IMA Journal of Management Mathematics Page 1 of 34
doi:10.1093/imaman/dpnxxx

Revenue Management of Airport Carparks in Continuous Time

ANDREAS PAPAYIANNIS*
PAUL JOHNSON†
School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK
DMITRY YUMASHEV‡
Lancaster University Management School, Lancaster University, Lancaster, LA1 4YX, UK
PETER DUCK§
School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK

We study the revenue management (RM) problem encountered in airport carparks, with the primary objective to maximize revenues under a continuous-time framework. The implementation of pre-booking systems for airport carparks has spread rapidly around the world and pre-booking is now available in most major airports. Currently, most RM practices in carparks are simple adjustments of those developed for hotels, exploiting the similarities between the two industries. However, airport carparks have a distinct setting where the *price-per-day* of a parking space is heavily discounted by the *length-of-stay* of the booking. This is because the customer decision tends to be made after the length of the trip is already set, and it becomes a choice between parking or alternative modes of transport. Consequently, the length-of-stay becomes a critical variable for revenue optimization. Since customers are able to book the parking by the minute, the resulting state space is very large, making a conventional network solution intractable. Instead, decomposed single-resource problems need to be considered. Here we develop a bid-price control strategy to manage the bookings and propose novel approaches to define such bid-prices depending on the length of stay, which could be utilized in real-time RM algorithms. Managing stochastic carpark bookings by length-of-stay in the decomposed single-resource approximation allowed us to achieve within 5% of the expected revenues for a multi-resource approximation, with a fraction of the computational effort. When expected demand exceeds the available parking capacity, the method increases the revenues by up to 45% relative to the first-come-first-serve acceptance policy.

Keywords: revenue management, dynamic programming, continuous-time, length-of-stay

1. Introduction

In this paper, we study the revenue management (RM) problem encountered in airport carparks, with our primary objective to maximize revenues under a continuous-time framework. RM has been accredited for the success in various industries. Originally developed and implemented for airlines (Smith et al., 1992), it quickly expanded to hotels (Goldman et al., 2002; Badinelli, 2000) and car rentals (Geraghty and Johnson, 1997; Li and Pang, 2016), while extending to other less traditional industries such as restaurants (Hwang et al., 2010; Guerriero et al., 2014) and cruises (Ladany and Arbel, 1991; Maddah et al., 2010). An overview of research on these can be found in Chiang et al. (2007). All these industries possess one common feature that led to their rapid growth: they maintain a pre-booking system where customers can book their seats, rooms etc prior to their arrival. This enabled revenue managers to use advanced demand information (ADI) (see Tan et al., 2009; Benjaafar et al., 2011, for a discussion on

*Email: `a.papayiannis@gmail.com`
†Corresponding author. Email: `paul.johnson-2@manchester.ac.uk`
‡Email: `d.yumashev@lancaster.ac.uk`
§Email: `peter.duck@manchester.ac.uk`

© The authors 2008. Published by Oxford University Press on behalf of the Institute of Mathematics and its Applications. All rights reserved.

imperfect ADI) to predict future demand and make proactive informed decisions on pricing or booking control.

Airport carparks cater for customers planning to fly and looking for a place to park their car until they return. Over the last decade, the implementation of pre-booking systems for airport carparks has spread rapidly around the globe and pre-booking of parking space is now available in all major airports in the UK and Europe through their official websites as well as through third parties. Recent reports reveal that airport carparking revenues add up to around 15-22% of the overall turnover within an airport organisation, with a trend towards online pre-booking currently at 70% (Steer, 2013a,b) and steadily increasing. As a result, this has created a whole new market which in turn has generated an ever growing interest from consulting firms to provide RM solutions. During the last couple of years, several companies have already seen the potential benefit and are offering RM systems tailored for airport carparks across Europe, Australia and the US (eg. IDeaS, Kowee).

While parking lot operations share common traits with other prominent RM problems, the academic literature has paid surprisingly little attention to this industry. However, noting that big international airports in the UK carry tens of millions of passengers per year with an average parking revenue estimated between £1-2 per passenger (Steer, 2013a), we believe that RM techniques in airport carparks may prove to be highly beneficial.

One may notice that among the range of industries for which RM is applied to, airport carparks are most closely related to hotels. In particular, the main product that is sold in both cases is “space” in the facility (room in the hotel, parking spot in the carpark). Therefore, the major challenges are similar, as the manager needs to address the *network* problem arising from *multi-day* stays Bitran and Mondschein (1995); Liu et al. (2006, 2008). Consequently, most RM practices in carparks have been simple adjustments of those developed for hotels. However, in what follows, a number of distinct characteristics render airport carparks a unique setting for RM. Commonly, no customer makes a decision on the length of their holiday or business trip based on the cost of parking. In most cases, a decision on carparking is made after the arrival and departure date is already set, so the majority of customers do not experiment with arrival or departure dates to compare prices; rather customers make the decision based on a comparison between other parking options (with same LoS) and alternative travel (e.g. bus, train, taxi). The alternative travel options play an important role because the price for them is independent of the LoS. Furthermore, both the start time and the end time of each booking is completely free to be chosen by the customer (in hotels there are usually standard check in and check out times), which means that there can be very short bookings (20 minutes) and very long ones (30 days) overlapping with each other. For the reasons outlined above we believe that an operator would not be able to successfully sell products by offering different prices on different days, a practice commonly seen in hotels. Hence, the LoS pricing results in the need for a different approach to be developed, whereby the price stays fixed and we must optimize the revenue by controlling the bookings according to the LoS in a more continuous setting.

The complexity of the network problem in RM comes down to the ability to estimate the corresponding *value function*. This is an enormous task and even when optimal solutions exist, these are unrealistically time consuming, if not impossible to solve. Therefore, the only realistic option in practice is to rely on approximation methods. The two main approaches of solving network problems is either by mathematical programming (MP), or by dynamic programming (DP). In the hotel context, the network RM problem is usually formulated as a mathematical program, which preserves its network structure and intra-day dependence. At its simplest the stochastic demand for each product is replaced by its mean and the resulting deterministic linear program (widely known as DLP) is solved to give optimal partitioned allocations for each product. According to Weatherford (1995), when DLP is frequently resolved it can generate good control policies. The biggest limitation of DLP is its deterministic nature as it ignores all demand uncertainty and only considers the expected demand. Stochastic mathematical programs have been proposed (see Goldman et al., 2002; De Boer et al., 2002) but the extra complexity as well as the resulting large number of variables and constraints renders these less appealing in practice. Finally, improved mathematical programs have also been studied by Chen and Homem-de Mello (2010), under the choice-based setting.

Dynamic programming can be employed to solve the network problem instead; however the state space formation suffers from Bellman's curse of dimensionality. The common approach in practice is to use some decomposition technique to reduce the problem down to a set of single-resource dynamic programs, one for each day (resource), and approximate the value function of the network by the sum of these single-resource value functions (Cooper and Homem-de-Mello, 2007; Bertsimas and De Boer, 2005). A traditional way to achieve this incorporates the static dual values from a DLP to adjust the fare contribution of the products in each of the resources these products require; this is either done additively or multiplicatively. This methodology has first been introduced by Liu and Van Ryzin (2008) in the context of the choice-based RM and further extended by Zhang (2011). Another way to break up the inter-dependencies between the resources is by a Lagrangian relaxation across the capacity constraints of the mathematical program (Jiang, 2008; Topaloglu, 2009). Later variants of these approaches aim to improve the proration schemes by updating the fare allocations iteratively (Kunnumkal and Topaloglu, 2010).

The main limitation of the aforementioned decomposition approaches is that once the initial network adjustments are made, all resulting single-resource programs are solved in parallel with no further information being exchanged between them. To overcome this Zhang (2011) proposed an improved methodology. Given that under the traditional DLP-based decomposition approach we can obtain an upper bound to the value function from each of the single-resource programs at every time-step (see Zhang and Adelman, 2009, for details), he proceeds as before but uses the tightest bound (out of all resources) as the value function approximation at each time step. In a further paper, Kemmer et al. (2012) propose an iterative algorithm that solves the dynamic programs simultaneously, while at each time-step information on the marginal capacity values is exchanged in order to update the proration fares. Their computational results suggest that the new algorithm performs very similarly to the traditional iterative DP decomposition scheme, but it achieves this at a fraction of the computation cost.

The concept of dynamic time-dependent bid-prices is fairly new and is based on the ability to write the dynamic program as a linear program and then make an affine functional approximation to the value function to obtain prices that allow the marginal value to change with time; the methodology has been introduced by Adelman (2007), extended by Zhang and Adelman (2009) and Meissner and Strauss (2012a,b) to the customer-choice setting, and studied by Kirshner and Nediak (2015) under a continuous-time framework.

All of the above studies focus on how to deal with the network problem case and, in particular, how to decompose the network down to a set of single-resource programs, exchange information between them, solve them and then build the network value function up again, with promising results and also motivation for further research. However, these single-resource dynamic programs are always implemented under the traditional dynamic programming formulation (employed in either the standard or the choice-based setting) where each product i arrives with intensity λ_i and corresponds to the adjusted fare f_i.

This paper proposes a new formulation for the resulting single-resource programs, which leads to a Hamilton-Jacobi-Bellman (HJB) equation that is optimised by the LoS variable. In contrast to the traditional revenue management models where the formulation is in discrete-time, our methodology implements a continuous-time setting which seems more appropriate for airport carparks. In major airport carparks, the LoS of a customer can be anything between 20 minutes and 30 days, with the sophistication in the payment methods leading to a potential for pricing car park stays by the minute. Therefore, traditional techniques such as the DLP are difficult to formulate let alone to be solved, as the number of product combinations in the incidence matrix grows exponentially large. For instance, when time is viewed in 20-minute slots, there are $3 \times 24 = 72$ time periods in each day and for a 10-day network the total number of products (arrival-day/LoS combinations) grows from $10 \times 30 = 300$ in the hotel case to $(10 \times 72) \times (30 \times 72) = 1555200$! Therefore, in airports experiencing high booking volumes and short LoS durations, the continuous-time model becomes a reasonable approximation. From an operational point of view, continuous time allows a better utilisation of space since more cars can be accommodated in the same parking spot each day by reducing the minimum occupancy time buffer which is implemented to avoid overlaps between the arriving and departing cars within a given time slot. As in any realistic application reducing the calculation time is important, the continuity and smoothness of the underlying functions make it possible to speed up the computation times and enable the solution to be calculated/updated in real time.

Our model follows that in Papayiannis et al. (2012) and Papayiannis (2013) and assumes that demand follows a Poisson process with time-varying intensity, with exponential distributions to model the lead-time and LoS. Based on these distributions we simulate booking requests which we then use in a formulated *test framework*, that is our model representation of a real reservation system, to model the customer flow and the resulting occupancy in the carpark.

We present novel approaches to generate the bid-prices, as functions of capacity and time left, and construct the resulting bid-price tables that maximise the carpark revenue. The bid prices are generated in three distinct optimisation problem settings, each with a different set of assumptions and technical challenges.

We subsequently examine the three optimisation problem settings under a time invariant framework, a natural starting point that enables tractability of the solution while preserving its key elements. In each setting, our model yields an estimate of the value function that is then used to approximate the displacement cost of capacity and ultimately build up the bid-price table. The resulting bid-prices are examined in a special test framework to (a) assess the improvement in performance compared to a *first-come-first served* policy (FCFS) and (b) establish whether the two single-resource models could be used instead of the a more generic multi-resource model to speed up the computations.

The paper is structured as follows. Section 2 introduces our unique pricing formulation for airport carparks and describes the test framework. Sections 3, 4 and 5 present the three optimisation problem settings, namely the SMR, SSR and DSR methods, respectively. Numerical results are presented and analysed in section 6 and our conclusions are found in section 7.

2. Framework

2.1 *List of Notations*

Here we summarise the key notations used in the paper:

- j or t denotes the current time in a discrete or continuous setting, respectively;
- k or T denotes the future time of interest in a discrete or continuous setting, respectively;
- η is the *lead* time between booking and arrival, and ξ is the length of stay (LoS) between arrival and departure, respectively;

- $\Psi(\xi)$ is the price rate for a particular LoS;
- $\phi(\eta,\xi)$ is the probability density function which describes how far in advance the booking is likely to have been made and how long we expect a customer to stay;
- $Q_{j,k}$ or $Q(t,T)$ denotes the number of customers present in the carpark in a discrete or continuous setting;
- $x_{j,k}$ or $x(t,T)$ denotes the number of spaces remaining at the present time for the period of interest;
- $V_{x,j,k}$ or $V(x,t,T)$ denotes the future expected revenue generated during the period k in the discrete setting, or the future expected revenue generated per unit time at the moment T in the continuous setting;
- $\pi_{x,j,k}$ is the bid-price table used to control capacity in the model.

2.2 *Pricing function*

We assume the existence of a price rate function that describes the relationship between the price-rate-per-day and the LoS, denoted by ξ. The function follows an exponential form and is monotonically decreasing in ξ, namely

$$\Psi(\xi) = \psi_\infty + (\psi_0 - \psi_\infty)\, e^{-\mu\xi}, \tag{1}$$

where ψ_∞ is the long term price rate that is achieved in the limit, ψ_0 is the highest price rate associated with the theoretical stay of 0 days and $\mu > 0$ is the decreasing rate which controls the shape of the curve between these two extremes. Thus, for a customer staying for ξ days, the *total* price to charge is $\xi\,\Psi(\xi)$. The exact form of the price function has been in accordance to that being implemented by carpark managers of major UK airports. Usually, parameters ψ_0, ψ_∞ and μ are set manually at the beginning of the year, based on the seasonal cycles, the price elasticity of the customers, the taxi market and offsite competitors. Common values for ψ_0 range between £10-35 while ψ_∞ lies in the range of £2-6. Once set, these parameters are kept fixed within the year and reviewed once at the beginning of the next calendar year. We also note that often the price rate function in (1) is used to describe the *additional* price rate on top of a *base* price rate, as to link the quoted prices across the available carpark alternatives and seasons.

2.3 *Booking process*

We assume that bookings can be described by three time variables; these relate to the *arrival time* in the carpark, the *lead time*, i.e. the time span between pre-booking and arrival, and the *length-of-stay*.

In particular, under a general setting we assume that the probability of a customer *arriving* at time t follows an nonhomogeneous Poisson process with time-varying intensity $\lambda_a(t)$, so that the total number of customers $X(t)$ that have arrived at the carpark by time t is

$$X(t) = \int_0^t dX(t_a) \tag{2}$$

and the expected number is

$$\mathrm{E}[X(t)] = \int_0^t \lambda(t_a)dt_a. \tag{3}$$

Let us now focus on a particular time t. For each arriving customer at time t there is an associated lead time η and a required length-of-stay ξ at the carpark which will ultimately lead into the actual occupancy in the carpark, since not all of the customers who have arrived will still be present at time t. So we assume that there is a probability density function $\phi(\eta,\xi;t)$ which describes the distributions of customer requirements which is independent of the arrival process. The quantity $\phi(\eta,\xi;t)\,d\eta\,d\xi$ indicates the probability customers who arrive at t, to have booked between η and $\eta+d\eta$ days before and to stay between ξ and $\xi+d\xi$ days, and this function can easily fitted to booking data. We have noted that in the real world data the distribution of lead times η is best modelled with a gamma distribution, whilst the LoS distribution can only be captured with an empirical fit. However for this present paper we choose the distributions of η and ξ to be a linear combination of independent Exponential distributions, namely

$$\eta(t) \sim Exp\,(\lambda_b(t)) \tag{4}$$

$$\xi(t) \sim Exp\,(\lambda_s(t)) \tag{5}$$

where the parameters $\lambda_b(t)$ and $\lambda_s(t)$ are possibly time-varying functions.

2.4 *Booking classes*

In our study, we adopt a time-invariant framework, a natural starting point that enables tractability of the solution while preserving its key elements. Under this framework, all arrival days are assumed to follow the same distribution; thus, the explicit reference to the arrival time t can be removed, and as a result the time parameters in equations (3), (4), (5) become fixed constants, i.e. $\lambda_a, \lambda_b, \lambda_s$, while the joint density function is simplified to $\phi(\eta,\xi)$ for all arrival times t.

The definition of the joint density function $\phi(\eta,\xi)$ can easily be extended to accommodate N customer classes. In particular, we assume two customer classes ($N=2$), the *Business* and the *Leisure* customers. By definition, these two sets of customers express different types of customer behaviour and thus they are described by different average arrival intensities λ_a, different lead times, λ_b and different length-of-stays, λ_s. Therefore, the n^{th} customer class could be expressed as

$$\mathscr{B}_n \sim \begin{pmatrix} \lambda_{b_n} \\ \lambda_{a_n} \\ \lambda_{s_n} \end{pmatrix}. \tag{6}$$

Note that there are three sources of uncertainty as the number of arrivals per day from the n^{th} class, the lead-times and length-of-stay are all stochastic variables. Using equation (6) we can define the total booking profile as a combination of the single booking classes, namely

$$\mathscr{B} = \sum_n \mathscr{B}_n. \tag{7}$$

Consequently, each class n has its own distribution $\phi_n(\cdot)$ and thus the total distribution $\phi(\cdot)$ for all N classes combined is given by the weighted linear combination

$$\phi(\eta,\xi) = \sum_n \alpha_n\,\phi_n(\eta,\xi), \quad \alpha_n = \frac{\lambda_{a_n}}{\Lambda_a} \quad \forall n \in 1,\ldots,N, \tag{8}$$

with the total arrival intensity defined as

$$\Lambda_a = \sum_n \lambda_{a_n}. \tag{9}$$

The formulation of $\phi(\cdot)$ in (8) allows any type of distributions to be used. In some cases, the type of distributions used can lead to interesting analytical results. In particular, under Exponential lead times and length-of-stays, we have

$$\phi(\eta,\xi) = \sum_n \alpha_n \, \lambda_{b_n} \, \lambda_{s_n} \, e^{-\lambda_{b_n} \eta} \, e^{-\lambda_{s_n} \xi} \tag{10}$$

and in Appendix B we derive the corresponding analytic formulae.

Under this setting we price a single product (i.e. a single pricing function $\Psi(\cdot)$) that is offered simultaneously to both classes, irrespective of their class characteristics. Thus, a unified treatment of the combined booking flow $\mathscr{B}$ can be achieved and as long as the capacity restrictions are met, this product remains open to both classes.

We note that we are not concerned with deriving the optimal pricing-function parameters $(\psi_0, \psi_\infty, \mu)$. Instead, we consider that these have been derived or pre-set by the management and reflect the current market demand to the best possible degree. However, it is important to understand that although the pricing function is fixed, the price-rate-per-day charged *does* vary according to the customer's required duration of stay.

2.5 *Carpark setting*

Now assume that we are on day 0 with an empty carpark that is about to begin its operations. Our goal is to maximise the expected revenues on a future day T. We split the continuous time line into K fixed interval time periods of size Δt. The k^{th} period is defined as $[t_k, t_{k+1})$, where

$$t_k = (k-1)\,\Delta t \quad \forall\, k = 1, 2, ..., K.$$

We assume also that each booking request i has an associated booking time t_b^i, arrival time t_a^i and departure time t_d^i. Therefore, to indicate whether a customer requests to stay in period k we introduce the following binary variable

$$\delta_k^i = \begin{cases} 1 & \text{if} \quad t_k \leqslant t_a^i < t_{k+1} \quad \text{OR} \quad t_k < t_d^i \leqslant t_{k+1} \quad \text{OR} \quad (t_a^i < t_k \text{ and } t_d^i > t_{k+1}) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

This variable is set to 1 only if the customer is requesting to be present over period k, and 0 otherwise. Next, for each request we make a decision as to whether we should admit or deny entry in the carpark. Any decision made, affects the entire booking in the sense that the customer is admitted entry *to either all or none* of the days requested. Therefore, for each booking request i there is an associated binary control variable u^i that indicates just this;

$$u^i = \begin{cases} 1 & \text{if booking } i \text{ is accepted} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Now, we can define the number of cars present in period k as of period j, $j \leqslant k$ by

$$Q_{j,k} = \sum_{\substack{i \\ t_b^i < t_{j+1}}} u^i \, \delta_k^i, \quad \text{s.t} \quad Q_{j,k} \leqslant C, \tag{13}$$

where C denotes the carpark size. Note that in this definition, we account only for bookings that have been made *before* the end of period j.

Next we consider the number of spaces remaining in period k as of period j, according to (13) this is denoted by

$$x_{j,k} = C - Q_{j,k}. \tag{14}$$

Therefore, the *revenue remaining to be generated* for a carpark in period k and initial capacity C, $\boldsymbol{x} = (x_{j,1}, x_{j,2}, \ldots, x_{j,K})$ spaces remaining as of period j, under policy u, $u \in U$ is given by

$$J^u_{\boldsymbol{x},j,k,C} = \sum_{\substack{i \\ t^i_b \geqslant t_{j+1}}} u^i \, \delta^i_k \, \Psi(\xi^i) \, \Delta t, \tag{15}$$

subject to capacity constraint

$$\sum_{\substack{i \\ t^i_b \geqslant t_{j+1}}} u^i \, \delta^i_k \leqslant x_{j,k} \qquad \text{for } k \geqslant j. \tag{16}$$

Note that in the above definition, we account only for bookings that have been made *after* period j, and that ξ^i is still perceived as the length-of-stay in days for the i^{th} request[1].

FULL NETWORK PROBLEM Assume that the *expected revenue remaining* V in period k for a carpark of initial capacity C, with $\boldsymbol{x}$ spaces remaining as of period j, can be expressed by

$$V^u_{\boldsymbol{x},j,k,C} = \mathrm{E}\big[J^u_{\boldsymbol{x},j,k,C}\big] \tag{17}$$

where $u \in U$ is an admission control policy chosen from the set of all admissible policies. The problem of the carpark manager, is that they must *jointly* manage the available resources in each and every period and optimise revenues over the full network of days with a single admission control policy. The total expected future network revenue (NR) over all K periods at time j may be calculated in an additive manner, so the full network problem can be written as

$$NR_{\boldsymbol{x},j} = \max_{u \in U} \left[\sum_{k=1}^{K} V^u_{\boldsymbol{x},j,k,C} \right] \tag{18}$$

subject to

$$x_{j,k} \geqslant 0.$$

Under the given setting, customers can book for any duration of stay, from as little as one period of time Δt, up to 30 days. Thus, if for example, the minimum stay is set to 20-minutes, ($\Delta t = 1/72$), then a 10-day network consists of $K = 10/\Delta t = 720$ time periods with $(10 \times 72) \times (30 \times 72) = 1555200$ distinct products (arrival-day/LoS combinations). Since users of airport carparks may book up to 18 months in advance this is obviously going to be a problem.

Given the large state space of the full network problem, an exact network solution is not available. Rather, this paper is primarily focused on the decomposed single-resource problem and the associated bid-prices. In particular, three distinct optimisation models are studied, each with a different set of assumptions and technical challenges, and all lead to bid-prices which are defined as functions of capacity and time left. The objective of our study is ultimately to optimize the expected revenue rate that is related to a given future day. More precisely we have,

[1] To calculate the required length-of-stay in days of the i^{th} request, ξ^i, we need to follow a two-step process. We first derive the length-of-stay in terms of Δt-periods, D^i, as $D^i = \lfloor t^i_d/\Delta t \rfloor - \lfloor t^i_a/\Delta t \rfloor + 1$, and then calculate the required quantity by $\xi^i = D^i \Delta t$.

1. Stochastic Multi Resource (SMR)
   This is the simplest model intuitively, as it naturally arises from the formulation of the test framework which is described in section 2. This method uses Monte Carlo to simulate bookings; it begins with a zero bid-price table and iteratively refines the estimates until convergence is achieved. The major advantage of this method (which rendered its name) is that it treats the network as a multi-resource problem and proceeds in deriving bid prices that better reflect the true network structure of the problem. On the downside, however, are the high computation times required for the algorithm to converge.

2. Stochastic Single Resource (SSR)
   The network problem can be simplified when considered as a set of single-resource problems. In other words, we may treat each resource (day) separately and derive a set of independent single-resource dynamic programs, one for each day in the network. Under a continuous-time formulation, the single resource refers to an infinitesimal instant of time and thus it uses the price *rate* per day, as opposed to the total price of the booking. The problem still remains stochastic (Poisson demand) but it leads to a partial differential equation (PDE) to generate deterministic booking policies for which appropriate schemes can be solved relatively quickly.

3. Deterministic Single Resource (DSR)
   This is again a continuous-time model where the interdependence within days is also ignored. By relaxing the assumption of Poisson demand we can treat the parking spaces as *continuous* quantities, which results in a “fluid” approximation formulation with deterministic and continuous dynamics. This approach is developed to examine how well a deterministic model can perform in a stochastic environment. Pontryagin’s maximum principle is used to obtain the optimal solution (Bertsekas, 1995).

The booking RM system goes through every booking request and decides whether the current request should be admitted/denied entry. If the request is admitted into the carpark, then the system allocates a parking space on *all* periods required by that request. We can then assume booking control is driven by an underlying table of *bid-prices*,

$$\pi_{x,j,k}$$

quoted as a function of capacity remaining for period k, denoted by x, the time j, and the period of interest k. Therefore, the i^{th} booking request made in period j to arrive in period l and stay for D^i periods (or $\xi^i = D^i \Delta t$ days), is accepted only if there are spaces available *and* its total price is greater than the sum of the bid-prices over the periods required, namely

$$\xi^i \Psi(\xi^i) \geqslant \sum_{k=l}^{l+D^i-1} \pi_{x,j,k}. \tag{19}$$

Note that in the case of a *zero* bid-price table, a special case of control arises, the FCFS policy, as booking requests are only restricted based on capacity availability.

TIME INVARIANT BID-PRICES In the sections above we introduced the time-dependent framework for the sake of generality. We focus on the time-invariant framework for the remainder of the paper in order to obtain clear tractable steady-state solutions of the optimisation problem. Under a time invariant framework, the explicit reference to the actual time is removed, with only the time lag between booking and being present matters, $m = k - j$, in which case the *value function*, the *maximum expected revenue remaining* of the carpark with x spaces remaining and m periods left, is denoted by $v_{x,m}$. All methods that we consider in this study calculate the *expected marginal values* $\Delta v_{x,m} = v_{x,m} - v_{x-1,m}$ of the spaces as functions of the periods left m and capacity remaining x. The bid-price control is then generated by setting the bid-price of each resource equal to its corresponding expected marginal value, namely

$$\pi_{x,j,k} = \Delta v_{x,k-j} \quad \forall x, j, k. \tag{20}$$

Thus, the resulting set of bid-prices defines the *bid-price table*, which is also referred to as the *admission control policy* in this paper.

SIMULATION ENVIRONMENT Our simulation experiment proceeds as follows. We implement the admission control policy at time 0 when the carpark is entirely empty. We then simulate a number of booking reservations sets according to the distributions in (2), (4) and (5). For each reservation set we proceed by looping through its bookings one at a time and admit/deny entry to the carpark based on the rule in equation (19). The procedure estimates the *expected total revenue* to be generated in a given period k, denoted as

$$R_k = \mathrm{E}\big[J_{\boldsymbol{C},0,k,C}\big] = \mathrm{E}\left[\sum_i u^i\, \delta_k^i\, \Psi(\xi^i)\, \Delta t\right]. \tag{21}$$

Note that in this equation all possible bookings are accounted for, and we start with an empty carpark $\boldsymbol{x} = \boldsymbol{C} = \{C, C, \ldots, C\}$. To eliminate all initialization effects, we let this run for a sufficient number of periods out in the future and build up accordingly. Then, we summarise our results by averaging over an interval of 20 days (or more precisely $20/\Delta t$ consecutive periods) in order to obtain a *perpetual* quantity for the *maximum expected revenue per period,* R_∞, and for the *maximum expected revenue rate per day*, $r_\infty = R_\infty/\Delta t$. Finally, the algorithm is applied to a range of carpark sizes C.

3. Stochastic Multi Resource (SMR) Method

Motivated by the previous section we can derive an intuitive approach to calculate the underlying set of bid-prices. Let j be the current period, and we wish to estimate the *maximum* expected revenue remaining V to be generated in the future period k when there are $\mathbf{x}$ spaces remaining in a carpark of capacity C. This quantity is given by

$$V_{\boldsymbol{x},j,k,C} = \mathrm{E}\big[J^{u^*}_{\boldsymbol{x},j,k,C}\big] \tag{22}$$

where u^* is the *optimal* admission control policy, which means that the actual value here will depend on the effect the policy has on days other than k and the spaces remaining on those days as well (the full network problem).

We propose to approximate V in the following way, using a particular policy u we assume that

$$V_{\boldsymbol{x},j,k,C} \approx V^u_{x,j,k} \tag{23}$$

where $V^u_{x,j,k}$ is the expected value at period k given j is the current period, the carpark is in an empty state for all $k \geqslant j$ and the capacity of the carpark is $C = x_{j,k} = x$. Given that we can simulate values of J from this initial state, we can calculate V as

$$V^u_{x,j,k} = \mathrm{E}\big[J^u_{\boldsymbol{x},j,k,x} \mid \boldsymbol{x} = \{x, x, \ldots, x\}\big] \tag{24}$$

for a given control policy u. From a single simulation of the booking set we can calculate $J^u_{\boldsymbol{x},j,k,x}$ for all values of x and $k \geqslant j$ given a fixed j. By choosing a different value of j and returning the carpark to an empty state, we can then calculate $V^u_{x,j,k}$ for all values of x, j and k to build up the bid price table $\pi_{x,j,k}$. Using an iterative scheme we would successively update both V and the bid price table π in turn to maximise the expected revenues. Using an empty carpark to calculate the expected values appears problematic at first but it greatly simplifies the problem and allows for the results to be reused in different scenarios. We found that the values of R_k in (21) do not change greatly for large k, meaning that using the policy starting from a genuine empty carpark is not too different to when we apply it over longer periods, so this indicates that the proposed method captures the primary network effects.

Under the time-invariant framework, the problem becomes much simpler as the expected revenue remaining of a carpark of $x_{j,k}$ spaces remaining in period k as observed in period j, $j \leqslant k$ is equivalent to

$$V^u_{x,j,k} = V^u_{x,0,m}, \tag{25}$$

where $m = k - j$ is the time lag in periods between booking and being present. This means that to find the expected revenue remaining of a carpark of $x_{j,k}$ spaces remaining in period k as observed in period j, we can simply evaluate an initial empty carpark, on the $m^{th} = k - j$ period with size $C = x$ and we can write

$$\pi_{x,m} = V_{x,0,m} - V_{x-1,0,m}.$$

A full description of our implementation of the method is given in Appendix A. Results are presented later in section 6 and compared with the alternative methodologies introduced in the next two sections.

4. Stochastic Single Resource (SSR) Approach

Previously, we described the simulation process for a set of reservations that was characterised by the intensity parameters denoted λ_b, λ_a and λ_s, using Exponentially distributed lead times and length-of-stays. Over the current and following section we develop two models formulated in continuous time. Each of these models investigate the single resource approximation, in which we try to calculate the value of the car park over a small time interval between T and $T + dT$, and in the continuous setting this is infinitely small. This can only be done by ignoring the capacity at all other times, hence the single resource approximation.

Instead of simulating the underlying probability distributions numerically, we now build our model based on the relevant analytical expressions. Let us now evaluate the *occupancy* over the instant T generated from bookings up to time t, $Q(t,T)$. If the stochastic process $dX(t_a)$ from (2) captures a customer arriving at t_a, then by selecting only those customers that will be present at T we arrive at the process for occupancy at T,

$$Q(t,T) = \int_{t_a=-\infty}^{T} 1_{\{t_b \leqslant t \text{ and } t_d \geqslant T\}}\, dX(t_a). \tag{26}$$

Equation (26) involves all bookings that have either already arrived (i.e. $t_a \in (-\infty, t]$) or that are about to arrive sometime in the future (i.e. $t_a \in (t, T]$) and sums up only those that booked before t and stay long enough to be present over the time instant T. The corresponding intensity of this process is therefore given by

$$q(t,T) = \Lambda_a \int_{\xi=0}^{\infty} \int_{\eta=(T-t-\xi)^+}^{T-t} \phi(\eta,\xi)\, d\eta\, d\xi \tag{27}$$

assuming that the time-invariant booking intensities are given by (8) and (9).

Now as a single resource approximation we are only interested in the revenue generated over the instant T, which is given by the price rate $\Psi(\xi)$, rather than the total revenue generated for the whole time the booking is present, given by $\Psi(\xi)\xi$. Now consider that each customer will pay a different price rate, then the revenue generated is calculated by integrating the probability density of different LoS multiplied by the price rate they each would pay. The resulting revenue rate is given by

$$r(t,T) = \Lambda_a \int_{\xi=0}^{\infty} \int_{\eta=(T-t-\xi)^+}^{T-t} \phi(\eta,\xi)\, \Psi(\xi)\, d\eta\, d\xi. \tag{28}$$

It is important to note that equation (28) is a *double* rate in the sense that it relates to the contribution to the revenue at the instant T (price rate) multiplied by rate of bookings made at the instant t.

Next, define $x(t,T)$ to be the number of parking spaces remaining for the time instant T as of time t. Hence $V(x,t,T)$ is defined as the value rate function at T with x spaces remaining as of time t. To recover the total revenue in the carpark one would need to integrate V w.r.t T. More precisely, x is modelled as a jump process with jump size -1 (corresponding to one sale), with $x(t=-\infty;T)=C$ the initial carpark size. In particular, over the next time interval dt, we sell one space with probability $q(t,T)\,dt + o(dt)$, we do not sell a space with probability $1-q(t,T)\,dt - o(dt)$ and we sell more than one space with probability $o(dt)$. Omitting the terms of order less than dt we may write the change in x as

$$dx = \begin{cases} 0 & \text{with probability } 1-q(t,T)\,dt \\ -1 & \text{with probability } q(t,T)\,dt. \end{cases} \tag{29}$$

Then dV is given by,

$$dV(x,t,T) = \frac{\partial V}{\partial t} dt + q(t,T)\big[V(x-1,t,T) - V(x,t,T)\big]\,dt. \tag{30}$$

Equating equation (30) to the revenue rate in (28) and dividing by dt we obtain

$$\frac{\partial V(x,t,T)}{\partial t} + q(t,T)\big[V(x-1,t,T) - V(x,t,T)\big] = -r(t,T), \tag{31}$$

with $V(x,t=T,T)=0$ and $V(x=0,t,T)=0$, for all choices x and t, respectively.

The equation in (31) assumes that the value function at T only depends on the current time t and the number of spaces remaining for T, $x(t,T)$, is completely independent of the carpark state on the nearby days. This is a rather strong assumption as one would normally assume that network effects from the nearby days are present; if the neighbouring days are sold out then any customers for day T that plan to park for more than a day, will be denied entry. However, the interdependence among neighbouring days is less apparent when most bookings are for a single-day stay, in which case our independent-day assumption is reasonable (see Ladany, 1976; Bitran and Gilbert, 1996). Note the minus sign on the RHS of equation (31) indicates the fact that the remaining value decreases at the *opposite* rate at which revenue is generated. This describes the intuitive fact that as time progresses the value remaining for the carpark on day T reduces. However, the value that is lost when going from t to $t+dt$ is "replaced" by the revenue which is generated from customers who book within this period. Therefore, the rate at which the *value* reduces is exactly inversely proportional to the rate at which the *revenue* is generated as time passes.

Next, by setting $\tau = T - t$ the value function of x spaces remaining and time τ to go, $V(x,\tau)$, is the solution to

$$\frac{\partial V(x,\tau)}{\partial \tau} + q(\tau)\big[V(x,\tau) - V(x-1,\tau)\big] = r(\tau), \tag{32}$$

with $V(x,\tau=0)=0$ and $V(x=0,\tau)=0$, for all choices of x and τ, respectively. Note that this PDE is forward looking in τ, which is the reason for the imposed initial condition at $\tau=0$, as opposed to the final condition for (31).

An admission policy should ensure that in order to maximise the expected value rate we accept a customer only if its corresponding price rate Ψ is higher than or equal to some optimal minimum price rate Ψ^* set by the carpark manager. Since there is a one-to-one correspondence between the price $\Psi(\cdot)$ and the length-of-stay ξ, with the price rate monotonically decreasing in ξ, we can apply the admission policy with regards to the latter. In other words, we accept a customer only if its duration of stay ξ is *less than or equal to* some optimal maximum duration of stay ξ^*.

In fact, the intensity of customers that are present τ days later and stay for no more than ξ^* days is given precisely by

$$q(\tau|\xi^*) = \Lambda_a \int_{\xi=0}^{\xi^*} \int_{\eta=(\tau-\xi)^+}^{\tau} \phi(\eta,\xi)\, d\eta\, d\xi. \tag{33}$$

These customers will bring the revenue rate of

$$r(\tau|\xi^*) = \Lambda_a \int_{\xi=0}^{\xi^*} \int_{\eta=(\tau-\xi)^+}^{\tau} \phi(\eta,\xi)\,\Psi(\xi)\, d\eta\, d\xi. \tag{34}$$

In other words, out of the set of customers that would have arrived to be present over T, the optimal admission rule selects only those customers that will be staying at most ξ^* days, or better those that contribute at least $\Psi(\xi^*)$. Note the notation using the $|$ symbol to indicate that $q(\cdot)$ and $r(\cdot)$ now are evaluated for a given upper limit value on ξ.

Therefore, the SSR model, expressed in a Hamilton-Jacobi-Bellman (HJB) form, reads

$$\frac{\partial V(x,\tau)}{\partial \tau} = \max_{\xi} \Big\{ r(\tau|\xi) - q(\tau|\xi)\big[V(x,\tau) - V(x-1,\tau)\big] \Big\}. \tag{35}$$

The SSR model appears to be closer to a dynamic pricing model seen in the literature (eg. Gallego and Van Ryzin, 1994; Zhao and Zheng, 2000) because the price varies continuously as a result of changing the maximum allowed stay ξ. However, we notice that the demand is not directly affected by the price but it rather gets *truncated* so that only customers who pay more than a minimum amount are accepted. This might then indicate some form of dynamic capacity control as the entire demand flow is still observed but we limit the available products (by restricting the length-of-stay) in such a way that only part of this demand gets through (eg. Zhao and Zheng, 2001).

It is important to note that, as opposed to the SMR method that calculates the expected revenue generated over a finite Δt period, the SSR model in equation (35) (and the DSR model which is presented in the next section) solves for the expected revenue *rate* per day that must then be multiplied by Δt, before treating the two quantities on the same scale to allow a valid comparison between them.

One of the big advantages we found using the continuous-time model was in the derivation of analytic results for the functions q and r. The justification is that approximating the value function in the limit with simple functions as input, was preferable to introducing more complex inputs to derive the value over a small time interval. In particular, further details on this matter can be found in Papayiannis et al. (2013), where the authors present a methodology to derive this value over a finite time interval. It should also be noted under this setting all small intervals have a similar revenue potential so the bid-price table can be reused over a long period once it has been solved. This is particularly useful if bookings within a season were to exhibit similar patterns.

4.1 *Structure of the optimal price*

The optimal price Ψ^* at which we accept customer requests can be found by differentiating the PDE (35) w.r.t ξ^*. Doing so, we conclude that

$$\Psi^* = \max\left\{V(x,\tau) - V(x-1,\tau),\ \psi_\infty\right\}, \tag{36}$$

indicating that at optimality the price-per-day equals the expected opportunity cost. By then combining (1) and (36) and solving w.r.t. ξ^* we obtain

$$\xi^* = \log\left(\frac{\Psi^* - \psi_\infty}{\psi_0 - \psi_\infty}\right)^{-1/\mu}. \tag{37}$$

4.2 *Numerical considerations*

In order to solve the PDE (35) we use a finite-difference scheme (see Smith, 1985). The main challenge we are faced with is the presence of multiple integrals that need to be calculated at every step on the mesh; this would escalate computation times. Thus, a pre-processing step is employed, where the values of $q(\tau|\xi)$ and $r(\tau|\xi)$ are precomputed for all combinations of τ and ξ, and stored in 2D matrices. Details on the computations of this matrix are presented in Appendix B.

4.3 *Bid Prices*

We construct the discrete bid prices from the continuous value rate function V as

$$\pi_{x,m} = V(x,\tau = m\Delta t)\Delta t - V(x-1,\tau = m\Delta t)\Delta t.$$

A full description of how the bid-price table is calculated and used is presented in Appendix C.

5. Deterministic Single Resource (DSR) Approach

We now consider the associated deterministic (fluid) formulation of the problem by assuming that the remaining capacity x is a continuous quantity. Then the problem can be expressed as

$$\max_{\xi} \int_0^T r(\tau|\xi)\, d\tau \tag{38}$$

$$\text{subject to} \quad \frac{dx}{d\tau} = q(\tau|\xi) \quad \text{and} \quad x(\tau = T) = C, \quad x(\tau) \geqslant 0 \tag{39}$$

namely, to maximise the revenue rate over the booking horizon subject to x changing deterministically and continuously according to the controlled intensity $q(\cdot)$.

In this representation, time is in reverse, meaning that T should be regarded as the length of the booking horizon. The idea here is that at the beginning of the booking horizon (T days before) no spaces have been sold yet which means that the number of spaces remaining at this point is the entire carpark capacity C i.e. $x(\tau = T) = C$. As we move towards the target day, spaces are getting sold to customers, effectively reducing the remaining spaces in the carpark. On the target day, we would have either sold the entire number of spaces or we would be left with a few unsold spaces ($x(\tau = 0) \geqslant 0$). The objective is to find the optimal selling rate of the spaces (optimal trajectory for x) so that the generated revenue is maximised. Consequently, to solve this model we apply the Pontryagins maximum principle (Bertsekas, 1995).

When the problem is deterministic we can show that the optimal price policy is a fixed-price policy; that is when the total demand to come is greater than the carpark capacity, we choose the run-out price, Ψ^0, at which we sell precisely the entire capacity, and when the total demand is less than the carpark capacity we enact the revenue maximising price, Ψ^*; this result is also verified in Gallego and Van Ryzin (1994).

5.1 *Bid Prices*

We construct the discrete bid prices using the optimal price rate which is calculated as part of the solution. Consider that the optimal price rate ψ^0 has a corresponding LoS ξ^0, we can write

$$\pi_{x,m} = \begin{cases} \Psi\left(\xi^0\right) & \text{such that } \int_0^{m\Delta\tau} q(s|\xi^0)ds = x \\ \Psi^* & \text{when total demand is less than capacity} \end{cases} .$$

A full description of how the run-out price is calculated and used to construct the bid-price table is presented in Appendix D.

6. Results

Informal discussions with various airports carpark managers in UK indicates that the size of Premium carparks ranges from 50 to 500 spaces and can sell out during peak times. In contrast, Long Stay carparks have huge capacity which rarely sells out. Thus in practice, a Premium carpark naturally leads to constrained optimization problem, where ‘denied’ customers are redirected to the unconstrained Long Stay carparks.

In our default scenario, we assume that we operate a Premium carpark with capacity up to $C = 100$ spaces, in order to ensure that results can be obtained and comparisons can be achieved across all three methods (SMR, SSR, DSR). Later, for the two methods, SSR and DSR, this restriction is relaxed and we consider a Premium carpark of capacity $C = 500$ and assess their performance.

The parameters for the Business and the Leisure class are given by

$$B \sim \begin{cases} \lambda_b = 1/3 \\ \lambda_a = 25 \\ \lambda_s = 1 \end{cases} \text{and} \quad L \sim \begin{cases} \lambda_b = 1/14 \\ \lambda_a = 5 \\ \lambda_s = 1/7 \end{cases} , \tag{40}$$

respectively. The parameters for the two sets are carefully selected in a manner that best reflects the likely customer behaviour within each segment; note the higher arrival rate of B which is 25 as opposed to only 5 for set L. Also note that Leisure customers tend on average to book around two weeks in advance ($\mathrm{E}[\eta_L] = 1/\lambda_b = 14$ days) and stay for around a week in the carpark ($\mathrm{E}[\xi_L] = 1/\lambda_s = 7$ days), whereas the Business customers book relatively close to the arrival day ($\mathrm{E}[\eta_B] = 3$ days) and stay on average for just a day ($\mathrm{E}[\xi_L] = 1$). This is a typical situation in the airport carpark where we encounter a high-percentage of shorter length-of-stay customers averaging one day per customer. We have of course ignored the "roll-up" customers (those who do not prebook in advance) which greatly simplifies our problem setting. These customers typically stay for a shorter amount of time (dropping off or picking up other customers) and are subject to a different and very high price-rate.

The *expected total demand* per day can be calculated by $\mathrm{E}[\textit{total demand}] = \sum_n \lambda_{a_n}/\lambda_{s_n}$, which in our model scenario is 60 customers per day. In other words, we would *expect* that carparks of size 60 and larger should be adequate in satisfying the customer demand.

The price-rate function parameters are set to $\psi_0 = 15$, $\psi_\infty = 5$ and $\mu = 1/5$, chosen specifically to replicate the behaviour of pricing functions that are commonly used in practise. Under this set of parameters, the quoted prices start from 15 units per day (when $\xi \to 0$), exponentially decrease in length-of-stay and asymptotically achieve the long term rate of 5 units per day (when $\xi \to \infty$).

Recall that the SMR method uses the parameter Δt to set the size of the smallest time period that can be monitored. The smaller Δt is, the more refined the price structure and the occupancy becomes. In fact, as $\Delta t \to 0$, the SMR solution converges to the solution given by the two continuous-time approaches and ultimately enables comparisons to be made and meaningful conclusions to be deduced. For this study, we have used $\Delta t = 6.25 \times 10^{-3}$ (9-minute intervals) for all three methods SMR, SSR, DSR, a parameter value which has been shown to be sufficient for numerical accuracy at a reasonable computational cost (Papayiannis, 2013).

Under the absence of an optimal solution, A FCFS policy is implemented to serve as a benchmark against our three developed methodologies. Under this strategy the only condition that is required for a request to be accepted is capacity availability. Recall that this simple strategy can also be seen as a special case of a bid-price policy where all the bid-prices in the table are set to zero.

6.1 *Inspection of Methods*

Figure 1 presents the expected revenues to be generated, from the FCFS (left) and SMR (right), as a function of the time left. We recall that for the SMR case, these expected revenues (of x spaces remaining on period k as of period j) are derived by setting an empty carpark and evaluate the revenue generated on the m^{th} period ($m = k - j$) using capacity $C = x$. Thus for the *SMR* method, the terms capacities C and remaining capacities x, are used interchangeably.

When demand is unconstrained ($\mathrm{E}[\textit{total demand}] < x$) both approaches seem to generate identical revenues; this is the expected outcome as all customers are allowed entry irrespective of their length-of-stay. In this case, the expected revenue is a concave function of the time-left. At the maximum lead time, the expected revenues rise up to 530 currency units. When expected demand is constrained ($\mathrm{E}[\textit{total demand}] > x$), the expected revenues under the FCFS strategy fail to be concave and are lower revenues than those of the SMR. This happens due to the poor management decisions taken in the booking process; when customers are served on a first-to-come basis, the system tends to accept too many leisure customers which results in displacing some lucrative business customers that would have arrived later in the booking horizon. In contrast, under the SMR policy the bid-price algorithm works well in controlling the leisure stream of customers and thus ensuring that spaces will remain available for the business ones who will arrive later in the booking process.

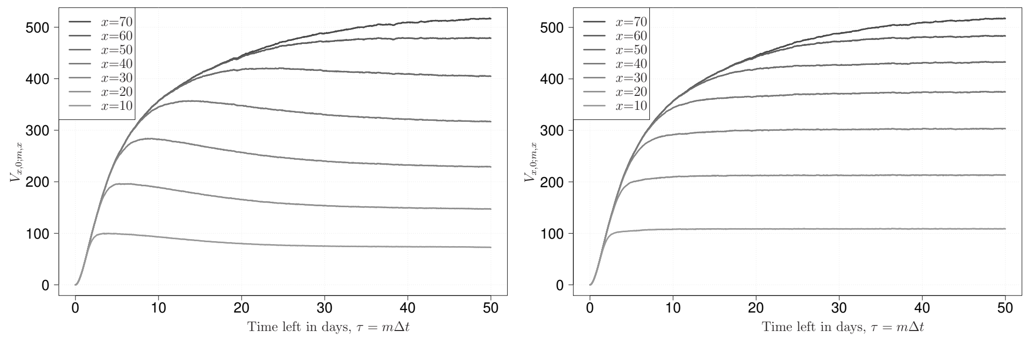

FIG. 1. Expected values as functions of the time left, for the FCFS (left) and SMR (right) policy. Time left is measured in periods of size Δt, though here we present it in terms of full days. The different curves show various carpark sizes, ranging from 70 to 10, top to bottom. The values in each period are divided by Δt to represent rates.

Figure 2 compares the solutions of the SSR and the DSR methodologies. Recall that the latter is the "fluid" formulation of the former and it is deterministic. In particular, we compare the expected revenues to be generated in the carpark when there are 100 days left until utilisation. The expected revenue under the DSR approach increases in x and it reaches its maximum at the point when the remaining capacity equals the expected total demand ($x = 60$). Beyond this size, no additional revenue can be achieved and hence the sharp horizontal line. This is to be contrasted to the SSR method where the expected revenue slowly reaches that maximum. Thus, we conclude that the DSR method serves as an upper bound to our SSR model, formalising the idea that uncertainty in sales results in lower expected revenues.

6.2 *Comparisons within the test framework*

6.2.1 *BidPrice Policies* Let us now examine the performance of the three derived methodologies, SMR, SSR and DSR when implemented within our test framework. Figure 3 presents the expected marginal values of the resource days, as these are calculated by the three developed methods plus that of the FCFS strategy. Note that under the assumption of time-invariance, these bid-price tables are identical for all days in the carpark; in the general non-stationary case there would be a distinct table for every single day (resource) in the carpark. The expected marginal values of the resource are presented as 2D tables w.r.t the capacity remaining and time left until utilisation. For the purpose of this study, each of these tables forms an admission policy, or equivalently a bid-price table which is implemented and tested within the test framework. Beginning from the FCFS policy (top left), the resources are associated with no value, in which case the admission algorithm accepts/denies requests based just on whether capacity is available. Looking at the SMR policy (top right) the expected marginal values show the anticipated behaviour as they increase in time-left and decrease in capacity remaining. Recalling that the expected total demand is 60 and that the problem is stochastic, the curves become zero on and around that capacity size. Around 15 iterations and 12000 paths were needed for the algorithm to converge to the solution. Even after many simulations the surfaces still show some variation and are not completely smooth. The SSR approach (bottom left) achieves a similar policy to that of the SMR approach. The results are not only much smoother here but the time taken to compute these has been limited to seconds as opposed to hours. Finally, for the DSR approach the deterministic nature of the model is reflected in the resulting policy, where there is a sharp cut-off in the valuations; when the demand is greater than

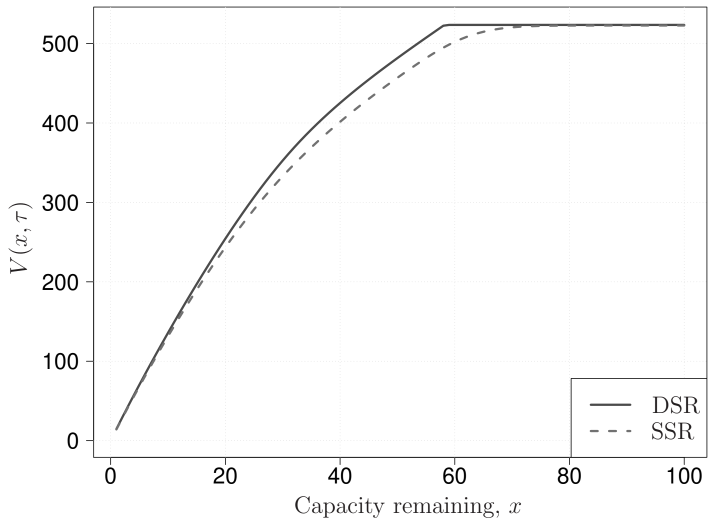

FIG. 2. Expected values as functions of the capacity remaining, x, with $\tau = 100$ days left, for the SSR and DSR models.

capacity these are positive and similar to that of the SSR model and when capacity becomes greater than demand they become zero.

A closer inspection of the policies reveals some interesting facts; this can be seen in figure 4 that shows the expected marginal values as a function of capacity remaining, for a fixed expected demand of 60 and with $\tau = 50$ days remaining. The SSR and DSR results are very similar in shape and absolute value, with the difference being the cut-off of the DSR once the total expected demand exceeds capacity. Both curves retain the monotonicity property in capacity since they increase monotonically as the available spaces are reduced. In contrast, the SMR policy seems to have lost this property when the capacity remaining drops below 20, as the curve appears to be bending downwards. This indicates that the last available spaces are not associated with the maximum price rate for stay for a single period. This is explained by the fact that when Δt is small, spaces are more efficiently utilised with a fewer number of customers occupying each period simultaneously - as a result, this reduces the expected marginal values.

6.2.2 *Revenue Performance Comparison* Next, we seek to examine how the bid-price policies perform in the test framework. For this purpose we have used 5000 booking reservations sets in the simulation. Let us denote by r_∞^{FCFS} the expected perpetual revenue rate per day after using the FCFS admission policy. Also let us define by r_∞^{SMR}, r_∞^{SSR} and r_∞^{DSR} the expected perpetual revenue rates that result after implementing the SMR, SSR and the DSR policy, respectively. Therefore, by dividing over r_∞^{FCFS} gives us a first indication of the methods' performance against the FCFS method.

Figure 5 and table 1 illustrate the relative performance of the methods for different carpark sizes. The key observations are summarised below;

(a) When capacity is congested, i.e when the total expected demand is much larger than the available capacity, all methods outperform the FCFS for all capacities. The added benefit is more pronounced

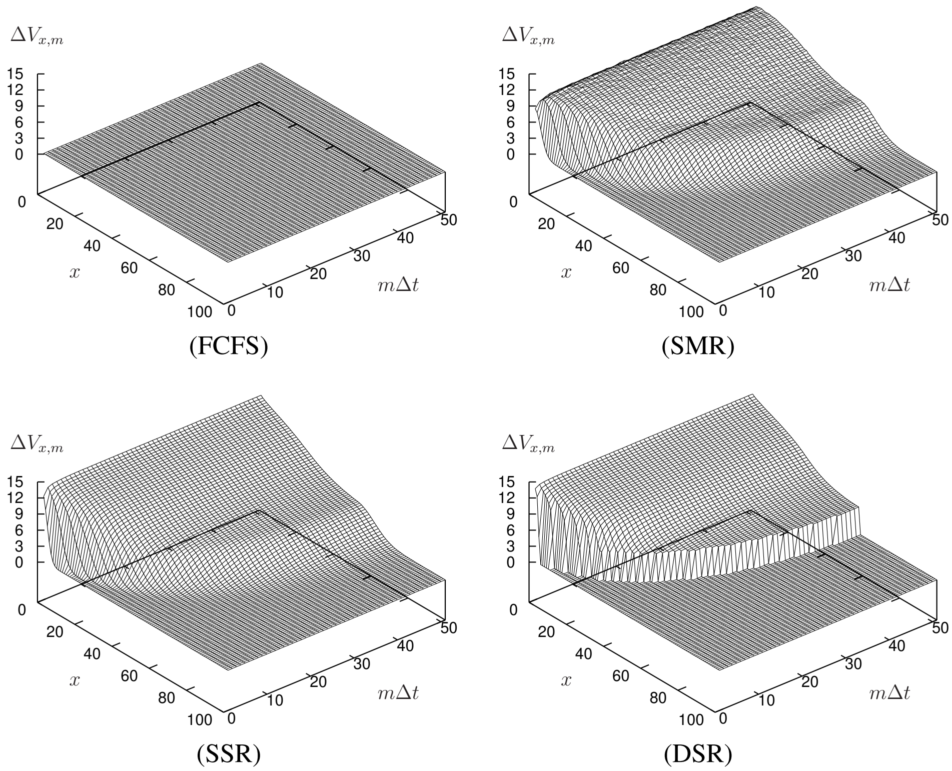

FIG. 3. Expected marginal values as functions of the capacity remaining, x and time remaining, in days $m\Delta t$, for the four approaches.

for relatively small carparks. This is because the booking flow in the carpark is controlled such that there will always be available spaces for the business customers who normally arrive later in the booking horizon. Under the FCFS policy, nothing restricts the leisure low-paying customers in filling the carpark early in advance, and consequently the carpark sells out long before the business and more profitable customers arrive. As the carpark size increases, the additional benefit from implementing an admission policy reduces; and eventually for relatively big carparks they all perform identically to the FCFS policy. The maximum benefit of the methods is observed at carpark size 10 when the SMR, SSR and DSR policies generate 48.4%, 47.0% and 44.2% higher expected revenues, respectively.

Given that our comparison is based on simulations, we can examine the statistical significance of the results obtained. In particular, we can test whether the difference in the perpetual revenues of each method against the FCFS method, is statistically significant. This is undertaken using a *two-sample t-test*; we run a test for each carpark size and assess it using a confidence level of $\alpha = 0.05$ while we also generate the 95% confidence intervals around these differences. The results for these tests are shown in tables 2, 3 and 4. For capacities up to 40, all three methods achieve statistically higher perpetual revenues than the FCFS policy, while for larger carpark sizes the improvement is no longer statistically significant.

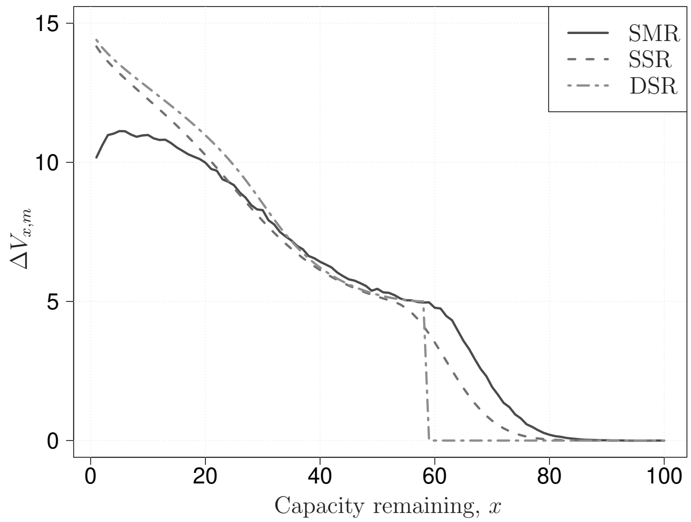

FIG. 4. Relative performance comparison of the three methods against the FCFS policy, for various carpark sizes with the expected demand fixed at 60 and $\tau = 50$ days (or $m = 50/\Delta t$ periods) remaining.

(b) Among all methods the SMR seems the superior method. Again, the impact of implementing the SMR policy as opposed to the others is more apparent for small capacities, when the booking control over the last few spaces is more appropriately managed; this is because the calculated marginal values of the last few spaces better reflect the true network structure of the problem when the SMR method is implemented. Since the marginal expected values take into consideration the neighbouring days the resulting bid-prices are lower in absolute value and tend to reject fewer short stay customers. For all other carpark sizes the performance of the three methods is similar. Putting this in a statistical context, and by looking at tables 5 and 6, the superiority of the SMR method cannot be generalised fully for all carpark sizes, at the selected confidence level.

(c) Comparing the SSR and the DSR methods, the former generates higher expected revenues for all carpark sizes. Evidently, the fact that the DSR method makes a deterministic statement on the total demand (assuming that the total demand equals its expectation) renders the sharp cut in the resulting bid-price table and inevitably deteriorates the resulting revenues. The maximum difference is observed for carpark size 5 where the SSR generates 4.0% higher expected revenues. Again, by looking at table 7, any difference in the performance between these two is not statistically significant.

Recall that for the DSR and SSR methods, we can obtain results for more realistic capacities. Thus we have experimented with carpark sizes up to $C = 500$ spaces, while adjusting for the expected demand accordingly; we scale the initial maximum carpark size of $C = 100$ by a factor of 5 to get $C^* = 500$. To ensure consistency, we thus apply the same factor on the arrival rates for the two classes while keeping the rest unchanged, namely

$$B^* \sim \begin{cases} \lambda_b = 1/3 \\ \lambda_a = 125 \\ \lambda_s = 1 \end{cases} \quad \text{and} \quad L^* \sim \begin{cases} \lambda_b = 1/14 \\ \lambda_a = 25 \\ \lambda_s = 1/7 \end{cases} \tag{41}$$

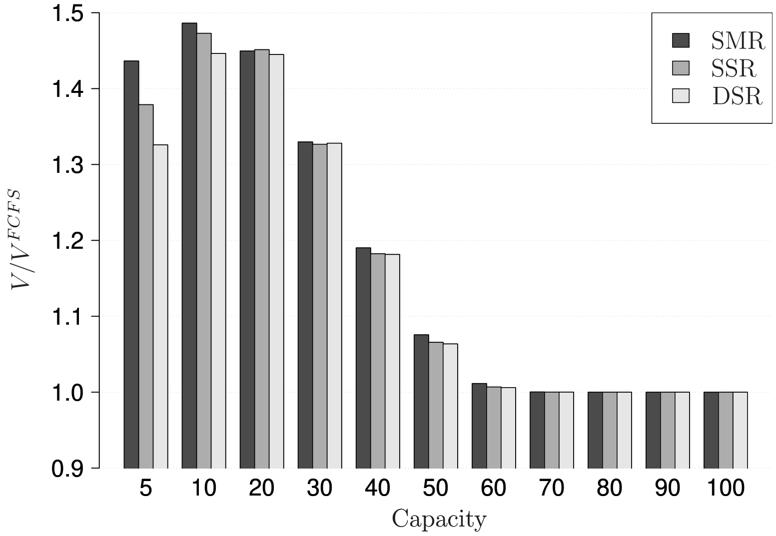

FIG. 5. Relative performance comparison of the three methods against the FCFS policy, for various carpark sizes.

Table 8 demonstrates our findings; the relative performance of the two methods in the scaled-up model scenario remains similar to the default setting, with any differences between the generated perpetual revenues being not statistically significant.

Table 1. Comparative performance of the SMR, SSR and DSR, relative to the FCFS policy, for various carpark sizes C.

C	r_∞^{FCFS}	$r_\infty^{SMR}/r_\infty^{FCFS}$	$r_\infty^{SSR}/r_\infty^{FCFS}$	$r_\infty^{DSR}/r_\infty^{FCFS}$
5	37.406	1.432	1.373	1.319
10	73.090	1.484	1.470	1.442
20	146.593	1.450	1.452	1.445
30	227.351	1.330	1.327	1.328
40	314.298	1.190	1.183	1.182
50	402.032	1.075	1.066	1.063
60	477.796	1.011	1.007	1.006
70	519.869	1.000	1.000	1.000
80	529.987	1.000	1.000	1.000
90	530.768	1.000	1.000	1.000
100	530.781	1.000	1.000	1.000

7. Conclusions

Three approximation RM methods have been studied in the context of airport carpark sales, which we refer to as Stochastic Multi Resource (SMR), Stochastic Single Resource (SSR) and Deterministic Single Resource (DSR). All these methods attempt to estimate the expected marginal values of the carpark spaces so that they can be used in a bid-price control in a running test framework. We have demonstrated

Table 2. Statistical Comparison of the SMR against the FCFS policy, for various carpark sizes C.

C	r_∞^{FCFS}	r_∞^{SMR}	$s.e.^{FCFS}$	$s.e.^{SMR}$	*Diff. in means*	*Diff. in means* *95% CI*	*P-value* *(t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	37.406	53.573	2.793	0.881	16.167	[10.425, 21.909]	0.000	significant
10	73.090	108.475	4.132	1.506	35.385	[26.763, 44.006]	0.000	significant
20	146.593	212.536	7.000	3.132	65.943	[50.911, 80.975]	0.000	significant
30	227.351	302.351	10.302	5.573	75.000	[52.039, 97.961]	0.000	significant
40	314.298	374.054	12.678	8.270	59.756	[30.084, 89.428]	0.000	significant
50	402.032	432.236	12.624	10.264	30.204	[-1.689, 62.097]	0.063	not significant
60	477.796	483.083	11.308	12.106	5.287	[-27.186, 37.76]	0.750	not significant
70	519.869	520.009	21.171	21.277	0.140	[-58.697, 58.977]	0.996	not significant
80	529.987	529.989	28.542	28.543	0.002	[-79.122, 79.126]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

Table 3. Statistical Comparison of the SSR against the FCFS policy, for various carpark sizes C.

C	r_∞^{FCFS}	r_∞^{SSR}	$s.e.^{FCFS}$	$s.e.^{SSR}$	*Diff. in means*	*Diff. in means* *95% CI*	*P-value* *(t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	37.406	51.356	2.793	1.237	13.950	[7.961, 19.939]	0.000	significant
10	73.090	107.415	4.132	1.936	34.325	[25.38, 43.269]	0.000	significant
20	146.593	212.809	7.000	3.316	66.216	[51.033, 81.399]	0.000	significant
30	227.351	301.715	10.302	5.240	74.364	[51.706, 97.022]	0.000	significant
40	314.298	371.720	12.678	7.208	57.422	[28.834, 86.01]	0.000	significant
50	402.032	428.411	12.624	8.377	26.379	[-3.32, 56.078]	0.082	not significant
60	477.796	481.040	11.308	10.682	3.244	[-27.248, 33.736]	0.835	not significant
70	519.869	519.930	21.171	21.208	0.061	[-58.68, 58.802]	0.998	not significant
80	529.987	529.988	28.542	28.543	0.001	[-79.122, 79.124]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

Table 4. Statistical Comparison of the DSR against the FCFS policy, for various carpark sizes C.

C	r_∞^{FCFS}	r_∞^{DSR}	$s.e.^{FCFS}$	$s.e.^{DSR}$	*Diff. in means*	*Diff. in means* *95% CI*	*P-value* *(t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	37.406	49.355	2.793	1.352	11.950	[5.866, 18.033]	0.000	significant
10	73.090	105.428	4.132	2.159	32.338	[23.199, 41.476]	0.000	significant
20	146.593	211.861	7.000	3.659	65.268	[49.785, 80.751]	0.000	significant
30	227.351	301.996	10.302	5.704	74.645	[51.561, 97.729]	0.000	significant
40	314.298	371.364	12.678	7.324	57.066	[28.365, 85.767]	0.000	significant
50	402.032	427.545	12.624	8.270	25.513	[-4.071, 55.097]	0.091	not significant
60	477.796	480.598	11.308	10.575	2.802	[-27.547, 33.151]	0.856	not significant
70	519.869	519.920	21.171	21.200	0.051	[-58.679, 58.781]	0.999	not significant
80	529.987	529.988	28.542	28.543	0.001	[-79.122, 79.124]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

the methodology of each approach explicitly and presented a number of important numerical results.

7.1 *Remarks*

The real network structure of the problem is difficult to model and to solve. Through a number of simplifications the underlying network problem was reduced to the SMR setting, which allowed us to calculate the expected marginal values for each space individually. The SMR model still retains some network features as the optimisation rule is applied to the booking request in total. Under this setting, a

Table 5. Statistical Comparison of the SSR against the SMR policy, for various carpark sizes C.

C	r_∞^{SSR}	r_∞^{SMR}	$s.e.^{SSR}$	$s.e.^{SMR}$	*Diff. in means*	*Diff. in means 95% Conf. Int.*	*P-value (t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	51.356	53.573	1.237	0.881	-2.217	[-5.194, 0.76]	0.144	not significant
10	107.415	108.475	1.936	1.506	-1.060	[-5.869, 3.749]	0.666	not significant
20	212.809	212.536	3.316	3.132	0.273	[-8.669, 9.215]	0.952	not significant
30	301.715	302.351	5.240	5.573	-0.636	[-15.631, 14.359]	0.934	not significant
40	371.720	374.054	7.208	8.270	-2.334	[-23.839, 19.171]	0.832	not significant
50	428.411	432.236	8.377	10.264	-3.825	[-29.796, 22.146]	0.773	not significant
60	481.040	483.083	10.682	12.106	-2.043	[-33.691, 29.605]	0.899	not significant
70	519.930	520.009	21.208	21.277	-0.079	[-58.966, 58.808]	0.998	not significant
80	529.988	529.989	28.543	28.543	-0.001	[-79.127, 79.125]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

Table 6. Statistical Comparison of the SMR against the DSR policy, for various carpark sizes C.

C	r_∞^{DSR}	r_∞^{SMR}	$s.e.^{DSR}$	$s.e.^{SMR}$	*Diff. in means*	*Diff. in means 95% Conf. Int.*	*P-value (t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	49.355	53.573	1.352	0.881	-4.217	[-7.38, -1.054]	0.009	significant
10	105.428	108.475	2.159	1.506	-3.047	[-8.207, 2.113]	0.247	not significant
20	211.861	212.536	3.659	3.132	-0.675	[-10.117, 8.767]	0.889	not significant
30	301.996	302.351	5.704	5.573	-0.355	[-15.987, 15.277]	0.964	not significant
40	371.364	374.054	7.324	8.270	-2.690	[-24.345, 18.965]	0.808	not significant
50	427.545	432.236	8.270	10.264	-4.691	[-30.529, 21.147]	0.722	not significant
60	480.598	483.083	10.575	12.106	-2.485	[-33.995, 29.025]	0.877	not significant
70	519.920	520.009	21.200	21.277	-0.089	[-58.965, 58.787]	0.998	not significant
80	529.988	529.989	28.543	28.543	-0.001	[-79.127, 79.125]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

Table 7. Statistical Comparison of the SSR against the DSR policy, for various carpark sizes C.

C	r_∞^{SSR}	r_∞^{DSR}	$s.e.^{SSR}$	$s.e.^{DSR}$	*Diff. in means*	*Diff. in means 95% Conf. Int.*	*P-value (t-test)*	*t-test outcome* ($\alpha = 0.05$)
5	51.356	49.355	1.237	1.352	2.000	[-1.592, 5.593]	0.275	not significant
10	107.415	105.428	1.936	2.159	1.987	[-3.697, 7.671]	0.493	not significant
20	212.809	211.861	3.316	3.659	0.948	[-8.732, 10.628]	0.848	not significant
30	301.715	301.996	5.240	5.704	-0.281	[-15.465, 14.903]	0.971	not significant
40	371.720	371.364	7.208	7.324	0.356	[-19.787, 20.499]	0.972	not significant
50	428.411	427.545	8.377	8.270	0.866	[-22.209, 23.941]	0.941	not significant
60	481.040	480.598	10.682	10.575	0.442	[-29.023, 29.907]	0.977	not significant
70	519.930	519.920	21.208	21.200	0.010	[-58.771, 58.791]	1.000	not significant
80	529.988	529.988	28.543	28.543	0.000	[-79.125, 79.125]	1.000	not significant
90	530.768	530.768	29.601	29.601	0.000	[-82.057, 82.057]	1.000	not significant
100	530.781	530.781	29.624	29.624	0.000	[-82.121, 82.121]	1.000	not significant

Monte Carlo approach has been implemented to simulate the booking sets, and an admission rule based on additive bid-prices was used for the booking control. An iterative scheme was designed in order to gradually converge to the correct estimates. The approach was found to be computationally intensive and the resulting surfaces still retained some degree of variance. Further, the resulting bid-price policy was found to be outperforming the policies from SSR and DSR, especially for small constrained capacity carparks. However, the feasibility of applying the SMR method in practice depends on a trade-off between the speed in calculating the policy and the added gains in performance. Note that in a real-time optimization keeping computation times in a practically low level is very desirable.

Table 8. Statistical Comparison of the SSR against the DSR policy, under a more realistic setting of larger capacities.

C	r_∞^{SSR}	r_∞^{DSR}	$s.e.^{SSR}$	$s.e.^{DSR}$	*Diff. in means*	*Diff. in means* *95% Conf. Int.*	*P-value* *(t-test)*	*t-test outcome* ($\alpha = 0.05$)
50	613.321	611.132	3.719	3.978	2.189	[-8.486, 12.864]	0.688	not significant
100	1180.440	1179.090	6.585	7.250	1.350	[-17.849, 20.549]	0.890	not significant
150	1651.090	1652.170	11.499	12.599	-1.080	[-34.517, 32.357]	0.950	not significant
200	2008.440	2007.640	16.488	16.817	0.800	[-45.366, 46.966]	0.973	not significant
250	2291.080	2288.870	19.009	18.823	2.210	[-50.227, 54.647]	0.934	not significant
300	2552.500	2551.920	26.106	25.989	0.580	[-71.628, 72.788]	0.987	not significant
350	2651.240	2651.240	65.056	65.056	0.000	[-180.345, 180.345]	1.000	not significant
400	2651.650	2651.650	65.749	65.749	0.000	[-182.266, 182.266]	1.000	not significant
450	2651.650	2651.650	65.749	65.749	0.000	[-182.266, 182.266]	1.000	not significant
500	2651.650	2651.650	65.749	65.749	0.000	[-182.266, 182.266]	1.000	not significant

Next, we have shown how to construct a continuous-time stochastic model, named SSR. The model is formulated as an HJB PDE model and the solution is calculated by using simple finite-differences techniques. Although this model no longer solves the network problem, the resulting surfaces have been shown to be smooth and they were actually computed in less than a minute. One potential approach that would exploit the speed advantage of the SSR model while preserving the SMR bid-price generation algorithm is to use them both in a joint scheme whereby the smoothed SSR admission policies serve as the starting point in the SMR iterations. In this way, we aim to speed up convergence without losing valuable information.

The final model we studied, namely DSR, was a direct extension of the previous SSR model formulated under the assumption of a continuous quantity. Our results verified that the DSR solution forms an upper bound to the SSR solution. The biggest advantage of the DSR method compared to SSR is the computation time which has now been limited to just a few seconds. However, further information is lost due to the assumption of the deterministic sales process, particularly when the capacity is close to the demand.

The bid-price policies resulting from the three models were found to perform significantly better than the naive FCFS policy. In particular, the SMR has shown the best overall performance with the added gains to be more apparent on low capacity carparks. The SSR and DSR did similarly well but there is room for further improvement.

Taking into account each of the methods' efficiency, the computation times and the resulting expected revenues, a hybrid approach that uses either SSR or DSR as a default and then switches to SMR in times of congestion would appear to be the preferred method. Each of the three methods showcase the merits of using the continuous time RM framework for real-time booking systems.

7.2 *Implications in real-world booking systems*

Our study exploits an RM setting under the traditional Poisson arrival Process, but with booking dynamics modelled by a two-dimensional probability density function, $\phi(\eta,\xi)$, with η and ξ being the lead time and length-of-stay, respectively. The versatility of the formulation allows for any mix of marginal distributions and an arbitrary number of customer classes with different characteristics, which together can model a variety of carpark or other RM scenarios.

In contrast to most RM problems, the length-of-stay dimension is also modelled in continuous-time, leading to a huge state space and intractable mathematical problems. We showed that applying a single-resource decomposition is feasible for this type of RM problem, which makes it possible to utilise a continuous-time formulation and exploit the associated algorithmical benefits. Also, the unique relationship between price and length-of-stay, a distinct feature of airport carparks, enables the management to apply dynamic capacity optimization by imposing a limit on the maximum length-of-stay allowed. The successfully tested idea of using the length-of-stay as a control variable to optimise customer demand hints at how other RM problems could be tackled in the near future, with extensions of the current model to incorporate spot (roll-up) demand and optimisation under multiple carparks to naturally go to future work.

References

Adelman, D. (2007). Dynamic bid prices in revenue management. *Operations Research*, 55(4):647–661.

Badinelli, R. D. (2000). An optimal, dynamic policy for hotel yield management. *European Journal of Operational Research*, 121(3):476–503.

Benjaafar, S., Cooper, W. L., and Mardan, S. (2011). Production-inventory systems with imperfect advance demand information and updating. *Naval Research Logistics (NRL)*, 58(2):88–106.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont.

Bertsimas, D. P. and De Boer, S. V. (2005). Simulation-based booking-limits for airline revenue management. *Operations Research*, 53:90–106.

Bitran, G. R. and Gilbert, S. M. (1996). Managing hotel reservations with uncertain arrivals. *Operations Research*, 44(1):35–49.

Bitran, G. R. and Mondschein, S. V. (1995). An application of yield management to the hotel industry considering multiple day stays. *Operations Research*, 43(3):427–443.

Chen, L. and Homem-de Mello, T. (2010). Mathematical programming models for revenue management under customer choice. *European Journal of Operational Research*, 203(2):294–305.

Chiang, W., Chen, J., and Xu, X. (2007). An overview of research on revenue management: Current isuues and future research. *International Journal of Revenue Management*, 1:97–128.

Cooper, W. and Homem-de-Mello, T. (2007). Some decomposition methods for revenue management. *Transportation Science*, 41(3):332–353.

De Boer, S. V., Richard, F., and Nanda, P. (2002). Mathematical programming for network revenue management revisited. *European Journal of Operational Research*, 137(1):72–92.

Gallego, G. and Van Ryzin, G. J. (1994). Dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020.

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

Goldman, P., Freling, R., Pak, K., and Piersma, N. (2002). Models and techniques for hotel revenue management using a rolling horizon. *Journal of Revenue and Pricing Management*, 1(3):207–219.

Guerriero, F., Miglionico, G., and Olivito, F. (2014). Strategic and operational decisions in restaurant revenue management. *European Journal of Operational Research*, 237(3):1119–1132.

Hwang, J., Gao, L., and Jang, W. (2010). Joint demand and capacity management in a restaurant system. *European Journal of Operational Research*, 207(1):465–472.

Jiang, H. (2008). A lagrangian relaxation approach for network inventory control of stochastic revenue management with perishable commodities. *Journal of the Operational Research Society*, 59(3):372–380.

Kemmer, P., Strauss, A. K., and Winter, T. (2012). Dynamic simultaneous fare proration for large-scale network revenue management. *Journal of the Operational Research Society*, 63(10):1336–1350.

Kirshner, S. N. and Nediak, M. (2015). Scalable dynamic bid prices for network revenue management in continuous time. *Production and Operations Management*.

Kunnumkal, S. and Topaloglu, H. (2010). A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management*, 19(5):575–590.

Ladany, S. P. (1976). Dynamic operating rules for motel reservations. *Decision Sciences*, 7(4):829–840.

Ladany, S. P. and Arbel, A. (1991). Optimal cruise-liner passenger cabin pricing policy. *European Journal of Operational Research*, 55(2):136–147.

Li, D. and Pang, Z. (2016). Dynamic booking control for car rental revenue management: A decomposition approach. *European Journal of Operational Research*.

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

Liu, S., Lai, K. K., Dong, J., and Wang, S. (2006). A stochastic approach to hotel revenue management considering multiple-day stays. *International Journal of Information Technology and Decision Making*, 5(3):545–556.

Liu, S., Lai, K. K., and Wang, S. Y. (2008). Booking models for hotel revenue management considering multiple-day stays. *International Journal of Revenue Management*, 2(1):78–91.

Maddah, B., Moussawi-Haidar, L., El-Taha, M., and Rida, H. (2010). Dynamic cruise ship revenue management. *European Journal of Operational Research*, 207(1):445–455.

Meissner, J. and Strauss, A. (2012a). Improved bid prices for choice-based network revenue management. *European Journal of Operational Research*, 217(2):417–427.

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

Olver, P. J. and Shakiban, C. (2006). *Applied Linear Algebra*. Prentice Hall.

Papayiannis, A. (2013). *On revenue management techniques: A continuous-time application to airport carparks*. PhD thesis, School of Mathematics, University of Manchester, UK.

Papayiannis, A., Johnson, P., Yumashev, D., and Duck, P. (2013). Continuous-Time Revenue Management in Carparks - Part Two: Refining the PDE. In *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems*, pages 76–81. ICORES, SciTePress Digital Library.

Papayiannis, A., Johnson, P., Yumashev, D., Howell, S., Proudlove, N., and Duck, P. (2012). Continuous-Time Revenue Management in Carparks. In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems*, pages 73–82. ICORES, SciTePress Digital Library.

Smith, B. C., Leimkuhler, J. F., and Darrow, R. M. (1992). Yield management at american airlines. *Interfaces*, 22(1):8–31.

Smith, G. D. (1985). *Numerical solution of partial differential equations: finite difference methods*. Oxford University Press.

Steer, D. G. (2013a). Assessment of commercial revenues at Gatwick airport. Technical report, Gatwick Airport.

Steer, D. G. (2013b). Assessment of commercial revenues at Heathrow airport. Technical report, Heathrow Airport.

Tan, T., Güllü, R., and Erkip, N. (2009). Using imperfect advance demand information in ordering and rationing decisions. *International Journal of Production Economics*, 121(2):665–677.

Topaloglu, H. (2009). Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, 57(3):637–649.

Weatherford, L. R. (1995). Length of stay heuristics: do they really make a difference? *The Cornell Hotel and Restaurant Administration Quarterly*, 36(6):70–79.

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

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

Zhao, W. and Zheng, Y. (2000). Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science*, 46(3):375–388.

Zhao, W. and Zheng, Y. (2001). A dynamic model for airline seat allocation with passenger diversion and no-shows. *Transportation Science*, 35(1):80–98.

A. Numerical Implementation of the MSR method

We describe the method under the time invariant framework which simplifies our problem greatly. In practice this means that the valuation drops by one dimension, from three dimensions to two. Thus, we define the matrix v to store our results; this is a $2D$ array composed of the expected values at every time and state of the carpark, defined by

$$V_{x,0,m} := v_{x,m}. \tag{42}$$

As before, this is interpreted as the expected value of the carpark with x spaces remaining and $m = k - j$ periods remaining until all spaces are occupied.

The procedure to determine the optimal solution v and the optimal admission control policy is based on iteration and Monte-Carlo simulation. Recall that two customer classes are assumed with different booking characteristics. Reservations from each customer class are simulated based on a Poisson arrival process, with Exponential lead times and length-of-stays, with class-specific intensities. Let us define a *sample path* as one simulated booking set, where each booking set is comprised of thousands of individual reservations. Given a large number of such paths, the methodology works by taking each sample path, one at a time, applying the booking control policy (bid-price table) on all the simulated reservations of that sample path and calculating the resulting revenue. The procedure is repeated for all paths and then the expected revenue v is estimated by taking their average value. Once the expected revenue is calculated, we also calculate the expected marginal values Δv, which are used to update the estimates of the bid-prices in the booking control policy; this completes one iteration. For the second iteration, the same paths are examined, but now under the new updated policy, with the entire procedure repeated, gradually improving on the values, until some convergence criterion is met. Note that at the first iteration the booking control assumes all bid-prices are zero.

Note that we use the same sample path to generate the revenue $v_{x,m}$ for all possible values of x and m. This guarantees that the p^{th} sample path gives a marginal value that is non-negative (i.e. $\Delta v^{(p)}_{x,m} \geqslant 0$). This will mean that over a total of P sample paths, our estimate of the expected marginal value, namely

$$\frac{1}{P}\sum_{p=1}^{P} \Delta v^{(p)}_{x,m} \geqslant 0,$$

is strictly non-negative as well. However, even if we use a large number of paths the variance in the calculated values will still be large. Consequently, the scheme might not converge, as the convergence relies on accurate estimates of the expected revenues. If the opportunity costs are overestimating the spaces, we will be rejecting too many customers in the next iteration, which will thus decrease the expected revenues. Then, the resulting policy will now be underestimating the spaces, leading to too many customers being accepted and thus push the expected revenues up. This oscillatory behaviour may potentially persist before converging.

Therefore, we propose an improved algorithm that will prevent us encountering such an undesirable situation. The algorithm is based on the idea that we should iterate towards the correct solution by using an *under-relaxation* scheme. As opposed to an over-relaxation scheme where the scheme's relaxation parameter ω is often set to be greater than one in order to speed up convergence, in this scheme we choose this parameter to lie between $0 \leqslant \omega < 1$. In this manner, the solution should steadily improve at every iteration. A reasonable choice for ω is to have $\omega = {}^{1}\!/_{P}$. Motivated by the *Gauss-Seidel* iteration method (details in Olver and Shakiban, 2006), within this algorithm new estimates on the expected values made available from one sample path are used directly in the evaluation for the next sample path. Under these assumptions we can observe *experimentally* that the algorithm converges to the true solution in a finite number of iterations.

As an extra measure for reducing the variance from the sample paths, we propose a scheme whereby the number of sample paths is multiplied by a factor of $\sqrt{2}$ at every iteration. Therefore a stopping criterion on the maximum total number of paths at each iteration, $P_{\max}$, is imposed in order to prevent the algorithm from excessive use of time and RAM.

The algorithm is as follows

1. Choose the booking horizon T and split it into K finite time periods of size Δt.

2. Set the value matrix v equal to 0. This implies that all spaces in the carpark are initially assumed worthless and therefore all customers can be accepted as long as capacity is available. Since this is our initial guess for v we denote it as $v^{(0)}$, where $v^{(r)}$ is the r^{th} estimate for the solution.

3. Choose the number of paths $P_{(r)}$ and the relaxation parameter $\omega_{(r)} = 1/P_{(r)}$.

4. Use Monte-Carlo simulation to generate booking sets within the pre-specified time interval $[0, T]$.

5. Evaluate the expected value of the carpark in period k for all time periods k, $k = 1, 2, \ldots, K$ and all possible capacities $0 \leqslant x_k \leqslant C$ (beginning with capacity 0) to generate the matrix y from

$$y_{x,k} = J^u_{\mathbf{x},0,k,x} \tag{43}$$

given that for the i^{th} booking made in the period j to arrive in period l and stay for D^i periods (or $\xi^i = D^i \Delta t$ days) the equation to satisfy is

$$\xi^i \Psi(\xi^i) \geqslant \sum_{k=l}^{l+D^i-1} \left(v^{(r)}_{x,k-j} - v^{(r)}_{x-1,k-j} \right). \tag{44}$$

6. Under-relax to obtain the next estimate of the value

$$v^{(r+1)}_{x,k} = v^{(r)}_{x,k} + \omega \left(y_{x,k} - v^{(r)}_{x,k} \right). \tag{45}$$

7. Go to step 4 P times.

8. Go to step 3, update P and ω to $P_{(r+1)} = P_{(r)}\sqrt{2}$ and $\omega_{(r+1)} = 1/P_{(r+1)}$, respectively, and repeat until $||v^{(r+1)} - v^{(r)}|| < \varepsilon$ or until $P > P_{\max}$, whichever occurs first.

Finally, recall that this *discrete*-time methodology is to be compared against two *continuous*-time alternatives in sections 4 and 5. Since the time period in a continuous-time setting is infinitesimal, the SMR method is solved for the limiting case $\Delta t \to 0$; in practice we have used $\Delta t = 6.25 \times 10^{-3}$, a sufficient size that balances out accuracy and computation times.

B. Computation of q and r rates for SSR and DSR

Recall that our goal is to pre-compute the occupancy and revenue rates $q(\xi|\tau)$ and $r(\xi|\tau)$ given by

$$q(\xi^*|\tau) = \Lambda_a \int_{\xi=0}^{\xi^*} \int_{\eta=(\tau-\xi)^+}^{\tau} \phi(\eta,\xi)\, d\eta\, d\xi = \int_{\xi=0}^{\xi^*} \rho_s(\eta,\xi)\, d\xi \tag{46}$$

and

$$r(\xi^*|\tau) = \Lambda_a \int_{\xi=0}^{\xi^*} \int_{\eta=(\tau-\xi)^+}^{\tau} \phi(\eta,\xi)\, \Psi(\xi)\, d\eta\, d\xi = \int_{\xi=0}^{\xi^*} \rho_s(\eta,\xi)\, \Psi(\xi)\, d\xi, \tag{47}$$

where

$$\rho_s(\eta,\xi) = \Lambda_a \int_{\eta=(\tau-\xi)^+}^{\tau} \phi(\eta,\xi)\, d\eta.$$

These rates are functions of two parameters, length-of-stay ξ and time τ, for all pairwise combinations. Suppose that the domain we will work on is rectangular with τ ranging from 0 to T and x ranging from 0 to $\xi_{\max}$. Divide $[0, T]$ into K equally spaced intervals at $K+1$ values indexed by $k = 0, 1, \ldots, K$. Similarly, we divide $[0, \xi_{\max}]$ into I integer intervals at $I+1$ values indexed by $i = 0, 1, \ldots, I$. The length of these intervals is $\Delta\tau$ in the time direction and $\Delta\xi$ in the length-of-stay direction such that $\tau^k = k\Delta\tau\ \forall k$ and $\xi^i = i\Delta\xi\ \forall i$. We then define

$$\mathbf{q}_i^k = q(i\Delta\xi | k\Delta\tau) \tag{48}$$

and

$$\mathbf{r}_i^k = r(i\Delta\xi | k\Delta\tau), \tag{49}$$

where $\mathbf{q}$ and $\mathbf{r}$ are the corresponding arrays with dimensions $(I+1) \times (K+1)$.

Notice that equation (48) can also be expressed as

$$\mathbf{q}_i^k = \int_0^{(i-1)\Delta\xi} \rho_s(\tau^k, \xi)\, d\xi + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\tau^k, \xi)\, d\xi = \mathbf{q}_{i-1}^k + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\tau^k, \xi)\, d\xi. \tag{50}$$

Thus, the next value of $\mathbf{q}_i^k$ can be expressed in terms of the previous value $\mathbf{q}_{i-1}^k$ plus an integral term corresponding to the increment $\Delta\xi$. Similarly, for equation (49) we have

$$\mathbf{r}_i^k = \mathbf{r}_{i-1}^k + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\tau^k, \xi)\, \Psi(\xi)\, d\xi. \tag{51}$$

Again, the next value of $\mathbf{r}_i^k$ may be computed as the previous value $\mathbf{r}_{i-1}^k$ plus an integral term corresponding to the increment $\Delta\xi$.

Now, if the difference between the two limits is sufficiently small (i.e. $\Delta\xi \to 0$) then the integral term can be approximated by the *one-step Simpsons Rule*. Therefore the above formulations for $\mathbf{q}$ and $\mathbf{r}$ are powerful as they enable us to recursively calculate the values q and r, one after the other. The recursion begins from the trivial case $\mathbf{q}_0^k = 0$ and $\mathbf{r}_0^k = 0$ for all k.

In the time-invariant case when the distribution of bookings are of the form (10) we can derive the following analytic formula:

$$q(\xi|\tau) = \begin{cases} q_1(\xi|\tau) & \text{if } \tau < \xi \\ q_1(\tau|\tau) + \frac{1}{P_a(\tau) - P_d(\tau)} \sum_n \alpha_n (e^{-\lambda_{a_n}\tau} - 1)(e^{-\lambda_{s_n}\xi} - e^{-\lambda_{s_n}\tau}) & \text{if } \tau \geqslant \xi \end{cases} \tag{52}$$

where

$$\begin{aligned} q_1(\xi|\tau) &= \frac{1}{P_a(\tau) - P_d(\tau)} \sum_n \alpha_n \left(\frac{\lambda_{s_n}}{\lambda_{a_n} - \lambda_{s_n}} e^{-\lambda_{a_n}\tau} \left(e^{(\lambda_{a_n} - \lambda_{s_n})\tau} - 1 \right) + e^{-\lambda_{a_n}\tau} \left(e^{-\lambda_{s_n}\xi} - 1 \right) \right) \\ P_a(\tau) &= 1 - \sum_n \alpha_n e^{-\lambda_{a_n}\tau} \\ P_d(\tau) &= 1 - \sum_n \alpha_n \left[\frac{\lambda_{a_n} e^{-\lambda_{s_n}\tau} - \lambda_{s_n} e^{-\lambda_{a_n}\tau}}{\lambda_{a_n} - \lambda_{s_n}} \right]. \end{aligned}$$

C. Numerical Implementation of the SSR method

We use an *explicit* finite difference scheme to solve this PDE. For details on this numerical technique the reader is referred to Smith (1985).

We first construct the mesh. In the stationary case the mesh has two dimensions only, the advance-time τ and the capacity remaining x. Suppose that the domain we will work on is rectangular with τ ranging from 0 to T and x ranging from 0 to C. Divide $[0,T]$ into K equally spaced intervals at $K+1$ values indexed by $k=0,1,\ldots,K$. Similarly, we divide $[0,C]$ into C integer intervals at $C+1$ values indexed by $j=0,1,\ldots,J$, so that we move with integer steps in space as parking spaces cannot be sold in fractions. The length of these intervals is $\Delta\tau$ in the time direction and $\Delta x=1$ in the state direction such that $\tau^k = k\Delta\tau\,\forall k$ and $x_j = j\,\forall j$. We seek an approximation to the values of V at the $(K+1)\times(C+1)$ grid points.

Therefore,

$$V(x_j,\tau^k) = V(j,k\Delta\tau) \approx v_j^k,$$

where v is a $2D$ array.

Similarly, if $[0,\xi_{\max}]$ is the domain for the length of stay ξ, we may divide it into I equally spaced intervals of length $\Delta\xi$ such that we have $\xi^i = i\Delta\xi$ for every $i=0,1,\ldots,I$.

The next step is to approximate the partial derivative of v at each grid point. More precisely, we use a forward divided difference in time to write it as

$$\frac{\partial v}{\partial \tau} = \frac{v_j^{k+1}-v_j^k}{\Delta\tau}.$$

Combining, the above we may write the numerical scheme as

$$v_j^{k+1} = v_j^k + \max_i \left\{ \mathbf{p}_i^k\,(v_{j-1}^k - v_j^k) + \mathbf{r}_i^k \right\}\Delta\tau,$$

with the boundary conditions

$$v_j^0 = 0 \quad \forall j \tag{53}$$

$$v_0^k = 0 \quad \forall k. \tag{54}$$

This is an explicit scheme as one could then proceed to calculate *explicitly* all the (unknown) v_j^{k+1}'s from the already computed (and thus known) v_j^k's and recursively obtain u for the entire grid. Note that such a numerical scheme will lead to a first order convergence in time.

The algorithm reads

1. Calculate the integrand and fill in the matrices $\mathbf{p}$ and $\mathbf{r}$ for all combinations of $i=0,1,\ldots,I$ and $k=0,1,\ldots,K$, as shown in the Appendix B
2. Set the time τ^k starting from $k=0$ to $k=K$.
3. Set the capacity j starting from $j=0$ (boundary condition) to $j=C$, where C the largest carpark size considered.
4. Set the optimal price $\Psi^* = \psi_0$ at the beginning. Calculate the optimal price rate as

$$\Psi^* = \max\left\{ v_{j-1}^{k-1} - v_j^{k-1}, \psi_\infty \right\}$$

5. Calculate the respective optimal length of stay by

$$\xi^* = \left(\frac{\Psi^* - \psi_\infty}{\psi_0 - \psi_\infty} \right)^{-1/\mu}$$

6. Derive the index i using

$$i = \left\lfloor \frac{\xi^*}{\Delta \xi} \right\rfloor$$

7. Calculate

$$v_j^k = v_j^{k-1} + \left(\mathbf{q}_i^{k-1} \, (v_{j-1}^{k-1} - v_j^{k-1}) + \mathbf{r}_i^{k-1} \right) \Delta \tau$$

8. Go to step 3 C times

9. Go to step 2 K times.

D. Numerical Implementation of the DSR method

Under the deterministic setting the problem is greatly simplified. We now examine our strategy for the two possible scenarios. When the expected total demand to come is less than the carpark capacity C then the problem is trivial because we maximise the revenues by letting $\xi^* \to \infty$ so that $\Psi^* \to \psi_\infty$. In this case all customers are accepted irrespective of their duration of stay. Conversely, when the expected total demand to come is greater than the carpark capacity C, there should exist a constant pricing policy under which we can sell all the spaces exactly. Mathematically this implies that the optimal policy should result in $x(\tau = 0) = 0$.

Let us, consider the initial value problem that arises from equations (39), evaluated at ξ, while using the results in Appendix B,

$$x(\tau)\Big|_{\xi} = C - \int_{\tau}^{T} q(\tau'|\xi)\, d\tau' \tag{55}$$

where $T = K\Delta\tau$ and $\tau = k\Delta\tau$ for some k and $\xi = i\Delta\xi$ for some i.

At expiry ($\tau = 0$) the remaining unsold spaces are given by $x(\tau = 0)\big|_{\xi}$. Let us define the function $\mathscr{F}$ as

$$\mathscr{F}(\xi) \equiv x(\tau = 0)\Big|_{\xi} = C - \int_{0}^{T} q(\tau'|\xi)\, d\tau'$$

We know that, if we had used the optimal length of stay ξ^* we would have sold the entire inventory exactly and therefore we would have

$$\mathscr{F}(\xi^*) = 0.$$

This suggests the implementation of a root-finding method (e.g. Newton-Raphson) to find the root of $\mathscr{F}$, i.e. the optimal value ξ^*. Finally, we can substitute the optimal ξ^* in equation (38) to obtain the maximised total revenue rate generated within the time horizon T; more precisely, we evaluate the integral

$$V(x = C, \tau = T) = \int_{0}^{T} r(\tau|\xi^*)\, d\tau \tag{56}$$

FIND THE OPTIMAL LENGTH-OF-STAY AND CALCULATE THE RESULTING REVENUE RATE We define the one dimensional array $\mathbf{l}^*$ to store the optimal duration of stay for each carpark size. To do so, we divide the range of carparks, $[0, C]$, into C integer intervals at $C+1$ values indexed by $j = 0, 1, \ldots, J$, so that we move with integer steps as carparks cannot have fractional capacity.

$$\mathbf{l}_j^* = \xi^*(x_j)$$

The procedure to find the optimal durations ξ^* for each carpark size considered, works as follows

1. Calculate the integrand and fill in the matrices **p** and **r** for all combinations of $i = 0, 1, \ldots, I$ and $k = 0, 1, \ldots, K$, as shown in the Appendix B

2. For carpark size x_j do

   (i) Set $i = 0$, the case when the allowed duration is set to 0 and thus no customers are allowed to enter.

   (ii) Solve the IVP problem in (55) using the numerical approximation

   $$\mathscr{F}(\xi) = x_j - [\mathbf{q}_i^K - \mathbf{q}_i^0]$$

   .

   (iii) If $||\mathscr{F}(\xi)|| < \varepsilon$, where ε is the tolerance level, ξ^* has been found. Go to 3

   (iv) Otherwise, update guess on ξ as

   $$\xi^{r+1} = \xi^r - \frac{\mathscr{F}(\xi^r)}{\mathscr{F}'(\xi^r)},$$

   where in the updating equation, $\mathscr{F}$ is defined using the analytical formula for $q(\xi|\tau)$ as shown in (52) which allows for the derivative to be calculated, and update i by

   $$i = \left\lfloor \frac{\xi^{r+1}}{\Delta\xi} \right\rfloor$$

   (v) Go to step ii

3. Set

   $$i^* = \left\lfloor \frac{\xi^*}{\Delta\xi} \right\rfloor$$

   and calculate the revenue rate generated as

   $$v_j^K = \sum_{k=0}^{K} \mathbf{r}_{i^*}^k. \tag{57}$$

4. Store the optimal length of stay, $\mathbf{l}_j^* = \xi^*$, to use later on

5. Go to 2

DERIVE CORRESPONDING BID-PRICES FOR EACH TIME AND CAPACITY REMAINING Our goal is to construct bid-prices as functions of time and capacity remaining, and not to use the actual solution (revenues) directly. Unlike the SSR method whereby the resulting expected values are used directly to obtain the expected marginal values for all x and τ and hence construct the bid-price control policy, in the DSR method this requires a rather more work. The DSR method calculates the revenue to be generated from a carpark with C available spaces and $\tau = T$ time remaining to sell them. Thus, if we wish to calculate the revenue to be generated from a carpark with C spaces and $\tau = T'$ time remaining we should resolve the problem starting from the new time T'. After integrating for the values for all C and

T' we could evaluate the marginal values and thus create the bid-price control policy. This procedure requires resolving the problem for all possible C and τ which is incredibly inefficient.

Fortunately, we do not have to do this because we can use the optimal durations in $\mathbf{l}^*$ directly. For example, if we want to calculate the marginal value when there are x_j spaces remaining and τ time left ($\tau < T$), we can navigate to the required time point and search through the different trajectories to find the one passing closest to x_j. Once this is identified we can take $\xi^* = \mathbf{l}_j^*$ (or the interpolated ξ^* between the two closest trajectories) to calculate $\Psi(\xi^*)$, which is the corresponding expected marginal value and hence the bidprice at that point.

We first construct the mesh. In the stationary case the mesh has two dimensions only, the advance-time τ and the capacity remaining x. Suppose that the domain we will work on is rectangular with τ ranging from 0 to T and x ranging from 0 to C. Divide $[0,T]$ into K equally spaced intervals at $K+1$ values indexed by $k = 0,1,\ldots,K$. Similarly, we divide $[0,C]$ into C integer intervals at $C+1$ values indexed by $j = 0,1,\ldots,J$, so that we move with integer steps in space as parking spaces cannot be sold in fractions. The length of these intervals is $\Delta\tau$ in the time direction and $\Delta x = 1$ in the state direction such that $\tau^k = k\Delta\tau\ \forall k$ and $x_j = j\ \forall j$.

Next define

$$\mathbf{p}_j^k = \pi(x_j, \tau^k),$$

a $(K+1)\times(C+1)$ array to store the resulting bid prices and proceed as follows

1. Iterate through all times $k = 0,1,...,K$
2. Iterate through all remaining capacities $j = 0,1,...,J$
3. Find the sales trajectory S that passes *through* the point $(\tau^k = k\Delta\tau, x_j = j)$ (or else, the two trajectories S_1, S_2 closest to that point) and note down the carpark size m (or sizes m_1, m_2) it has initiated from, along with the resulting optimal length-of-stay used $\mathbf{l}_m^*$ (or $\mathbf{l}_{m_1}^*, \mathbf{l}_{m_2}^*$).
4. The optimal length-of-stay to use is

$$\xi^* = \begin{cases} \mathbf{I}_m^* & \text{if trajectory S passes through the point} \\ \mathbf{I}_{m_1}^* + (j - m_1)\frac{\mathbf{I}_{m_2}^* - \mathbf{I}_{m_1}^*}{m_2 - m_1} & \text{if trajectories } S_1 \text{ and } S_2 \text{ pass closest to the point} \end{cases} \tag{58}$$

5. Set $\mathbf{p}_j^k = \Psi(\xi^*)$
6. Go to 2 J times
7. Go to 1 K times