noche de la opción reservada
- Código de la tarifa reservada: las tarifas se agrupan en distintos tipos según si se trata de un precio promocional por compra adelantada, si es el precio general, o si incluye algún servicio adicional.
- Lista de alternativas observadas por el cliente al momento de la reserva. Por cada alternativa se especifica el tipo de habitación, el código de tarifa y el valor de la tarifa.

Por último, el dataset provee por cada reserva un indicador denominado *Merge Indicator* que marca las reservas para las cuales se pudo obtener la lista de alternativas correspondiente. La existencia de este indicador se debe a que a veces los autores encontraron inconvenientes técnicos para obtener la mencionada lista de alternativas, y optaron por incluir esa información en el dataset, pero demarcándola adecuadamente.

## 2.2. Estimación de máxima verosimilitud

Los parámetros de los modelos de elección se suelen estimar frecuentemente mediante métodos de *estimación de máxima verosimilitud*. Sea  $Z$  una variable aleatoria, y sea  $\mathbf{x}$ , un vector de atributos conocidos que influyen la distribución de  $Z$ . Denotamos esta dependencia como  $Z \sim f(\mathbf{x}; \theta)$ , donde  $f$  es la distribución de  $Z$  y  $\theta$  un vector de parámetros, algunos de los cuales son desconocidos a priori. Usando una muestra de observaciones  $z_1, \dots, z_N$  del proceso que se está modelando, se construye un *estimador*: una función de las observaciones para estimar los parámetros desconocidos.

Denotamos con  $\hat{\theta}$  al estimador de  $\theta$ . Hasta que se se obtiene la muestra,  $\hat{\theta}$  constituye una variable aleatoria. Pero, luego de que se realiza el proceso y se observa la muestra,  $\hat{\theta}$  es simplemente un vector de números escalares. Para distinguir ambas posibilidades, nos referimos a la variable aleatoria como el *estimador* y a cualquier realización como una *estimación*.

Los estimadores de máxima verosimilitud están basados en inferir los parámetros  $\theta$  que maximizan la probabilidad de observar los datos que se obtuvieron



en la muestra. La verosimilitud de observar el  $n$ -ésimo valor de la muestra  $z_n$ ,  $1 \leq n \leq N$ , se denota  $f(z_n|x_n, \theta)$ . La verosimilitud de observar resultados independientes  $((z_1, x_1), \dots, (z_N, \mathbf{x}_N))$  condicionado a los parámetros  $\theta$  es simplemente

$$\mathcal{L}(x, \theta) = \prod_{n=1}^N f(z_n|x_n, \theta)$$

El problema de obtener el estimador de máxima verosimilitud consiste en obtener la estimación  $\hat{\theta}$  que maximiza  $\mathcal{L}(x, \theta)$ . El enfoque tradicional es aprovechar la monotonía estricta del logaritmo y maximizar el logaritmo de  $\mathcal{L}$ , lo que no cambia el valor del estimado resultante.

### 2.3. Descripción del modelo

En el modelo descrito por Vulcano et al.[VvRR08],  $n$  productos se venden durante  $T$  períodos, con índices  $t = 1, 2, \dots, T$ . No se presupone nada respecto del orden o la duración de estos períodos. Por ejemplo, un período de compra puede ser un día, en cuyo caso tendríamos datos de  $T$  días, o puede ser una semana, en cuyo caso tendríamos observaciones de compras realizadas durante  $T$  semanas. Los períodos incluso pueden tener distintas duraciones y no necesariamente estar en orden cronológico.

La única información disponible para cada período son las compras efectuadas (es decir, cuántas unidades se vendieron de cada producto en cada período), y un indicador binario de la disponibilidad de cada producto durante el período. El número de clientes que llega en cada período no se conoce; equivalentemente, tampoco observamos la cantidad de clientes que deciden no comprar.

Al conjunto de todos los productos se los denota  $\mathcal{N} = 1, \dots, n$ . El número de compras del producto  $i$  observadas en el período  $t$  se denota como  $z_{it}$ , y se define  $z_t = (z_{1t}, \dots, z_{nt})$ . El modelo asume que  $z_{it} \geq 0$  para todo  $i, t$ . Además asume que para todo producto  $i$  existe un período  $t$  tal que  $z_{it} > 0$ .

Asimismo, se asume que en cada período, ocurren  $A_t$  arribos con distribución Poisson y una tasa media  $\lambda_t$ . Por otra parte, se asume que sólo un subconjunto  $S_t \subset \mathcal{N}$  de productos se ofrece en el período  $t$ , que se conoce  $S_t$  para cada  $t$  y que este conjunto no varía a lo largo del período.

Los clientes que llegan en el período  $t$  eligen entre las alternativas disponibles  $S_t$  de acuerdo a un modelo MNL. Bajo este modelo, la probabilidad de compra de un

producto se define basada en un vector de preferencias  $\mathbf{v} \in \mathcal{R}^n, \mathbf{v} > 0$ , donde cada escalar  $v_j$  representa el “peso” que se le asigna a cada producto, como función de la preferencia del cliente. Estas preferencias se asumen uniformes para todos los clientes. Sin embargo, abrimos un paréntesis para destacar que en general no es cierta esta suposición para todos los clientes que lleguen al hotel sino que seguramente vamos a poder detectar distintos *segmentos*. Estos segmentos representan distintos tipos de clientes (viajeros de negocios, familias de vacaciones, etc.) para los cuales sí vamos a poder asumir que las preferencias son uniformes. Para resolver esto, es necesario dividir los datos disponibles y estimar por separado los parámetros para cada una de estas particiones. Volveremos sobre este tema en la sección 2.5 y lo profundizaremos en la sección 2.5.3.

Además, debemos definir una preferencia de no compra, cuyo valor es fijado en forma arbitraria por nosotros en  $v_0 = 1$ . De esta manera, las preferencias por los productos se escalan en relación a este valor de referencia. Estas magnitudes determinan las probabilidades de compra de la siguiente manera: sea  $P_j(S, \mathbf{v})$  la probabilidad de que un cliente elija comprar el producto  $j \in S$  cuando las alternativas disponibles están dadas por  $S$  y el vector de preferencias es  $\mathbf{v}$ . Entonces,

$$P_j(S, \mathbf{v}) = \frac{v_j}{\sum_{i \in S} v_i + 1}$$

Si  $j \notin S$ , entonces  $P_j(S, \mathbf{v}) = 0$ .

A su vez, se define la probabilidad de no compra, es decir la probabilidad de que el cliente dadas las alternativas elija comprar a un competidor o no comprar nada, a la que se denota  $P_0(S, \mathbf{v})$  y está definida de la siguiente manera:

$$P_0(S, \mathbf{v}) = \frac{1}{\sum_{i \in S} v_i + 1}$$

Se puede pensar que la opción de no compra es otro producto (etiquetado con el número 0) cuya preferencia es  $v_0 = 1$  y que está siempre disponible.

Finalmente, se debe tener en cuenta que si en un período no se observan compras puede deberse a dos causas distinta: o bien llegó un cliente y decidió no comprar, o bien no llegó ningún cliente. Para separar qué fracción de estos períodos corresponde a uno y otro caso, el modelo requiere que se provea la participación en el mercado (o *market share*) de la firma, que se denota con la letra  $s$ .

## 2.4. El método *expectation-maximization*

El objetivo de la estimación es el de encontrar el valor de los parámetros del modelo subyacente dados por: las tasas de arribo  $\lambda_t$  para cada período  $t$  y el vector de preferencias  $\mathbf{v}$ . Es importante notar que la dificultad más importante de este problema subyace en que no tenemos observaciones de los clientes que llegaron y eligieron no comprar, como así tampoco de aquellos que compraron una opción alternativa dado que la que hubieran preferido no estaba disponible. Para resolver este problema, necesitamos maximizar la función de verosimilitud para información incompleta definida de la siguiente manera (ver [VvRR08]):

$$\mathcal{L}_I = \prod_{t=1}^T \left( \mathbb{P}(m_t \text{ compras en el período } t | \mathbf{v}, \lambda) \frac{m_t!}{z_{1t}! z_{2t}! \dots z_{nt}!} \prod_{j \in S_t} \left[ \frac{P_j(S_t, \mathbf{v})}{\sum_{i \in S_t} P_i(S_t, \mathbf{v})} \right]^{z_{jt}} \right)$$

Por su parte, la cantidad de clientes que compran en el período  $t$ ,  $m_t$ , vienen dados por una realización de una variable Poisson con media  $\lambda_t \sum_{i \in S_t} P_i(S_t, \mathbf{v})$ , con lo que:

$$\mathbb{P}(m_t \text{ compras en el período } t | \mathbf{v}, \lambda) = \frac{[\lambda_t \sum_{i \in S_t} P_i(S_t, \mathbf{v})]^{m_t} e^{-\lambda_t \sum_{i \in S_t} P_i(S_t, \mathbf{v})}}{m_t!}$$

Dada la complejidad de esta función, su maximización (o la de su logaritmo) no es un enfoque muy práctico. Para sortear este problema, Vulcano et al. [VvRR08] propone descomponer el número (aleatorio) de compras  $Z_{it}$  en las compras que correspondían a la demanda primaria  $X_{it}$  y a la demanda por sustitución  $Y_{it}$ :

$$Z_{it} = X_{it} + Y_{it}$$

La demanda primaria  $X_{it}$  es el número de clientes en el período  $t$  que preferían el producto  $i$  por sobre cualquier otro y la demanda por sustitución  $Y_{it}$  es la cantidad de clientes en el período  $t$  que eligieron comprar el producto  $i$  dado que el producto que hubieran elegido no estaba disponible.  $N_j$  representa la demanda primaria del producto  $j$ . Es decir, la cantidad de clientes que hubieran elegido el producto  $j$  si todos los productos hubieran estado disponibles. Por lo tanto,  $N_j = \sum_{t=1}^T X_{jt}$ . Por su parte,  $N_0$  representa la cantidad de clientes que eligen no comprar aún cuando todos los productos hubiesen estado disponibles. Gracias a plantear el problema

como una estimación de los parámetros  $\mathbf{v}$  y  $\lambda$  como un problema de estimación con observaciones incompletas de la demanda primaria  $X_{it}$ , el logaritmo de la función de verosimilitud se torna muy simple y con una fórmula cerrada para su máximo (ver [VvRR08]):

$$L(\mathbf{v}) = \sum_{j=1}^n N_j \ln \left( \frac{v_j}{\sum_{i=1}^n v_i + 1} \right) + N_0 \ln \left( \frac{1}{\sum_{i=1}^n v_i + 1} \right)$$

Sin embargo, dado que no se observa  $N_j$  de forma directa se utiliza el método de *expectation-maximization*(EM) de Dempster et al.[DLR77] para estimar los parámetros. El mismo consiste en iterar entre dos pasos: comenzando de valores iniciales arbitrarios de los parámetros, calcula el valor esperado condicional a dichos parámetros de la función ( $E[L(v)|\hat{\mathbf{v}}]$ ) –el paso *expectation*(E)–, y luego maximiza la nueva función para generar nuevos estimadores – el paso *maximization*(M). Este procedimiento continúa hasta que los valores convergen con precisión arbitraria. El algoritmo completo se presenta en la figura 2.1.

Notar que **lambda** no aparece dentro del problema de optimización, pero se puede obtener su valor con sólo sumar los valores de las demandas primarias  $X_{it}$  para todo producto  $i$ , inclusive la alternativa de no compra:

$$\hat{\lambda}_t = \hat{X}_{0t} + \sum_{i=1}^n \hat{X}_{it}$$

[Inicialización]: Dada la participación en el mercado  $s$ , asignar  $r := (1 - s)/s$ . Para todo producto  $j$  y período  $t$ , asignar  $X_{jt} := z_{jt}$ , con  $X_{jt} := 0$  si  $j \notin S_t$ . Luego, inicializar las variables  $N_0, N_1, \dots, N_n$ , de la siguiente manera:

$$N_j := \sum_{t=1}^T X_{jt}, \quad j = 1, \dots, n, \quad N_0 := r \sum_{j=1}^n N_j, \quad X_{0t} := N_0/T, \quad \text{y} \quad v_j := N_j/N_0, \quad j = 1, \dots, n.$$

**Repetir**

[Paso E]:

Para  $t := 1, \dots, T$

Para  $j := 1, \dots, n$

Si  $j \notin S_t$ , entonces asignar

$$X_{jt} := \frac{v_j}{\sum_{i=1}^n v_i + 1} \frac{\sum_{h \in S_t} v_h + 1}{\sum_{h \in S_t} v_h} \sum_{h \in S_t} z_{ht}$$

sino (i.e.,  $j \in S_t$ ), entonces asignar

$$Y_{jt} := \frac{\sum_{h \notin (S_t \cup \{0\})} v_h}{\sum_{i=1}^n v_i + 1} z_{jt}, \quad \text{and} \quad X_{jt} := z_{jt} - Y_{jt}.$$

Fin "Si"

Fin "Para"

Asignar

$$X_{0t} := \frac{1}{\sum_{i=1}^n v_i} \sum_{i=1}^n X_{it}, \quad \text{and} \quad Y_{0t} := \frac{1}{\sum_{i \in S_t} v_i + 1} \sum_{h \notin S_t \cup \{0\}} X_{ht}.$$

Fin "Para"

[Paso M]:

Asignar  $N_0 := \sum_{t=1}^T X_{0t}$ .

Para  $j := 1, \dots, n$

Asignar  $N_j := \sum_{t=1}^T X_{jt}$ .

Asignar  $v_j := N_j/N_0$ .

Fin "Para"

**hasta** que se alcance el criterio de parada.

Figura 2.1: Algoritmo EM para estimar demanda primaria.

## 2.5. Procedimiento

Como mencionamos, nuestro primer trabajo fue el de construir un modelo de demanda para los hoteles del dataset. Cabe destacar que de los datos sobre los que trabajamos, sólo utilizamos aquellos que correspondían a reservas hechas por exactamente una noche. Esto se debe a que estudiamos cada día de check-in en forma independiente de los demás. Por otra parte, notamos que existen registros de reservas que indican una lista de alternativas, pero en las que ningún producto fue comprado. Estos registros también fueron descartados. Por último, utilizamos el

*Merge Indicator* (ver sección 2.1) para descartar los registros que no contaran con una lista de alternativas.

A continuación nos dedicamos a estimar los parámetros del modelo de demanda. A fin de poder aplicar el algoritmo EM, primero fue necesario definir los productos (sección 2.5.1) y los períodos (sección 2.5.2) de nuestro escenario para poder modelarlo en los términos que se explicó en la sección 2.3. Una vez establecidas ambas definiciones, pudimos construir los datos de entrada del algoritmo EM (matriz de compras  $Z$ , disponibilidad de los productos en cada período  $t$  representado por el conjunto  $S_t$ , etc.) para cada hotel presente en el dataset y así estimar los parámetros  $v_j$  y  $\lambda_t$  del modelo.

Sin embargo, antes de poder calcular estos parámetros, fue necesario tomar en cuenta un hecho importante: la distribución de las preferencias no es uniforme entre todos los clientes, sino que existen distintos tipos de clientes (viajeros de negocios, familias de vacaciones, etc.) con comportamientos diversos, pero que se pueden agrupar en *segmentos*. Este paso se explica en la sección 2.5.3. Finalmente, utilizando el algoritmo EM, obtuvimos estimaciones tales para los parámetros del modelo, que logran explicar en buena medida los datos observados (ver los resultados obtenidos en la sección 2.6). En todos los casos utilizamos un valor de participación en el mercado  $s = 0,2$ .

Por último, consideramos un hecho común de los modelos MNL y es que el peso  $v_j$  con el que se pondera la preferencia del producto  $j$  proviene de la ponderación de distintos atributos de los productos (precio, fecha, etc.) a los que el cliente dará mayor o menor importancia. En la sección 2.7 hacemos algunas consideraciones de cómo deberíamos obtener las preferencias, esta vez, descompuestas según atributos.

### 2.5.1. Definición de Producto

Mencionamos en la sección 2.1 que en el dataset se encuentran registradas todas las alternativas entre las que el cliente pudo elegir (incluida la alternativa elegida) y que estas alternativas constaban de diversos atributos, entre los que podíamos distinguir: el tipo de habitación, el tipo de tarifa, el largo de la estadía, etc. Debimos definir, entonces, cuál iba a ser el conjunto de productos que consideraríamos en la estimación y de qué manera se correspondían las alternativas registradas con los productos en consideración. Para ser coherentes con el trabajo de Ferguson et al.[FGMN10] donde se analiza el mismo dataset que utilizamos en este trabajo, optamos por definir los productos por el espacio físico asociado al producto. Es decir, distinguimos por el tipo de habitación, y no consideramos adicionales como ser servicios accesorios,

visitas, tickets para eventos, si es fumador o no fumador, etc. Tampoco distinguimos los productos por el largo de la estadía. Los tipos de habitaciones de cada hotel -que son entonces, por definición, los tipos de producto disponibles- son los que se presentan en la tabla 2.1. Observar que cada uno de estos nombres (“King Room 3”) corresponden a un espacio físico con ciertas comodidades cuyos detalles desconocemos. Sin embargo, sabemos que constituyen alternativas distinguibles por parte de los clientes. Asimismo, no asumimos ninguna relación entre tipos de habitación de distintos hoteles que tienen la misma denominación.

2 Double Beds Room 1	2 Queen Beds Room 1	2 Double Beds Room 1
King Room 1	2 Queen Beds Room 2	2 Double Beds Room 2
King Room 2	King Room 1	King Room 1
King Room 3	King Room 2	King Room 3
King Room 4	King Room 3	Regular Bed Room 1
Queen Room 1	Standard Room	Standard Room
Queen Room 2	Suite 1	(c) Hotel 3
Special Type Room 1	Suite 2	
Suite 1	(b) Hotel 2	
Suite 2		
(a) Hotel 1		
2 Queen Beds Room 1	2 Double Beds Room 1	
2 Queen Beds Room 2	2 Double Beds Room 2	
King Room 1	2 Double Beds Room 3	
King Room 2	Double Bed Room 1	
Standard Room	King Room 1	
Suite 1	King Room 2	
Suite 2	Suite 1	
(d) Hotel 4	Suite 2	
	(e) Hotel 5	

Tabla 2.1: Tipos de habitación/Productos por hotel

### 2.5.2. Definición de Período

Tal como se define en el modelo, existen  $T$  períodos indexados con  $t = 1, 2, \dots, T$  sin ninguna restricción respecto de la duración o el orden los mismos. En cada período  $t$  llegan  $A_t$  clientes, donde  $A_t$  es una variable aleatoria con distribución Poisson de media  $\lambda_t$ . Cada uno de estos clientes elige entre los productos disponibles (es decir, los productos pertenecientes a  $S_t$ ), o bien, elige no comprar nada.

Ahora bien, el único requisito respecto de estos períodos, es que la lista de alternativas disponibles no varíe en el transcurso de un mismo período (todos los  $A_t$

clientes que llegan en el período  $t$  observan la lista  $S_t$ ). Con lo cual, esta hipótesis del modelo fue algo a tener en cuenta al definir los períodos.

En el dataset (ver la sección 2.1 para una descripción del mismo), disponíamos del día en que efectuó cada reserva ( $t^B$ ) y el día para el cual se reservó –fecha de check-in– ( $t^C$ ). Dado esto, definimos un período como un par ordenado de ambas fechas,  $t = (t^B, t^C)$ . Esta definición, es la más natural, ya que contempla que las disponibilidades para cierto día de ingreso al hotel van a ser muy distintas si se reserva con 2 días de anticipación, que si se reserva con 10. Análogamente, contempla que las alternativas observadas en un día en particular en el que se intenta efectuar la reserva, pueden ser distintas en función del día para el que se busca reservar.

Ahora bien, con esta definición en mano, quedaba un problema por resolver, y es que nadie garantizaba que las listas de alternativas para las reservas hechas dentro de cada uno de estos períodos fueran siempre iguales. Con lo cual, en una primera aproximación, consideramos que podía suceder que hubiera listas de alternativas distintas para reservas hechas en un mismo período. Por lo tanto, optamos por construir una lista de alternativas única para el período  $t$  de la siguiente manera. Denotamos  $B(t)$  al conjunto de las listas de alternativas para las reservas hechas durante el período  $t$ . Definimos entonces la lista de alternativas  $S_t$  como la unión de todas las listas de alternativas observadas en ese período:

$$S_t = \bigcup_{R_b \in B(t)} R_b$$

Asimismo, definimos la siguiente medida para medir cuánto se parecen las listas de alternativas  $R_b$  que unimos en  $S_t$ :

$$w_t = \frac{\left| \bigcap_{R_b \in B(t)} R_b \right|}{\left| \bigcup_{R_b \in B(t)} R_b \right|}$$

Obsérvese que cuanto más similares entre sí son los conjuntos  $R_b$ ,  $w_t$  más se acerca a 1. Mientras que, cuanto más disímiles son, más se acerca a 0. En particular, en los puntos extremos, se observa que si todos los conjuntos son iguales,  $w_t = 1$ ; mientras que si hay al menos un par de conjuntos disjuntos,  $w_t = 0$ . Por ejemplo, supóngase que en el período  $t_1$  se observan 3 reservas, cada una de las cuales, posee como lista de alternativas  $\{A, B, C\}$ ,  $\{A, C, D\}$ ,  $\{A, D\}$ . En este caso  $w_{t_1} = \frac{|\{A\}|}{|\{A, B, C, D\}|} = \frac{1}{4}$ . Supongamos ahora un período  $t_2$  con igual cantidad de reservas, pero con listas de alternativas  $\{A, B, C\}$ ,  $\{C, D\}$ ,  $\{A, D\}$ . Dado que  $\{A, B, C\} \cap \{C, D\} = \emptyset$ ,  $w_{t_2} = 0$ .



Calculamos esta medida sobre todos los períodos considerados en el dataset, para todos los hoteles, y los resultados son los que se vuelcan en la figura 2.2. Como se puede observar, los valores están casi siempre en 1 o muy cerca. Con lo cual, se puede pensar que la hipótesis de que la lista de productos disponibles no varía a lo largo de un período es, en general, correcta. Sin embargo, dado que aquí se agregaron los datos de todos los hoteles, hicimos también una prueba comparando esta magnitud diferenciando por los distintos hoteles, y los resultados son los que se observan en el boxplot de la figura 2.3. Acá se ve como el valor de  $w_t$  está fuertemente pegado al 1, indistintamente de cuál sea el hotel en consideración. Este buen comportamiento se puede atribuir no sólo al hecho de que los conjuntos de alternativas para un mismo período no fueran muy distintos entre sí, sino también al hecho de que la cantidad de reservas por período fuera baja (ver figura 2.4), y por lo tanto, si por ejemplo hubiera una sola reserva,  $w_t$  valdría, por definición, 1. Esto último se ve confirmado por la figura 2.5 en donde se observa la distribución de  $w_t$  en función de la cantidad de reservas del período, en donde se puede apreciar cómo los valores se mantienen relativamente confiables hasta con 4 reservas por período, pero a partir de la quinta, se tornan más inestables. Sin embargo, por lo que refleja la figura 2.6, pudimos concluir que alrededor del 80 % de los datos se mantiene en estos márgenes relativamente aceptables. Con lo cual, optamos por eliminar las reservas pertenecientes a listas de alternativas que no fueran iguales a la lista que más aparece a lo largo del período, para que estos datos no introduzcan ruido en la estimación.

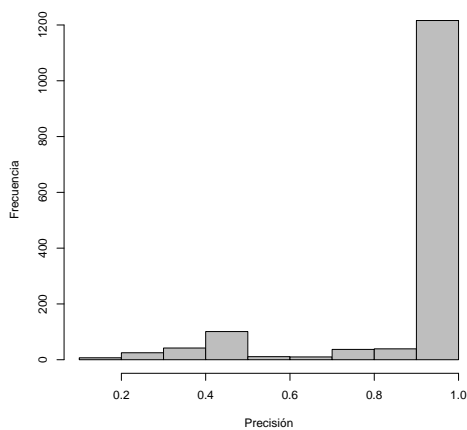


Figura 2.2: Histograma de la medida  $w_t$  tomada sobre todo el espectro de períodos del dataset para todos los hoteles

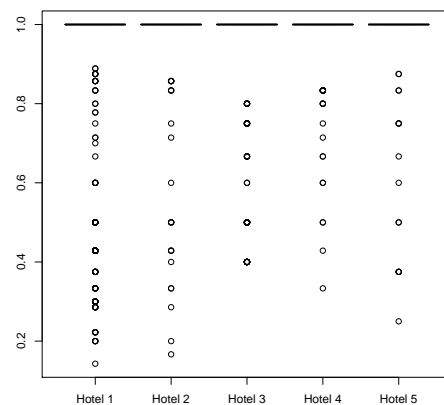


Figura 2.3: Box-plot de valores de  $w_t$  para todos los períodos observables de un determinado hotel

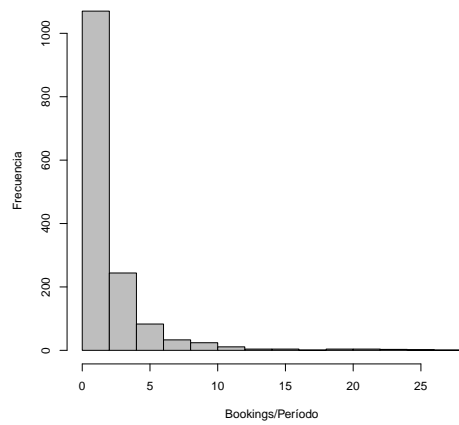


Figura 2.4: Histograma de la cantidad de reservas observadas en un determinado período

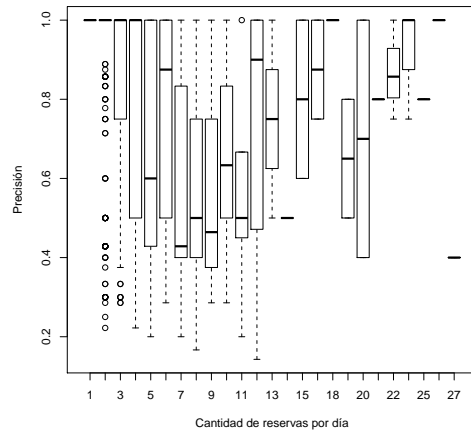


Figura 2.5: Box-plot de  $w_t$  en función de la cantidad de bookings  $B(t)$  del período

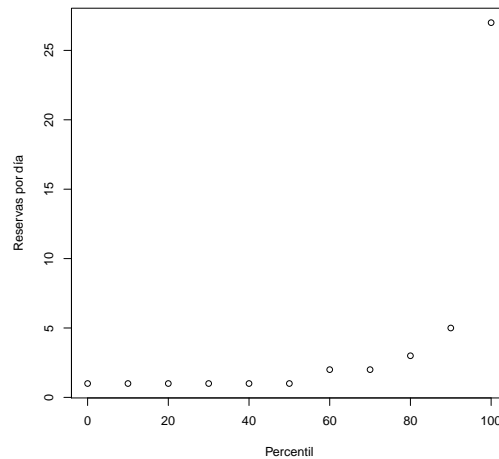


Figura 2.6: Distribución de las reservas por día

### 2.5.3. Segmentación de los datos

Como ya se explicó en la Sección 2.5, los datos tuvieron que ser segmentados dado que las preferencias de los clientes no eran homogéneas. Sin embargo, podemos agrupar tipos de clientes con comportamientos similares en *segmentos*, para los cuales encontraremos distintos vectores de preferencias  $\mathbf{v}$ . Por ejemplo, es de esperar que en los fines de semana lleguen fundamentalmente turistas que pueden traer a sus familias y preferir habitaciones de 4 camas, mientras que en la semana quizás lo más predominante sean viajeros de negocios, que preferirán habitaciones simples o dobles.

Por lo tanto, a fin de capturar las preferencias de estos distintos tipos de clientes, tuvimos que decidir de qué manera íbamos a dividir los datos de reservas.

En primer lugar, decidimos efectuar esta división sobre dos dimensiones: día de la semana y tiempo de antelación en la reserva. Ambas magnitudes sirvieron para caracterizar a los distintos tipos de clientes. La primera ya la ilustramos antes, con el ejemplo de los turistas en el fin de semana. Algo similar sucede con el tiempo de antelación: mientras que una familia de vacaciones planifica con cierta antelación su viaje, el viajero con motivos de negocios probablemente haya decidido el viaje con mucha menos anticipación.

Por lo tanto, considerando estos dos atributos de la reserva, comenzamos a buscar los cortes que mejor ajustaran a los datos (por ejemplo, un posible corte son las reservas hechas para un día de semana hechas con 5 a 10 días de antelación). Esto lo llevamos a cabo elaborando estimaciones independientes con los datos de compras restringidos a una determinada partición para luego comprobar el ajuste de la estimación obtenida. En esta búsqueda, no hicimos una exploración exhaustiva de todas las posibles particiones, sino que mediante una interfaz gráfica desarrollada para facilitar este proceso, probamos distintos particionados de los datos en forma manual. En cada caso, verificamos qué tan bien ajustaba nuestro modelo los datos observados mediante un test de  $\chi^2$  y seleccionamos las particiones que mejoraran el estadístico de dicho test. El test utiliza como hipótesis nula que los clientes compran siguiendo un modelo MNL con los parámetros estimados. Bajo esta hipótesis, se comparó la cantidad de clientes que se esperaba que compraran el producto  $j$  en el período  $t$  con las observaciones de compras  $z_{jt}$  para el mismo período y producto. Sin embargo, para que el test fuera más significativo agrupamos todos los períodos en 3 períodos de igual longitud.

A continuación, comparando estos mismos valores esperados y observados, calculamos también el RMSE (*root mean square error*). Por último, dado que el test de  $\chi^2$  utiliza una aproximación de la distribución del estadístico que no es muy precisa para valores esperados bajos, confirmamos nuestros resultados calculando el p-valor con simulación monte carlo de 10000 observaciones.

## 2.6. Resultados de la estimación

Como resultado del proceso de estimación obtuvimos los segmentos que se detallan en la tabla 2.2. Podemos observar que en general, el modelo ajusta muy bien para las particiones obtenidas. Sin embargo, el caso más conflictivo fue el del Hotel 3, para el cual debimos construir un número importante de particiones y aún así el

ajuste no parece ser muy robusto. Probablemente esto se deba a que sólo mediante los días de check-in y la antelación no estamos capturando apropiadamente los distintos segmentos y, para este caso en particular, podría existir alguna otra dimensión que segmente mejor los datos.

Con esto, obtuvimos las estimaciones de los parámetros del modelo de demanda para cada una de las particiones. En la tabla 2.3, se presenta a modo de ejemplo los resultados correspondientes a las preferencias del segmento con día de check-in de lunes a viernes, para reservas hechas con una antelación de 0 a 1 días. En esta tabla se pueden observar la cantidad de compras ( $z_j$ ) registradas para cada producto, la demanda primaria ( $N_j$ ), el *asymptotic standard error* (ASE) y la preferencia de los clientes pertenecientes al segmento por cada producto ( $v_j$ ).  $v_j$  y  $N_j$  se encuentran vinculadas por la expresión  $v_j = N_j/N_0$ , donde  $N_0$  es el número de clientes con la no-compra como su opción primaria. Por su parte, el orden de magnitud de los valores de  $v_j$  está estrictamente relacionado con el valor de la preferencia de no compra  $v_0$  que fijamos en 1 y la participación en el mercado  $s$  según la ecuación:

$$s = 1 - \frac{1}{1 + \sum_{j=1}^n v_j}$$

Merece destacarse que para ciertos casos, como el “King Room 3”, la demanda primaria es muy superior a la cantidad de compras observadas. Este resultado implica que una gran proporción de clientes que tenían como primera opción este tipo de habitación no la encontraron disponible, por lo que debieron adquirir un reemplazo o directamente optaron por no comprar. En otros casos, como la “King Room 4”, la demanda primaria es inferior a la cantidad de compras. Esto quiere decir que al menos una parte de las compras de este producto no provinieron de clientes que lo tuvieran como primera opción, sino de clientes que encontraron agotada su primera opción y se inclinaron por este producto a modo de reemplazo. Esto es consistente con el modelo de demanda primaria sumada a la demanda por sustitución con el que se explican las compras que se observan.

El resto de los resultados se incluye en el apéndice A.

Hotel	Partición	$\chi^2$	$\chi^2$ MC	RMSE
	Lunes a Jueves reservas hechas de 0 a 1 días del día de check-in	0.93	0.92	1.02
Hotel 1	Lunes a Jueves reservas hechas de 2 días en adelante del día de check-in	0.84	0.84	2.20
	Viernes y Sábado en todo el horizonte	0.94	0.94	1.27

2.6. RESULTADOS DE LA ESTIMACIÓN

	Domingo en todo el horizonte	0.98	0.98	0.78
Hotel 2	Cualquier día en todo el horizonte	0.99	0.99	0.95
	Lunes a Jueves de 0 a 1 día del día de check-in	0.42	0.47	4.5
	Lunes a Jueves de 2 a 8 días del día de check-in	0.60	0.63	1.62
	Lunes a Jueves de 9 a 19 días del día de check-in	0.91	0.93	0.90
	Lunes a Jueves de 20 a 27 días del día de check-in	0.79	0.79	0.95
	Lunes a Jueves de 28 días en adelante del día de check-in	0.34	0.40	1.24
Hotel 3	Viernes y Sábado a 0 días del día de check-in	0.29	0.3	1.55
	Viernes y Sábado a 1 día del día de check-in	0.80	0.86	0.93
	Viernes y Sábado de 2 a 4 días del día de check-in	0.60	0.64	1.2
	Viernes y Sábado de 5 a 10 días del día de check-in	0.77	0.8	1.13
	Viernes y Sábado de 11 días en adelante del día de check-in	0.53	0.53	2.1
	Domingo en todo el horizonte	0.86	0.84	1.85
Hotel 4	Lunes a Jueves en todo el horizonte	0.66	0.68	2.16
	Viernes, Sábado y Domingo en todo el horizonte	0.97	0.98	1.34
	Lunes a Jueves de 0 a 2 días del día de check-in	0.59	0.59	0.83
Hotel 5	Lunes a Jueves de 3 días en adelante del día de check-in	1.00	1.00	0.48
	Viernes, Sábado y Domingo en todo el horizonte	0.94	0.94	1.74

Tabla 2.2: División en segmentos del horizonte de booking

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	1	16.5923	0.00528846	0.0221147
King Room 3	13	100.16	0.0110905	0.133496
2 Double Beds Room 1	8	18.682	0.00559253	0.0248999
Queen Room 2	1	5.33144	0.0030521	0.00710591
Special Type Room 1	5	10.1081	0.00417097	0.0134724
King Room 4	28	23.8318	0.00626306	0.0317637
Suite 1	4	4.53106	0.00281722	0.00603913
Suite 2	9	8.33417	0.003798	0.011108

Tabla 2.3: Hotel 1. Lunes a Jueves, reservas hechas de 0 a 1 día previo al check-in.

## 2.7. Descomposición de las preferencias

Con este procedimiento obtuvimos el vector de preferencias  $\mathbf{v}$ . En su forma más general, el modelo MNL asume que cada escalar  $v_j$  que pondera la preferencia por el producto  $j$  puede escribirse como una función de la forma  $v_j = e^{u_j}$  donde  $u_j = \beta^\top \mathbf{x}_j$  es la utilidad media de la alternativa  $j$ ,  $\mathbf{x}_j$  es un vector de atributos de la alternativa  $j$ , y  $\beta$  es un vector de coeficientes que asigna una utilidad a cada atributo. Expresado de esta manera, el problema consiste en estimar el vector  $\beta$ . Conocer dicho vector puede ser de importancia para evaluar cómo impactan distintas propiedades de los productos (el precio, por ejemplo) en las preferencias de los clientes.

En el trabajo de Vulcano et al. [VvRR08] se propone buscar los valores de  $\beta$  que mejor aproximen el sistema  $u_j = \beta^\top \mathbf{x}_j, j = 1 \dots n$ . Dado que este sistema probablemente se encuentra sobredeterminado, se propone aproximarlos mediante cuadrados mínimos. Aquí queremos agregar algunas precisiones al respecto. Sabemos que tenemos  $n$  productos y supongamos que elegimos estudiar  $m$  atributos. De esta manera, la matriz  $X = [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \dots, \mathbf{x}_n^\top]$  tiene  $n$  filas y  $m$  columnas. Suponemos a las columnas de  $X$  linealmente independientes, por lo que  $\text{rango}(X) = m$ . En caso contrario, debemos eliminar todo atributo redundante.

Entonces, como decíamos antes, si  $n = m$ , entonces  $X\beta = \mathbf{u}$  está determinado y el sistema tiene solución única. Si  $m > n$  el sistema es indeterminado, y bastará reducir el número de atributos cuyo peso queremos estimar. En cambio, si  $m < n$ , tenemos dos posibilidades:

- $\text{rango}([X|\mathbf{u}]) = m$ : en este caso el sistema sigue estando determinado y tenemos solución única, con lo cual, no nos preocupa.
- $\text{rango}([X|\mathbf{u}]) > m$ : en este caso tenemos un sistema incompatible, pero como decíamos antes, se puede encontrar el  $\beta$  que minimice  $\|X\beta - \mathbf{u}\|_2$ . Sin embargo, en este proceso estamos perdiendo información, dado que estamos proyectando el vector  $\mathbf{u} \in \mathbb{R}^n$  sobre un espacio de menor dimensión ( $\mathbb{R}^m$ ). Por esta razón, los nuevos pesos de preferencia de los productos  $v'_j = e^{\beta^\top \mathbf{x}_j}$  van a ser distintos de los originales.

Por lo tanto, llegado el caso de tener un sistema indeterminado, utilizar mínimos cuadrados para obtener los pesos de los atributos no es quizás la mejor estrategia. En su lugar proponemos el siguiente procedimiento:

1. Agregar para cada producto  $j$ , un atributo  $I_j$  que vale 1 para el producto  $j$ , y 0 para todos los demás. A este tipo de atributos se los denomina ASC

(*alternative specific constant*). Este paso se puede ver como adjuntar a  $X$  la matriz identidad y de esta manera, nos garantizamos  $\text{rango}(X) = n$ .

2. Eliminar las columnas correspondientes a los ASC agregados en el paso anterior que sean combinación lineal de otras columnas de la matriz de atributos  $X$ , hasta que ésta quede con  $n$  columnas.

Mediante este nuevo conjunto de atributos podemos resolver directamente el sistema  $X\beta = \mathbf{u}$ . Así, podemos siempre reconstruir en forma exacta los pesos relativos a la preferencia de cada producto a partir de los pesos de los atributos y sus respectivos valores.





## Capítulo 3

# Optimización

En este capítulo abordamos el segundo módulo de un sistema de *Revenue Management*: la optimización de la política para el control de capacidad. Como se mencionó en la sección 1.1.5 el problema operativo consiste en decidir qué productos ofrecer dada la capacidad remanente y el tiempo restante a la fecha de check-in. Para llevar a cabo esta tarea, internalizamos el modelo de demanda descrito en el capítulo anterior en distintos algoritmos de optimización. El capítulo se encuentra organizado de la siguiente manera. Primero, en la sección 3.1, explicamos qué tipo de modelo utilizamos y las razones que nos llevaron a tomar dicha elección. Luego, describimos el modelo en la sección 3.2, para proseguir explicando, en la sección 3.3, cómo lo aplicamos para modelar nuestro problema particular. En la sección 3.4, abordamos los distintos algoritmos que consideramos para resolver el problema. Estudiamos en profundidad uno de estos algoritmos, CDLP, en la sección 3.5. Por último, en la sección 3.6 presentamos los experimentos computacionales que llevamos a cabo para comparar los distintos métodos.

### 3.1. Consideraciones preliminares

Siguiendo los lineamientos de Ferguson et al.[FGMN10], consideramos que un cliente que adquiere un cierto tipo de habitación, quedaría también satisfecho utilizando un recurso (es decir, una habitación) de igual o mayor calidad. Para esto, se asume que existe una relación que indica si un tipo de recurso es sustituto –es de mayor o igual calidad– de otro. Dicha relación (denominada *Demand Satisfaction* o DS) es observable, determinística, estática y no depende de las preferencias individuales de cada cliente en particular. Además, es transitiva.

Utilizando esta relación DS, permitimos que en caso de agotamiento de un recurso, pudiera utilizarse otro que también satisficiera al cliente para vender el producto. Por ejemplo, si un cliente compró una habitación *standard* pero ésta ya no estaba disponible, era posible entregarle una *suite* por el mismo precio de la habitación adquirida. Esto genera una interdependencia entre los recursos que resolvimos utilizando un modelo de red (ver sección 1.1.9). Por su parte, seguimos utilizando un modelo de elección para modelar la demanda (ver sección 1.1.6.2).

## 3.2. Descripción del modelo

La red contiene  $m$  recursos y provee  $n$  productos (un producto se define como una combinación de un recurso y una clase de tarifa). Las capacidades iniciales se denotan  $\mathbf{c} = (c_1, \dots, c_m)$ . El conjunto de productos se denota  $N = \{1, \dots, n\}$ . Cada producto  $j$  tiene un precio asociado  $r_j$ . Se define una matriz de incidencia  $A = [a_{ij}]^{m \times n}$ . Si el recurso  $i$  es utilizado por el producto  $j$ ,  $a_{ij} = 1$ ; de lo contrario,  $a_{ij} = 0$ . El vector columna  $A_j$  es el vector de incidencia del producto  $j$  y el vector fila  $A^i$  es el vector de incidencia para el recurso  $i$ . Haciendo un pequeño abuso de notación, vamos a decir que  $i \in A_j$  o  $j \in A^i$  para decir que el producto  $j$  consume el recurso  $i$ .

Se consideran períodos discretos indexados hacia adelante por un índice  $t$ . Se asume un sólo arribo de un cliente como máximo por período. Es decir, el tiempo está subdividido en períodos lo suficientemente breves de modo que la probabilidad de que ocurra más de un arribo en un período es despreciable. Esto es consistente con un modelo Poisson, muy común en la literatura. Los períodos se agrupan en  $K$  ventanas de tiempo de longitud  $T_k$  en las cuales la tasa de arribos es constante. La probabilidad de arribo en un período  $t$  que se encuentra en la ventana  $k$  se denota  $\lambda_k$ . El horizonte de reservas tiene una longitud total  $\sum_{k=1}^K T_k = T$ . Notar que estos períodos no mantienen relación con los definidos en la sección 2.3.

La política de control viene dada por el conjunto de productos que se ofrecen (es decir, que se hacen disponibles) en cada período. A este subconjunto de productos abiertos  $S \subseteq N$  se lo denomina *offer set*. Dado un *offer set*  $S$ , un cliente que llega dentro de la ventana de tiempo  $k$  elige el producto  $j \in S$  con probabilidad  $P_j^k(S)$ . Si  $j \notin S$ ,  $P_j^k(S) = 0$ .  $P_0^k(S)$  denota la probabilidad de no compra. Y, por probabilidad total,  $\sum_{j \in S} P_j^k(S) + P_0^k(S) = 1$ .

El estado de la red viene dada por la capacidad disponible denotada por el vector  $\mathbf{x} = (x_1, \dots, x_m) \geq 0$ . El problema operativo entonces consiste en encontrar para cada instante el *offer set* que maximice el ingreso esperado dado el estado de la red

$\mathbf{x}$  en el instante  $t$ .

### 3.2.1. Formulación basada en programación dinámica

Este problema de decisión se puede expresar como un problema de programación dinámica. Se define la función de valor, denotada  $V_t(\mathbf{x})$ , como el ingreso óptimo esperado que se puede obtener desde el período  $t$  hasta el instante final  $T$  dado que el vector de capacidades remanentes al momento  $t$  es  $\mathbf{x}$ . La ecuación de Bellman (ver [BD66]) es entonces

$$\begin{aligned} V_t(x) &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda_k P_j^k(S) (r_j + V_{t+1}(x - A_j)) + \right. \\ &\quad \left. (\lambda_t P_0^k(S) + 1 - \lambda_t) V_{t+1}(x) \right\} + V_{t+1}(x) \\ &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda_k P_j^k(S) (r_j - (V_{t+1}(x) - V_{t+1}(x - A_j))) \right\} + V_{t+1}(x) \end{aligned} \quad (3.1)$$

donde  $k$  es la ventana de tiempo que corresponde al período  $t$ . La segunda ecuación proviene de utilizar que  $\sum_{j \in S} P_j(S) + P_0(S) = 1$ . No se permite *overbooking*, con lo cual  $x - A_j \geq 0$  para todo  $j \in S$ . Las condiciones de borde son

$$V_t(0) = 0, t = 1, \dots, T$$

$$V_{T+1}(x) = 0, \forall x \geq 0$$

### 3.3. Instanciación del modelo

Para optimizar el control de capacidad asumimos disponer de los siguientes datos que forman parte de la información básica del hotel:

- $H$  es el conjunto de los tipos de habitaciones.
- $\hat{r}_j : j \in H$  es el precio de la habitación  $j$ .
- $\hat{c}_j : j \in H$  es la capacidad disponible para las habitaciones de tipo  $j$ .

- $U_j \subseteq H : j \in H$  es el conjunto de tipos de habitación que pueden substituir a una habitación de tipo  $j$ . Este conjunto se define en base a la relación DS (ver sección 3.1).

Por otra parte, disponíamos también de las estimaciones obtenidas del modelo de demanda:

- $\hat{v}_j^t : j \in H$  es la preferencia asociada al producto  $j$  en el período  $t$ . La dependencia de las preferencias del período proviene de haber considerado que existían distintos segmentos de clientes según la antelación con la cual efectuaban la reserva (ver sección 2.5.3).
- $\hat{\lambda}_{(t^B, t^C)}$  es la tasa de arribos en el día  $t^B$  de clientes que desean reservar para el día  $t^C$ .

Como ya mencionamos, optamos por utilizar un modelo de elección, donde un cliente que llega en un cierto instante, observa la lista de habitaciones disponibles y elige cuál adquirir o bien elige no comprar ninguna. Sin embargo, queríamos desacoplar la clase de tarifa que el cliente vería del recurso que efectivamente podría llegar a consumir para posibilitar el reemplazo de recursos agotados. Es decir, quizás la clase de tarifa correspondiente a la habitación *standard* se encontrase disponible en dicho instante, pero el recurso que efectivamente se le entregaría podría ser una *suite* si la primera se encontrase agotada.

Para esto, definimos que las clases de tarifas corresponden a los distintos tipos de habitaciones. A su vez, el conjunto de recursos es también el conjunto de tipos de habitaciones. Un producto entonces se define como la combinación del tipo de habitación que se le ofrece al cliente con su correspondiente tarifa, con el tipo de habitación que efectivamente se le va a entregar.

Así, el modelo descrito en la sección anterior se instancia de la siguiente manera:

- Para cada ventana de tiempo  $k$  se calcula  $\bar{\lambda}_k$  como el promedio de todos los  $\hat{\lambda}_{(t^B, t^C)}$  tales que  $t^C - t^B = K - k$ . Decimos  $K - k$  en lugar de  $k$  para indexar las ventanas de tiempo hacia adelante. Cada ventana de tiempo se subdivide a su vez en  $T_k = \bar{\lambda}_k / 0,1$  períodos, por lo que la probabilidad de arribo en un determinado período es  $\lambda_k = 0,1$ .
- Si  $j \in H$ , entonces definimos  $(j, j) \in N$  como un producto de la red.
- Si  $j \in H$  y  $k \in U_j$ , entonces definimos  $(j, k) \in N$  como un producto de la red. Decimos que  $k$  “implementa”  $j$  porque es el recurso que se utiliza para vender la clase de tarifa  $j$ .

- La ganancia de vender una unidad de  $(j, i)$  es  $r_{(j,i)} = \hat{r}_j$ .
- $(j, i)$  consume una unidad del recurso  $i$  y nada más. Es decir,  $A_{(j,i)} = \mathbf{e}_i$ , donde  $\mathbf{e}_i \in \mathbb{R}^n$  vale 1 en la posición  $i$  y 0 en todas las demás. Este tipo de red se conoce como “*single leg*”.
- Definimos la capacidad del recurso  $i$ ,  $c_i = \hat{c}_i$ .
- La probabilidad de que el cliente que arriba en la ventana de tiempo  $k$  elija  $(j, i)$  dado que se ofrece  $S \subseteq N$  viene dada por  $P_{(j,i)}^k(S) = \hat{v}_j^k / (\sum_{(p,q) \in S} \hat{v}_p^k + 1)$ .
- La probabilidad de que el cliente que arriba en la ventana de tiempo  $k$  elija no comprar cuando se ofreció  $S \subseteq N$  es  $P_0^k(S) = 1 / (\sum_{(p,q) \in S} \hat{v}_p^k + 1)$ .

Para simplificar la notación utilizamos en forma indistinta como índices a una variable  $z \in N$  y a  $(j, i) \in N$ , si el producto  $z$  corresponde a la clase de tarifa  $j$  implementada sobre el recurso  $i$ .

Agregamos, además, una nueva restricción: un *offer set* válido no puede contener  $(j, i_1)$  y  $(j, i_2)$  para  $i_1 \neq i_2$ . Esto se debe a que la decisión de qué recurso se utiliza al vender la clase de tarifa  $j$  debía definirse al construir el *offer set*, pues esto era parte fundamental del problema operativo que intentábamos resolver. Definimos el conjunto de todos los *offer sets* válidos como

$$\mathcal{P}_\Lambda(N) = \{S \in \mathcal{P}(N) : (j_1, i_1) \in S \wedge (j_2, i_2) \in S \Rightarrow (j_1 = j_2 \Rightarrow i_1 = i_2)\}$$

## 3.4. Descripción de los algoritmos utilizados

### 3.4.1. FCFS

FCFS (*First Come-First Serve*) no es un método de optimización para el control de capacidad, sino una política en sí. Esta política consiste en aceptar cualquier pedido de reserva de los clientes que llegan o, dicho de otro modo, que todas las clases de tarifa se encuentran abiertas –siempre que haya disponibilidad de recursos. En nuestro caso en particular, utilizamos siempre el recurso de menor costo para dicha clase.

Expresado según el modelo que acabamos de describir, para cada  $j$  se elige un  $r$  que implemente  $j$  cuya capacidad no se encuentre agotada tal que el precio con que el hotel ofrece normalmente ese tipo de habitación sea mínimo. Formalmente, buscamos

$$\min_{i=1, \dots, m} \hat{r}_i \quad : \quad (j, i) \in N \wedge x_i > 0$$

Si algún  $i$  es solución, entonces se incluye  $(j, i)$  en el *offer set*  $S$ .

Esta política, que corresponde a no hacer ningún tipo de control de capacidad, fue el punto de referencia con el que comparamos los demás métodos.

### 3.4.2. CDLP

Dado el espacio de búsqueda de tamaño exponencial del modelo de programación dinámica, resulta poco práctica su resolución exacta para una red de tamaño real. Por lo tanto, el único enfoque realista es el de intentar aproximar el problema de decisión. Un enfoque popular es el de usar una aproximación determinística, en la cual, las cantidades estocásticas son reemplazadas por sus valores esperados y la capacidad y la demanda se asumen continuos. Esto reduce el problema a un problema de programación lineal, que es mucho más fácil de resolver. Este es el enfoque tomado por [GIPD04].

CDLP se formula utilizando una tasa de arribos  $\lambda$  constante. En este trabajo, lo adaptamos a una tasa de arribos variable de la siguiente manera. Se esperan arribos durante  $T$  períodos divididos en  $K$  ventanas de tamaño  $T_k$  ( $\sum_{k=1}^K T_k = T$ ). Durante un período perteneciente a la ventana  $k$ , los clientes llegan con probabilidad  $\lambda_k$ . Cuando llega un cliente en este período y observa un conjunto de productos abiertos  $S$ , se vende una unidad del producto  $j$  con probabilidad  $P_j^k(S)$ . Esta probabilidad puede entenderse como que se vende una cantidad determinística  $P_j^k(S)$ , que es la demanda media del producto  $j$  en un período perteneciente a la ventana  $k$  cuando se ofrece  $S$ . Llamamos  $R^k(S)$  el ingreso esperado generado por un cliente que llega en algún instante de la ventana  $k$  cuando se ofrece  $S$ . Es decir,

$$R^k(S) = \sum_{j \in S} r_j P_j^k(S)$$

De manera similar, denominamos  $Q_i^k(S)$  a la probabilidad de utilizar una unidad de capacidad de un recurso  $i$ ,  $i = 1, \dots, m$ , dado que se ofrece el conjunto  $S$  en la ventana  $k$ , y denominamos al vector de probabilidades de consumo de capacidades  $Q^k(S) = (Q_1^k(S), \dots, Q_m^k(S))^T$ . Entonces

$$Q^k(S) = AP^k(S)$$

donde  $P^k(S) = (P_1^k(S), \dots, P_n^k(S))^T$ . CDLP trata  $Q^k(S)$  como una tasa de consumo determinística de los recursos. De manera similar,  $\lambda_k$  constituye una tasa de arribo determinística de los clientes por cada período.

Por último, definimos las variables de decisión del problema, a las que denominamos  $t_k(S)$  que representan la cantidad de períodos que se ofrece el conjunto  $S$  en la ventana  $k$ . A los fines de la formulación del PL, es irrelevante exactamente en qué orden se ofrece: el orden lo fijaremos luego mediante una heurística.

De esta manera, el problema se reduce al siguiente programa lineal:

$$\begin{aligned}
 V^{\text{CDLP}} &= \max \sum_{S \in \mathcal{P}_\Lambda(N)} \sum_k R^k(S) \lambda_k t_k(S) \\
 &\text{s.a.} \\
 &\sum_{S \in \mathcal{P}_\Lambda(N)} \sum_k Q^k(S) \lambda_k t_k(S) \leq c \\
 &\sum_{S \in \mathcal{P}_\Lambda(N)} t_k(S) \leq T_k, \quad k = 1, \dots, K \\
 &t_k(S) \geq 0, \quad k = 1, \dots, K
 \end{aligned}$$

#### 3.4.2.1. Construcción de la política

Una vez resuelto el problema de programación lineal planteado por CDLP, se obtiene la cantidad de períodos durante la ventana de tiempo  $k$  que se debe ofrecer cada conjunto  $S$ . Esto no determina aún una política ya que falta definir en qué momento exacto —es decir, en qué orden— se van a ofrecer estos conjuntos de productos.

Para construir la política, tomamos por cada ventana de tiempo  $k$  todos los conjuntos  $S$  tales que  $t_k(S) > 0$ . Luego, los ordenamos en forma decreciente en función de  $R^k(S)$  y los establecimos como *offer sets* de la política en este mismo orden durante  $t_k(S)$  períodos consecutivos, comenzando desde el instante en que se inicia la ventana de tiempo  $k$  (es decir, en el momento  $\sum_{i=1}^{k-1} T_i$ ). Optamos por esta forma de ordenamiento por ser consistente con la idea de que a medida que avanza el tiempo, los productos tienden a cerrarse.

También puede ocurrir que  $\sum_{S \subseteq N} t_k(S) < T_k$ , es decir que la solución indique ofrecer conjuntos durante un tiempo inferior a la duración de la ventana. En este caso, el período en que no se ofrece ningún producto se encuentra al final de la ventana de tiempo. Esto podría verse como que se está ofreciendo el conjunto vacío, cuyo ingreso esperado es nulo, y por lo tanto se debe ubicar al final de acuerdo al ordenamiento propuesto.

### 3.4.2.2. Generación de columnas

CDLP tiene un número exponencial de variables primales ( $K2^n$ ). Sin embargo, tal como mencionan Liu y van Ryzin ([LvR08]) y Gallego et al. ([GIPD04]), admite métodos de generación de columnas para intentar resolver el programa lineal en forma eficiente. Una columna se corresponde con una variable de decisión  $t_k(S)$ , identificada por el período  $k$  y el conjunto  $S$  al que hace referencia. La idea general es comenzar con un número limitado de columnas (en nuestro caso, sólo con el *offer set* vacío en cada uno de los períodos) y resolver este PL reducido. A continuación, se usa la solución dual para verificar si hay columnas que no hayan sido incluidas y que tengan un costo reducido positivo. En caso afirmativo, se agrega alguna de ellas al problema y se vuelve a resolver el PL. Este procedimiento se repite hasta que no se encuentren más columnas con costo positivo reducido, en cuyo caso la solución encontrada es óptima. Se espera que el número de columnas generado sea relativamente modesto y aunque en el peor caso se puede llegar a incorporar un número exponencial de columnas, efectivamente se observa una mejora considerable de la eficiencia en la práctica.

Específicamente, el algoritmo de generación de columnas funciona del siguiente modo. Sean  $\pi$  y  $\sigma$  los vectores de variables duales asociadas al primer y segundo grupo de restricciones, respectivamente. Los valores de estas variables se obtienen de resolver CDLP sujeto a un conjunto reducido de columnas. A continuación, se debe determinar si algún conjunto  $S$  en un período  $k$  que no estaba incluido tiene un costo reducido positivo. Eso se logra resolviendo el siguiente subproblema de generación de columnas:

$$\max_{k \in \{1, \dots, K\}, S \in \mathcal{P}_\Lambda(N)} \left\{ \lambda_k(R^k(S) - \pi^\top Q^k(S)) - \sigma_k \right\}$$

Si el valor óptimo del funcional en este problema no es positivo, entonces  $\pi$  y  $\sigma$  son duales factibles y la solución actual es entonces óptima para el problema original. De lo contrario la solución óptima  $(S^*, k^*)$  corresponde a un conjunto en un período con costo reducido positivo ( $\lambda_{k^*}(R^{k^*}(S^*) - \pi^\top Q^{k^*}(S^*)) - \sigma_{k^*} > 0$ ). Por lo tanto, si se agrega la columna dada por  $(S^*, k^*)$  y se vuelve a resolver el PL, el valor de la función objetivo se va a incrementar. De esta manera, se obtiene un nuevo conjunto de valores para las variables duales que se utiliza para repetir este procedimiento.

La principal dificultad radica en resolver el problema de generación de columnas. Este problema se puede resolver en forma eficiente para un caso particular en que  $\mathcal{P}_\Lambda(N) = \mathcal{P}(N)$ , es decir, cuando cualquier subconjunto de  $N$  forma un *offer set*



factible. La solución eficiente a este caso particular del problema se puede encontrar en [LvR08] y [GIPD04]. El caso más general, que corresponde a un problema de programación hiperbólica 0-1, se sabe que es NP-Hard (ver [BMDV09]). Por su parte, para este problema restringido no pudimos dar con una solución óptima polinomial y, por lo tanto, desarrollamos una heurística. Para ello, se define un vector binario  $\mathbf{y}$  como el vector característico de  $S$ . Cada componente  $y_j$  indica si el producto  $j$  se incluye en conjunto de productos:

$$y_j = \begin{cases} 1 & \text{si el producto } j \text{ está disponible, } j \in N \\ 0 & \text{si el producto } j \text{ no está disponible, } j \in N \end{cases}$$

De esta manera, el problema se puede reescribir en función de estas nuevas variables  $y_j$ :

$$\begin{aligned} & \max_{k \in \{1, \dots, T\}, S \in \mathcal{P}_\Lambda(N)} \left\{ \lambda_k (R^k(S) - \pi^\top Q^k(S)) - \sigma_k \right\} \equiv \\ & \max_{k \in \{1, \dots, T\}, \mathbf{y} \in \{0,1\}^n} \left\{ \lambda_k \sum_{j=1}^n r_j P_j^k(\mathbf{y}) - \lambda_k \pi^\top A P^k(\mathbf{y}) - \sigma_k \right\} \\ & \text{s.a.} \\ & \sum_{r:(j,i) \in N} y_{(j,i)} \leq 1, \quad j = 1, \dots, m. \end{aligned}$$

Donde,

$$P_j^k(\mathbf{y}) = \frac{v_j^k y_j}{\sum_{i=1}^m v_i^k y_i + 1}$$

Reemplazando, podemos escribir el funcional como:

$$\begin{aligned} & \max_{k \in \{1, \dots, T\}, \mathbf{y} \in [0,1]^n} \left\{ \lambda_k \frac{\sum_{j=1}^n r_j v_j y_j}{\sum_{i=1}^n v_i y_i + 1} - \lambda_k \sum_{j=1}^n \frac{\pi^\top A_j v_j y_j}{\sum_{i=1}^n v_i y_i + 1} - \sigma_k \right\} \\ = & \max_{k \in \{1, \dots, T\}, \mathbf{y} \in [0,1]^n} \left\{ \lambda_k \frac{\sum_{j=1}^n (r_j - A_j^\top \pi) v_j y_j}{\sum_{i=1}^n v_i y_i + 1} - \sigma_k \right\} \end{aligned}$$

Utilizamos entonces la siguiente estrategia. Sea  $w_j = r_j - A_j^\top \pi$ , elegimos de cada partición (es decir, por cada clase de tarifa) el producto  $j$  con  $w_j$  máximo. Luego ordenamos los productos elegidos de cada una de las particiones de manera tal que  $i$  antecede a  $j$  si  $w_i \geq w_j$ . Por último, agregamos un producto a la vez al *offer set* en este orden, verificando que en ningún caso disminuyera el valor del funcional. Si esto ocurriera, no se agregaban más productos. Este algoritmo de generación de

columnas se presenta en la figura 3.1. En [LvR08] y [BMDV09] se abordan problemas de generación de columnas similares.

Notar que este procedimiento no necesariamente encuentra el óptimo. Por ejemplo, el caso en el que  $m = 1$ ,  $n = 2$ ,  $\mathcal{C}_1 = \{1, 2\}$ ,  $w = [w_1, w_2] = [1, 2]$ ,  $v = [v_1, v_2] = [1, 0, 1]$ . La solución óptima es  $y = [y_1, y_2] = [1, 0]$ . Sin embargo, la estrategia de elegir sólo el máximo encontraría  $y = [y_1, y_2] = [0, 1]$  dado que  $w_2 > w_1$ .

De todos modos, en la práctica esta heurística dio resultados satisfactorios. Tal es así que para las pruebas que hicimos no encontramos diferencia entre los resultados obtenidos resolviendo el problema primal de CDLP utilizando el conjunto completo de columnas (que, dado el tamaño de nuestras instancias era abordable en la práctica) y los obtenidos utilizando esta heurística para la generación de columnas.

[Inicialización]: Construir  $w_j = r_j - A_j^\top \pi$  y sea

$$\text{Valor}(S) = \frac{\sum_{j \in S} w_j v_j}{\sum_{j \in S} v_j + 1}$$

$S := \text{buildOfferSet}()$   
 Para  $k := 1, \dots, T$ :  
   Si  $\lambda_k \text{Valor}(S) - \sigma_k > 0$ :  
     Devolver la columna dada por  $k, S$ .  
   Fin “Si”  
 Fin “Para”

Donde el procedimiento  $\text{buildOfferSet}()$  es:

  Para  $h := 1, \dots, m$ :  
     Calcular  $\omega(h) := j \in \mathcal{C}_h$  tal que  $w_j = \max_{i \in \mathcal{C}_h} w_i$   
   Fin “Para”  
 Construir  $L := [\omega(h) : h = 1, \dots, m]$   
 Ordenar  $L$  de manera tal que  $L[i] \geq L[j]$  sii  $i < j$  y  $w_{L[i]} \geq w_{L[j]}$ .  
 Inicializar  $S := \emptyset$ .  
 Para cada  $j$  en  $L$ :  
   Si  $\text{Valor}(S \cup \{j\}) > \text{Valor}(S)$ :  
     Asignar  $S := S \cup \{j\}$   
   Si no:  
     Devolver  $S$ .  
   Fin “Si”  
 Fin “Para”

Figura 3.1: Heurística para generación de columnas

### 3.4.3. Heurística: agregación de estados del problema de programación dinámica

Otra forma equivalente de ver el problema 3.1 es la de pensarlo como un Proceso de Decisión de Markov cuyos estados son los pares  $(t, \mathbf{x}), t = 1, \dots, T, \mathbf{x} \in \mathbb{Z}_{c_1+1} \times \mathbb{Z}_{c_2+1} \times \dots \times \mathbb{Z}_{c_m+1}$  y donde la función de transición viene dada de la siguiente manera: la probabilidad de transición del estado  $(t, \mathbf{x})$  al estado  $(t+1, \mathbf{x} - \mathbf{e}_i)$  es  $P_{(j,i)}^k(S)$ , siendo que  $t$  está en la ventana de tiempo  $k$  y  $\mathbf{e}_i \in \mathbb{R}^m$  es un vector tal que la coordenada  $i$  vale 1 y todas las demás 0. Por su parte, con probabilidad  $P_0^k(S)$ , el proceso pasa al estado  $(t+1, \mathbf{x})$ . Cualquier otra transición tiene probabilidad 0. Como ya mencionamos, optimizar este proceso en forma exacta no es práctico dada la cantidad exponencial de estados que se deben explorar ( $T \prod_{i=1}^m c_i$ ). Por lo tanto, nos propusimos desarrollar una heurística alternativa. La idea fue la de reducir la complejidad del problema agrupando estados que representasen momentos similares de la capacidad remanente (ver [DF04] para una aplicación distinta de esta técnica). Por ejemplo, si el sistema tiene 15, 16 o 17 unidades de un determinado recurso en un momento  $t$ , es probable que la decisión óptima sea igual en cualquier caso. Desde luego, esto no es lo mismo que ocurre cuando las disponibilidades son de 0, 1 o 2 unidades. Con lo cual, la forma de agregación de los estados debía ser flexible para distintos estados de capacidad. Por ejemplo, los estados correspondientes a las capacidades 0, 1 y 2 pueden permanecer desagrupados mientras que las disponibilidades de 15 a 20 unidades pueden quedar agrupadas en único estado. Esta agrupación de estados se efectúa en forma uniforme para todos los períodos  $t = 0, \dots, T$ , con lo cual ignoramos la dimensión temporal durante el siguiente análisis.

A fin de agrupar los distintos estados de capacidad, para cada recurso  $i$ , construimos un nuevo conjunto de estados de tamaño arbitrario<sup>1</sup>  $\hat{c}_i + 1$ . Denotamos a este conjunto  $\hat{S}^i = \{\hat{s}_0^i, \dots, \hat{s}_{\hat{c}_i}^i\}$ . Cada uno de estos estados agrupa un cierto número posibles capacidades remanentes del recurso  $i$  determinado por una función  $\phi_i : \hat{S}^i \rightarrow \{1, \dots, c_i + 1\}$ . Las capacidades que se agrupan son siempre secuenciales y adyacentes: es decir, no se pueden agrupar en un único estado las capacidades 1, 3 y 5 sin incluir también las capacidades 2 y 4. La única restricción que se aplica sobre esta función consiste en que la cantidad total de estados agrupados tiene que coincidir con la cantidad de estados que existían en problema inicial, es decir

$$\sum_{\hat{s}_j^i \in \hat{S}^i} \phi_i(\hat{s}_j^i) = c_i + 1$$

Por ejemplo, en la figura 3.2 se presenta una posible agrupación de los estados

<sup>1</sup>Esta variable  $\hat{c}_i$  no tiene relación con la homónima de la sección 3.3

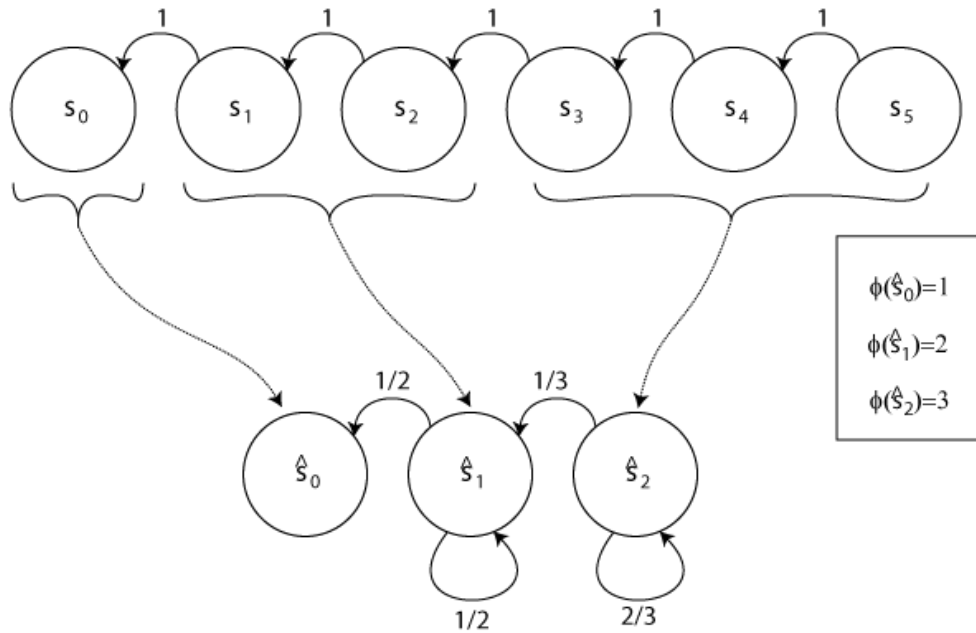


Figura 3.2: Agregación de estados para un recurso de capacidad 5. Las probabilidades de transición entre estados están condicionadas a que se observó un consumo del recurso en cuestión.

de un sistema con un único recurso de capacidad 5. Los nuevos estados son  $\hat{s}_2$ ,  $\hat{s}_1$  y  $\hat{s}_0$  (dado que hay un único recurso descartamos el superíndice), mientras que con  $s_0, \dots, s_5$  se representan los estados de capacidad del problema inicial. Queremos que  $\hat{s}_0$  represente el mismo estado que  $s_0$ , y para ello, definimos  $\phi_1(\hat{s}_0) = 1$  —es decir, que agrupa un único estado, y como el agrupamiento se hace en forma secuencial, se comienza por  $s_0$ . En  $\hat{s}_1$  agrupamos  $s_1$  y  $s_2$ , por lo que definimos  $\phi_1(\hat{s}_1) = 2$ , puesto que de esta manera se agrupan los dos estados que suceden a  $s_0$ :  $s_1$  y  $s_2$ . Por último queremos que en  $\hat{s}_2$  se agrupen los 3 estados restantes, por lo que definimos  $\phi_1(\hat{s}_2) = 3$ . De esta manera,  $\hat{s}_0$  representa la capacidad agotada del recurso;  $\hat{s}_1$ , la disponibilidad de 1 o 2 unidades del recurso; y  $\hat{s}_2$  las disponibilidades de 3, 4 o 5 unidades del recurso.

Ahora bien, cuando se consume una unidad del recurso  $i$ , el proceso expresado en función de los nuevos estados no necesariamente se va a efectuar una transición de  $\hat{s}_j^i$  a  $\hat{s}_{j-1}^i$ . Para el ejemplo que mencionábamos antes, si existían dos unidades de capacidad del recurso y se consume 1, el sistema debería permanecer aún en el estado  $\hat{s}_1$ . Sin embargo si había una sola unidad en existencia y esta se consume, entonces se debería verificar una transición de  $\hat{s}_1$  a  $\hat{s}_0$ . Por esta razón, modelamos como un evento estocástico la transición de un estado  $s_j^i$  a  $s_{j-1}^i$  cuando se consume una unidad del recurso  $i$ . Definimos entonces  $P_i^\downarrow(\hat{x}) = 1/\phi_i(\hat{s}_{x_i})$ , como la probabilidad de que

se verifique esta transición. Observar por ejemplo, las probabilidades de transición condicionadas al consumo de una unidad del recurso ilustradas en la figura 3.2.

En este punto, estamos en condiciones de escribir la nueva función de transición de la siguiente manera: sea  $\hat{\mathbf{x}} \in \mathbb{Z}_{\hat{c}_1+1} \times \mathbb{Z}_{\hat{c}_2+1} \times \cdots \times \mathbb{Z}_{\hat{c}_m+1}$  el vector de capacidad remanente en base a los nuevos estados (es decir, que el sistema se encuentra en la conjunción de los estados  $s_{\hat{\mathbf{x}}_i}^i$ ) y, asumiendo que el período  $t$  se encuentra en la ventana de tiempo  $k$ , el sistema evoluciona a  $(t+1, \hat{x} - \mathbf{e}_i)$  con probabilidad  $\sum_{j \in S: j \in A^i} P_j^k(S) P_i^\downarrow(\hat{x})$ , donde  $\sum_{j \in S: j \in A^i} P_j^k(S)$  es la probabilidad de consumo de un producto que utilice el recurso  $i$ . En cambio, el sistema evoluciona sin cambios en la capacidad hacia  $(t+1, \hat{x})$ , con probabilidad  $P_0^k(S) + \sum_{i=1}^m \sum_{j \in S: j \in A^i} P_j^k(S) (1 - P_i^\downarrow(\hat{x}))$ . Notar que la suma de estas probabilidades es 1:

$$\begin{aligned} & \sum_{i=1}^m \sum_{j \in S: j \in A^i} P_j^k(S) P_i^\downarrow(\hat{x}) + P_0^k(S) + \sum_{i=1}^m \sum_{j \in S: j \in A^i} P_j^k(S) (1 - P_i^\downarrow(\hat{x})) \\ = & \sum_{i=1}^m \sum_{j \in S: j \in A^i} P_j^k(S) + P_0^k(S) \\ = & \sum_{j \in S} P_j^k(S) + P_0^k(S) = 1 \end{aligned}$$

Luego, podemos escribir la ecuación de Bellman para este proceso de la siguiente manera:

$$\begin{aligned} V_t(\hat{x}) &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda_k P_j^k(S) (r_j + P_i^\downarrow(\hat{x}) V_{t+1}(\hat{x} - \mathbf{e}_i) + (1 - P_i^\downarrow(\hat{x})) V_{t+1}(\hat{x})) \right. \\ & \quad \left. + \lambda_k P_0^k(S) + 1 - \lambda_k V_{t+1}(\hat{x}) \right\} \\ &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda_k P_j^k(S) (r_j + P_i^\downarrow(\hat{x}) V_{t+1}(\hat{x} - \mathbf{e}_i) + (1 - P_i^\downarrow(\hat{x})) V_{t+1}(\hat{x})) \right. \\ & \quad \left. + \underbrace{\left( \lambda_k \left( 1 - \sum_{j \in S} P_j^k(S) \right) + 1 - \lambda_k \right)}_{\left( \cancel{\lambda_k} - \lambda_k \sum_{j \in S} P_j^k(S) + 1 - \cancel{\lambda_k} \right)} V_{t+1}(\hat{x}) \right\} \end{aligned}$$

Donde el último paso proviene de utilizar que  $\sum_{j \in S} P_j^k(S) + P_0^k(S) = 1$ . Entonces,

$$\begin{aligned} V_t(\hat{x}) &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda_k P_j^k(S) (r_j + P_i^\downarrow(\hat{x}) V_{t+1}(\hat{x} - \mathbf{e}_i) \right. \\ &\quad \left. + (\mathcal{I} - P_i^\downarrow(\hat{x})) V_{t+1}(\hat{x}) - V_{t+1}(\hat{x}) \right\} + V_{t+1}(\hat{x}) \\ &= \max_{S \in \mathcal{N}} \left\{ \sum_{j \in S} \lambda P_j^k(S) (r_j - P_i^\downarrow(\hat{x}) (V_{t+1}(\hat{x}) - V_{t+1}(\hat{x} - \mathbf{e}_i))) \right\} + V_{t+1}(\hat{x}) \end{aligned}$$

A este nuevo problema de programación dinámica lo resolvemos de forma exacta, sobre un espacio de búsqueda varios órdenes más chico (aunque aún exponencial).

Para ver cómo se comporta este algoritmo, supongamos el siguiente escenario. Existen dos recursos con 20 unidades de capacidad cada uno. Hay dos clases de tarifa con precios  $r_1 = 2, r_2 = 4$ . La clase 1 se puede implementar sobre el recurso 1 o 2, mientras que la clase de tarifa 2 sólo se puede implementar sobre el recurso 2. Se observan arribos durante 400 períodos con tasa  $\lambda = 0,1$ . Resolvemos el programa dinámico para este problema en forma exacta y también agrupando estados. Para esto último, definimos  $\phi_i(j) = \text{fib}(j)$ ,  $i = 1, 2$  —es decir, el primer estado representa la capacidad 0; el segundo estado, la capacidad 1; el tercero, las capacidades 2 y 3; el cuarto, 4, 5 y 6; el quinto, 7, 8, 9, 10 y 11; y así sucesivamente. Luego comparamos las funciones de valor resultantes. Dada la multidimensionalidad de las mismas, mostramos un corte fijando la disponibilidad del recurso 2,  $x_2 = 1$ , y exhibimos en simultáneo 40 valores de  $t$  separados cada 10 períodos. Los resultados se pueden observar en la figura 3.3. Como se puede apreciar, obtuvimos una buena aproximación de la función de valor. Además, la ecuación de Bellman que este método nos permite hallar puede aproximar la del problema original con distintos grados de precisión en función de cómo se decidan agrupar los estados. Por ejemplo, si  $\phi_i(j) = 1$  para todo recurso  $i$  y para cualquier capacidad  $j$ , el problema que se obtiene de aplicar este método sigue siendo el problema exacto. De esta manera, se puede ajustar la simplificación que se efectúa en función de la complejidad del sistema a resolver.

### 3.4.3.1. Selección eficiente de los *offer sets*

Hay una dificultad adicional en la resolución del programa dinámico además de la cantidad exponencial de estados y es que en cada uno de ellos, la cantidad de subconjuntos que se debe explorar es también exponencial. Sin embargo, podemos evitar hacer una búsqueda exhaustiva. Para ello, observamos que a  $V_{t+1}(\hat{x}) - V_{t+1}(\hat{x} -$

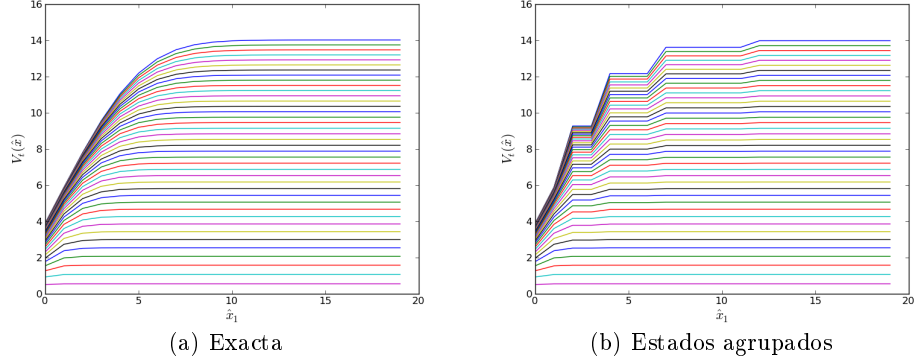


Figura 3.3:  $V_t(\hat{x})$  en función de  $\hat{x}_1$  con  $\hat{x}_2 = 1$  fijo para distintos valores de  $t$  (representados por cada una de las trazas)

$\mathbf{e}_i$ ) se lo puede escribir como  $\Delta V_{t+1}^\top(\hat{x}) \cdot \mathbf{e}_i$  donde  $\Delta V_{t+1}^\top(\hat{x}) = (V_{t+1}(\hat{x}) - V_{t+1}(\hat{x} - \mathbf{e}_1), V_{t+1}(\hat{x}) - V_{t+1}(\hat{x} - \mathbf{e}_2), \dots, V_{t+1}(\hat{x}) - V_{t+1}(\hat{x} - \mathbf{e}_m))$ . Entonces, el problema de maximización se puede escribir como

$$\sum_{j \in S} \lambda P_j^k(S) (r_j - P_i^\downarrow(\hat{x}) (\Delta V_{t+1}^\top(\hat{x}) \mathbf{e}_i)) \quad (3.2)$$

Este problema tiene la misma forma que el problema de generación de columnas visto en la sección 3.4.2.2 (utilizando  $\Delta V_{t+1}^\top(\hat{x})$  en lugar de  $\pi$ ), con lo cual se puede resolver (heurísticamente) de la misma manera en forma eficiente. Además, con el fin de lograr alguna mejora adicional en los tiempos computacionales, decidimos no efectuar esta búsqueda en todos los instantes  $t$ , sino cada un intervalo fijo, utilizando como *offer set* en los períodos intermedios el último  $S$  óptimo calculado. Dado que el  $S$  óptimo no suele presentar demasiada variabilidad en el tiempo, esta simplificación no afecta apreciablemente el resultado.

#### 3.4.4. MEMUR

MEMUR (por Multi-resource Expected Marginal Unit Revenue) es una extensión al popular EMSR-b<sup>2</sup> propuesto por Ferguson et al. en [FGMN10]. MEMUR utiliza un enfoque de demandas independientes, donde la demanda total de un producto  $j$  se modela con una variable aleatoria independiente denotada  $D_j$ . La política obtenida se expresa mediante niveles de protección (ver sección 1.1.5.1).

Sea  $S(j)$  el conjunto de los productos sustitutos de  $j$ , excluido  $j$ . Calcular la demanda agregada de los sustitutos de cada producto:

<sup>2</sup>Detalles de EMSR-b se pueden encontrar en [TvR04]

$$\tilde{D}_j = \sum_{k \in S(j)} D_k$$

Calcular la capacidad agregada de los substitutos de cada producto:

$$\tilde{c}_j = \sum_{k \in S(j)} c_k$$

Y por último, calcular la ganancia media de los substitutos ponderada según el volumen esperado de demanda:

$$\bar{r}_j = \frac{\sum_{k \in S(j)} r_k E[D_k]}{\sum_{k \in S(j)} E[D_k]}$$

Con estas definiciones, se formula el cálculo de los niveles de protección para cada producto  $j$  de la siguiente manera.

$$y_j = \min \left\{ \tilde{c}_j, y : \mathbb{P}(\tilde{D}_j > y) = \frac{r_j}{\bar{r}_j} \right\}$$

### 3.5. Análisis de CDLP

Para comprobar la performance del método CDLP, efectuamos una simulación sobre la base de datos reales. Para ello, tomamos las preferencias correspondientes al Hotel 1 para días de semana y manteniendo 15 días de arribo con las tasas de llegada calculadas, redimensionamos las capacidades de modo tal que el *Demand Ratio*(DR) fuera de 4. El DR se define como el cociente entre la cantidad media esperada total de clientes y la capacidad total del hotel (ver la sección 3.6 para mayores detalles sobre los experimentos con datos reales). Luego, construimos la política de aceptación basada en CDLP y la comparamos con una política FCFS utilizando 1000 tiras de clientes simuladas. El resultado es el que observamos en la figura 3.4.

Sorprendentemente, FCFS se comporta notablemente mejor que CDLP. Para explicar esto, nos detuvimos a analizar dos escenarios más simples en los cuales esto también ocurre y que nos ayudaron a identificar las razones por las cuales CDLP se comporta de manera notablemente subóptima.



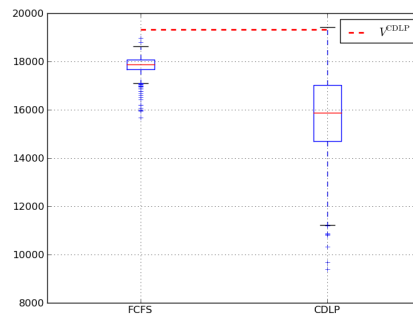


Figura 3.4: Ingresos obtenidos bajo una política creada por CDLP vs. una política FCFS en comparación con el valor del funcional óptimo para CDLP ( $V_{CDLP}$ ).

### 3.5.1. Un problema de CDLP: la política estática

La política generada por CDLP es una política estática, en el sentido de que el conjunto de productos que se ofrece no es sensible a la capacidad remanente de los productos, y si el consumo de la misma se desvía de la media esperada, la política puede alejarse sensiblemente del óptimo. Para observar esto, construimos un escenario ad-hoc en el que se evidenciasen las dificultades de CDLP. Así, definimos una tasa de arribo constante,  $\lambda = 0,01$  durante 12000 períodos, es decir, un promedio de 120 clientes. Estos clientes deciden siempre comprar el único producto disponible, del cual se disponen de 120 unidades divididas en 2 recursos con 60 unidades de capacidad cada uno y da 1 unidad de ganancia por compra. Con este modelo, se efectúa una simulación utilizando una política obtenida mediante CDLP y una política “First Come, First Served” (FCFS) en la cual se intenta satisfacer cada cliente que llega, utilizando cualquier recurso disponible. Cabe destacar que, dado que en este caso la política óptima viene dada por aceptar todas las ofertas, la política FCFS es óptima. A continuación, se agregan 2 recursos, haciendo que el único producto se pueda implementar sobre 4 posibles recursos, cada uno de 30 unidades. Nuevamente, se simula con la política que se obtiene con CDLP y la política FCFS. De nuevo, se agregan 2 recursos más, obteniendo como total 6 recursos de 20 unidades, y se procede igual que antes. Así continuamos hasta llegar a dividir el producto entre 12 recursos de 10 unidades. El resultado de los ingresos comparados de ambas políticas es el que se muestra en la figura 3.5.

Esta ineficiencia de CDLP se explica de la siguiente manera. En cada caso, la política obtenida consiste en abrir siempre el único producto, utilizando en forma uniforme, cada uno de los recursos. Es decir, cada recurso es utilizado durante igual cantidad de periodos para abrir el producto. El problema surge cuando la disponibili-

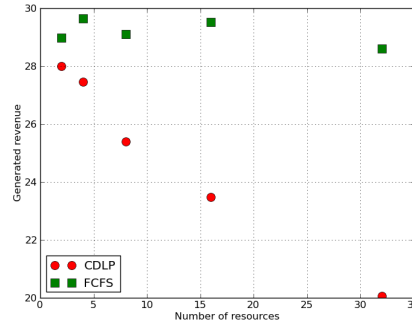


Figura 3.5: Revenue generado en función de la cantidad de recursos en los que se divide un único producto para una política obtenida por CDLP contra una FCFS

dad de un recurso llega a 0: dado que la política es estática, cuando un cliente solicite el producto, no se sustituirá el recurso sobre el que se implementa el recurso, y por tal razón, no se efectivizará la transacción, mientras que la política FCFS resuelve esta sustitución sin problemas.

### 3.5.2. Segundo problema de CDLP: subutilización de recursos

El otro problema que presenta CDLP es que el tiempo que permanecen abiertos los productos que utilizan un determinado recurso está acotado por el tiempo esperado (en promedio) para que este recurso se agote. Sin embargo, esto genera una subutilización de los recursos disponibles, ya que si el recurso no se consume en el tiempo esperado, el remanente queda sin utilizar pues los productos que lo utilizan permanecen cerrados.

Si la restricción de CDLP asociada a la capacidad de un cierto recurso no se cumple por igualdad, tal ineficiencia no se verifica, pues la disponibilidad del recurso no limita el tiempo que sea óptimo mostrar el producto que lo utiliza. Pensamos entonces, el caso en el cual la restricción —escrita nuevamente en 3.3— se verifica por igualdad.

$$\sum_{S \subseteq N} \sum_k Q_k^i(S) \lambda_k t_k(S) \leq c_i \quad (3.3)$$

En cualquier caso, ya sea que se verifique por igualdad o por una diferencia mayor a 0, podemos pensar que el consumo de un recurso es una variable aleatoria—llamémosla  $W_i$ — cuya tasa de arribos está dada por la parte izquierda de la inecuación anterior. Cabe destacar que las variables  $t_k(S)$  ya tienen un valor fijo dado por

la resolución del LP. Decimos entonces:

$$W_i \sim \text{Poisson}\left(\sum_{S \subseteq N} \sum_k Q_k^i(S) \lambda_k t_k(S)\right)$$

Particularmente, cuando la restricción se cumple por igualdad

$$W_i \sim \text{Poisson}(c_i)$$

Por lo tanto, para  $W_i$ —es decir, la utilización del recurso  $i$ —deberíamos observar una media  $c_i$ , que se correspondería al consumo total del recurso. Esto, sin embargo, no se verifica por la razón de que la variable  $W_i$  está truncada por  $c_i$ : no se puede consumir más recursos de lo que hay en existencia. Por lo tanto,  $W_i$  no tiene una distribución Poisson por estar truncada en  $c_i$  y su valor esperado se debe buscar en la siguiente fórmula:

$$\begin{aligned} E[W_i] &= \sum_{k=0}^{c_i} P(W_i = k) \cdot k + \sum_{k=c_i+1}^{\infty} P(W_i = k) \cdot c_i \\ &= \sum_{k=0}^{c_i} P(W_i = k) \cdot k + (1 - P(W_i \leq c_i)) \cdot c_i \end{aligned}$$

A modo de ilustración de este hecho, generamos un escenario en el cual se ofrecen  $K$  productos, cada uno de los cuales utiliza un recurso independiente, con capacidades  $c_1 \cdots c_K$ . Efectuamos una simulación con 10 períodos de llegada, con tasas de arribo tales que el DR resultante sea de 10 promediando 1000 instancias aleatorias. En la figura 3.6 se pueden observar los consumos observados para los recursos y los predichos por la fórmula anterior.

Para hacer más visible la subutilización de recursos respecto del óptimo (que estaría dado por el consumo del total de los recursos), calculamos el *load factor* de cada recurso que está dado por el consumo observado dividido la capacidad total y lo comparamos también con el valor esperado  $E[W_i]/c_i$ . El resultado es el que se aprecia en la figura 3.7.

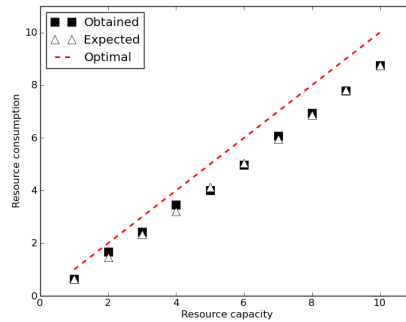


Figura 3.6: Comparación entre los consumos esperados y los simulados para recursos de distintas capacidades bajo una política CDLP. Se marcan con “Obtained” los valores obtenidos por simulación y con “Expected” los valores teóricos predichos por  $E[W_i]$

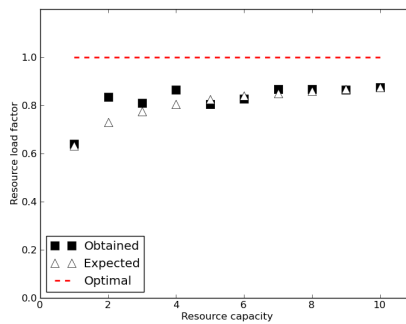


Figura 3.7: Comparación entre los load factors esperados y los simulados para recursos de distintas capacidades bajo una política CDLP. Se marcan con “Obtained” los valores obtenidos por simulación y con “Expected” los valores teóricos predichos por  $E[W_i]/c_i$

### 3.5.3. Explicación del caso real

Volviendo al caso que había motivado esta discusión, buscamos explicar por qué CDLP se comportaba peor que FCFS, de acuerdo a los resultados observados en la figura 3.4. En esta figura se graficaba en línea punteada el valor óptimo del funcional de CDLP. Dicho funcional expresaba el ingreso esperado de ofrecer cada conjunto durante el tiempo que determinasen las variables de decisión del problema (ver sección 3.4.2). Ya mencionamos que CDLP es una aproximación heurística en la que las magnitudes estocásticas se reemplazan por sus valores esperados y que no fija el orden en que se deben ofrecer los conjuntos de productos. A pesar de esto, se esperaba que el valor del funcional para una solución de CDLP reflejase en cierta medida los valores que se obtenían por simulación de aplicar una política construida

según los parámetros de dicha solución. Sin embargo, esto no es lo que se observa en la figura 3.4: si bien el valor del funcional supera a los ingresos obtenidos de una política FCFS, estos últimos superan a los valores obtenidos mediante la política generada a partir de CDLP. Por lo tanto, buscamos identificar qué ocurría con el ingreso que se esperaba obtener pero que no se verificaba en la práctica.

Hasta el momento, identificamos dos problemas que inciden en el rendimiento de las políticas obtenidas por CDLP. Por un lado, el problema descrito en la sección 3.5.2, que como ya se vio, tiene un impacto directo sobre el funcional del problema; con lo cual, no explica la diferencia entre el funcional y los valores obtenidos por simulación. Sugerimos entonces, que la razón por la cual el ingreso es tan inferior al valor del funcional se debe al problema explicado en la sección 3.5.1: es decir que la diferencia entre el ingreso que se esperaba obtener y el finalmente observado es debido a que existen productos que la política mantiene abiertos pero cuyos recursos ya se encuentran agotados.

El funcional de CDLP no tiene en cuenta las capacidades remanentes de los recursos pues, por restricción, el consumo esperado no puede superar la capacidad de ningún recurso. Sin embargo, en una realización del proceso, los consumos pueden ser superiores al valor esperado, agotándose el recurso antes de lo previsto. Entonces, quisimos ver qué ocurría si el funcional se actualizaba teniendo en cuenta que los productos cuyos recursos ya se habían agotado no se podían ofrecer. Para comprobar esto, definimos una nueva función  $G_t^{\text{CDLP}}(\mathbf{x}_t)$  que representase el ingreso esperado en el período  $t$  actualizado en función de las capacidades remanentes  $\mathbf{x}_t$ :

$$G_t^{\text{CDLP}}(\mathbf{x}_t) = \lambda_k R^k(S_\mu(t)|\mathbf{x}_t)$$

donde  $\mu$  es la política que obtuvimos por CDLP,  $S_\mu(t)$  es el conjunto seleccionado en dicha política para ofrecer en el instante  $t$  y se asume que  $t$  se encuentra dentro de la ventana de tiempo  $k$ . A su vez, se define  $R^k(S_\mu(t)|\mathbf{x}_t)$  como el ingreso esperado de ofrecer en el período  $t$  los productos de  $S_\mu(t)$  que aún tienen disponibilidad de recursos:

$$R^k(S_\mu(t)|\mathbf{x}_t) = R^k(\{j \in S_\mu(t) : j \text{ está disponible según } \mathbf{x}_t\})$$

A todos estos valores  $G_t^{\text{CDLP}}(\mathbf{x}_t)$  los sumamos sobre todos los períodos para obtener un valor del funcional actualizado con las disponibilidades de los recursos en cada período. Lo llamamos a este valor  $G^{\text{CDLP}}$ :

$$G^{\text{CDLP}} = \sum_{t=0}^T G_t^{\text{CDLP}}(\mathbf{x}_t)$$

Entonces, dada una realización de un proceso,  $G^{\text{CDLP}}$  calcula el valor del funcional de CDLP actualizado en función de las capacidades agotadas. De esta manera, calculamos este valor para cada realización del proceso que generamos por simulación en la sección 3.5, cuyos resultados graficamos en la figura 3.4, y lo comparamos con el valor efectivamente obtenido en dicha simulación por una política CDLP. El resultado es el que se puede apreciar en la figura 3.8.

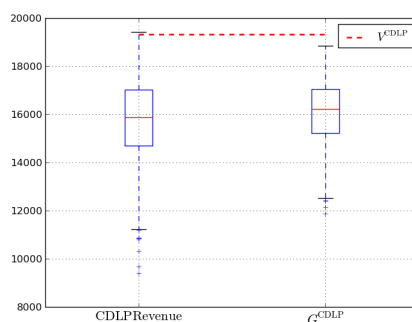


Figura 3.8: Comparación entre el revenue observado y el revenue esperado actualizado en función de las capacidades remanentes del sistema en cada instante del proceso, para una política CDLP en una simulación de 1000 tiras de clientes.

Como se puede observar, la aproximación resulta muy buena, lo que sugiere que el ingreso que según el funcional se esperaba obtener, se está perdiendo a causa de recursos que se encontraban agotados. Esto dá un fuerte elemento de prueba para sospechar que el problema de los productos abiertos con recursos agotados es la causa del bajo rendimiento de CDLP en este caso de simulación sobre datos reales.

### 3.5.4. Reoptimización de CDLP

Como ya vimos, el principal problema que ofrece CDLP consiste en que la política obtenida es estática, en el sentido que no depende de las capacidades remanentes, generando así las ineficiencias que ya describimos. Una solución posible consiste en reoptimizar el problema en puntos intermedios del proceso a fin de ajustar el error y justamente éste fue el enfoque que utilizamos. Para ello efectuamos reoptimizaciones periódicas al inicio de cada ventana de tiempo  $k$  y siempre que se violara alguna restricción de capacidad. Podemos ver la mejora que esto implica sobre las instancias para las que habíamos graficado los resultados de CDLP en las figuras 3.5 y 3.7. El

resultado de aplicar la reoptimización, se puede apreciar en 3.9 y 3.10.

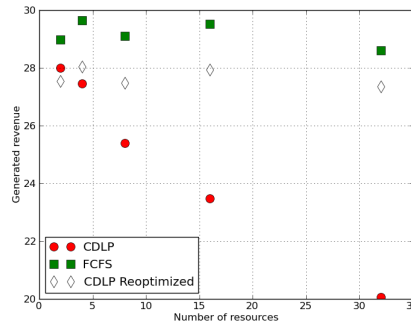


Figura 3.9: Revenue generado en función de la cantidad de recursos en los que se divide un único producto para una política obtenida por CDLP (con y sin reoptimizaciones) contra una FCFS

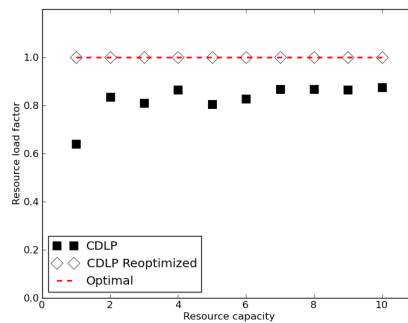


Figura 3.10: Comparación entre los load factors esperados y los simulados para recursos de distintas capacidades bajo una política CDLP (con y sin reoptimizaciones)

### 3.5.5. Comparación con otros trabajos del área

Este resultado negativo de CDLP respecto a una política FCFS es particularmente llamativo si se lo compara con otros trabajos del área como por ejemplo [LvR08], [BMDV09] o [CV09] que reportan valores relativamente buenos para CDLP. Cabe pensar entonces que el escenario que aquí analizamos puede estar particularmente mal condicionado para este método de control de capacidad. En particular, detectamos en la sección 3.5.1 que un número elevado de recursos para implementar pocas clases de tarifa genera un caso patológico para CDLP. Conjeturamos que tener igual número de recursos que clases de tarifas, como tenemos en nuestro escenario, contribuye a degradar la performance de CDLP, mientras que por su parte, FCFS resuelve en forma exitosa las situaciones en que un recurso se agota, reemplazándolo.

### 3.6. Experimentos computacionales

Para probar los distintos métodos de optimización simulamos distintas realizaciones del proceso de arribo de clientes con los parámetros estimados en el capítulo anterior, aplicando distintas políticas de control de capacidad.

Para ello, debíamos conocer los parámetros del modelo mencionados en la sección 3.3, a saber: el valor de las tarifas para las distintas clases (es decir, para cada tipo de habitación), el número de habitaciones disponibles y la relación de sustitución entre los tipos de habitaciones. Para el caso del hotel 1 pudimos obtener esta información a través del contacto directo con los autores de [FGMN10]. Sin embargo, en el caso de los hoteles restantes, no disponíamos de los valores de lista de las habitaciones, de las capacidades, ni de los reemplazos válidos. De modo que tuvimos que construirlos de la siguiente manera. El precio de cada tipo de habitación lo calculamos como el valor medio de los registros de la base de datos donde este tipo de habitación se vendía a una tarifa de tipo “*rack rate*”. Por su parte, las capacidades las fijamos como la cantidad máxima de ocupación observada para cualquier día de check-in multiplicada por 1,25. Por último, asumimos que todo tipo de habitación es reemplazable por otra de valor igual o mayor.

Un caso excepcional es el del hotel 4, para el cual, todas las habitaciones tenían igual tarifa y, por lo tanto, no había ninguna optimización posible. Por tal razón, decidimos construir a partir de éste un escenario posible, utilizando las mismas tarifas del hotel 2.

Los datos utilizados fueron, entonces, los que se muestran en las tablas 3.1, 3.2, 3.3, 3.4 y 3.5.

Tipo de habitación	Precio	Capacidad
King Room 1	448	31
King Room 3	357	17
King Room 4	405	28
2 Double Beds Room 1	410	26
Queen Room 1	405	12
Suite 1	609	13
Suite 2	508	8
Special Type Room 1	405	17

Tabla 3.1: Datos utilizados para la simulación en el hotel 1



### 3.6. EXPERIMENTOS COMPUTACIONALES

Tipo de habitación	Precio	Capacidad
Standard Room	183	1
King Room 3	156	7
King Room 1	166	3
King Room 2	175	4
2 Queen Beds Room 1	156	6
2 Queen Beds Room 2	156	14
Suite 1	192	2
Suite 2	205	1

Tabla 3.2: Datos utilizados para la simulación en el hotel 2

Tipo de habitación	Precio	Capacidad
Standard Room	123	19
King Room 3	106	47
King Room 1	115	5
2 Double Beds Room 1	115	41
2 Double Beds Room 2	110	3
Regular Bed Room 1	110	9

Tabla 3.3: Datos utilizados para la simulación en el hotel 3

Tipo de habitación	Precio	Capacidad
Standard Room	183	1
King Room 1	166	16
King Room 2	175	1
2 Queen Beds Room 1	156	14
2 Queen Beds Room 2	156	1
Suite 1	192	3
Suite 2	205	11

Tabla 3.4: Datos utilizados para la simulación en el hotel 4

Tipo de habitación	Precio	Capacidad
Double Bed room 1	256	3
King Room 1	150	8
King Room 2	167	11
2 Double Beds Room 1	203	9
2 Double Beds Room 2	165	5
2 Double Beds Room 3	95	1
Suite 1	135	1
Suite 2	244	1

Tabla 3.5: Datos utilizados para la simulación en el hotel 5

Con estos datos, construimos las simulaciones respectivas a cada hotel. Para ello establecimos una ventana de 15 días en la cual llegaban los compradores. Cabe destacar que, dado que habíamos definido distintos segmentos según a qué día de semana correspondía la fecha de check-in, hicimos simulaciones independientes en función de lo que indicaban las particiones que se habían definido en el modelo de demanda. Por ejemplo, en el hotel 1 definimos distintos segmentos según si el cliente llegaba de lunes a jueves con 0 o 1 día de antelación, o bien de lunes a jueves con 2 o más días de antelación, o de viernes a sábado o si llegaba un domingo. Por lo tanto, efectuamos simulaciones para días de arribo de lunes a jueves, viernes a sábado y domingos. La división según la antelación no implicó realizar simulaciones separadas: simplemente, en el ejemplo del hotel 1, para los períodos correspondientes a 0 o 1 día de antelación se utilizaron las estimaciones de un segmento, y para el resto las del otro. Por último, combinamos los resultados para reflejar el ingreso diario a lo largo de una semana. A fin de conseguir esto, siguiendo con el ejemplo del hotel 1, se incluyeron 4 veces los resultados de la partición de lunes a jueves, 2 veces los de la partición de viernes a sábado y una los de la correspondiente al domingo, para de esta manera, ponderar los ingresos de cada partición en función de la cantidad de días de la semana que abarcase.

Así, simulamos 100 realizaciones para cada proceso y calculamos el ingreso resultado de aplicar políticas obtenidas con cada uno de los métodos explicados: CDLP, CDLP con reoptimización y la heurística de simplificación del programa dinámico (rotulada “DP Heur.” en los gráficos). Para esta última heurística utilizamos  $\phi_i(0) = 1, \phi_i(1) = c_i$ , es decir, agregamos todos los estados que implican alguna disponibilidad del recurso en uno sólo.

Cada uno de los procesos los adaptamos a distintas tasas de arribo de los clientes,

correspondientes a distintos *Demand Ratios*(DR). Como ya se mencionó en la sección 3.5, el DR se define como el cociente entre la cantidad media esperada total de clientes y la capacidad total del hotel. En nuestros experimentos simulamos con DR de 0.5, 1, 2 y 4. En cada caso, redimensionamos proporcionalmente las tasas de arribo de modo de obtener el DR deseado.

Los resultados los expresamos en porcentaje de mejora respecto del ingreso obtenido por una política FCFS y los presentamos mediante *box plots*. Estos gráficos permiten visualizar la distribución de los resultados graficando una caja que se extiende entre el primer y el tercer cuartil de los datos. A la diferencia entre los valores de estos dos cuartiles se lo denomina *rango intercuartílico*. Asimismo, la caja se encuentra dividida por un trazo rojo en el valor de la mediana. Por último, se extienden dos brazos, denominados “bigotes”, uno desde el primer cuartil hacia abajo hasta el menor valor que no sea inferior al valor del primer cuartil menos 1.5 del rango intercuartílico, y otro desde el tercer cuartil hasta arriba hasta el mayor valor que no exceda el valor del primer cuartil sumado a 1.5 del rango intercuartílico.

#### 3.6.1. Resultados

Es posible efectuar diversas observaciones sobre los resultados obtenidos. Por un lado, se puede apreciar que, en general, el método CDLP con reoptimizaciones y la heurística basada en programación dinámica tienden a mejorar su performance a medida que se incrementa el DR. Esto es consistente con la idea intuitiva de que al incrementarse el número de clientes los controles de capacidad se tornan más efectivos para optimizar los ingresos. Además, comparando estos dos métodos, se puede observar —salvando los casos del hotel 5 donde tienen un desempeño similar y el hotel 3, donde ningún método de control de capacidad ofrece un buen resultado— que la heurística basada en programación dinámica ofrece resultados ligeramente superiores. Por su parte, CDLP sin reoptimizaciones es dominado por FCFS y su rendimiento empeora a medida que aumenta el DR.

En cuanto a MEMUR, se puede apreciar en todos los casos una pérdida de ingresos respecto a FCFS para DR 2, cosa que no se observa ni con DR 1 ni con DR 4. Sin embargo, existen casos particulares que impiden extraer una conclusión general sobre su desempeño: en la figura 3.16 se puede observar que su rendimiento empeora a medida que aumenta el DR, mientras que en la figura 3.17 se ve una mejora sostenida para DR 1 o mayores. De todos modos, se puede afirmar que para los casos analizados la heurística basada en programación dinámica domina a MEMUR.

Por último, cabe destacar además el caso del hotel 3 en el que, como ya se

mencionó, ningún método logra obtener mejor rendimiento que FCFS. En la tabla 3.6 se puede apreciar el desvío estándar de los precios ponderados por número de habitaciones. Se puede apreciar que el hotel 3 presenta una muy baja dispersión de precios, lo cual puede explicar el bajo rendimiento de las políticas obtenidas. Si se piensa que cuando los productos tienen igual precio FCFS es una política óptima, no es difícil pensar que si los precios son muy similares, la política FCFS no debe estar muy lejos de ser la óptima. Por su parte, el hotel 1 que presenta los mejores resultados tiene una muy alta dispersión de precios. Mientras que los hoteles 2, 4 y 5 que ofrecen rendimientos regulares presentan dispersiones intermedias.

Hotel	Desvío estándar de los precios
1	71.74
2	14.27
3	6.18
4	18.50
5	34.57

Tabla 3.6: Desvíos estándar de los precios para cada uno de los hoteles, ponderados por cantidad de habitaciones correspondientes a cada precio

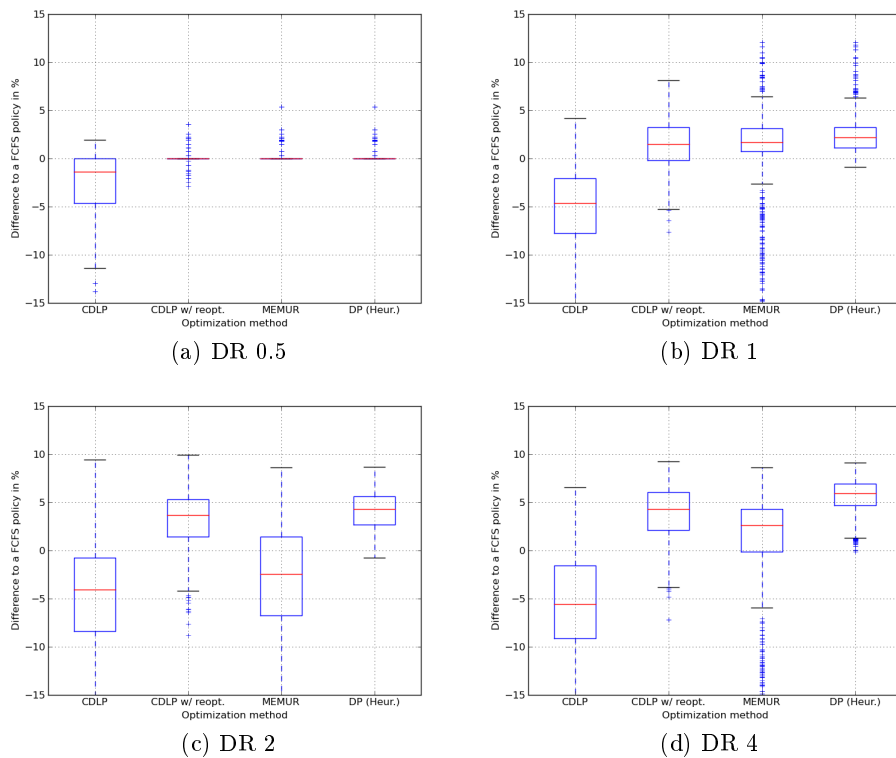


Figura 3.11: Resultados del ingreso diario a lo largo de una semana para el hotel 1

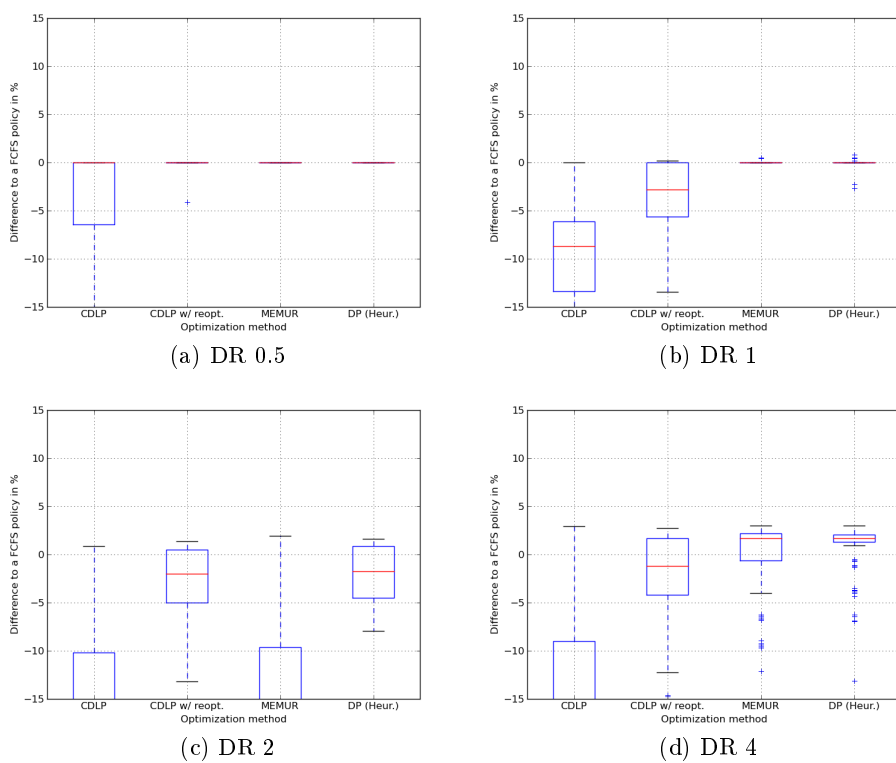


Figura 3.12: Resultados del ingreso diario a lo largo de una semana para el hotel 2

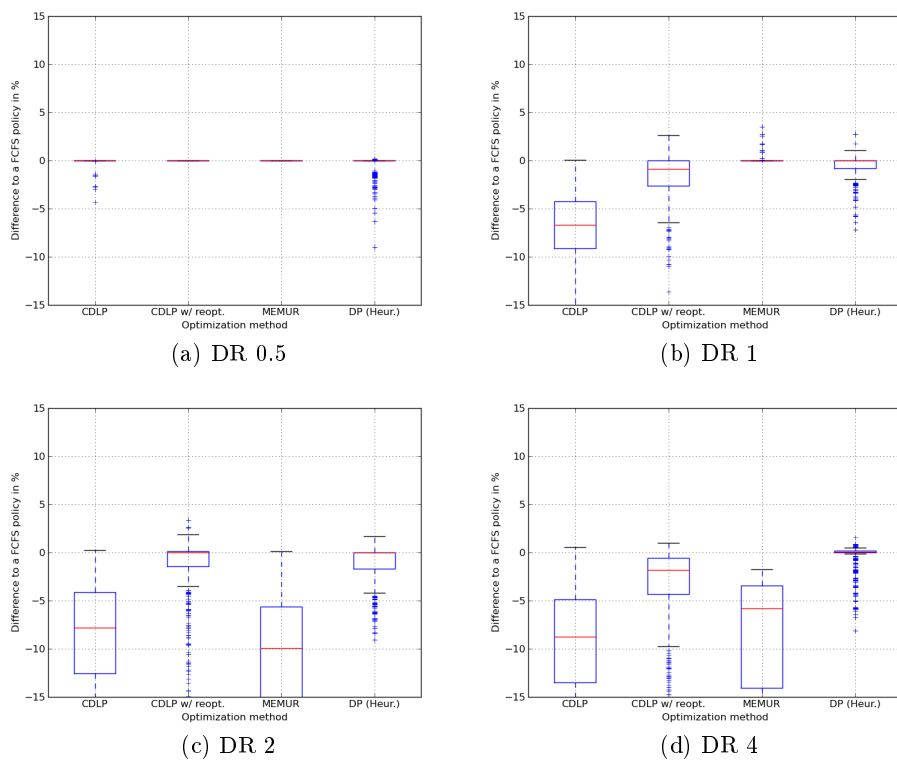


Figura 3.13: Resultados del ingreso diario a lo largo de una semana para el hotel 3

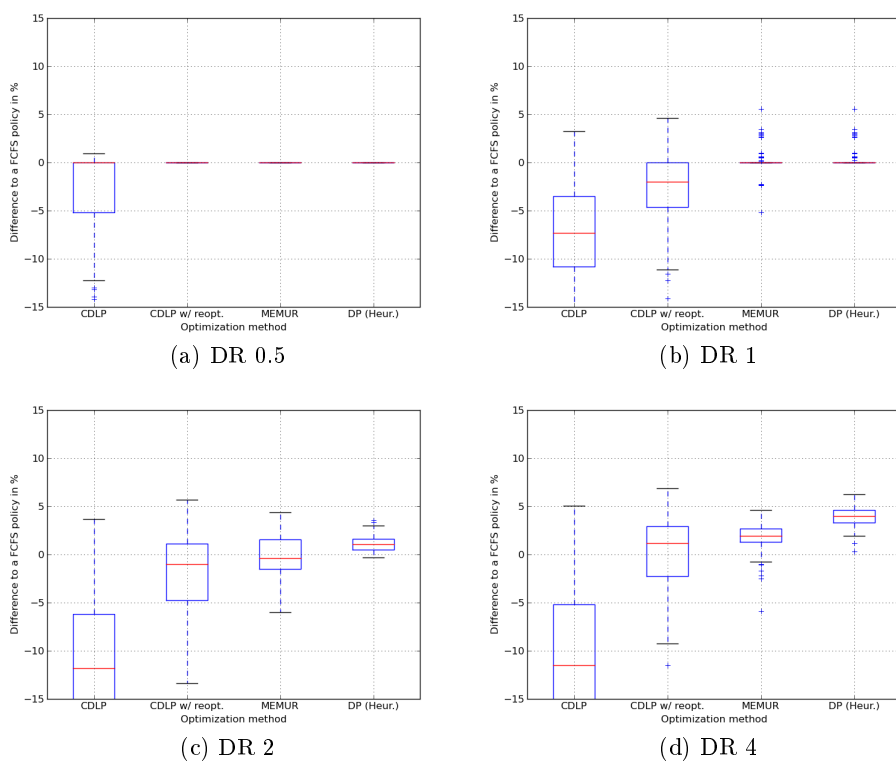


Figura 3.14: Resultados del ingreso diario a lo largo de una semana para el hotel 4

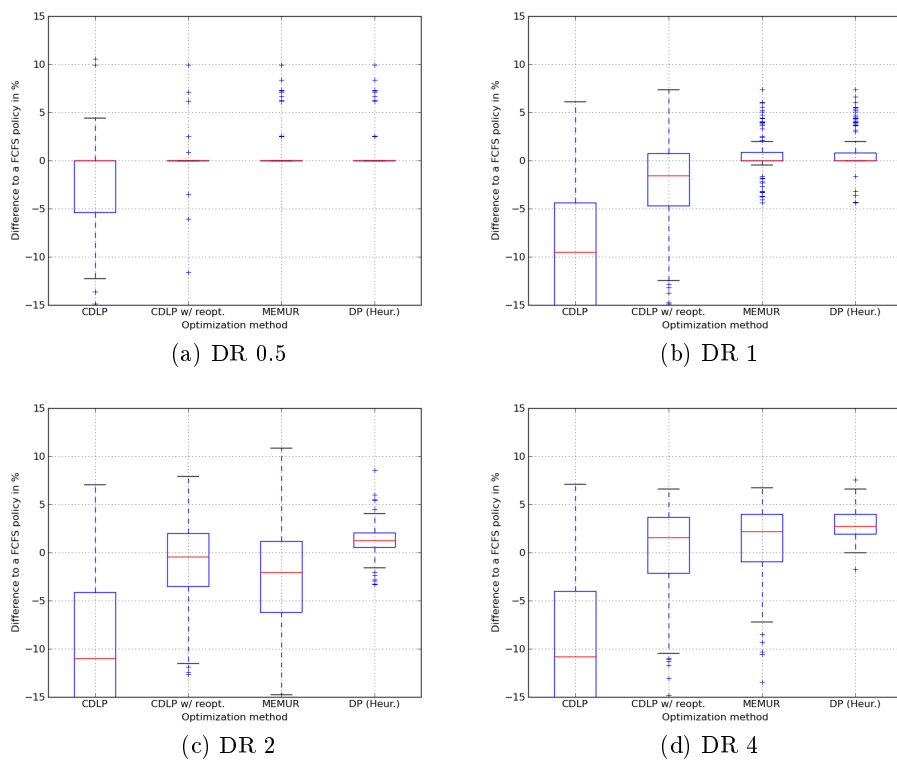


Figura 3.15: Resultados del ingreso diario a lo largo de una semana para el hotel 5



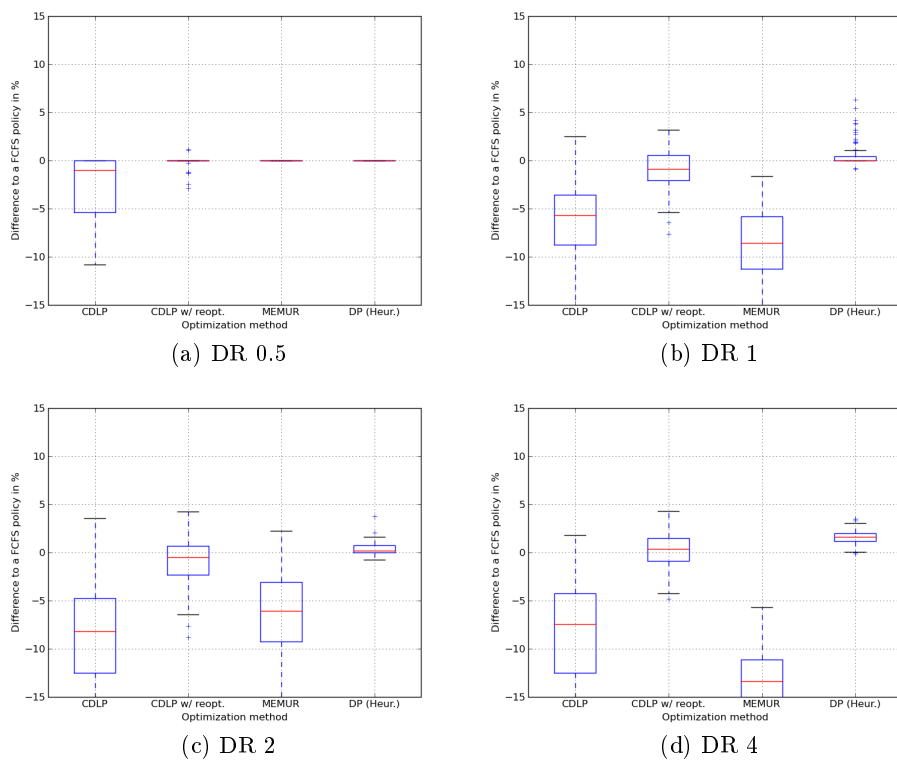


Figura 3.16: Resultados para check-in en día domingo en el hotel 1

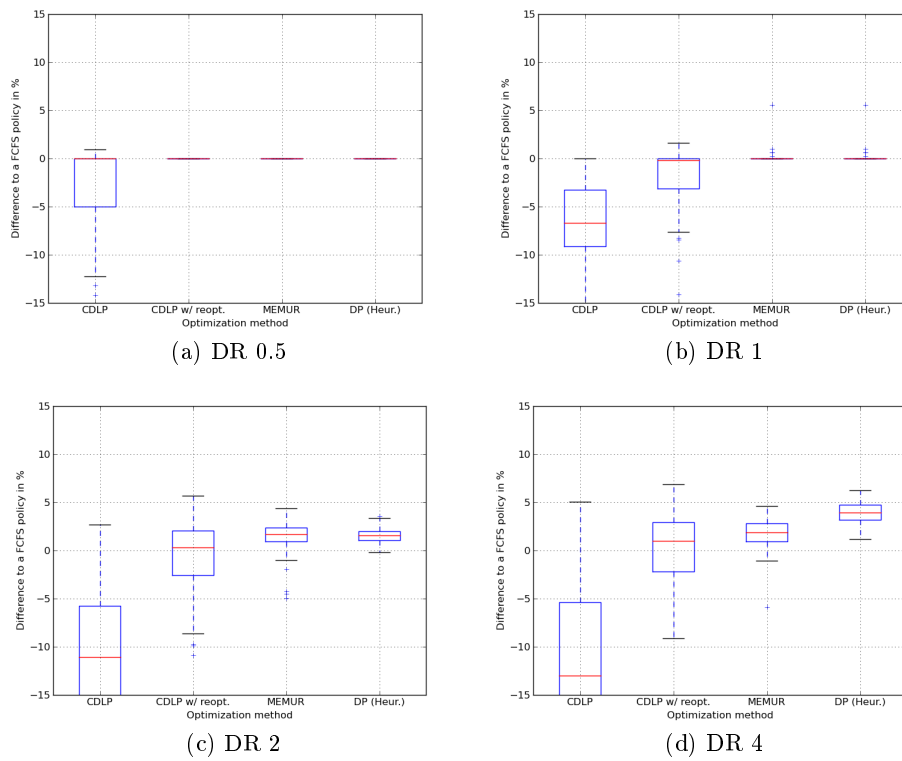


Figura 3.17: Resultados para check-in en días viernes, sábado o domingo en el hotel 4

### 3.6.1.1. Tiempos de ejecución

Generalmente el tiempo de ejecución de los algoritmos viene dado por un procedimiento inicial donde se calcula la política para toda la simulación. Sin embargo, CDLP con reoptimización es una excepción a esto, donde al aplicarse la política, el algoritmo puede volver a ejecutarse. Con lo cual medimos el tiempo de ejecución de los algoritmos como la suma del tiempo de inicialización más el promedio del tiempo de aplicación de la política. Los resultados son bastante similares entre las distintas simulaciones, con lo cual mostramos solamente el tiempo de ejecución para la simulación del Hotel 1 de Lunes a Jueves con DR 1 (ver figura 3.18).

Se puede observar que el tiempo de ejecución de la heurística aquí desarrollada es muy superior a la de los otros algoritmos. Sin embargo, cabe destacar que este cálculo se efectúa una única vez, obteniendo una política que define para cualquier instante y capacidad remanente, qué *offer set* mostrar. Por su parte, CDLP con reoptimizaciones, vuelve a recalculer la política periódicamente a lo largo de todo el proceso. El tiempo de ejecución de CDLP sin reoptimizaciones es muy inferior a los dos métodos anteriores, y el de MEMUR, es prácticamente despreciable en comparación a los otros.

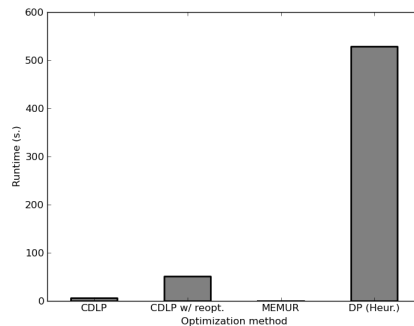


Figura 3.18: Tiempo de ejecución de los algoritmos en la simulación correspondiente al Hotel 1 de Lunes a Jueves con DR 1



## Capítulo 4

# Conclusiones

En este trabajo transitamos todo el proceso de un sistema de *Revenue Management*. Comenzamos por elaborar un modelo de demanda y estimar sus parámetros a partir de los datos de ventas recolectados de los sistemas de cinco hoteles. Para interpretar estos datos, debimos definir qué significaba para nosotros un producto y cómo se definía un período. Además, buscamos caracterizar los clientes en distintos segmentos no solapados, de manera tal de poder ajustar lo mejor posible los datos observados. Abordamos también, un procedimiento para descomponer los valores escalares con los que estimamos las preferencias de los clientes por los distintos productos en valores de preferencias por atributos de los productos. Esto último podía ser de interés para analizar el impacto de modificar alguna característica de un producto sobre las preferencias de los clientes.

Luego, utilizamos el modelo de demanda para optimizar el control de capacidad y, posteriormente, simular el proceso de arribo de clientes a fin de observar el resultado de aplicar las distintas políticas. Consideramos distintas heurísticas, ya que el problema de control de capacidad es un problema difícil de resolver en forma exacta. Estudiamos en profundidad CDLP, para el cual propusimos una heurística de generación de columnas y estudiamos los factores que pudieran explicar los resultados subóptimos que observamos. Además propusimos una nueva heurística basada en la simplificación de los estados del problema de programación dinámica.

Al observar los resultados de las simulaciones, sobresale, por un lado, el rendimiento negativo de la política basada en CDLP respecto a una política FCFS, y por otro, las mejoras de 2 % a 5 % que ofrecen las políticas de CDLP con reoptimizaciones y la heurística basada en programación dinámica para DR 2 y 4. Por el contrario, para DR 0.5, ningún método mejora a FCFS, lo cual es esperable en condiciones de baja demanda. En cuanto a MEMUR, los resultados son muy variables como para

decidir una tendencia clara, aunque en ningún caso mejora los resultados obtenidos por el DP heurístico.

Sin embargo, si se considera que el punto de referencia fue el de no hacer ningún tipo de control de capacidad (es decir, utilizar una política FCFS), las mejoras obtenidas no han sido tan dramáticas como podía esperarse. Una posible explicación a esto puede encontrarse en que las decisiones de control, en función de los productos que definimos, se limitan a ofrecer o no un tipo de habitación por su precio normal y si se acepta reemplazar el producto por uno mejor. Sin embargo, es de esperarse que si se definen, por ejemplo, tres clases de tarifas por cada tipo de habitación, la aplicación de controles de capacidad efectivos generen un mayor margen ya que podrían aprovechar situaciones de elevada demanda abriendo solamente los productos de tarifa más alta. Queda, pues, como trabajo a futuro explorar esta posibilidad. Además, sería interesante probar el rendimiento de la heurística basada en programación dinámica con otras funciones de agrupación de estados. También se podría comparar el rendimiento de esta heurística con la heurística de descomposición propuesta por Liu y Van Ryzin en [LvR08], también basada en una simplificación del problema de programación lineal.

## Apéndice A

# Resultados de la Estimación

Se presentan aquí los resultados remanentes de la sección 2.6.

### A.1. Hotel 1

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	1	16.5923	0.00528846	0.0221147
King Room 3	13	100.16	0.0110905	0.133496
2 Double Beds Room 1	8	18.682	0.00559253	0.0248999
Queen Room 2	1	5.33144	0.0030521	0.00710591
Special Type Room 1	5	10.1081	0.00417097	0.0134724
King Room 4	28	23.8318	0.00626306	0.0317637
Suite 1	4	4.53106	0.00281722	0.00603913
Suite 2	9	8.33417	0.003798	0.011108

Tabla A.1: Hotel 1. Lunes a Jueves, reservas hechas de 0 a 1 día previo al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	4	317.033	0.00288964	0.0555404
Queen Room 1	5	160.126	0.00213597	0.0280522
King Room 3	79	500.236	0.0034521	0.0876354
2 Double Beds Room 1	9	30.1845	0.000955971	0.00528797
Queen Room 2	18	74.5425	0.00148711	0.013059
Special Type Room 1	29	84.4358	0.00157909	0.0147921
King Room 4	75	110.388	0.00179461	0.0193387
Suite 1	11	16.3313	0.000705381	0.00286106
Suite 2	18	16.8989	0.000717441	0.00296049
King Room 2	3	116.861	0.00184366	0.0204726

Tabla A.2: Hotel 1. Lunes a Jueves, reservas hechas de 2 días previos al check-in en adelante.

APÉNDICE A. RESULTADOS DE LA ESTIMACIÓN

---

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	13	157.646	0.00625905	0.0894689
Queen Room 1	5	58.4555	0.00415126	0.0331753
King Room 3	47	87.6643	0.00496474	0.0497522
2 Double Beds Room 1	28	65.5769	0.004372	0.0372169
Queen Room 2	3	3.67749	0.00108544	0.00208709
Special Type Room 1	4	4.72361	0.00122923	0.00268079
King Room 4	14	16.7037	0.00229126	0.00947988
Suite 1	9	15.8669	0.00223452	0.00900496
Suite 2	11	9.84905	0.00176833	0.00558963
King Room 2	3	20.3416	0.00252165	0.0115445

Tabla A.3: Hotel 1. Viernes y Sábados, reservas hechas con cualquier antelación.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	17	53.1573	0.0163614	0.143616
Queen Room 1	5	8.9831	0.00788251	0.0242697
King Room 3	13	15.27	0.0100763	0.0412551
Special Type Room 1	6	5.30639	0.00612648	0.0143363
King Room 4	6	5.6126	0.00629495	0.0151636
Suite 2	1	1.38178	0.00316301	0.00373316
King Room 2	1	2.82285	0.00450171	0.00762652

Tabla A.4: Hotel 1. Domingos, reservas hechas con cualquier antelación.

## A.2. Hotel 2

Producto	Compras	$N_j$	ASE	$v_j$
King Room 3	7	13.6403	0.0047177	0.0177372
King Room 1	10	14.1709	0.00480519	0.0184272
King Room 2	2	2.51222	0.0020544	0.00326678
2 Queen Beds Room 1	25	68.9524	0.00979524	0.0896627
2 Queen Beds Room 2	63	92.0678	0.0109027	0.119721
Standard Room	1	0.911467	0.00124001	0.00118523

Tabla A.5: Hotel 2. Cualquier día, reservas hechas con cualquier antelación.



Producto	Compras	$N_j$	ASE	$v_j$
2 Queen Beds Room 2	34	52.4359	0.0152534	0.130289
Standard Room	1	0.920278	0.00237774	0.00228665
King Room 3	4	7.80813	0.00679613	0.0194011
King Room 1	5	6.12633	0.00604817	0.0152223
King Room 2	2	2.89931	0.00419782	0.00720401
2 Queen Beds Room 1	12	29.3989	0.0123644	0.0730486
Suite 1	1	1.02543	0.0025092	0.00254791

Tabla A.6: Hotel 2. Viernes y Sábados, reservas hechas con cualquier antelación.

### A.3. Hotel 3

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	77	76.3375	0.00860873	0.0812069
King Room 3	5	13.6598	0.00388135	0.0145311
2 Double Beds Room 1	131	130.152	0.010564	0.138454
Regular Bed Room 1	12	11.8838	0.00362639	0.0126418
Standard Room	3	2.97675	0.00183029	0.00316663

Tabla A.7: Hotel 3. Lunes a Jueves, reservas hechas de 0 a 1 día previo al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	79	79.4026	0.0102313	0.100418
King Room 3	8	11.5816	0.00424838	0.0146469
2 Double Beds Room 1	101	100.449	0.0111777	0.127035
Regular Bed Room 1	2	2.09028	0.0018242	0.00264352
Standard Room	4	4.15598	0.00256631	0.00525596

Tabla A.8: Hotel 3. Lunes a Jueves, reservas hechas de 2 a 8 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	72	71.9647	0.0180155	0.177797
King Room 3	1	1.12416	0.00261416	0.00277737
2 Double Beds Room 1	27	26.989	0.0121904	0.0666792
Standard Room	1	1.11175	0.00259974	0.00274671

Tabla A.9: Hotel 3. Lunes a Jueves, reservas hechas de 9 a 19 días previos al check-in.

APÉNDICE A. RESULTADOS DE LA ESTIMACIÓN

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	20	19.9704	0.0331778	0.171402
King Room 3	1	1.07878	0.00885369	0.00925901
2 Double Beds Room 1	7	7	0.021683	0.0600798
Standard Room	1	1.07878	0.00885369	0.00925901

Tabla A.10: Hotel 3. Lunes a Jueves, reservas hechas de 20 a 27 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	13	12.9111	0.0398922	0.157783
2 Double Beds Room 1	3	3	0.0207329	0.0366622
Standard Room	4	4.54592	0.025242	0.0555544

Tabla A.11: Hotel 3. Lunes a Jueves, reservas hechas de 28 días previos al check-in en adelante.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	30	29.9355	0.0168809	0.0999957
King Room 3	5	6.0355	0.00808396	0.0201608
2 Double Beds Room 1	37	36.8871	0.0183603	0.123217
Regular Bed Room 1	2	1.98387	0.00468196	0.00662687

Tabla A.12: Hotel 3. Viernes y Sábados, reservas hechas 0 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	21	21	0.0427127	0.21
2 Double Beds Room 1	4	4	0.0197484	0.04

Tabla A.13: Hotel 3. Viernes y Sábados, reservas hechas 1 día previo al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	11	10.8833	0.0276034	0.0963469
2 Double Beds Room 1	14	16.4733	0.0329025	0.145833
Standard Room	1	0.883334	0.00828419	0.00781989

Tabla A.14: Hotel 3. Viernes y Sábados, reservas hechas de 2 a 4 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	14	13.953	0.0351787	0.143256
2 Double Beds Room 1	9	8.96473	0.0291324	0.0920414
2 Double Beds Room 2	1	1.43203	0.0121859	0.0147027

Tabla A.15: Hotel 3. Viernes y Sábados, reservas hechas de 5 a 10 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	33	32.9893	0.0223396	0.144575
King Room 3	2	2.03733	0.00621419	0.00892857
2 Double Beds Room 1	21	21	0.0186765	0.0920321
Standard Room	1	1.01867	0.00440867	0.00446429

Tabla A.16: Hotel 3. Viernes y Sábados, reservas hechas de 11 días previos al check-in en adelante.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	78	77.9923	0.0148782	0.153672
King Room 3	3	2.9547	0.00336958	0.00582179
2 Double Beds Room 1	43	43.9877	0.0120362	0.0866711
Regular Bed Room 1	1	1.0082	0.00197497	0.00198651
Standard Room	1	0.938297	0.0019055	0.00184877

Tabla A.17: Hotel 3. Domingos, reservas hechas con cualquier antelación.

## A.4. Hotel 4

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	53	75.1786	0.0156533	0.159939
King Room 2	4	4.94603	0.00468761	0.0105224
2 Queen Beds Room 1	30	28.749	0.0107792	0.061162
Suite 2	7	7.58872	0.00577723	0.0161446
Standard Room	1	1.04932	0.00217501	0.00223237

Tabla A.18: Hotel 4. Lunes a Jueves, reservas hechas con cualquier antelación.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 1	29	28.229	0.0171793	0.102162
King Room 2	2	2.11445	0.00522258	0.00765229
2 Queen Beds Room 1	23	28.9359	0.0173379	0.10472
2 Queen Beds Room 2	1	2.1324	0.00524436	0.00771724
Suite 1	4	5.83316	0.00855652	0.0211104
Suite 2	2	1.8342	0.00486912	0.00663802

Tabla A.19: Hotel 4. Viernes, Sábado y Domingo; reservas hechas con cualquier antelación.

### A.5. Hotel 5

Producto	Compras	$N_j$	ASE	$v_j$
King Room 2	6	23.9788	0.0168804	0.0913004
King Room 1	8	27.7689	0.0178469	0.105732
2 Double Beds Room 1	5	4.2373	0.00771185	0.0161337
2 Double Beds Room 2	2	3.76897	0.00728641	0.0143505
Double Bed Room 1	2	1.8465	0.00513788	0.00703065
Suite 2	3	4.05854	0.00755267	0.0154531

Tabla A.20: Hotel 5. Lunes a Jueves, reservas hechas de 0 a 2 días previos al check-in.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 2	9	92.3866	0.0169039	0.213735
King Room 1	3	7.90604	0.00638642	0.0182905
2 Double Beds Room 1	4	3.47357	0.00427742	0.00803607
2 Double Beds Room 2	2	2.0949	0.00333243	0.00484652
Double Bed Room 1	1	1.10303	0.00242362	0.00255185
Suite 2	1	1.09778	0.00241787	0.0025397

Tabla A.21: Hotel 5. Lunes a Jueves, reservas hechas de 3 días previos al check-in en adelante.

Producto	Compras	$N_j$	ASE	$v_j$
King Room 2	40	44.5416	0.0127659	0.096692
King Room 1	21	20.8789	0.00938503	0.0453245
2 Double Beds Room 1	32	33.016	0.011393	0.0716719
2 Double Beds Room 3	3	3.11331	0.00380028	0.00675844
2 Double Beds Room 2	7	6.98779	0.0056369	0.0151693
Double Bed Room 1	1	0.959447	0.00212123	0.00208279
Suite 2	2	2.15063	0.00316629	0.00466864
Suite 1	3	3.51591	0.00403438	0.00763243

Tabla A.22: Hotel 5. Viernes, Sábado y Domingo; reservas hechas con cualquier antelación.

## Apéndice B

# Resultados de la Optimización

Se presentan aquí los resultados de la aplicación de las distintas políticas de control de capacidad, desagregados según el día de semana de la fecha de check-in.

### B.1. Hotel 1

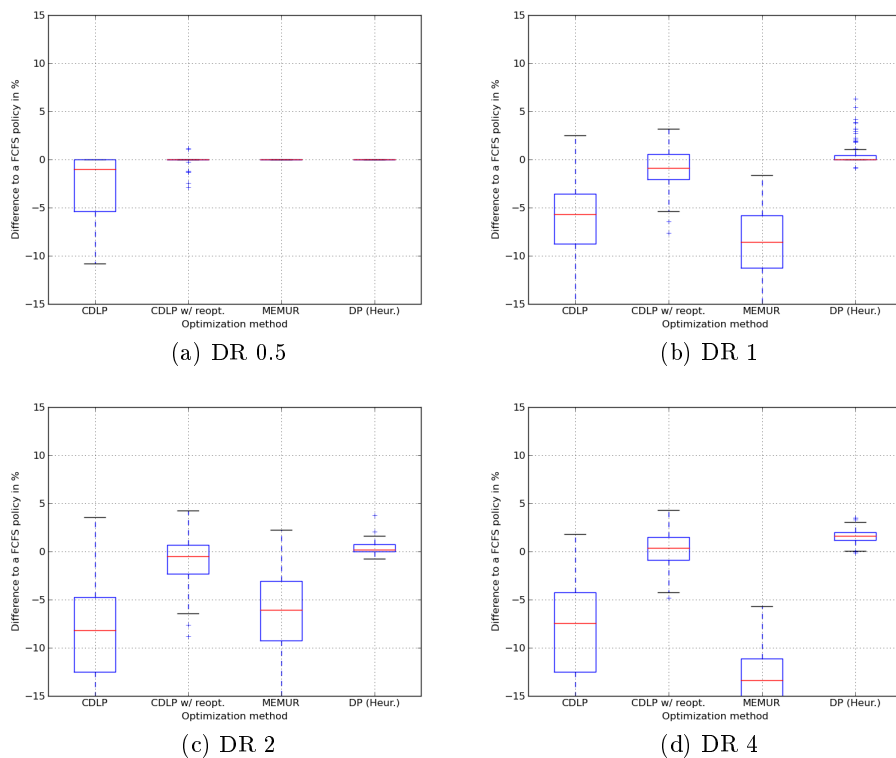
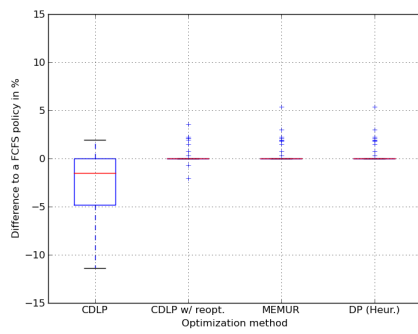
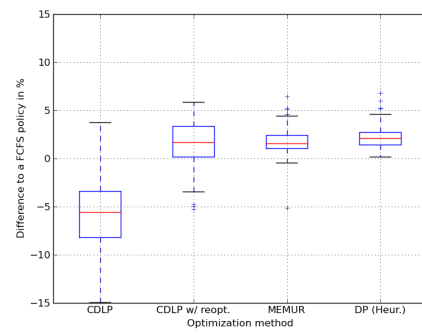


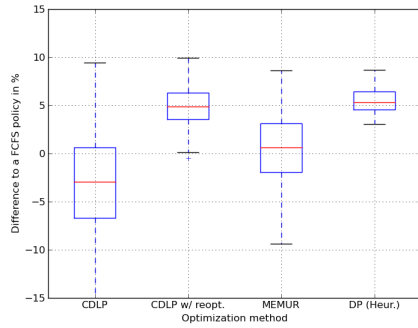
Figura B.1: Resultados para check-in en día domingo en el hotel 1



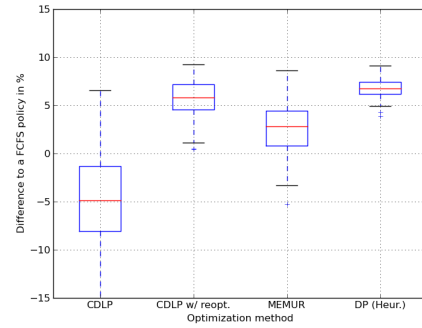
(a) DR 0.5



(b) DR 1



(c) DR 2



(d) DR 4

Figura B.2: Resultados para check-in en días lunes, martes, miércoles o jueves en el hotel 1

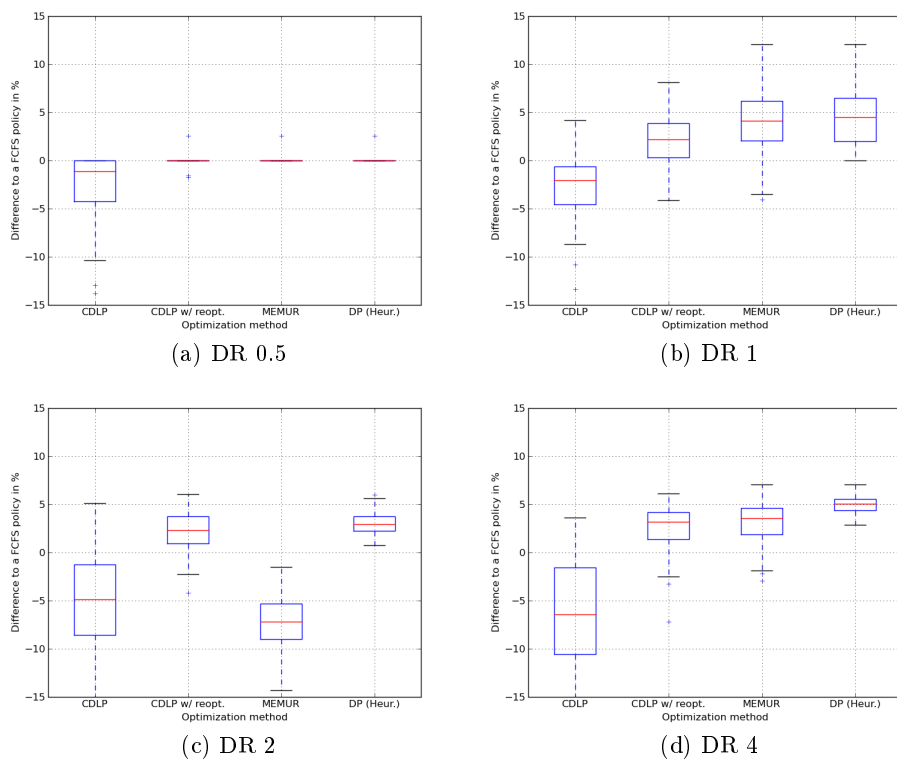
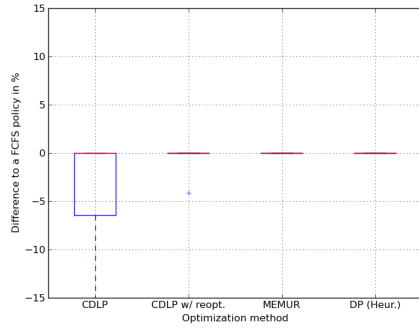
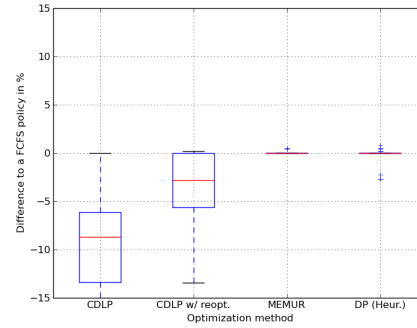


Figura B.3: Resultados para check-in en días viernes o sábado en el hotel 1

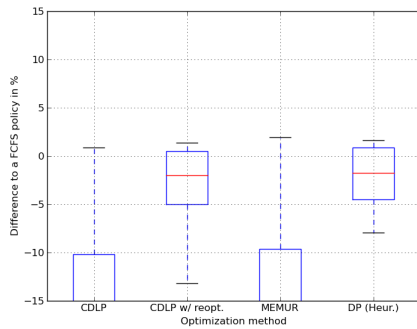
## B.2. Hotel 2



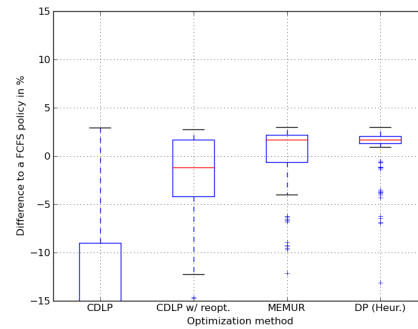
(a) DR 0.5



(b) DR 1



(c) DR 2



(d) DR 4

Figura B.4: Resultados para check-in en cualquier día en el hotel 2

## B.3. Hotel 3



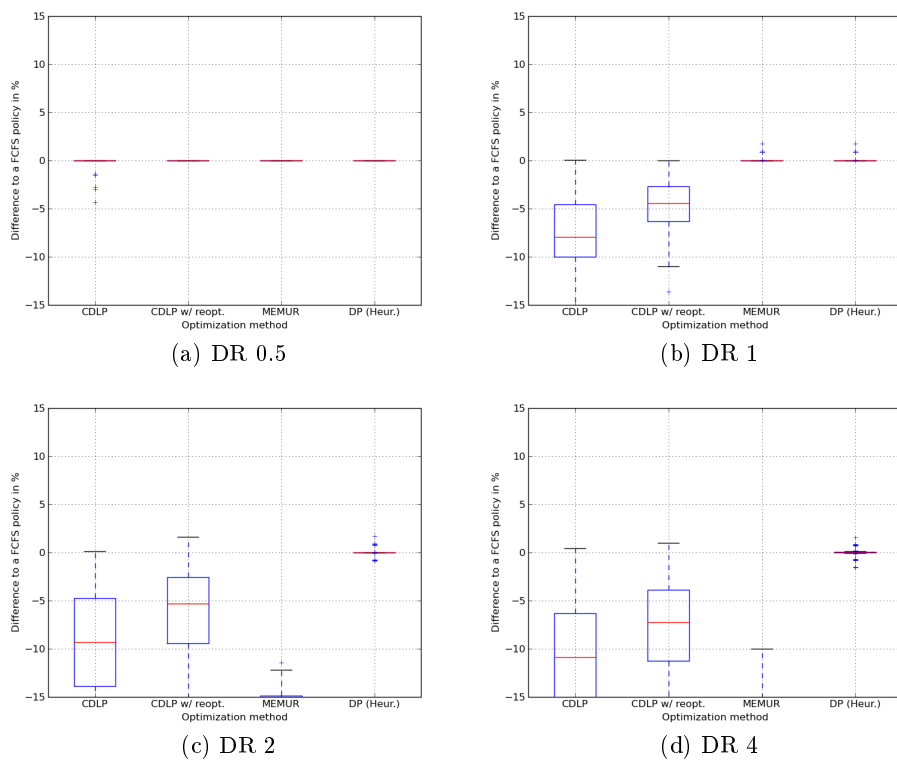


Figura B.5: Resultados para check-in en día domingo en el hotel 3

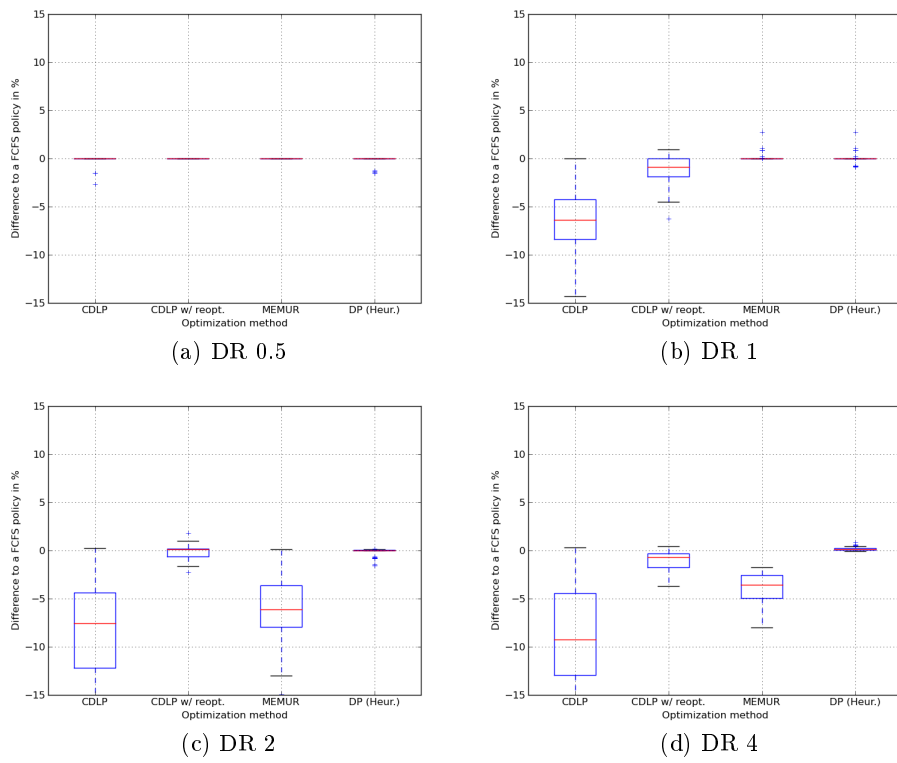


Figura B.6: Resultados para check-in en días lunes, martes, miércoles o jueves en el hotel 3

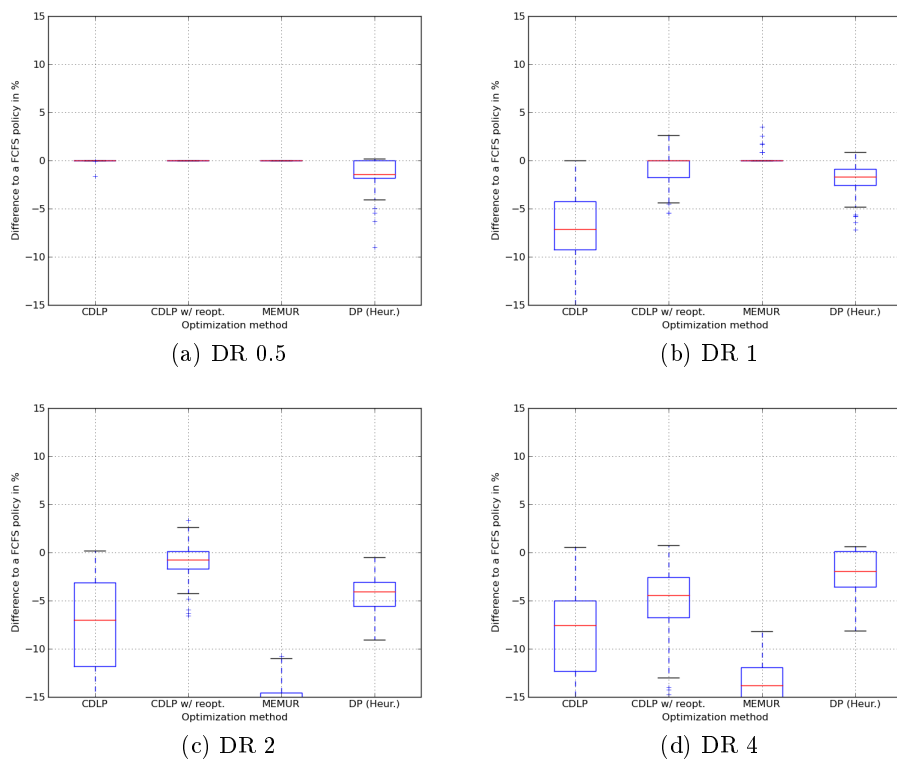
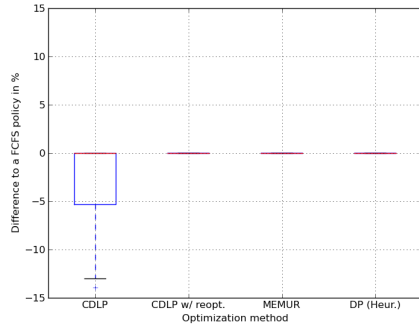
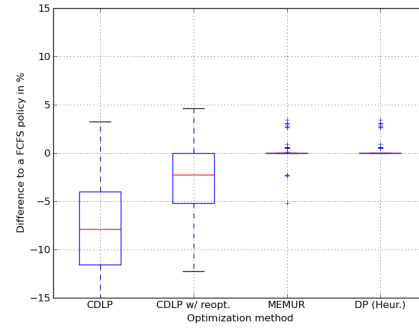


Figura B.7: Resultados para check-in en días viernes o sábado en el hotel 3

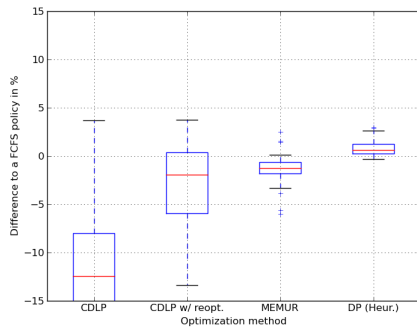
B.4. Hotel 4



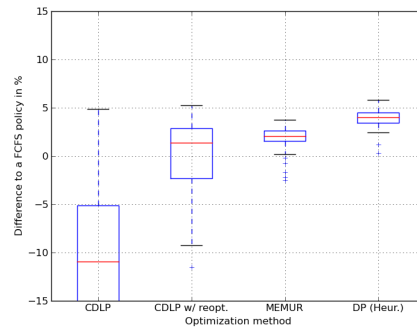
(a) DR 0.5



(b) DR 1



(c) DR 2



(d) DR 4

Figura B.8: Resultados para check-in en días lunes, martes, miércoles o jueves en el hotel 4

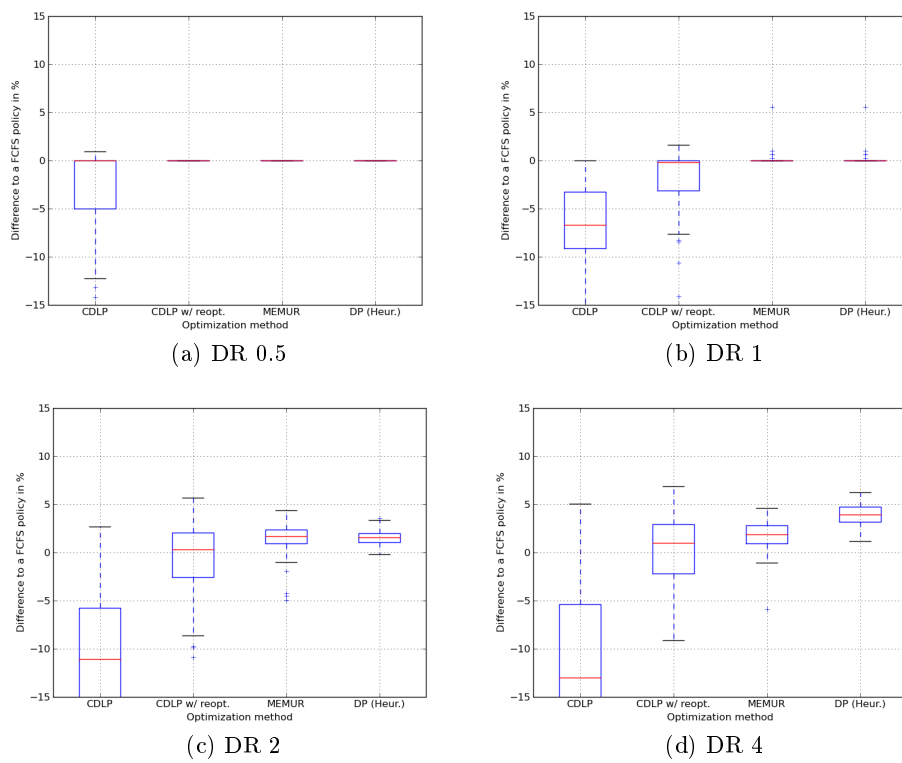
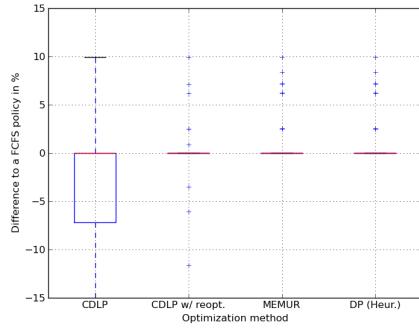
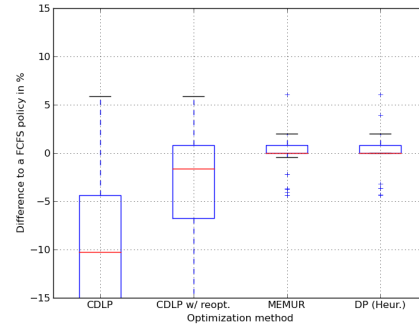


Figura B.9: Resultados para check-in en días viernes, sábado o domingo en el hotel 4

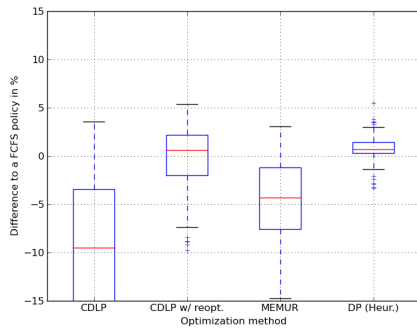
B.5. Hotel 5



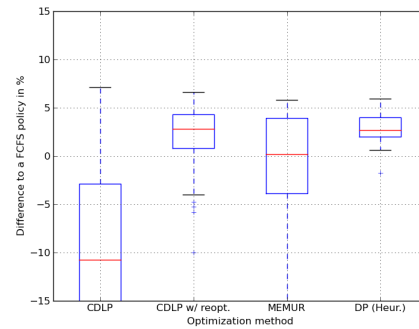
(a) DR 0.5



(b) DR 1



(c) DR 2



(d) DR 4

Figura B.10: Resultados para check-in en días lunes, martes, miércoles o jueves en el hotel 5

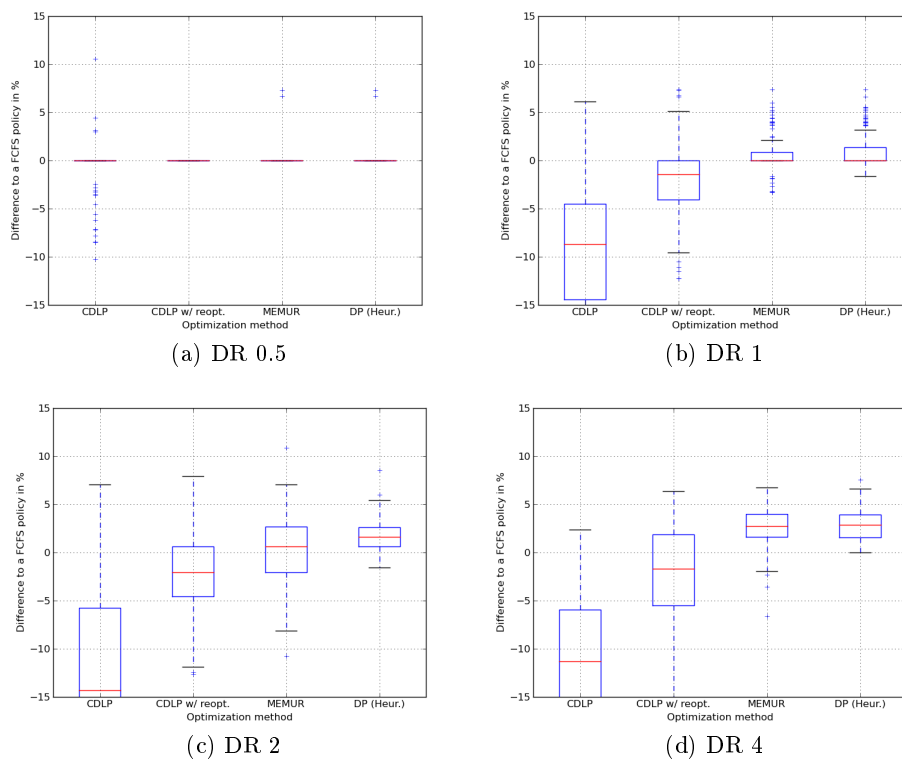


Figura B.11: Resultados para check-in en días viernes, sábado o domingo en el hotel 5





# Bibliografía

- [BD66] R.E. Bellman and S.E. Dreyfus. *Applied dynamic programming*. Princeton University Press, 1966.
- [Bel89] P. P. Belobaba. Or practice—application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37:183–197, 1989.
- [BFG09] T. Bodea, M. Ferguson, and L. Garrow. Choice-based revenue management: Data from a major hotel chain. *Manufacturing & Service Operations Management*, 11:356–361, 2009.
- [BMDV09] J. Miranda Bront, I. Mendez-Diaz, and G. Vulcano. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57:769–784, 2009.
- [CV09] J. M. Chaneton and G. Vulcano. Computing bid-prices for revenue management under customer choice behavior. 2009. Forthcoming in *Operations Research*.
- [DF04] Zhengzhu Feng Department and Zhengzhu Feng. Dynamic programming for structured continuous markov decision problems. In *In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 154–161, 2004.
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
- [FGMN10] M. Ferguson, L. Garrow, M. Meterelliyoz, and J. Newman. Multi-resource revenue management with upgrades: A comparison of esmr-based and choice-based algorithms. Working paper, College of Management, Georgia Institute of Technology, Atlanta, 2010.
- [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.
- [LvR08] Q. Liu and G.J. van Ryzin. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10:288–310, 2008.

## BIBLIOGRAFÍA

---

- [TvR04] K. Talluri and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, New York, NY, 2004.
- [VvRR08] G. Vulcano, G. van Ryzin, and R. Ratliff. Estimating primary demand for substitutable products from sales transaction data. Working paper, Stern School of Business, New York University, New York, 2008.

# Agradecimientos

Quiero agradecer a mis directores, Gustavo y Paula, por su tiempo, ayuda y dedicación; y a los jurados, Juan José Miranda Bront y Tomás Tetzlaff. También a la Universidad Nacional de General Sarmiento, por brindarme el marco laboral para llevar a cabo este trabajo.

También quiero hacer un agradecimiento especial

- A toda mi familia. Principalmente a mi viejo, que me apoyó en todo momento y fue fundamental para que pudiera llegar hasta acá. También, agradezco especialmente a mi hermana Liliana por acompañarme en tantas etapas de mi vida y a mi prima/sobrina Maru por los domingos pochocleros.
- A Alessandra, que a pesar de la distancia, me acompañó y sostuvo en estos últimos años de la carrera.
- A todos mis amigos y compañeros. A los de siempre: Jhompi, Gabo, Mariano, Andrés, Pablo –el bueno– y Pablo –el malo–. Y a los de la carrera: Lucio, Daniel, David, El Pocho, Fran, Manix, El Marino, Arti, Nigel, Wawa, Nina, Pachi, Cesaro y Gaboto.
- Repito el agradecimiento para los miembros del grupo de TP de ayer, de hoy y de siempre: David, Lucio y Daniel.
- Otro agradecimiento repetido para los que me escucharon atentamente con sugerencias y consejos útiles mientras trabajaba en esta tesis: Nina, Mariano, Pablo –el bueno– y Diego.