Article:

Li, Dong orcid.org/0000-0003-3883-8688 and Pang, Zhan (2017) Dynamic Booking Control for Car Rental Revenue Management:A Decomposition Approach. European Journal of Operational Research. 850–867. ISSN: 0377-2217

https://doi.org/10.1016/j.ejor.2016.06.044

Accepted Manuscript

Dynamic Booking Control for Car Rental Revenue Management: A Decomposition Approach

Dong Li, Zhan Pang

PII: S0377-2217(16)30484-2
DOI: 10.1016/j.ejor.2016.06.044
Reference: EOR 13796

To appear in: *European Journal of Operational Research*

Received date: 10 March 2015
Revised date: 19 June 2016
Accepted date: 20 June 2016

Highlights

- We consider dynamic booking controls for car rental revenue management.
- A decomposition approach and two dynamic heuristic policies are proposed.
- Their strong revenue performance is in contrast to the PNLP policy used in practice.
- The proposed policies are robust to vehicle transhipment cost while PNLP not.
- The PNLP policy might still work well in peak seasons and downtown rental stations.

Dynamic Booking Control for Car Rental Revenue Management: A Decomposition Approach

Dong Li[a,*], **Zhan Pang**[b]

[a]*The York Management School, University of York, York YO10 5GD, United Kingdom*
[b]*Department of Management Science, City University of Hong Kong, Hong Kong*

Abstract

This paper considers dynamic booking control for a single-station car rental revenue management problem. Different from conventional airline revenue management, car rental revenue management needs to take into account not only the existing bookings but also the lengths of the existing rentals and the capacity flexibility via fleet shuttling, which yields a high-dimensional system state space. In this paper, we formulate the dynamic booking control problem as a discrete-time stochastic dynamic program over an infinite horizon. Such a model is computationally intractable. We propose a decomposition approach and develop two heuristics. The first heuristic is an approximate dynamic program (ADP) which approximates the value function using the value functions of the decomposed problems. The second heuristic is constructed directly from the optimal booking limits computed from the decomposed problems, which is more scalable compared to the ADP heuristic. Our numerical study suggests that the performances of both heuristics are close to optimum and significantly outperform the commonly used probabilistic non-linear programming (PNLP) heuristic in most of the instances. The dominant performance of our second heuristic is evidenced in a case study using sample data from a major car rental company in the UK.

Keywords: Revenue Management, Car Rental, Approximate Dynamic Programming, Decomposition

1. Introduction

Revenue Management (RM), also known as *Yield Management*, has been widely adopted in various industries such as airlines, hotels, car rentals and cruise lines. See Cross et al. (2011) for an excellent review of the history and developments of RM in practice. Driven by its prevalence in service industry, the research interest in RM has been growing rapidly over the last two decades; see Talluri and van Ryzin (2004) for a comprehensive introduction to the theoretical developments of RM.

RM was first introduced to the car rental industry in the early 1990's. Carroll and Grimes (1995) introduce the implementation of car rental RM system in 1990-1991 at Hertz. Another well-known successful case of RM in rental car industry is that in 1993-1994 RM saved the large rental car company, National Car Rental, from near bankruptcy into a profitable business within two years (Geraghty and Johnson, 1997). Despite its early success, the increasingly competitive market requires more sophisticated models and systems to support the RM decisions and operations.

*Corresponding author
Email addresses: dong.li@york.ac.uk (Dong Li), zhan.pang@cityu.edu.hk (Zhan Pang)

The common characteristics of RM across various industries are perishable inventory, customers booking in advance and customer segmentation. Apart from these characteristics, car rental operations feature multiple length of rentals, which is corresponding to multiple length of stay in hotels or the number of consecutive flights with the same itinerary in airlines. The capacity decisions in car rental RM need to take into account not only the remaining inventory for each day in the future but also the length of rental, pick-up and return times of existing bookings. A high-dimensional state space is therefore required to describe dynamics of the system. A booking consisting of a pick-up day and a length of rental can be treated as a product. Hence, the car rental RM problem by its nature is a multi-product or network problem.

A distinctive feature in car rental operations that differs itself from that of airlines and hotels is the capacity flexibility. Unlike the other two industries which have fixed inventories either being aircraft seats or hotel rooms, cars are moved between rental stations constantly to adjust local capacities, which is often referred to as *shuttling* by car rental companies. Jens Utech, the former director of back-office and station systems at Avis Budget EMEA, has stated, "Every Friday at 7am, one transporter of cars would go from London Heathrow to Mayfair, due to demand in downtown London on weekends."(FICO, 2009) Flexible capacity provides car rental companies an extra lever to meet customer demand and raise revenue. In the meantime it imposes challenges on booking control decisions which drive the shuttling movements.

In this paper, we consider a dynamic booking control problem for a single car rental station with flexible capacity. At the beginning of each booking period and in advance of knowing the true demand realisations, the station's revenue manager needs to decide a booking limit for each product based on the current inventory levels and their knowledge on future demand arrivals. *Booking limits* specify the maximum number of customers to be accepted for each product during a booking period. In the meantime, a shuttling decision is made and extra cars are moved in from nearby stations if the benefit is deemed higher than the cost incurred. These booking limits are then updated and new shuttling movements proposed periodically at each of the following booking periods. The objective of the revenue manager is the determination of a booking limit policy which maximises the total discounted expected revenue over an infinite horizon. For simplicity, our model does not address issues of product upgrade or overbooking. These simplifications allow us to focus on and tackle the core challenges in the current car rental RM, as elaborated in the next section. We have also assumed that customer demand is stationary and does not change with the pick-up date, for ease of exposition and notation convenience. In the appendix we show that our approach can be readily extended to the setting in the presence of seasonality.

We first formulate the problem as a stochastic dynamic program. Due to the intractability of this formulation, we propose a decomposition approach to tackle the computational issue.The idea is to treat the demand of multiple rental days as multiple single-day equal-valued demands which are perfectly positively correlated. Our approach then relaxes this correlation and replaces the multiple rental day demand by multiple independent single-day demands, which allows us to decompose the original problem into single-day booking control problems.

Based on the solutions to the decomposed problems, we develop two heuristics that can address the interdependency between multiple rental days. The first one uses the value function of the single-day booking control problems to approximate the value function of the original dynamic problem and constructs policies via an approximation dynamic programming (ADP) approach. However, such an ADP approach only applies to small size problems as the optimal booking limits are obtained by solving a nonlinear program in the periodic-review setting. The second one exploits the optimal booking policies from the decomposed problems with consideration of their interdependence, which can be readily applied to the original problem. Our numerical results suggest that the latter performs very well against the ADP approach in the small-size problems and is more scalable. The results also suggest that both heuristics perform comfortably better than the classical probabilistic non-linear programming (PNLP) approach. We also identify situations in which PNLP approach might work well.

The approach we have proposed is readily applicable to hotel revenue management. Like length of rentals in rental cars, hotel businesses feature length of stay controls. Hotels normally have fixed capacities, which can be accommodated by simply setting the number of shuttling to zero in our model. Similarly, our method could also be applied to apartment renting or resort industry. Another stream of potential applications of our model is equipment rentals, which include other vehicle rental businesses such as truck rentals. These businesses are particularly relevant as they feature both a range of rental durations and the opportunity to temporarily change local capacities via transshipment.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 formulates the problem as a stochastic dynamic program. Section 4 presents a resource decomposition approach via which we develop heuristic policies and an analytical upper bound. We also introduce the PNLP heuristic in this section. Section 5 provides an extensive numerical study, followed by a case study with industry data from a major car rental company in the UK in Section 6. Section 7 concludes the paper with a few remarks. In the appendix, we extend our model to the cyclic dynamic programming framework and show that our approach applies to seasonal demand scenarios.

2. Literature Review

Despite the long history of adopting RM in the car rental industry, there is a limited literature on car rental RM, which is perhaps due to its complexity in modeling and analysis. Most of the existing models in the literature approximate the car rental RM problems by either treating the car rental systems as multi-product passenger airline systems or by ignoring the dynamics of advance booking processes.

Schmidt (2009) formulates the car rental booking control problem into a finite-horizon Markov decision process. The decision epoch is upon each arrival and the action is to either accept or reject the booking. He also includes shuttling and upgrade in his model. However, he does not consider the solution to the formulated dynamic program and instead approximates the problem with network models. Specifically he proposes a PNLP heuristic and develops a static booking limit policy. Similarly, Haensel et al. (2012) address a capacity allocation and shuttling decision problem for a car rental network within a fixed pick-up period. A two stage stochastic programming (SP) is proposed to the development of booking limit policies. The demand uncertainty in the second stage is approximated by a finite number of scenarios. Both works report the improved performance of their proposed policies over the deterministic linear programming (DLP) heuristic. In a more recent work, Guerriero and Olivito (2014) incorporate one way rentals into the booking controls, along with consideration of shuttling and upgrade. Like Schmidt (2009), they present a dynamic program for the problem without considering the solution methods. DLP approximation is used to develop booking limit and bid price control policies. Their results suggest that the former policies outperform the latter and the gap between them increases with the demand volume. As far as we know the only work on dynamic booking control policies in a car rental setting is Steinhardt and Gönsch (2012) who suggest two decomposition methods to capacity control and product upgrade decisions in the same time. As the others they follow the models of typical airline RM and only consider bid price policies. The flexible capacity is not considered.

We are not aware of any contributions in the literature which have actually solved the dynamic booking control problems with flexible capacity, especially in a car rental setting. Most of the previous literature work around this rather daunting challenge and approximate the problem via easier-to-solve network models. Our results suggest that the static booking control policies developed via such approximation could lead to significant loss of performance in compared with the dynamic policies we have proposed. Moreover, our model differs from theirs in that we consider an infinite time horizon and allow more general demand arrival processes, which underscores the generality of our approach.

The literature on fleet planning is also related to our work, as they also address the flexible capacity in the car rental operation, but from a strategic or tactical perspective. Pachon et al. (2003) consider tactical fleet deployment of rental cars. Vehicles within a fleet pool- a group of stations within one district- are geographically redistributed on a daily basis to achieve better supply and demand balance. They formulate the problem as a stochastic program that is then decomposed into two separate subproblems. The first one concerns the optimal inventory levels in each station while the second concerns the optimal shuttling movements. Fink and Reiners (2006) address the short term national rental car deployment from the logistics perspective. Apart from shuttling among stations, they also consider fleeting (to add new cars into the fleet) and de-fleeting (to remove old cars from the fleet) decisions, and formulate the problem as a minimum cost network flow model.

An alternative approximation is to treat the rental system as a queueing system with multiple servers and exponentially distributed service times. The advance booking process is ignored. Savin et al. (2005) address the capacity rationing problem of rental services with multiple customer classes. They formulate the problem as a continuous-time loss queueing system where arriving customers are lost immediately if no capacity is available. They characterize the structure of the optimal policy. Gans and Savin (2007) further extend their analysis to the model that integrates the capacity rationing and pricing decisions. Both models have only one-dimensional state space and they are able to analytically characterize the structure of the optimal policies. In the car rental RM problem, different customers may book different pick-up date and length of rental combinations, which leads to a multi-dimensional state space and it is therefore intractable to derive the optimal policy for a large size problem.

Also related is the literature that applies approximate dynamic programming techniques to network RM. Comprehensive references on approximate dynamic programming are offered by Powell (2007). In the network RM literature, there are two commonly used ADP approaches: linear programming approach (see, e.g., Meissner and Strauss, 2012, Vossen and Zhang, 2014 and references therein) and decomposition approach (see, e.g., Talluri and van Ryzin, 2004, Liu and Van Ryzin, 2008, Zhang and Adelman, 2009, Zhang, 2011 and references therein). In a typical dynamic programming decomposition approach (Talluri and van Ryzin, 2004, Chapter 3), the value function is approximated by an affine function of the state with the coefficients representing the opportunity costs of the resources. Zhang (2011) proposes a nonlinear, separable functional approximation to the value function and shows that his approximation leads to a tighter upper bound than some known bounds in the literature. Their problem is solved using a simultaneous dynamic programming algorithm.

Those decomposition approaches in airline network RM literature (e.g., Zhang, 2011) are in spirit similar to ours in that we also decompose the problem with multiple resources into multiple single-resource problems. However, they are not readily applicable to our model due to the distinctive features and challenges in car rental operation. In airline RM, the product is a combination of multiple legs (resources) within a flight network, while in car rental the product is a combination of pick-up day and length of rental. As time moves on, new rental days become open for booking and hence more products available which include the existing rental days as part of the new products. In contrast, the range of products in the airline RM is determined by the flight network and does not change with time. This rolling time window feature necessitates an infinite horizon model for car rental problems. For the similar reason the decomposition is not a straightforward extension of those in the airline RM literature.

Another challenge in our model is the construction of booking control policies. Typical airline RM models (e.g., Talluri and van Ryzin, 2004) usually assume that the capacity allocation decisions are made upon the arrival of each customer, which is however unrealistic and costly in car rental. Due to this difference, the airline RM literature typically considers the bid-price heuristics while in car rental booking limits are widely implemented (e.g., Schmidt, 2009, Haensel et al., 2012). To construct the dynamic booking limit policies a complex non-linear programming problem must be solved, whereas the construction of bid price policies is straightforward after the decomposition (see, e.g., Talluri and van Ryzin, 2004). The study of dynamic booking limit policies is relatively rare in the literature. Actually, to the best of our knowledge we are the first to propose decomposition approaches to the development of booking limit policies with flexible capacity.

3. The Model

3.1. Problem Description

Consider a single rental car station with one car group. The fleet size, or the total number of cars of that car group, is C. If necessary cars of the same group at proximity stations could be shuttled in to meet extra demand. At most M extra cars are available each day. All the cars shuttled in must be returned back to their original stations at the end of their rental period. The focal station needs to pay a daily charge of o for each shuttling. Cars are always picked up at the beginning of a day and returned at the end of the last day of their rental period. The cars returned today are ready for pick-up tomorrow and cannot be picked up again on the same day. Assume the car rental company has the perfect information about the actual length of rental (LoR) that customers will drive for. In other words, cars are returned always on time.

Each arriving customer will request a car to be picked up on day n from the time of booking ($0 \le n \le N$) and driven for l days ($1 \le l \le L$), where N is the maximum advance booking days allowed and L the maximum length of rental. Each combination of (n, l) is defined as a product. For each product, denote by $d_{n,l}$ the total demand arrived during a single booking period, which is a discrete random variable. For simplicity, we assume that the demands for different products are independent of each other and they are identically and independently distributed over time. Once accepted customers will pay $r_{n,l}$ at the point of booking. Otherwise they are lost from the system.

The revenue manager needs to decide which customer bookings to accept and which to reject, with the observation of the current inventory level and the distributions of the future demand on all the products. A booking limit $b_{n,l}$ is specified for each product at the beginning of each booking period, before their demand is materialised. A new customer booking is accepted as long as the corresponding booking limit is not yet reached. Otherwise the booking is rejected. All the booking limits are periodically updated. In other words they remain the same within each booking period. Along with the booking limits, the revenue manager also determines the number of future shuttling movements, with the objective of maximising the overall revenue generated from the fleet.

3.2. Stochastic Dynamic Programming Formulation

The problem described above can be modelled as a discrete-time Markov decision process.

1. Decision epochs are the beginning of each booking period, which is discrete and denoted by $t = 0, 1, 2, ..., T$, $T \to \infty$. Each booking period t represents a day in the model.

2. The system state at each decision epoch is described by an $N+L-1$ vector: $\mathbf{x} = (x_0, x_1, ..., x_z, ..., x_{N+L-2})$, where $0 \leq x_z \leq C + M$ denotes the total number of cars having been booked or already on the road on day z, $z = 0, ..., N+L-2$ ($z = 0$ represents the current booking period). Note that a customer can book at most N days into the future and the length of rental could be L days, which results $N + L$ days of interest. This dimension is reduced by one with the observation that the last day $N + L - 1$ always has no cars booked as it would have just rolled into the booking horizon at each decision epoch. Denote by Ω the state space, and we have

$$\Omega = \{\mathbf{x} : 0 \leq x_z \leq C + M, 0 \leq z \leq N + L - 2\}\,. \tag{1}$$

3. A booking limit decision $\mathbf{b} = (b_{n,l})$ is made at each decision epoch immediately after observing state $\mathbf{x}$. In other words, $b_{n,l}$ specifies the maximum amount of capacity that can be sold to product (n, l) demand. The admissible booking limit set is given by

$$\mathcal{A}(\mathbf{x}) = \left\{ \mathbf{b} \in \mathbb{Z}^{(N+1)\times L} : \sum_{n=0}^{\min(z,N)} \sum_{l=z-n+1}^{L} b_{n,l} \leq C + M - x_z, 0 \leq z \leq N + L - 2 \right\}, \tag{2}$$

where the condition imposes that for each day z the total allocated capacity for future demand should never exceed the number of available cars. Note that both Ω and $\mathcal{A}(\mathbf{x})$ are discrete and finite and they do not vary with the decision epoch t.

4. Once a decision $\mathbf{b}$ is made at state $\mathbf{x}$, the system evolves to the next state $\tilde{\mathbf{x}}$, subject to the demand arrivals of $d_{n,l}$,

$$\tilde{\mathbf{x}} = \left(x_1 + \sum_{n=0}^{1} \sum_{l=2-n}^{L} \min\{b_{n,l}, d_{n,l}\}, ..., x_{N+L-2} + \sum_{n=0}^{N} \sum_{l=N+L-1-n}^{L} \min\{b_{n,l}, d_{n,l}\}, \min\{b_{N,L}, d_{N,L}\} \right). \tag{3}$$

5. A reward of $r_{n,l}$ is received for each accepted booking of product (n, l). A unit shuttling cost o is incurred when the number of bookings accepted is greater than the available capacity and a shuttling movement is triggered.

6. A policy π is any non-anticipative rule for choosing booking limits. Let $\mathbf{b}^\pi(\mathbf{x})$ denote the booking limits at state $\mathbf{x}$ under the policy π. The objective is the determination of a policy π which maximises the total discounted revenue over an infinite horizon.

Let $V^\pi(\mathbf{x})$ be the total discounted revenue to be received starting from state $\mathbf{x}$ and under a policy π, which is given by

$$V^\pi(\mathbf{x}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(R(\mathbf{b}^\pi(\mathbf{X}_t)) - O(\mathbf{X}_t, \mathbf{b}^\pi(\mathbf{X}_t))\right) \middle| \mathbf{X}_0 = \mathbf{x}\right] \tag{4}$$

where $\gamma \in [0, 1]$ is the discount factor. Note that we have added an index t to denote the revenue (cost) received (incurred) in each booking period t. $\mathbf{X}_t$ denotes the system state at time t. In addition, $R(\cdot)$ and $O(\cdot, \cdot)$ are the expected revenue and shuttling cost per period such that

$$R(\mathbf{b}) = \sum_{n=0}^{N}\sum_{l=1}^{L} r_{n,l}\mathbb{E}\left[\min\{b_{n,l}, d_{n,l}\}\right], \tag{5}$$

$$O(\mathbf{x}, \mathbf{b}) = o\mathbb{E}\left[\left(x_0 + \sum_{l} \min\{b_{0,l}, d_{0,l}\} - C\right)^+\right]. \tag{6}$$

For a given stationary policy π its value function $V^\pi(\mathbf{x})$ can be evaluated by computing the following recursive equations.

$$V^\pi(\mathbf{x}) = R(\mathbf{b}^\pi(\mathbf{x})) - O(\mathbf{x}, \mathbf{b}^\pi(\mathbf{x})) + \gamma\mathbb{E}[V^\pi(\tilde{\mathbf{x}})]. \tag{7}$$

Denote by $V(\mathbf{x})$ the optimal value function achieved under an optimal policy π^*, which satisfies the following optimality equation (Bellman equation)

$$V(\mathbf{x}) = \max_{\mathbf{b}\in\mathcal{A}(\mathbf{x})} \left\{R(\mathbf{b}) - O(\mathbf{x}, \mathbf{b}) + \gamma\mathbb{E}[V(\tilde{\mathbf{x}})]\right\}. \tag{8}$$

Note that only the shuttling cost incurred at the current time period ($z = 0$) is included in equation (6). In other words, the shuttling cost is only charged when cars are delivered. Any shuttling scheduled for a future period is not paid for until delivery.

In principle the optimality equation (8) can be solved and the optimal policy can be obtained by standard algorithms such as value iteration. Unfortunately due to the notorious curse of dimensionality (Powell, 2007) this is intractable for problems of any practical size. For our model, not only the state space increases exponentially with N and L, the action space also increases exponentially at a higher rate. The dimension of the state space and action space is $N + L - 1$ and $(N + 1)L$, respectively, which lead to a size of $(C + M + 1)^{N+L-1}$ for state space and $(C + M + 1)^{(N+1)L}$ for action space. For a small problem with a fleet of 4 cars, $N = 2$ and $L = 3$, the state space is 5^4 and the action space increases sharply to 5^9. It is computationally prohibitive to solve equation (8) by value iteration or other exact methods. We thus seek to develop heuristic approaches which are close to revenue maximising.

Remark 1 (Special Cases). When $L = 1$, i.e., the length of rental is always one day, the booking controls for different rental days are independent of each other and hence the problem is decomposed into a series of advance-booking problems for each rental day. Since there is only one rental rate, it becomes apparent that the booking requests should be accepted on a first come first serve principle.

In another special case where $N = 0$, customers pick up the cars immediately after booking. They could still hire for l days, $1 \leq l \leq L$. This problem reduces to a multi-class multi-server admission control problem. Upon arrival each customer is either accepted and begins service (pick-up of the car) immediately, or is rejected by the controller or when all servers (cars) are busy (on rent). The servers spend a certain amount of time (LoR) with a customer and become available again for the next customers. It resembles the tactical capacity allocation model in Savin et al (2005), but differs in that the duration of rentals is discretely distributed rather than exponentially distributed as specified in their model.

It is notable that we formulate the car rental RM problem in a rolling time window fashion over an infinite horizon due the nature of the car rental businesses. Compared to their counterpart in airlines, the interdependencies between car rental products are much stronger. The length of rental could be 1 to 28 days, which means that some products might share resources across many days. Similarly, one rental day could be part of products with various pick-up days and length of rentals along the timeline. There are no obvious breaking points and thus the problem cannot be naturally partitioned as in the airline RM. Therefore, to capture the key feature of car rental RM, we need to address the booking dynamics via the rolling time window throughout the entire time horizon.

Our approach also applies to the finite period setting with non-stationary demands. Formulating the finite-period problem requires the specification of a terminal value function which is dependent on the system state at the end of planning horizon (the pipeline of existing bookings yet to pick up the cars and the number of cars already on rent). Unlike airline RM where one could assume that the terminal value function is zero or a function of overbookings (if any), it is unclear how to specify the terminal value function in the car rental setting. To the best of our knowledge, there is no literature discussing about this. Hence, this paper focuses on the infinite horizon setting. As we will show in the appendix, even in the infinite horizon setting, the seasonality can be readily addressed using a cyclic dynamic programming framework. We understand that the infinite horizon dynamic program is restricted to fully accommodate the absolute date dependent features, which therefore remains one limitation of this work. Nevertheless, the proposed cyclic dynamic programming is able to mitigate this limitation to a large extent.

Remark 2 (Multiple Demand Classes). When there are multiple independently distributed demand classes for each product, which are differentiated by the prices, and the demands of different classes arrive simultaneously, our model can be readily generalized to the multi-class case by treating a product as a combination of pick-up day, length of rental and demand class. It is notable that in the discrete-time RM literature it is suggested that nested booking limit policies are commonly used when there are multiple demand classes for each product and the demands of different classes arrive sequentially from low-priority to high-priority (see, e.g., Brumelle and McGill, 1993, and Netessine and Shumsky, 2005). However, when the demands of different classes arrive simultaneously in each period and the booking limit decisions must be made before demands are realized, it is unclear whether nested booking limit policies are still optimal in the discrete-time setting even for the airline RM models.

Remark 3 (Outbound Shuttling). Our model allows the station to increase capacity level temporarily using inbound shuttling with a daily operational cost. In reality, a car rental station may have both inbound and outbound shuttling. To explicitly model the outbound shuttling, we need to model the demands for the outbound shuttling (see, e.g., Haensel et al., 2012), which may significantly complicate the model and the dimension of the system state will be largely increased. Indeed, a network setting is more suitable to address the detailed inbound/outbound shuttling operations. To partially account for the outbound shuttling within the single station setting, one can consider them as another demand class, assuming that the outbound shuttled cars are returned to the station after service. Therefore for each (n, l) pair there exist two fare classes, one for outbound shuttling requests and the other normal customer bookings. The resulted problem can then be addressed as multiple demand class problems as described in Remark 2 (Multiple Demand Classes).

In our current treatment, we only allow inbound shuttling which can be seen as an emergent supply when in the shortage of capacity. In reality this could happen when a single-station car rental company (e.g. franchises) could access the excessive capacity of other companies by paying additional fees, if it is profitable to do so. Note that when the upper limit of the inbound shuttling, M, is zero, our model reduces to the case without inbound shuttling. Such a treatment allows us to model a single demand class for each product so that we can focus on handling the high dimensionality caused by the multiple length of rentals. We would like to retain the explicit consideration of the outbound shuttling decisions in our future research in the network settings.

3.3. An Illustrative Example of Optimal Policy

In this example we consider a rental station with a fleet of $C = 7$ cars. Two cars $(M = 2)$ are available to be shuttled in from nearby stations every day. Customers are allowed to pick up a car either today or tomorrow $(N = 1)$, and drive for one or two days $(L = 2)$. During each booking period, customers arrive according to Poisson processes. We assume the arrival rate is dependent on n and denoted by λ_n. We have $\lambda_0 = 1, \lambda_1 = 2$, which indicates that the demand for the same day pick-up is half of those for the following day pick-up. Upon arrival each customer announces a rental length of l, with a probability $p_l, \sum_{l=1}^{L} p_l = 1$. We have $p_1 = 0.6, p_2 = 0.4$. The demand for product (n, l) then follows a Poisson process with rate $\lambda_n p_l$.

The rental rates for all the four products are set at $r_{0,1} = r_{1,1} = £33, r_{0,2} = r_{1,2} = £52$. The one day rentals are more profitable as the daily rate for one day rental is higher than two day rentals. The shuttling cost is $o = £30$, which is in between of the daily rate of the two length of rentals.

For this small problem we are able to calculate the value functions and find the optimal booking limit policy via equation (8). The value functions in respect to the states are plotted in Figure 2, Section 4.2. It shows that the value function is decreasing and concave componentwise in $\mathbf{x}$.

State	Booking Limit			
(x_0, x_1)	$b_{0,1}$	$b_{0,2}$	$b_{1,1}$	$b_{1,2}$
$(0, 0)$	4	2	4	3
$(1, 0)$	4	2	4	3
$(0, 1)$	4	2	3	3
$(7, 6)$	2	0	1	2
$(6, 7)$	3	0	0	2
$(7, 7)$	2	0	0	2

Table 1: Optimal Booking Limits at Selected States for the Example.

The optimal booking limits for a number of selected states are listed in Table 1. It shows that when the system is empty $(\mathbf{x} = (0, 0))$, the cars allocated to one day rentals are more than those to two day rentals, due to the combined effect of higher daily rate and higher demand of the one day rentals. Also, more cars are reserved for the following day pick-up compared to the same day pick-up, the former of which has a higher arrival rate. On day 0 there are some spare capacities due to the low demand for same day pick-up. In contrast, shuttling of two cars are proposed on the following day (note that the total number of cars required is given by $b_{0,2} + b_{1,1} + b_{1,2}$). The same result is seen in state $(1, 0)$ when there is just one car booked on day 0. If the only car being booked is on day 1 the result is slightly different. The booking limit $b_{1,1}$ is reduced by one while all the others remain the same. Still no shuttling is proposed on day 0.

Towards the other end, for state $(7, 7)$ where cars are fully booked on both days, shuttling of two cars are proposed for both days to meet extra demand. On day 0 these two cars are allocated to one day rentals for their higher rental rate. On day 1, however, they are allocated to two day rentals. This is a typical example of the trade off between rental rates and demand uncertainties. Even though allocating cars to two day rentals leads to lower rental revenue per day, it reduces the risk of idle cars on day 2. When there is one more car available on day 1, as in state $(7, 6)$, it is allocated to one day rental to generate higher revenue of that day. Similar rationing is seen for state $(6, 7)$ as well.

As this small example illustrates, it is non-trivial to find the optimal booking limits for multiple car rental products. The trade off between rental rates and the demand uncertainties has been made particularly complex due to multiple length of rentals. In the following section we propose a decomposition approach to address this complexity.

4. The Decomposition Approach and Heuristic Policies

In this section, we propose an approach to decompose the problem with multiple length of rentals into multiple identical and independent single-day rental problems. Based on the decomposition, we propose two heuristics. The first heuristic is to approximate the value function using the value functions of the decomposition problems. The second heuristic constructs the booking policies using the booking limits derived from the decomposition problems.

4.1. Decomposition

In car rental, a product consists of the resources of multiple consecutive days from a future pick-up day. In other words, the demands for the resources of different days are highly correlated, which leads to the high dimensional state space. A natural decomposition technique is to ignore the correlation between the demands of different resources and treat a product with multi-day length of rentals as multiple single-day products with independent demands. Each of these single-day product can be offered or denied individually, without consideration of the resource availability in the other days. Therefore customers requesting a multiple day rental product could be offered a car only for part of its rental duration. Technically speaking, such a decomposition approach reduces the dimensionality of the state space to one.

As illustrated in Figure 1, the product (n, l) is decomposed into l single-day products $(n, l, j), j = 0, 1, \ldots, l - 1$, which are identical products available on each of the l days starting from n days from now. The demands of these products are independent of each other. To decide whether or not to accept the customer bookings for each of them, we only need to concern the resource availability on a single day. A finite horizon dynamic program could be developed to find the optimal booking limit policy for the decomposed problems, which are in spirit similar to single leg airline booking control problems. However, as discussed earlier, different from the standard airline RM models where products are defined by the combinations of multiple legs but with the same departure periods, the car rental products are defined by the combinations of (n, l) pairs within a rolling time window. As a result, the set of products that use cars on a particular day are not the same between different times prior to the pick-up day. This is illustrated in the next paragraph. To the best of our knowledge, it is the first time to employ such a decomposition approach to a car rental RM model with rolling time windows,

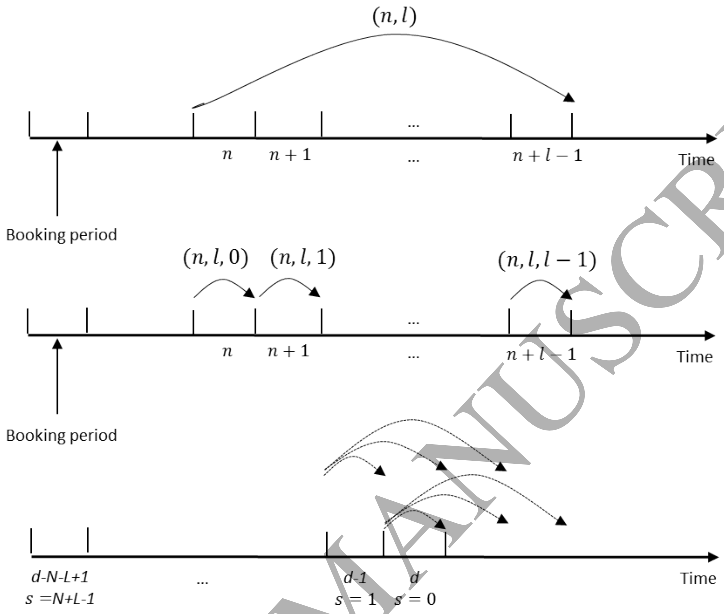


Figure 1: Decomposition Illustration.

For any specific rental day, say day d, we can identify all the demands that will book and use a car on this day, and then treat them as independent demands for this day. The earliest bookings to use a car on day d may be made $N + L - 1$ days prior to day d and the latest bookings may be made on the same day. We use s to denote the number of days prior to day d, or the booking period, as demonstrated in Figure 1. Clearly, the maximum length of the booking period s is $N + L - 1$, i.e., the earliest booking that will use a car on day d is made on day $d - N - L + 1$ with the pick-up day $d - L + 1$ and the length of rental L.

More specifically, on day d (or $s = 0$), all the bookings with a pick-up day d need to be counted, regardless of their *LoR*. All bookings which pick up cars from $d + 1$ or later do not use cars on day d and are thus excluded. In other words, any product satisfying $n = 0$ is included.

On the day prior to day d (or $s = 1$), there are two types of bookings that will use a car on day d. The first type of bookings pick up cars on d with any length of rental, while the second type pick up cars on $d - 1$ with the length of rental being 2 or more days. For both types of bookings, the corresponding (n, l) pair satisfies $n + l \geq 2$.

In such a manner, we can identify all the demands for this day for booking period $s = 0, 1, ..., N - L + 1$. The set of bookings of products that will use a car on day d can be characterized with $n + l \geq s + 1$. Denote by $\mathbf{K_s}$ the set of these products. We have,

$$\mathbf{K_s} = \{(n,l) : n + l \geq s + 1, n \leq s\}, \forall 0 \leq s \leq N + L - 1, \tag{9}$$

where $n \leq s$ requires that the cars are due to pick-up on day d or already on the road. The cardinality of $\mathbf{K_s}$ is determined by

$$|\mathbf{K_s}| = \sum_{n=0}^{\min\{s,N\}} (L + n - s)^+, \forall 0 \leq s \leq N + L - 1. \tag{10}$$

Since the decomposed products have exactly the same demand distributions as their original products, the set $\mathbf{K_s}$ is sufficient to represent the set of decomposed products at booking period s. Given such a demand stream, a finite horizon dynamic program can be formulated to find the optimal booking limits. Denote by $\hat{v}_s(x)$ the value function at booking period s when x cars have been already booked, and $b^s_{n,l}$ the corresponding booking limit for product (n,l) at s. We then have the following DP.

$$\hat{v}_s(x) = \max_{\mathbf{b^s}} \left\{ \sum_{(n,l)\in K_s} \mathbb{E}\left[\frac{r_{n,l}}{l} \min\{b^s_{n,l}, d_{n,l}\}\right] + \gamma \mathbb{E}[\hat{v}_{s-1}(\tilde{x})] \right\}, \forall s = 1, 2, ..., N + L - 1, \tag{11}$$

with a boundary condition

$$\hat{v}_0(x) = \max_{\mathbf{b^0}} \left\{ \sum_{(n,l)\in K_0} \mathbb{E}[\frac{r_{n,l}}{l} \min\{b^0_{n,l}, d_{n,l}\}] - o\mathbb{E}[x + \sum_{(n,l)\in K_0} \min\{b^0_{n,l}, d_{n,l}\} - C]^+ \right\}, \tag{12}$$

where $\tilde{x} = x + \sum_{(n,l)\in K_s} \min\{b^s_{n,l}, d_{n,l}\}$.

As in the original DP a shuttling cost is paid only when $s = 0$. Any idle cars beyond time $s = 0$ are perished and lose all their values. Note that we have chosen to use the *rate-per-day* for the decomposed products, for the following two reasons. Firstly, the rate-per-day is a term which is frequently and widely used in car rental businesses. It is easy to interpret and thus a natural choice for the decomposition which breaks multi-day rentals into single-day ones. In airlines, however, the legs of an itinerary usually have different origins/destinations and/or capacities, and thus there does not exist an obvious "rate-per-leg". Secondly, rate-per-day is straightforward to obtain compared with those sophisticated proration methods, such as the displacement adjusted revenue. A DLP (or other models) needs to be solved in each decision epoch to obtain opportunity cost of each leg. A different DP must then be solved every time due to the changing revenue for each decomposed product. Whereas with the choice of rate-per-day only one DP needs to be solved. One could also choose to solve a DLP just once and use the static opportunity cost (Talluri and van Ryzin (2004)), but the benefits of using the displacement adjusted revenue are limited. Nevertheless, which daily revenue to use remains an less important question in this work and has little impact to our main contributions.

The above formulation for the decomposed problem is similar to the classical revenue management problem of allocating a fixed capacity of a single resource among the demand classes (Robinson, 1995). We next characterize the structural properties of the decomposed problem.

In the conventional single-leg airline setting, the expected revenue function is normally concave and the dynamic programming iterations are concave programs. It is natural to ask the question whether the above revenue function for the decomposed problem is also a concave function. In fact, the above optimality equations involve objective functions of the form $g(x, u_1, ..., u_n) = \mathbb{E}[f(x, \min(d_1, u_1), ..., \min(d_n, u_n))]$ for some concave function f. Note that the concavity of f cannot be preserved under the expectation operations. It is not hard to show that when $d_1, ..., d_n$ are independent of each other, $g(x, u_1, ..., u_n)$ is componentwise concave in x and quasi-concave in $u_1, ..., u_n$. This class of problem is studied by Chen et al. (2015). They develop a transformation technique that converts the above program into an equivalent concave program and show that the concavity can be preserved under the maximization operation.

Lemma 1 (Chen et al., 2015). *If $f : \mathbb{Z}^{n+1} \to \mathbb{R}$ is a concave function, then the function*

$$g(x) = \max_{u_k \geq 0, k=1,...,n} E[f(x, \min(d_1, u_1), ..., \min(d_n, u_n))]$$

is concave in x, where d_k's are non-negative and independent random variables.

Applying Lemma 1, we can show the concavity of v_s.

Proposition 1. *For all s, $\hat{v}_s$ is concave, which ensures that the objective functions in the optimality equations (11)-(12) are quasi-concave.*

Proof. We first show that $\hat{v}_0$ is concave. In fact, as $\sum_{(n,l)\in K_0} \frac{r_{n,l}}{l} x_{n,l} - o(x + \sum_{(n,l)\in K_0} x_{n,l} - C)^+$ is jointly concave in $(x, x_{n,l}, (n,l) \in K_0)$, applying Lemma 1 we know that $\hat{v}_0(x)$ is concave in x.

By induction. For any $s = 1, ..., N + L - 1$, assume that $\hat{v}_{s-1}$ is concave. Clearly, $\sum_{(n,l)\in K_s} \frac{r_{n,l}}{l} x_{n,l} + \gamma \hat{v}_{s-1}(x + \sum_{(n,l)\in K_s} x_{n,l})$ is jointly concave in $(x, x_{n,l}, (n,l) \in K_0)$. Applying Lemma 1 again, we know that $\hat{v}_s$ is concave.

Apparently, the concavity of $\hat{v}_s$ ensures that the objective functions in the dynamic program (11)-(12) are quasi-concave in the booking limits. □

The concavity structural property of the value functions ensures the unimodularity of the optimization programs in the dynamic program, which is important for designing algorithms as it helps to judge whether the local optimum is a global optimum.

In comparison with the original dynamic program (8), the complexity of the decomposed dynamic program (11)-(12) is significantly reduced. The state space now has only a single dimension. The dimension of action space varies with s and is determined by $|\mathbf{K_s}|$, with the largest action space occurring when $s = N$ as follows.

$$|\mathbf{K_N}| = \begin{cases} (N+1)(L - \frac{1}{2}N), N < L \\ \frac{1}{2}L(L+1), N \geq L \end{cases} \tag{13}$$

In contrast to the dimension of $(N + 1)L$ in the original problem, the action space reduces significantly in the decomposed problems. Particularly when $N \geq L$ the action space becomes independent of N.

It is worth mentioning that the action space can be further reduced to L dimensions by product aggregation. All decomposed products with the same l (but different n) can be aggregated to a virtual product, with a demand $d_l = \sum_n d_{n,l}$ and a rental rate $r_l = \sum_n r_{n,l}/[(N + 1)l]$. The solution is then obtained via (11)-(12) but with the virtual products. The dimension of the action space is then at most L.

4.2. Approximate Dynamic Programming (ADP)

The first heuristic is to approximate the value function in (8) with the value functions of the decomposed problems. Denote by $\hat{V}(\mathbf{x})$ an approximation to the value function of the original problem $V(\mathbf{x})$. We have

$$\hat{V}(\mathbf{x}) = \sum_{s=0}^{N+L-2} \hat{v}_s(x_s) + \hat{v}_{N+L-1}(0) + \sum_{t=1}^{\infty} \gamma^t \hat{v}_{N+L-1}(0)$$

This approximation is obtained by summing over the discounted value functions for each day from the current day and onwards. The first term is the summed total for the first $N+L-1$ days, whose value functions are given by $\hat{v}_s(x_s), 0 \le s \le N+L-2$. The second term is for day $N+L-1$. Recall that no cars would have been booked yet for this day, and thus its value function is given by $\hat{v}_{N+L-1}(0)$. On all the subsequent days no cars have been booked either. The maximum possible daily revenue generated is bounded by $\hat{v}_{N+L-1}(0)$. A discount factor γ is applied to obtain their present values, as shown in the last term. After simplification we obtain the following equation.

$$\begin{aligned} \hat{V}(\mathbf{x}) &= \sum_{s=0}^{N+L-2} \hat{v}_s(x_s) + \sum_{t=0}^{\infty} \gamma^t \hat{v}_{N+L-1}(0) \\ &= \sum_{s=0}^{N+L-2} \hat{v}_s(x_s) + \left(\frac{\hat{v}_{N+L-1}(0)}{1-\gamma} \right). \end{aligned} \tag{14}$$

Similar to the airline RM literature (see, e.g., Zhang, 2011), we can show that the approximated value function $\hat{V}(\mathbf{x})$ serves as an upper bound to the original value function, as stated in Proposition 2.

Proposition 2. *$\hat{V}(\mathbf{x})$ is an upper bound of $V(\mathbf{x})$, i.e., $V(\mathbf{x}) \le \hat{V}(\mathbf{x})$ for all $\mathbf{x}$.*

Proof. For each product (n, l), the optimal booking limit in the original problem, $b_{n,l}(\mathbf{x})$, can be seen as a feasible solution of the decomposed problems corresponding to the days during the product's rental period, i.e., $b^s_{n,l} = b_{n,l}(\mathbf{x})$, $s = n, n+1, ..., n+l-1$. Note that the corresponding total expected revenue and the shuttling cost for each day during the rental period must be the same for both the original and the approximate problems under the above booking policies. Considering all the possible pairs of (n, l), the corresponding booking limits derived from the optimal booking limits of the original problem are suboptimal for the decomposed problems, which implies that $V(\mathbf{x}) \le \hat{V}(\mathbf{x})$ for all $\mathbf{x}$. □

This is exemplified in Figure 2 below where values of the exact value function $V(\mathbf{x})$ and approximation $\hat{V}(\mathbf{x})$ are plotted for the illustrated example whose details are given in Section 3.3. It clearly shows that $\hat{V}(\mathbf{x})$ is above $V(\mathbf{x})$ across all the states.

Following standard approximation dynamic programming approaches (see, e.g., Powell (2007)), we replace the value function $V(\mathbf{x})$ with its approximation $\hat{V}(\mathbf{x})$ within (8) and obtain a policy π^{ADP} via the formula below,

$$\pi^{ADP}(\mathbf{x}) = \underset{\mathbf{b} \in \mathcal{A}(\mathbf{x})}{\arg\max} \left\{ R(\mathbf{b}) - O(\mathbf{x}, \mathbf{b}) + \gamma \mathbb{E}[\hat{V}(\tilde{\mathbf{x}})] \right\}. \tag{15}$$

The performance of policy π^{ADP} can be evaluated by equation (7). Our experience is that it has strong performance when it is available. In support this claim we make reference to 1200 randomly generated

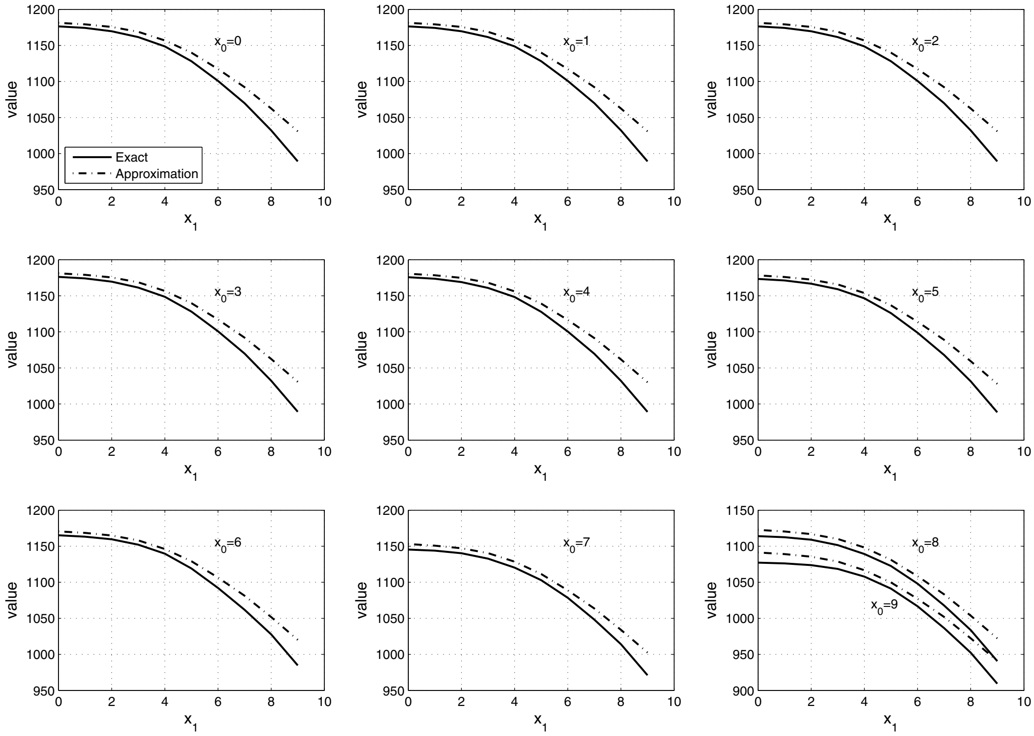


Figure 2: The Exact Value Function $V(\mathbf{x})$ and the Approximation $\hat{V}(\mathbf{x})$ in respect to States for the Example.

problems in which $N = 0$ and $L = 2, 3, 4$ and customer bookings arrive according to Poisson processes. The performance of three heuristics, which include π^{ADP}, when applied to these problems is given in Table 2, Section 5.1, together with details of the problems themselves. For each L, the results are presented in four groups labelled A,B,C and D according to the demand-supply ratio in the generated problems. In Table 2, the mean performances for π^{ADP} when $L = 2$, as measured by the percentage deviation from optimum, are 0.00%(A), -0.05%(B), -0.08%(C) and -0.12%(D). These figures do not change much for longer length of rentals. The worst performances across all the four categories are -0.89%($L = 2$), -0.85%($L = 3$) and -1.39%($L = 3$). Furthermore for problems with $N = 1$ and $L = 2$, results for 400 randomly generated problems (100 for each category) are given in Table 4. This may be found in Section 5.2 along with full problem details. In Table 4, the mean performances for π^{ADP} are 0.00%(A), 0.00%(B), -0.01%(C) and -0.02%(D), and the worst performances are 0.00%(A), -0.17%(B), -0.07%(C) and -0.23%(D).

The strong performance of this policy notwithstanding, its development is still computationally expensive other than for small problems. Even though the equation (15) only needs to be solved once for each state, the action space remains the same as in the original problem and it is non-trivial to find the solution. In light of this difficulty we proceed to propose a simple heuristic policy which is more scalable.

4.3. Decomposed Booking Limit (DB) Policy

The second approach is to construct the heuristic policy with the booking limits derived from the decomposed problems directly.

The optimal policy obtained via (11)-(12) determines booking limits $b^s_{n,l}(x_s)$ for product (n, l) on day s prior to the pick-up day when state x_s is occupied. In Figure 1, the booking limit $b^{n+j}_{n,l}(x_{n+j})$ specifies the number of cars allocated to product (n, l) for day $n + j$, or j^{th} day of its rental period. There are in total l such booking limits for this product, one for each of its rental days. As in the airline RM literature (see, e.g., Talluri and van Ryzin (2004)), a booking limit for (n, l) in the original problem can be obtained as the minimum of them. In other words a product (n, l) demand can only be accepted if none of these single-day booking limits has been reached. We have

$$b_{n,l}(\mathbf{x}) = \min_{s=n}^{s \leq n+l-1} b^s_{n,l}(x_s), \tag{16}$$

We denote the resulting policy by π^{DB}. In obtaining this policy most of the computational effort is consumed in solving the decomposed dynamic program (11)-(12). As we have discussed in Section 4.1, the complexity of (11)-(12) is significantly reduced compared to the original problem. The state space reduces to one dimension and the action space dimension could reduce down to L with product aggregation. In such a case the resulting booking limits need to be disaggregated for each product. A simple heuristic is to disaggregate proportionally to their expected demand.

Once (11)-(12) is solved, the determination of policy π^{DB} for each state is trivial. This feature is particularly attractive in practice. Even though the solution to the decomposed problem might well take some time, it can be left running over night or longer if necessary. Once finished the booking limits can be obtained real time according to the observed state at the beginning of each booking period, which makes π^{DB} applicable to more practical problems.

5. Numerical Study

In this section we conduct an extensive numerical study to investigate the performance of the two proposed heuristic policies π^{ADP} and π^{DB} as well as the upper bound. We also include the policy derived from a probabilistic non-linear program (π^{PNLP}) in the analysis as a competitive benchmark. PNLP is an important approach in solving network RM booking control problems. Different from the decomposition approaches, this approximation keeps the interdependency feature of the multiple length of rentals, but only models the demand at an aggregated level. The individual booking times are not considered. More details on the PNLP model can be found in Appendix A.

The demand streams for all products are generated randomly to create various test cases. In this numerical study we assume customers arrive according to an inhomogeneous Poisson process with an arrival rate λ_n for the advanced booking period n. Our experience is that λ_n may well increase when n reduces from N to 0. Indeed, car rental customers tend to book closer to the pick-up date. Each customer, upon arrival, requests a rental length of l days with probability p_l, with $\sum_{l=1}^{L} p_l = 1$. Assume the length of rental distribution is independent of the arrival rate. We then have $d_{n,l} \sim Pois(\lambda_n p_l)$, where λ_n and p_l are randomly generated as follows.

$$\lambda_n \sim U[1, 10], \tag{16a}$$

$$p_l \sim U(0, 1), \sum p_l = 1, 1 \leq l \leq L. \tag{16b}$$

The values of λ_n and p_l jointly determine an expected demand-supply ratio ρ for each rental day, which is defined as below.

$$\rho = \frac{\sum_{s=0}^{N+L-1} \sum_{(n,l)\in K_s} \lambda_n p_l}{C + M}, \tag{16c}$$

where K_s is the set of products as defined by (9). The numerator is the total expected on-rent that day, while the denominator is the corresponding capacity.

The test cases are generated into four categories or selling seasons, in the order of increasing ρ values.

$$\rho \in (0.0, 0.5], \qquad (\text{valley}, A), \tag{16d}$$

$$\rho \in (0.5, 1.0], \qquad (\text{low}, B), \tag{16e}$$

$$\rho \in (1.0, 1.5], \qquad (\text{high}, C), \tag{16f}$$

$$\rho \in (1.5, 2.0], \qquad (\text{peak}, D). \tag{16g}$$

The rental rates and shuttling cost are fixed values, as shown below. The rental rate per day is decreasing and convex with the length of rentals, which reflects the pricing strategy in practice.

$$r_{n,l} = \begin{cases} £33, l = 1, \forall n \\ £52, l = 2, \forall n \\ £66, l = 3, \forall n \\ £84, l = 4, \forall n \end{cases} \tag{16h}$$

$$o = £30. \tag{16i}$$

Without loss of generality, we have assigned the same rental rates to products with the same length of rentals but different pick-up dates. This can be readily relaxed.

Finally, the number of cars are sampled from discrete uniform distributions as follows.

$$C \sim DU[2, 8], M \sim DU[1, 2]. \tag{16j}$$

We first consider the special case of $N = 0$, where customers walk to the rental station and pick up the cars straightaway. Three sub-scenarios are considered with different values of L. We then extend to general cases where $N \geq 1$. Again three sub-scenarios are generated for different combinations of N and L. For each of these sub-scenarios, 400 problem instances are randomly generated, with 100 for each category. For small problems, it is possible to obtain the exact optimal policy π^*, which is used to evaluate the upper bound and heuristic policies' performances. For larger problems, the exact optimal policy is not possible to obtain within reasonable time. In such cases, the gap between the upper bound and each of the heuristics offers a representative indicator of their performance.

As suggested in Remark 1, the shuttling cost has significant influence on booking control policies. In the last section of numerical study a sensitivity analysis is conducted to investigate the impact of the shuttling cost on the performance of alternative heuristic policies.

5.1. Booking Upon Pick-ups ($N = 0$)

In such situations customers book a car and pick it up straightaway. We consider three scenarios, one for each $L \in \{2, 3, 4\}$.

For each category A, B, C and D, 100 problems are generated randomly according to (16a) - (16j). For all these problems the optimal policy π^* is obtained. The suboptimality of each heuristic is then calculated as a percentage deviation from the optimum from an initial empty state. The performance of the upper bound is reported in the similar way. We have added a $+/-$ sign to indicate explicitly their position to the optimum. Table 2 summarises all the results, with 95% confidence intervals depicted in Figure 3. For each policy π, the confidence intervals are calculated as $\mu_\pi \pm 1.96 s_\pi/10$, where μ_π and s_π are the sample mean and the sample standard deviation of the suboptimality across the 100 instances. The confidence intervals for the upper bound are calculated in the same way.

Category		$L=2$				$L=3$				$L=4$			
		π^{ADP}	π^{DB}	π^{PNLP}	ub	π^{ADP}	π^{DB}	π^{PNLP}	ub	π^{ADP}	π^{DB}	π^{PNLP}	ub
A	min	0.00	0.00	-0.57	+0.01	0.00	0.00	-2.59	+0.31	0.00	-0.06	-4.56	+0.86
	mean	0.00	-0.07	-3.71	+0.30	0.00	-0.25	-7.78	+1.01	0.00	-0.54	-11.92	+2.09
	max	0.00	-0.54	-13.38	+1.19	-0.01	-0.95	-16.88	+1.95	-0.03	-3.48	-21.43	+3.21
B	min	0.00	0.00	-3.27	+0.13	0.00	0.00	-6.74	+1.49	0.00	-0.20	-7.52	+2.45
	mean	-0.05	-0.44	-8.54	+1.53	-0.05	-0.91	-13.31	+3.24	-0.04	-1.26	-15.85	+5.42
	max	-0.89	-2.02	-29.76	+4.60	-0.61	-2.37	-29.52	+6.81	-0.44	-3.94	-25.79	+9.41
C	min	0.00	0.00	-1.89	+0.35	0.00	0.00	-3.32	+1.09	-0.01	-0.15	-6.23	+3.71
	mean	-0.08	-0.47	-6.22	+2.27	-0.14	-0.73	-10.01	+4.41	-0.20	-1.58	-14.49	+7.06
	max	-0.84	-1.76	-15.03	+5.03	-0.85	-2.99	-26.51	+9.40	-0.64	-5.14	-38.27	+11.81
D	min	0.00	0.00	-1.40	0.00	0.00	0.00	-1.94	+0.58	0.00	0.00	-3.53	+2.68
	mean	-0.12	-0.27	-4.40	+2.36	-0.12	-0.83	-7.10	+3.99	-0.25	-1.28	-10.67	+6.99
	max	-0.85	-2.08	-14.32	+4.66	-0.58	-9.82	-21.70	+8.02	-1.39	-6.16	-34.79	+17.34

Table 2: Percentage Deviation from the Optimum for Heuristic Policies with $N = 0$.

It is shown that policy π^{ADP} has very strong performance. Its mean suboptimality is -0.25% or less across all problem instances, with the worst performance just at -1.39%. π^{DB} also performs strongly. The mean suboptimality is always less than -1.58% and the worst case is at most -9.82%. In contrast, π^{PNLP} has a mean suboptimality ranging from -3.71 % to -15.85%. The worst case suboptimality could be as large as -38.27%. It is outperformed by π^{ADP} and π^{DB} in both the expectation and variability. The confidence intervals in Figure 3 have shown that the differences are significant. Those two alternatives are not only better on average, but also more robust. The upper bound is tight, with the mean distance above the optimum less than 7.06% across all problem instances.

A closer check reveals that the reason why π^{PNLP} does not perform well is due to its reluctance to shuttle cars in. With limited knowledge on the dynamics of future demand, the PNLP model views shuttling a risky decision. Its logic is that the cars being shuttled would sit in the car park should not enough demand realised. Therefore, π^{PNLP} is rather risk averse and decides not to move cars in. Potential revenue opportunities are lost. As we shall see in Section 5.3, reducing the shuttling cost will motivate π^{PNLP} to move more cars in and thus significantly improve its performance.

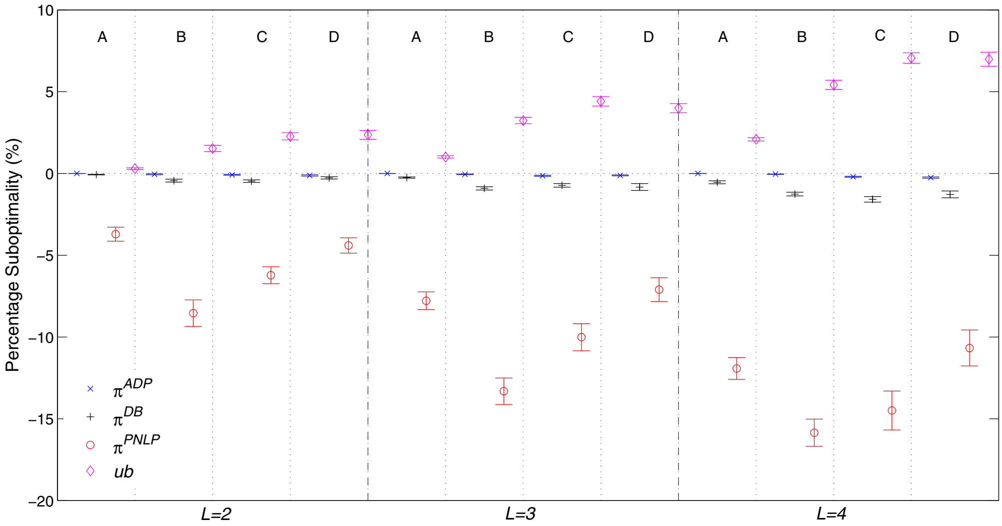


Figure 3: 95% Confidence Intervals of the Percentage Suboptimality for Heuristic Policies when $N = 0$

For all policies, the performance deteriorates with L. The tightness of the upper bound decreases as well. This is expected as the approximation is obtained by relaxing the interdependence between multiple rental days. The longer the length of rentals, the larger the approximation errors. However, it is shown that π^{PNLP}'s performance deteriorates more quickly than its alternatives.

Both π^{ADP} and π^{DB} demonstrate reduced performance with the increase of demand-supply ratio ρ. They perform very well when the demand is low. In these situations, there are normally enough cars compared to demand. In case the demand is high cars are moved in under both policies to meet extra demand. With the increase of the demand the capacity becomes relatively scarce, the rationing of capacity between multiple products more important, and the difference in the revenue performance resulted from different rationing policies more distinct.

In contrast, as shown in Figure 3, the performance of π^{PNLP} gets worse at first just as its competitors, but it then bounces back with high demand. The improved performance in such situations is explained as follows. Due to the large number of demand arrivals, the capacity becomes insufficient, especially on those days which have already been heavily booked. More shuttling is thus proposed by the PNLP model as the risk of cars sitting on the car park is deemed very low. The consequence due to risk averse in shuttling movement is mitigated, and performance improved.

The average computational time in seconds for the exact DP and the three heuristics is given in Table 3 below. All experiments were run on high performance computing clusters. Each node in the cluster has 256GB RAM and two Intel Xeon E5 processors, and each processor has multiple cores with a CPU speed of 2.50GHz. It is shown that quite long time has been spent on π^{PNLP}. Even though each PNLP takes little time to finish, one has to solve a PNLP program for every state and the accumulated time is significant. The computational time for the optimal policy π^* is relatively short for these small testing cases. Nevertheless, it increases exponentially with both the demand volume and length of rentals. In sharp contrast, our proposed heuristic π^{DB} takes almost no time in all problem cases. It should however be noted that the solution time for π^{ADP}, even though much less than π^{PNLP} and π^*, increases quickly with the problem size.

Category	$L=2$				$L=3$				$L=4$			
	π^{ADP}	π^{DB}	π^{PNLP}	π^*	π^{ADP}	π^{DB}	π^{PNLP}	π^*	π^{ADP}	π^{DB}	π^{PNLP}	π^*
A	0.00	0.00	2.65	0.01	0.08	0.03	33.93	1.15	3.46	0.14	387.35	48.40
B	0.00	0.00	2.09	0.23	0.13	0.08	26.90	1.55	3.59	0.20	262.93	52.61
C	0.00	0.00	1.99	0.01	0.18	0.04	26.90	3.09	8.65	0.49	246.66	125.44
D	0.00	0.00	1.73	0.19	0.21	0.09	21.16	3.55	15.84	0.79	259.50	215.70

Table 3: Average Computational Time (in seconds) for Cases with $N=0$.

5.2. *General Cases ($N \geq 1$)*

We consider different combinations of (N, L) with general values in the set of $\{(1,2),(1,3),(2,2)\}$. All the other problem parameters are sampled according to (16a) - (16j). Again, 100 problems are randomly generated for each category A, B, C and D.

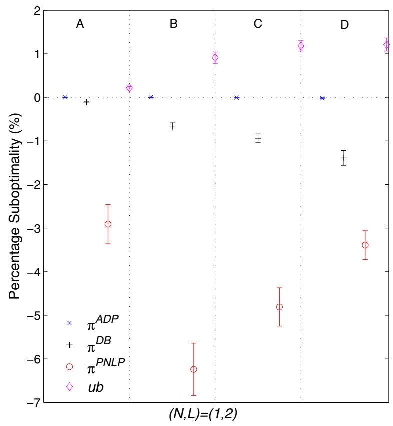
For the first combination $(1,2)$, it is possible to calculate the optimal policy. The results are reported in Table 4, in the same way as Table 2. For the other two combinations, it has not been possible to calculate the optimal policy or π^{ADP} within reasonable time. However we are still able to derive the upper bound and develop the other two heuristic policies. Their performances are then evaluated by $\Delta(\pi^{DB}, ub)$ and $\Delta(\pi^{PNLP}, ub)$, the percentage deviation from the upper bound, as shown in Table 4. The percentage difference in revenue received by π^{DB} compared with π^{PNLP}, denoted as $\Delta(\pi^{DB}, \pi^{PNLP})$, is also reported in this table. Hence positive values indicate stronger performance from π^{DB} while negative values stronger performance from π^{PNLP}. Again, the confidence intervals are plotted in Figure 4, where the last two scenarios are separated from the first one.

Category		$(N,L)=(1,2)$				$(N,L)=(2,2)$			$(N,L)=(1,3)$		
		π^{ADP}	π^{DB}	π^{PNLP}	ub	$\Delta(\pi^{DB}, ub)$	$\Delta(\pi^{PNLP}, ub)$	$\Delta(\pi^{DB}, \pi^{PNLP})$	$\Delta(\pi^{DB}, ub)$	$\Delta(\pi^{PNLP}, ub)$	$\Delta(\pi^{DB}, \pi^{PNLP})$
A	min	0.00	0.00	-0.50	+0.01	-0.04	-0.94	+0.87	-0.49	-3.14	+1.94
	mean	0.00	-0.11	-2.91	+0.22	-0.24	-2.50	+2.27	-2.12	-8.79	+6.82
	max	0.00	-0.63	-11.37	+0.64	-0.73	-7.72	+7.26	-3.69	-26.35	+24.63
B	min	0.00	-0.08	-2.33	+0.13	-0.35	-2.56	+2.00	-1.72	-7.82	+3.79
	mean	0.00	-0.66	-6.24	+0.91	-1.60	-6.02	+4.50	-4.73	-14.50	+10.29
	max	-0.17	-2.27	-25.79	+3.27	-3.71	-11.48	+10.50	-10.53	-37.59	+30.57
C	min	0.00	-0.04	-2.10	+0.21	-0.52	-2.67	+0.20	-1.52	-5.78	+1.12
	mean	-0.01	-0.94	-4.81	+1.18	-2.30	-6.32	+4.11	-5.73	-12.81	+7.53
	max	-0.07	-2.35	-15.27	+2.43	-5.26	-12.65	+10.20	-15.13	-39.24	+35.68
D	min	0.00	0.00	-0.45	0.00	-0.28	-1.02	**-0.30**	-1.72	-4.72	**-0.44**
	mean	-0.02	-1.39	-3.39	+1.21	-2.23	-5.03	+2.87	-5.17	-9.28	+4.35
	max	-0.23	-4.76	-9.37	+3.51	-6.18	-18.50	+15.93	-9.53	-19.85	+12.88

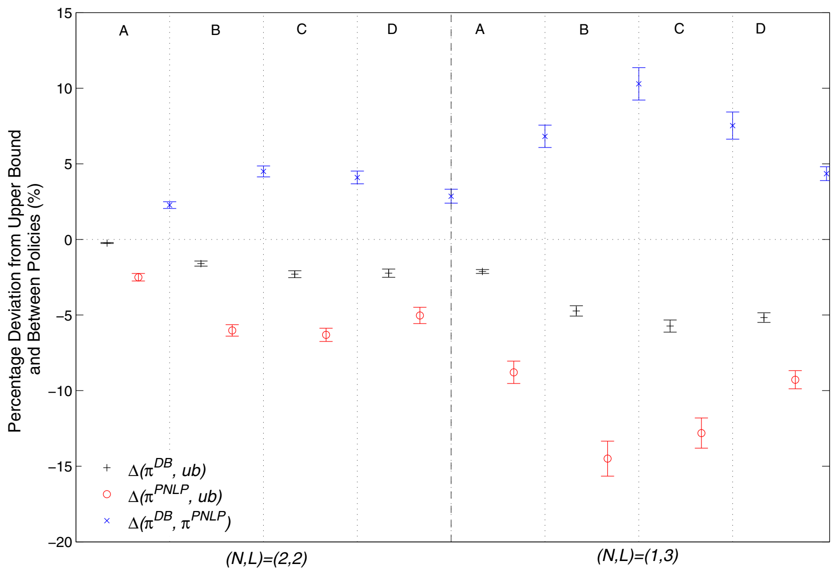
Table 4: Performance of Heuristic Policies and the Upper Bound When $N \geq 1$.

The evidence provided by scenario $(1,2)$ in Table 4 yields similar conclusions to those drawn from Table 2. π^{ADP} and π^{DB} continue to have strong performance and outperform comfortably π^{PNLP} in all the problem instances. The upper bound is still tight.

In both scenarios of $(1,3)$ and $(2,2)$, π^{DB} achieves a smaller distance from the upper bound and offers higher revenue performance than π^{PNLP} in almost all problem instances, which is in agreement with the previous results. Again, the performance of π^{PNLP} deteriorates first and then improves significantly with the demand-supply ratio, as shown in Figure 4. In both scenarios there is a remarkable new observation. In Category D, when the demand is significantly higher than capacity, π^{PNLP} could beat its competitor and



(a)



(b)

Figure 4: 95% Confidence Intervals of the Performance of Heuristic Policies and the Upper Bound when $N \geq 1$

this is evidenced by the minimum value of -0.30% and -0.44% for $\Delta(\pi^{DB}, \pi^{PNLP})$ in both scenarios, even though on average π^{DB} is still better (the mean of $\Delta(\pi^{DB}, \pi^{PNLP})$ is 2.87% and 4.35% respectively). It suggests that in some special situations π^{PNLP} could perform quite well. We include further discussions on such cases in the following section.

The results of the computational time for these scenarios, which are not included for the sake of brevity, are similar to those reported in the previous section.

5.3. Sensitivity Analysis on Shuttling Cost

In the numerical study so far the monetary data are fixed. In this section we investigate the performance of the heuristics in respect to the shuttling cost. The following shuttling cost values are considered, with the middle one having been used in the previous studies.

$$o = \{\pounds 0, \pounds 30, \pounds 50\}. \tag{17}$$

Two scenarios, $(0, 3)$ and $(1, 2)$, have been chosen for this analysis, which together cover most typical situations. The same problem instances in Section 5.1 and 5.2 are used except that the shuttling cost is replaced by each of those in (17). For all these problems the optimal policy can be obtained and the performance of the heuristics is evaluated in the same way as in Table 2. The results for the two scenarios are reported in Table 5a and 5b, respectively.

Category		o=£0			o=£30			o=£50		
		π^{ADP}	π^{DB}	π^{PNLP}	π^{ADP}	π^{DB}	π^{PNLP}	π^{ADP}	π^{DB}	π^{PNLP}
A	min	0.00	0.00	-0.50	0.00	0.00	-2.59	0.00	0.00	-2.58
	mean	0.00	-0.27	-3.02	0.00	-0.25	-7.78	0.00	-0.26	-7.59
	max	-0.01	-0.96	-9.44	-0.01	-0.95	-16.88	-0.02	-0.85	-16.41
B	min	0.00	0.00	-1.14	0.00	0.00	-6.74	0.00	0.00	-4.78
	mean	-0.02	-0.78	-4.71	-0.05	-0.91	-13.31	-0.11	-0.89	-10.70
	max	-0.24	-4.84	-11.55	-0.61	-2.37	-29.52	-1.20	-10.71	-23.56
C	min	0.00	0.00	0.00	0.00	0.00	-3.32	0.00	0.00	-0.52
	mean	-0.11	-0.96	-3.20	-0.14	-0.73	-10.01	-0.22	-0.73	-5.70
	max	-0.70	-4.60	-11.81	-0.85	-2.99	-26.51	-1.17	-3.22	-21.88
D	min	0.00	0.00	0.00	0.00	0.00	-1.94	0.00	0.00	0.00
	mean	-0.15	-0.74	-2.20	-0.12	-0.83	-7.10	-0.24	-0.94	-3.21
	max	-1.39	-5.09	-11.42	-0.58	-9.82	-21.70	-1.58	-14.81	-45.19

(a) Problems with $N = 0, L = 3$.

Category		o=£0			o=£30			o=£50		
		π^{ADP}	π^{DB}	π^{PNLP}	π^{ADP}	π^{DB}	π^{PNLP}	π^{ADP}	π^{DB}	π^{PNLP}
A	min	0.00	0.00	-0.01	0.00	0.00	-0.50	0.00	0.00	-0.49
	mean	0.00	-0.10	-0.40	0.00	-0.11	-2.91	0.00	-0.11	-2.64
	max	0.00	-0.53	-1.54	0.00	-0.63	-11.37	-0.01	-0.73	-9.66
B	min	0.00	-0.08	0.00	0.00	-0.08	-2.33	0.00	-0.06	-1.93
	mean	0.00	-0.66	-1.08	0.00	-0.66	-6.24	-0.01	-0.55	-4.20
	max	-0.21	-2.36	-28.53	-0.17	-2.27	-25.79	-0.12	-3.59	-18.35
C	min	0.00	-0.18	0.00	0.00	-0.04	-2.10	0.00	0.00	0.00
	mean	-0.01	-1.34	-1.11	-0.01	-0.94	-4.81	-0.02	-1.15	-1.70
	max	-0.08	-3.06	-5.78	-0.07	-2.35	-15.27	-0.12	-4.71	-6.90
D	min	0.00	0.00	0.00	0.00	0.00	-0.45	0.00	0.00	0.00
	mean	-0.03	-1.63	-0.84	-0.02	-1.39	-3.39	-0.02	-1.81	-1.08
	max	-0.30	-5.43	-5.24	-0.23	-4.76	-9.37	-0.20	-6.32	-6.68

(b) Problems with $N = 1, L = 2$.

Table 5: Percentage Deviation from the Optimum for Heuristic Policies with Different Shuttling Cost.

The results in Table 5a show that π^{ADP} has strong performance regardless of the shuttling cost. Similarly, π^{DB} works well across the board. There is no obvious pattern in respect to the shuttling cost for both policies. In clear contrast, π^{PNLP} achieves the best results when the shuttling cost is zero, then deteriorates when the shuttling cost increases to £30, before improves slowly at the cost of £50. Such a pattern is apparent in all four categories. No matter which situation it is still outperformed by the competitor policies.

In Table 5b, the overall performance improves for every policy compared to that in Table 5a, which agrees with the previous conclusions that the performance improves with shorter *LoR*. The similar pattern for π^{PNLP} is observed in respect to the shuttling cost and demand-supply ratio. The major new finding is that it outperforms π^{DB} at peak seasons when there is either free shuttling or a heavy shuttling charge. In such situations the mean suboptimality for π^{PNLP} is -0.84% (free shuttling) and -1.08% (heavy shuttling charges), while for π^{DB} it is -1.63% and -1.81%, respectively. This behaviour has been identified in Section 5.2 but it becomes more prevalent when the shuttling cost is either too high or too low compared to rental prices. Such situations are rare in real life. They could happen most likely during peak seasons in downtown rental stations, when the demand is significantly higher than normal and the shuttling is relatively cheap due to proximity to other stations.

In summary, the policy π^{PNLP} is sensitive to the shuttling cost, while both π^{ADP} and π^{DB} demonstrate robust performances. In most shuttling cost scenarios π^{PNLP} is beaten by its competitor policies, even though it could outperform π^{DB} in some special cases.

6. Case Study: Policy Performances Using Industry Data

Despite the strong performance of our proposed approach in the randomly generated scenarios, it is still unclear to us their scalability and performance in more realistic settings. In this section we study the performance of heuristic policies with industry data from a major car rental company in the UK. Problems of practical sizes are way beyond the capability of exact solutions. Neither is it possible to derive the policy π^{ADP} or even evaluate policy performances (via equation (7)) in reasonable time. However it is still possible to derive the policy π^{DB} and π^{PNLP} on the fly, which allows us to undertake the study via simulation. We only need to solve the decomposed DP (11)-(12) once before the simulation. The policy π^{DB} can then be quickly obtained via (16) for each visited state on the simulated sample path. Similarly, π^{PNLP} is obtained by solving a PNLP program (A.1)- (A.4) for each visited state.

We consider two rental stations from the case rental company. One station is in downtown London and the other an airport station in north-east England. The booking pattern in downtown is distinct from that in airport stations. Downtown customers usually book just before their rentals start, while airport customers tend to book further in advance of their journey. For both stations, the number of pick-ups per day are well fitted by Poisson distributions.

Despite the very different booking patterns between these two stations, the length of rental distribution is somewhat similar. In both stations, more demand is seen for shorter rentals. Moreover, the majority of them are returned on time, which well justifies our assumption on deterministic rental durations.

We have chosen $N = 6$ and $L = 4$ for the performance comparison study between the two heuristics. Such a choice includes the majority of demand and covers the most profitable rentals, and thus represents typical real situations. Since both heuristics concern a single car group, we have selected two car groups with different sizes in both stations. The rental rates are directly captured from the rental company's website for the same rentals in both stations. The shuttling costs are obtained as the average of historical shuttling charges. Fleet sizes are prorated for the selected length of rentals and the proportion of rental days. The number of cars available for shuttling each day, or M, is set to 2 for all instances.

A complete description of the case study setting can be found in Appendix B, where Table B.1 lists the detailed parameters which are used in the simulation.

In each simulation replication a 90 day time horizon is covered, which is long enough for convergence of the total discounted revenue. At the beginning of each time period, a state is observed for which booking limits under policy π^{DB} and π^{PNLP} are calculated. The number of bookings accepted are obtained as the minimum of the booking limits and the sampled customer arrivals during that period. Shuttling is triggered if the resulted cars on rent are more than the fleet size. At the end of each replication, the total discounted revenue is summed over 90 days, along with the total number of shuttling movements. We also calculate the utilisation as a percentage of total rental days out of the fleet days over the time horizon. Common random numbers are generated for comparison between these two policies. The simulation is replicated 5,000 times for each station and car group combination. The average results are presented in Table 6. The solution time spent on the decomposed DP is included in the last column. We have used high performance computing cluster for the simulation. It is clear that the run time increases exponentially with the problem size. The solution time for PNLP is negligible. Despite significantly longer computational time, the decomposed DP is still manageable, as in practice the booking limits usually update once a day and thus the computation may be run overnight.

Station	Car Group	Revenue (£)		Utilisation		Shuttling		No. Wins		Decomposed DP Solution Time
		π^{DB}	π^{PNLP}	π^{DB}	π^{PNLP}	π^{DB}	π^{PNLP}	π^{DB}	π^{PNLP}	
Airport	A	1518(1423, -6.3%)	1229(1220, -0.7%)	0.86(0.88)	0.71(0.75)	40(44)	0.1(0.1)	5000(4997)	0(3)	1572s(287s)
	B	1073(1143, 6.5%)	807(818, 1.4%)	0.60(0.68)	0.43(0.45)	10(15)	0(0)	4996(4990)	4(10)	30s(11s)
Downtown	A	1573(1569, -0.3%)	1246(1240, -0.5%)	0.85(0.80)	0.68(0.64)	43(33)	8(3.4)	4996(4999)	4(1)	100s(134s)
	B	1004(935, -6.9%)	644(628, -2.5%)	0.69(0.60)	0.43(0.40)	20(15)	0.2(0.2)	4987(4975)	13(25)	4s(7s)

(Note: Numbers in parentheses are the results with swapped booking patterns between the two stations).

Table 6: Simulation Results

It is shown that our proposed policy π^{DB} comfortably outperforms π^{PNLP} in revenue in all the four scenarios. Moreover, π^{DB} wins nearly all of the 5,000 simulation instances in every scenario. The winning number for π^{PNLP} is in single digit, with the only exception of 13 in downtown for car group B. The number of shuttling movements is in sharp contrast between these two policies. While π^{DB} proposes on average 40 movements of group A cars in the airport station within the 90 day period, no movement has been suggested by π^{PNLP}. The other scenarios have similar observations.

π^{DB} beats its competitor not only in revenue performance, but also in utilisation, a key performance indicator for car rental companies. It has achieved high utilisations for car group A in both stations, at 0.86 and 0.85 respectively. It is a different story for π^{PNLP}. The utilisation is universally lower under this policy, with the worst case of 0.43 for car group B in both stations. where the figure for π^{DB} is 0.60 (airport) and 0.69 (downtown) respectively.

The limitations of PNLP would have contributed to its inferior performance. Firstly, PNLP optimises booking limits over a finite booking horizon. The impact of the derived booking policies on the dynamics of the system beyond the booking horizon is ignored. Secondly, it only addresses the demand uncertainty at the aggregated level (the overall demand for a product (n, l)), while DP methods consider demand arrivals at each booking period and hence the booking decisions in each period take into account the demand information up to each period. To improve the performance of π^{PNLP}, a typical technique is to conduct the re-optimization more often so that the latest information can be fully used.

The booking patterns are different between airport and downtown. We are interested in understanding the impact of such a difference on the policy performance. We swap the booking patterns in Table B.1a between these two stations, while keep all the other parameters unchanged. The simulation results are reports in parentheses in Table 6.

It is clear that the booking pattern does have an impact on the policy performance, especially for π^{DB}. In the airport station, its revenue decreases by 6.3% for car group A and increases by 6.5% for car group B. Despite a small decrease of revenue for car group A in the downtown station, a notable 6.9% decrease is seen for car group B. In contrast the impact of the booking pattern on π^{PNLP} is much smaller, with the revenue changes within -1.4% to 2.5%. This is not surprising as the PNLP model does not consider booking patterns but only the aggregated final demand. Nevertheless its performance is always dominated by π^{DB}, as shown by the detailed simulation results in Table 6, which also indicate that the booking pattern has obvious impacts on the solution time of the decomposed DP.

7. Concluding Remarks

We study the dynamic booking control for a single-station car rental revenue management problem with flexible capacity. The problem is formulated into a discrete-time Markov decision process whose exact solution is not tractable due to the large state and action spaces. We propose a decomposition approach based on which two heuristic policies are developed. The first heuristic π^{ADP} uses the value functions of the decomposed problems with single-day length of rental to approximate the value function of the original problem and obtains booking policies via an approximate dynamic programming approach. Nevertheless, such an ADP approach only applies to small size problems. The second heuristic π^{DB}constructs a booking control policy directly from the optimal booking limits for the decomposed problems, which is more scalable compared to the ADP approach.

Our numerical study concludes that the proposed heuristics are close to optimum and outperform the commonly used PNLP heuristic π^{PNLP} in most instances. The case study with industrial data from a major car rental company in the UK confirms the dominant performance of π^{DB} over π^{PNLP} in practical situations. Our results further suggest that PNLP heuristic is conservative in shuttling movements and sensitive to the shuttling cost. It might work well in situations with large demand and low shuttling cost, such as the peak selling seasons in downtown rental stations within large cities where a cluster of proximity stations make shuttling much more affordable. In general situations car rental stations normally face sparse and intermittent demand over time, for which policy π^{DB} is capable of generating higher revenue as well as higher fleet utilisation. The case study also confirms that the booking pattern has a significant impact on the policy performance, which explains largely why the PNLP approach that ignores this critical information generally delivers poor results.

An important limitation of this paper is that only a single car group is considered. In practice product upgrade is often exercised in car rental operations. It is not uncommon that the demand for a lower car group is satisfied with vehicles from higher car groups. Hence an interesting work could extend to multiple car groups and thus incorporate product upgrade into the booking controls. Moreover, we have assumed the length of rental is deterministic. In other words, customers always drive the cars according to the agreed length of rental upon booking. Even though most customers would return cars on time, many of them may well return their cars either earlier or later, which then significantly changes the car availability over time. Uncertainty of length of rental is therefore another interesting problem to address in the future.

Acknowledgements

The authors are grateful to the valuable support of AVIS Budget EMEA for this research.

References

Brumelle, S. L., McGill, J. I., 1993. Airline seat allocation with multiple nested fare classes. Operations Research 41 (1), 127–137.

Carroll, W. J., Grimes, R. C., 1995. Evolutionary change in product management: Experiences in the car rental industry. Interfaces 25 (5), 84–104.

Chen, X., Gao, X., Hu, Z., 2015. A new approach to two-location joint inventory and transshipment control via $L^\natural$-convexity. Operations Research Letters 43 (1), 65–68.

Cross, R. G., Higbie, J. A., Cross, Z. N., 2011. Milestones in the application of analytical pricing and revenue management. Journal of Revenue & Pricing Management 10 (1), 8–18.

FICO, 2009. Optimization drives $19 million gains at Avis. Accessed: 06/2014.
URL `http://www.fico.com/en/products/fico-xpress-optimization-suite/xpress-optimization-suite-resources/#SuccessStories`

Fink, A., Reiners, T., 2006. Modeling and solving the short-term car rental logistics problem. Transportation Research Part E: Logistics and Transportation Review 42 (4), 272–292.

Gans, N., Savin, S., 2007. Pricing and capacity rationing for rentals with uncertain durations. Management Science 53 (3), 390–407.

Geraghty, M. K., Johnson, E., 1997. Revenue management saves national car rental. Interfaces 27 (1), 107–127.

Guerriero, F., Olivito, F., 2014. Revenue models and policies for the car rental industry. Journal of Mathematical Modelling and Algorithms in Operations Research 13 (3), 247–282.

Haensel, A., Mederer, M., Schmidt, H., 2012. Revenue management in the car rental industry: A stochastic programming approach. Journal of Revenue & Pricing Management 11 (1), 99–108.

Liu, Q., Van Ryzin, G., 2008. On the choice-based linear programming model for network revenue management. Manufacturing & Service Operations Management 10 (2), 288–310.

Meissner, J., Strauss, A., 2012. Network revenue management with inventory-sensitive bid prices and customer choice. European Journal of Operational Research 216 (2), 459–468.

Netessine, S., Shumsky, R. A., 2005. Revenue management games: Horizontal and vertical competition. Management Science 51 (5), 813–831.

Pachon, J. E., Iakovou, E., Ip, C., Aboudi, R., 2003. A synthesis of tactical fleet planning models for the car rental industry. IIE Transactions 35 (9), 907–916.

Powell, W. B., 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley and Sons, Hoboken, New Jersey.

Robinson, L. W., 1995. Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. Operations Research 43 (3), 252–263.

Savin, S. V., Cohen, M. A., Gans, N., Katalan, Z., 2005. Capacity management in rental businesses with two customer bases. Operations Research 53 (4), 617–631.

Schmidt, H., 2009. Simultaneous control of demand and supply in revenue management with flexible capacity. PhD thesis, Clausthal University of Technology, Germany.

Steinhardt, C., Gönsch, J., 2012. Integrated revenue management approaches for capacity control with planned upgrades. European Journal of Operational Research 223 (2), 380–391.

Talluri, K. T., van Ryzin, G. J., 2004. The Theory and Practice of Revenue Management. Kluwer, Boston.

Vossen, T. W., Zhang, D., 2014. A dynamic disaggregation approach to approximate linear programs for network revenue management. Production and Operations Management.

Zhang, D., 2011. An improved dynamic programming decomposition approach for network revenue management. Manufacturing & Service Operations Management 13 (1), 35–52.

Zhang, D., Adelman, D., 2009. An approximate dynamic programming approach to network revenue management with customer choice. Transportation Science 43 (3), 381–394.

Appendix A Probabilistic Non-linear Program (PNLP)

Here we present a PNLP model introduced in Schmidt (2009). Changes have been made where necessary to adapt to the setting of this work. A discount factor is included to be in line with the problem setting. Moreover, due to the different product definition in Schmidt (2009), the following mapping has been established.

We define an alternative product as the combination of the pick-up date z and length of rental l, or (z, l). It is essentially an aggregated product which comprises a group of product (n, l) with the same l but different n. Denote by $D_{z,l}$ the total demand still to come for product (z, l). The following relationship exists between the aggregated demand $D_{z,l}$ and the demand arrivals during each booking period $d_{n,l}$,

$$D_{z,l} = \sum_{n=0}^{min(z,N)} d_{n,l}, \forall 0 \leq z \leq N + L - 1, 1 \leq l \leq L.$$

A probabilistic non-linear program for a finite time horizon of $N + L$ can be formulated as below. The capacity outside of this time horizon is deemed as infinite.

$$\text{PNLP: maximise} \sum_{z=0}^{N+L-1} \gamma^z \sum_{l=1}^{L} r_{z,l} u_{z,l} - o \sum_{z=0}^{N+L-1} \gamma^z q_z, \tag{A.1}$$

$$\text{s.t. } u_{z,l} = \mathbb{E}\left[\min\{B_{z,l}, D_{z,l}\}\right], \forall 0 \leq z \leq N + L - 1, 1 \leq l \leq L, \tag{A.2}$$

$$\sum_{i=0}^{z} \sum_{l=z-i+1}^{L} B_i^l + x_z \leq C + q_z, \forall 0 \leq z \leq N + L - 1, \tag{A.3}$$

$$0 \leq q_z \leq M, \forall 0 \leq z \leq N + L - 1, \tag{A.4}$$

where $B_{z,l}$ is the overall booking limit for product (z, l), and q_z the number of cars to be shuttled in on day z. Both decision variables can only take integer values. Still x_z denotes the number of cars having been booked or already on the road for day z.

Note that $u_{z,l}$ is the expected demand being accepted for product (z, l), which is the expectation of the minimum of the booking limit and future demand. Constraint (A.3) enforces that at most $C + q_z$ cars are available for rent each day, and constraint (A.4) specifies the maximum number of cars available for shuttling from other stations. The objective function (A.1) maximises the total (discounted) expected revenue across the considered time horizon, with the first term the revenue and the second the shuttling cost.

The objective function and the constraints are all linear except (A.2), which can be linearised as follows.

$$u_{z,l} \leq \alpha^i_{z,l} B_{z,l} + \beta^i_{z,l}, \forall 1 \leq i \leq I, \tag{A.5}$$

where $\alpha^i_{z,l}, \beta^i_{z,l}$ are the parameters for the i^{th} linear function for product (z, l), and there are in total I such functions. We have set $I = C + M + 1$ in the numerical study .

Replacing (A.2) with (A.5) the PNLP reduces to a mixed integer program which can then be solved by standard algorithms. The solution to the PNLP model determines a vector of booking limits $B_{z,l}$. These cumulative booking limits need to be transformed back to the booking limits for a single booking period. Here we use a greedy heuristic and let

$$b_{n,l} = B_{n,l}, \forall n, l. \tag{A.6}$$

Essentially it tries to capture the bookings as early as possible. The resulted policy is denoted by π^{PNLP}.

We now provide more details on linearisation of (A.2). Since the demand variable $D_{z,l}$ is discrete, we can calculate (A.2) by

$$u_{z,l} = \mathbb{E}[\min\{B_{z,l}, D_{z,l}\}] = B_{z,l} - \sum_{s=0}^{B_{z,l}-1} F_{z,l}(s), \tag{A.7}$$

where $F_{z,l}$ is the cdf function for $D_{z,l}$. It is obvious that $u_{z,l}$ is an increasing and concave function of $B_{z,l}$. In light of this property and the finite bound for $B_{z,l}$, equation (A.2) can be approximated by a set of piecewise linear functions. Specifically, it can be replaced by the following constraints.

$$u_{z,l} \leq \alpha^i_{z,l} B_{z,l} + \beta^i_{z,l},$$

where $\alpha^i_{z,l}, \beta^i_{z,l}$ are the parameters for the $i^{th}(1 \leq i \leq I)$ linear function for product (z, l).

We next present how to determine these parameters. For a comprehensive account on this process refer to Talluri and van Ryzin (2004). For each product (z, l), sample $I + 1$ booking limit values in between 0 and $C + M - x_z$, denoted by $B^i_{z,l}$. Substitute each of them into equation (A.7) and denote the result by $u^i_{z,l}$. Essentially we have just calculated the expected demand to be accepted for $I + 1$ booking limit values. These $I + 1$ pairs of $(B^i_{z,l}, u^i_{z,l})$ determine I linear functions whose parameters are given by,

$$\alpha^i_{z,l} = \frac{u^{i+1}_{z,l} - u^i_{z,l}}{B^{i+1}_{z,l} - B^i_{z,l}}, \tag{A.8}$$

$$\beta^i_{z,l} = \frac{u^i_{z,l} B^{i+1}_{z,l} - u^{i+1}_{z,l} B^i_{z,l}}{B^{i+1}_{z,l} - B^i_{z,l}}. \tag{A.9}$$

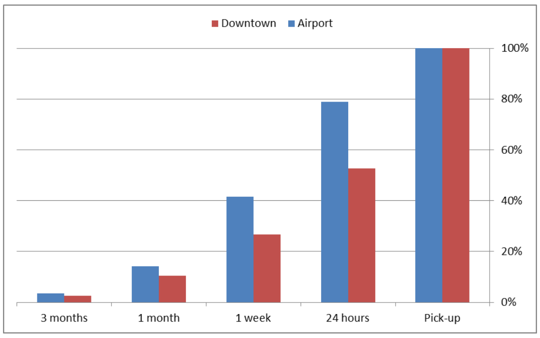


Figure B.1: Cumulative Bookings in both Downtown and Airport Stations.

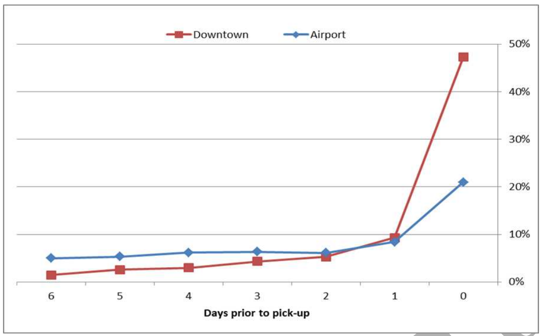


Figure B.2: Percentage Bookings in the Last Week in both Downtown and Airport Stations.

Appendix B The Case Study Setting

We consider a downtown station in London and an airport station in north-east England. Downtown customers usually book just before their rentals start. Indeed, as shown in Figure B.1, nearly half of the customer bookings are made within 24 hours of the pick-up date. More than 70% bookings are made within one week, and nearly 90% bookings within one month. On the contrary, airport customers tend to book further in advance of their journey. The biggest difference comes within one week prior to the pick-up day. More than 40% bookings have been made one week prior to the pick-up day and nearly 80% bookings before the last day, which are much more than and in sharp contrast to the figures in the downtown.

Figure B.2 zooms into the last week and shows the bookings in each of the seven days. For the airport station, customer arrivals are quite stable, within the range of 6-8% each day for the first 6 days, and then increase to 20% on the pick-up day. The downtown station has a clearly different patten. The number of arrivals starts from just 1-2% on 6 days prior to the pick-up day and increases steadily to 10% until 1 day prior to the pick-up day, before a sharp jump to 50% on the pick-up day.

For both stations, the number of pick-ups per day are well fitted by Poisson distributions. Due to the sparse and intermittent feature of the demand at the product level, it makes more sense to fit the distributions at the aggregated level over length of rentals and car groups. In Figure B.3 the observed aggregated pick-ups for a particular day of week and the fitted Poisson distribution are plotted for both the airport and downtown stations, along with the QQ plots. Chi-squared test has been carried out to assess the goodness of fit. In both situations the p-value is greater than 0.05 the significance level, and thus the null hypothesis that the data follow a Poisson distribution cannot be rejected. It is worth mentioning that the original data are de-seasonalised before fitting to remove the impact of seasonality on the distribution. The data entries during special events such as public holidays are discarded as well.

Note that we have so far approached arrival patterns backwards from pick-ups rather than forwards from bookings. The resulted arrival rates on each booking day are essentially the same. Remember we do not consider seasonality, upgrade and overbooking, and hence the average number of bookings made each day always equals the average daily pick-ups.

The length of rental distribution is similar. In both stations, more demand is seen for shorter rentals. More than a quarter customers book just one day rentals, and the total bookings for one to four day rentals are over 70%, as shown in Table B.1b below. Figure B.4 shows the distribution of returns for 1-4 day rentals in both stations. The majority of them are returned on time.

We have selected two car groups with different sizes in both stations. Car group A are mid-size cars such as Peugeot 308, and car group B large cars such as Seat Toledo. The average number of pick-ups per day is obtained from two years worth historical bookings for each station and car group combination. As shown earlier the arrival

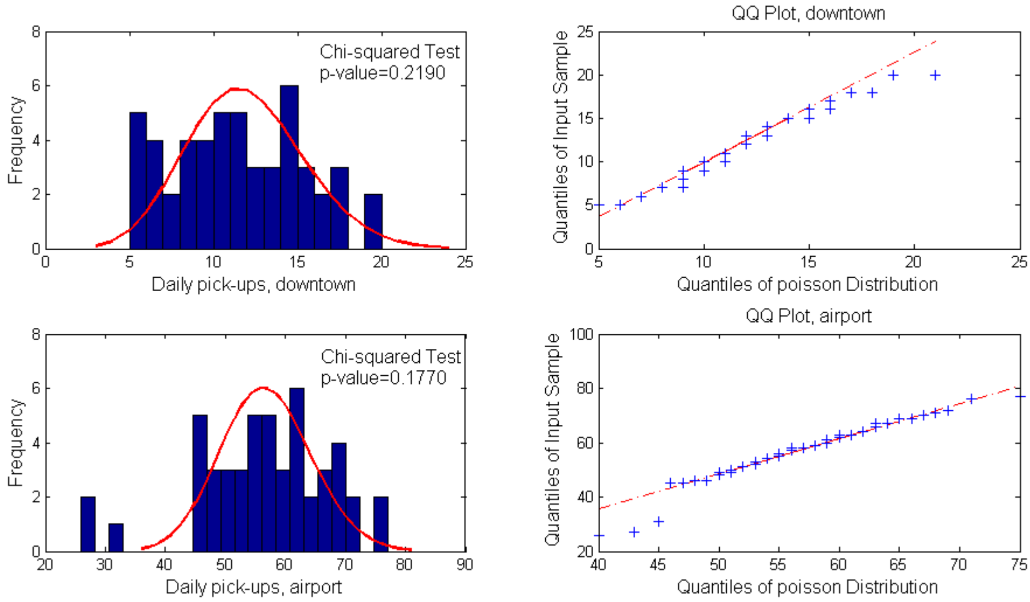
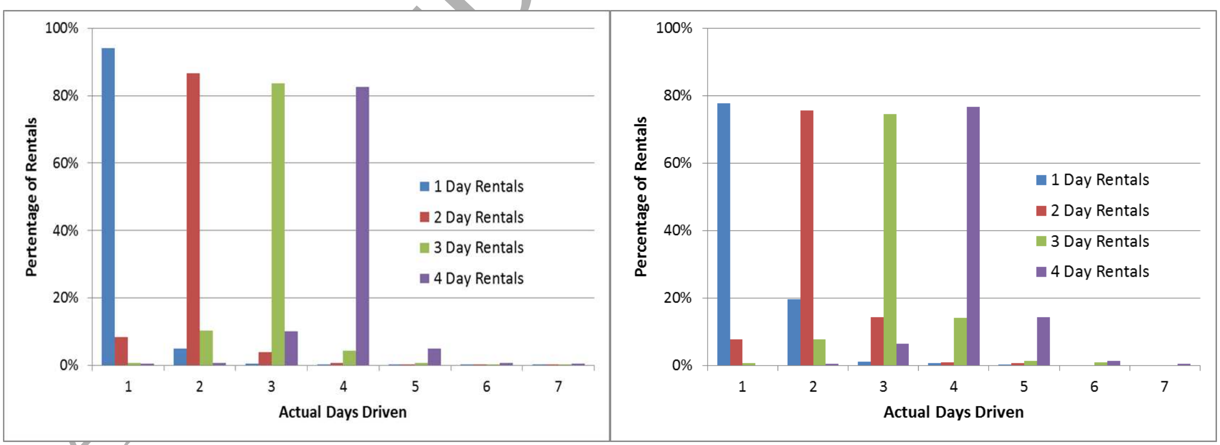


Figure B.3: The Fitted Daily Pick-up Distributions for both Downtown and Airport stations.



(a) Airport (b) Downtown

Figure B.4: The Distribution of Returns.

rate of customer bookings increases nearer the pick-up date. These arrivals are modelled by inhomogeneous Poisson processes. We calculate the arrival rate each day as a proportion of the mean number of pick-ups, according to the booking curves as illustrated in Figure B.2. The length of rental distribution is derived directly from the historical data, which along with the daily arrival rates determine the arrival rate for all the products. Note that the percentage of daily arrivals and the LoR distributions are both normalised in the calculation.

The detailed parameters are summarised in Table B.1 below. It shows that airport demand is much higher than the downtown station. In each station more demand is seen for car group A than car group B. In both stations car group B cars are more expensive than car group A, while between stations the north-east airport is significantly out-priced by London downtown station. Shuttling cost is the same between car groups but lower in London. This is because the selected downtown station is surrounded by other stations within London area, which makes it cheaper to share cars, while the airport station has just one fellow station in the nearby city centre.

	% of Bookings Each Prior Day							Mean Daily Pick-ups		Fleet Size	
Day prior	6	5	4	3	2	1	0	A	B	A	B
Airport	5.0	5.3	6.1	6.4	6.1	8.4	21.0	4.34	1.81	4	3
Downtown	1.5	2.6	3.0	4.3	5.3	9.3	47.3	3.13	1.54	3	2

(a) Demand and Fleet Size.

LoR	1	2	3	4
Percentage				
Airport	30%	22%	16%	10%
Downtown	28%	20%	17%	9%
Rental Rate				
Airport/A	£44	£66	£95	£123
Airport/B	£67	£101	£126	£164
Downtown/A	£66	£99	£127	£166
Downtown/B	£80	£121	£160	£208
Shuttling Cost				
Airport		£32		
Downtown		£27		

(b) *LoR* Distribution, Rental Rates and Shuttling Costs.

Station	LoR	Car Group A Day prior (n)							Car Group B Day prior (n)						
	(l)	6	5	4	3	2	1	0	6	5	4	3	2	1	0
Airport	1	0.14	0.15	0.18	0.18	0.18	0.24	0.60	0.06	0.06	0.07	0.08	0.07	0.10	0.25
	2	0.10	0.11	0.13	0.13	0.13	0.18	0.44	0.04	0.05	0.05	0.06	0.05	0.07	0.18
	3	0.08	0.08	0.09	0.10	0.09	0.13	0.32	0.03	0.03	0.04	0.04	0.04	0.05	0.13
	4	0.05	0.05	0.06	0.06	0.06	0.08	0.20	0.02	0.02	0.02	0.03	0.02	0.03	0.08
Downtown	1	0.02	0.04	0.05	0.07	0.09	0.15	0.77	0.01	0.02	0.02	0.03	0.04	0.07	0.38
	2	0.02	0.03	0.04	0.05	0.06	0.11	0.56	0.01	0.02	0.02	0.03	0.03	0.05	0.27
	3	0.01	0.03	0.03	0.04	0.05	0.09	0.46	0.01	0.01	0.01	0.02	0.03	0.04	0.23
	4	0.01	0.01	0.01	0.02	0.03	0.05	0.24	0.00	0.01	0.01	0.01	0.01	0.02	0.12

(c) Arrival Rates for Each Product.

Table B.1: The Case Study Parameters.

Appendix C Seasonality

Our approach also applies to the model with demand seasonality. Let $\mathcal{T}$ be the length of demand cycles. Car rental businesses usually observe a distinct weekly demand pattern. For example, the demand is higher during the weekdays and lower in the weekends for airport stations, while the pattern is reversed for downtown stations. Let $\tau \in \{0, 1, ..., \mathcal{T} - 1\}$ the index of each season within a cycle. For a decision epoch of season τ, the value function is redefined as $V_\tau(\mathbf{x})$. Therefore for the same state vector $\mathbf{x}$ there are in total $\mathcal{T}$ value functions, one for each season.

We redefine the key notations to be dependent on the season, therefore we have now $r_{n,l}(\tau), d_{n,l}(\tau)$, and $b_{n,l}(\tau)$, which represent the rental rate, customer demand, and booking limit for product (n,l) at a decision epoch or booking period of season τ. We also allow the shuttling cost to be dependent on the season, and thus have o_τ. Note that a booking in season τ for a rental with (n,l) will pick up the car in season $h(\tau+n)$ where $h(x) = (x \bmod \mathcal{T})$. Similarly, a pick-up in season τ with (n,l) was booked in season $h(\tau-n)$. The dynamics of the system state are given by

$$\tilde{\mathbf{x}} = \left(x_1 + \sum_{n=0}^{1}\sum_{l=2-n}^{L}\min\{b_{n,l}(\tau), d_{n,l}(\tau)\}, ..., x_{N+L-2} + \sum_{n=0}^{N}\sum_{l=N+L-1-n}^{L}\min\{b_{n,l}(\tau), d_{n,l}(\tau)\}, \min\{b_{N,L}(\tau), d_{N,L}(\tau)\}\right). \tag{C.1}$$

The cyclic dynamic program can be expressed as follows.

$$V_\tau(\mathbf{x}) = \max_{\mathbf{b}\in\mathcal{A}(\mathbf{x})}\{R_\tau(\mathbf{b}) - O_\tau(\mathbf{x},\mathbf{b}) + \gamma\mathbb{E}\left[V_{\tau+1}(\tilde{\mathbf{x}})\right]\}, \tau = 0, ..., \mathcal{T}-1, \tag{C.2}$$

where $V_\tau = V_0$ and

$$R_\tau(\mathbf{b}) = \sum_{n=0}^{N}\sum_{l=1}^{L} r_{n,l}(\tau)E\left[\min\{b_{n,l}(\tau), d_{n,l}(\tau)\}\right], \tag{C.3}$$

$$O_\tau(\mathbf{x},\mathbf{b}) = o_\tau E\left[\left(x_0 + \sum_l \min\{b_{0,l}(\tau), d_{0,l}(\tau)\} - C\right)^+\right]. \tag{C.4}$$

The decomposition approach also applies to the cyclic dynamic program which addresses the seasonality. For the cyclic dynamic program, rather than just solve a single finite horizon DP, we now need to solve one for each season. To this end, for a pick-up day in season τ, we define the decomposed single day value function as $\hat{v}_{s,\tau}(x)$ which satisfies the following optimality equations:

$$\hat{v}_{s,\tau}(x) = \max_{\mathbf{b^s}}\left\{\sum_{(n,l)\in K_s}\mathbb{E}\left[\frac{r_{n,l}(h(\tau-s))}{l}\min\{b^s_{n,l}(h(\tau-s)), d_{n,l}(h(\tau-s))\}\right] + \gamma\mathbb{E}[\hat{v}_{s-1,\tau}(\tilde{x})]\right\},$$
$$\forall s = 1, 2, ..., N+L-1, \tag{C.5}$$

with a boundary condition

$$\hat{v}_{0,\tau}(x) = \max_{\mathbf{b^0}}\left\{\sum_{(n,l)\in K_0}\mathbb{E}\left[\frac{r_{n,l}(\tau)}{l}\min\{b^0_{n,l}(\tau), d_{n,l}(\tau)\}\right] - o_\tau\mathbb{E}\left[x + \sum_{(n,l)\in K_0}\min\{b^0_{n,l}(\tau), d_{n,l}(\tau)\} - C\right]^+\right\}, \tag{C.6}$$

where $\tilde{x} = x + \sum_{(n,l)\in K_s}\min\{b^s_{n,l}(h(\tau-s)), d_{n,l}(h(\tau-s))\}$.

The solutions to all these DP are then used to construct an approximation to the original value function.

$$\hat{V}_\tau(\mathbf{x}) = \sum_{s=0}^{N+L-2}\hat{v}_{s,h(\tau+s)}(x_s) + \sum_{t=0}^{\infty}\gamma^t\hat{v}_{N+L-1,h(\tau+N+L-1+t)}(0).$$

With simplification we have

$$\hat{V}_\tau(\mathbf{x}) = \sum_{s=0}^{N+L-2}\hat{v}_{s,h(\tau+s)}(x_s) + \sum_{t=0}^{\mathcal{T}-1}\gamma^t\frac{\hat{v}_{N+L-1,h(\tau+N+L-1+t)}(0)}{1-\gamma^{\mathcal{T}}}. \tag{C.7}$$

Note that when $\mathcal{T} = 1$ or non-seasonality, the above equation reduces to the approximation function for stationary demand.

Using $\hat{V}_\tau(\mathbf{x})$ to approximate the value function, we can derive the heuristic policy π_τ^{ADP}, which is now dependent upon the season of the decision epoch.

$$\pi_\tau^{ADP}(\mathbf{x}) = \underset{\mathbf{b}\in\mathcal{A}(\mathbf{x})}{\arg\max} \left\{ R_\tau(\mathbf{b}) - O_\tau(\mathbf{x},\mathbf{b}) + \gamma\mathbb{E}[\hat{V}_{\tau+1}(\tilde{\mathbf{x}})] \right\}. \tag{C.8}$$

Similarly, the policy π_τ^{DB} can be generated as below. The optimal policy obtained via (C.5)-(C.6) for a season τ determines for each product $(n,l) \in K_s$ a booking limit $b_{n,l}^s(h(\tau - s))$, which specifies how many cars on this day are allocated to product (n,l) when there are s days left prior to usage. Note that the season for s is $h(\tau - s)$. In other words, to find a booking limit $b_{n,l}^s(\tau)$, one need to solve (C.5)-(C.6) for season $h(\tau + s)$. Remember this is only a single component of the requested l rental days for (n,l), and thus the booking limit for the entire rental length is given by

$$b_{n,l}(\tau,\mathbf{x}) = \min_{s=n}^{s\le n+l-1} b_{n,l}^s(\tau, x_s), \tag{C.9}$$

where $b_{n,l}^s(\tau, x_s)$ is solution of the decomposed DP for season $h(\tau + s)$. We have restored the state into the booking limit notation to emphasise their state dependency.