CONTINUOUS-TIME REVENUE MANAGEMENT IN CARPARKS

A. Papayiannis[1], P. Johnson[1], D. Yumashev[1], S. Howell[2], N. Proudlove[2] and P. Duck[1]
[1]*School of Mathematics, University of Manchester, Oxford Road, M13 9PL, Manchester, U.K.*
[2]*Manchester Business School, University of Manchester, Manchester, U.K.*

Keywords: Expected revenue, Rejection policy.

Abstract: In this paper, we study optimal revenue management applied to carparks, with primary objective to maximize revenues under a continuous-time framework. We develop a stochastic discrete-time model and propose a rejection algorithm that makes optimal decisions (accept/reject) according to the future expected revenues generated and on the opportunity cost that arises before each sale. For this aspect of the problem, a Monte Carlo approach is used to derive optimal rejection policies. We then extend this approach to show that there exists an equivalent continuous-time methodology that yields to a partial differential equation (PDE). The nature of the PDE, as opposed to the Monte Carlo approach, generates the rejection policies quicker and causes the optimal surfaces to be significantly smoother. However, because the solution to the PDE is considered not to solve the 'full' problem, we propose an approach to generate optimal revenues using the discrete-time model by exploiting the information coming from the PDE. We give a worked example of how to generate near-optimal revenues with an order of magnitude decrease in computation speed.

1 INTRODUCTION

Over the last twenty years, cars have formed the main transportation system for people worldwide, especially in developed countries. Since parking is essential for cars, this creates an opportunity for carparking owners to exploit the increased demand to maximize their turnover. This can be achieved through revenue management (RM) techniques, such as those used in the hotel industry, where customers purchase multiple units in one transaction. Nevertheless, in carparking, most research has focused upon the problem of reducing traffic congestion; For example, (Young et al., 1991) argue that carparks play a major role in the planning and management of transportation systems and thus, appropriate parking pricing polices can be used to reduce traffic congestion. A sample list of related work includes (Vickrey, 1969), (Vikrey, 1994), (Young et al., 1991), (Verhoef et al., 1995), (Teodorović and Vukadinović, 1998), (Arnott and Rowse, 1999) and (Zhao et al., 2010). Whilst the literature upon carparking RM is small, there are two worthy studies of mention. The first is (Teodorović and Lučić, 2006), who propose an "intelligent" parking space inventory control system, based on fuzzy logic and integer programming techniques. They study the problem of maximizing revenues when customer arrival and departure times are stochastic, assuming different parking tariffs (prices). For a parking request of a particular tariff, it is possible to know the percentage of the capacity remaining and the relative requests' revenue, as well as the percentage of all future requests that make less relative revenue than the current request; in this way, fuzzy rules are generated (for detailed information in creating fuzzy rules, the reader is referred to the work of (Wang and Mendel, 1992)). The problem is studied under several scenarios and the results show that the relative error (between the proposed algorithm and the optimal upper bound) never exceeds 10%. The initial setup of their objective function has strong similarities to the setup of our discrete-time model; however, their algorithm is assumed to be able to "recognize" the type of the request and to direct it to the appropriate fuzzy rule base, in which a decision is made. Our system combines all booking requests from both customers sets in consideration, so that it makes a decision to accept/reject a request without knowing the booking set each request comes from.

(Onieva et al., 2011) also study revenue management being applied in carparks. They consider the presence of a group of subscribers along with the individual customers while the arrivals are assumed to follow a non-homogeneous Poisson distribution. They examine the problem under both a deterministic and a stochastic environment and they develop three

Papayiannis A., Johnson P., Yumashev D., Howell S., Proudlove N. and Duck P..
CONTINUOUS-TIME REVENUE MANAGEMENT IN CARPARKS.
DOI: 10.5220/0003762800730082
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 73-82
ISBN: 978-989-8425-97-3

different algorithms for capacity allocation; a first-come-first-served, distinct and nested method. Using these simulation techniques, profit maximization were investigated, although little insight into the core dynamics of the model was supplied. Their results suggest that a stochastic model using nested allocation provides revenues that are closest to the optimal values.

Within this study we assume the following:

1. Their is a finite fixed number of spaces in a carpark. Fixed capacity means that no more revenue can be generated when there are no spaces left.
2. The inventory is perishable and it can be sold in advance or on arrival.
3. The demand for the product is time-invariant.

The main objective of our study is to *Maximize Profits by Optimally Managing the Bookings in a Continuous-Time Environment*. What makes it distinct in our carparking revenue maximization problem is the assumption of a continuous time framework.

We consider a carpark operating under the above conditions and a target time T for which the spaces must be used; for this two approaches are introduced. We begin by generating sets of bookings using a Poisson distribution. The bookings arrive continuously, but the cars are assumed to occupy the parking slots for discrete periods of time, Δt. The bookings are allocated a price rate per day according to their duration of stay (the more the stay days, the less the price paid per day). We develop a discrete-time model that makes a decision (accept/reject) for each one, in the order the bookings are recorded. The decision is based on the expected revenues generated in the carpark and on the opportunity cost that arises before each sale. In particular, we develop a rejection algorithm according to which, given there is capacity available, we do not sell any space for time T at any time prior, $t < T$, for less money than what we expect to receive for it in the future.

Then, a continuous-time model is introduced, leading to a partial differential equation (PDE); The methodology behind the derivation lies in the work of (Gallego and van Ryzin, 1994) who proposed a decision tree approach. The PDE approach aims to replicate the results of the discrete-time model when Δt tends to zero. The continuous model is based on the probability distributions used previously to generate the bookings. Instead of looking at the revenues generated within a finite time period, the model calculates the rate at which the value of the carpark changes during an instant of time. Again, bookings are allowed to request any length of stay, but the rejection policy will make a decision at each time period individually; given a number of periods requested by a booking, the policy may deny a parking slot for some of these periods, but still collect the revenues from the periods that are accepted. Thus, the PDE is assumed not to solve the 'full' problem.

Each approach will generate an optimal rejection policy, based on which the revenues will be maximized. The slight difference in the manner in which the rejection algorithms work, will generate slightly higher revenues for the PDE[1]. However, the use of the PDE is favourable as it produces much quicker and smoother results. Therefore, we examine the case of using the rejection algorithm in the Monte Carlo approach but with the opportunity costs (rejection policy) being calculated from the PDE. We show under which conditions, the use of the PDE rejection policy generates maximum revenues for the full problem and, in the case of near optimal revenues, we explain the adjustments that have to be implemented.

The remainder of this report is organized as follows. In section 2, we define the problem, list the set of assumptions used and develop the discrete-time model. The continuous-time PDE model is introduced and derived in section 3 with the numerical results from both approaches to follow in section 4. Section 5 presents our conclusions and thoughts for future research in this area.

2 PROBLEM FORMULATION

2.1 The Model

We begin by describing the structure of the bookings. Each booking consists of three characteristics, the time the booking is made, the time of arrival to the carpark and the time of departure from the carpark. Therefore, each booking i can be written as a vector, namely

$$B^i = \begin{pmatrix} t_b \\ t_a = t_b + \eta \\ t_d = t_a + \xi \end{pmatrix} \quad (1)$$

where t_b denotes the booking time, t_a the arrival time with η denoting the pre-booking time and t_d the departure time with ξ denoting the duration of stay.

Bookings arrive in a continuous time; each one requires a space in the carpark for a particular time period and, thus, the customer is required to pay an amount of money according to his/her duration of stay.

To achieve this, any given time interval $t \in [a,b]$ is

[1] Justification on this is shown in section 4.

split into K discrete time steps, each of length Δt, so that

$$t^k = a + k\Delta t. \quad \text{for} \quad k = 0,1,\ldots,K.$$

Then, if C^k denotes the number of cars present in the car park at any time during the period $t \in [t^k, t^{k+1})$, we have

$$C^k = \sum_i f(B^i, k)$$

where

$$f(B,k) = \begin{cases} 1 & \text{if} \quad t^k \le t_a < t^{k+1} \\ 1 & \text{if} \quad t^k < t_d \le t^{k+1} \\ 1 & \text{if} \quad t_a < t^k \quad \text{and} \quad t_d > t^{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Note that we discount cars departing at $t = t^{k+1}$ as being present during the period.
The duration of stay for a booking is, then, the number of periods at which this booking is present in the car park,

$$\xi^i = \sum_k f(B^i, k). \tag{3}$$

Now, suppose that the price rate per day (period) for a booking B^i changes according to a log-linear[2] pricing function of the form

$$\Psi(\xi^i) = \psi_1 + \psi_2 e^{-\mu\xi} \tag{4}$$

where ξ is the duration of the i^{th} booking in days (periods) and ψ_1, ψ_2 are positive constants and μ is the decaying coefficient.

Therefore, the revenue generated in the k^{th} period over all bookings is

$$V^k = \sum_i f(B^i, k)\Psi(\xi^i) \tag{5}$$

and the total revenue R for the carpark is

$$R^k = \sum_k \sum_i f(B^i, k)\Psi(\xi^i). \tag{6}$$

2.2 Generating Bookings

The bookings are generated using a Poisson distribution with constant intensity λ_b. Thus, λ_b indicates the average number of bookings made during a day (the standard unit of analysis in this paper). Even though we assumed that the average number of bookings made in a day is known, this is a stochastic problem because their exact number is still unknown.

[2]Our intuition indicates that a pricing function of this form is more common to be used in a real carpark, and it is easy to work with. A log-linear pricing function requires that the price rate per day (period) decreases monotonically in the number of days requested but, at the same time, it guarantees that the daily price never drops below a lower minimum we choose.

The time of the next booking can be calculated as follows

$$t^n = t^{n-1} - \frac{1}{\lambda_b}\log(u^n), \tag{7}$$

where u^n is a random variable from the uniform distribution and t^{n-1} the time of the last booking. If we know that the average time between booking and arrival is $\bar{\eta}$ units of time and the average time between arrival and departure is $\bar{\xi}$ units of time, then we can use Poisson processes with intensities $\lambda_a = 1/\bar{\eta}$ and $\lambda_s = 1/\bar{\xi}$ to model the arrival and duration of stays, respectively. Therefore, by knowing the last booking we can generate the next booking as:

$$B^{i+1} = \begin{pmatrix} t_b = t^i - (1/\lambda_b)\log(u^n) \\ t_a = t_b - (1/\lambda_a)\log(u^{n+1}) \\ t_d = t_a - (1/\lambda_s)\log(u^{n+2}) \end{pmatrix}. \tag{8}$$

2.2.1 Notation and Further Assumptions

(i) No discounting takes place, for simplicity. We do not consider the time-value of money as the report's objective is to examine the performance of the rejection algorithm.

(ii) There is no marginal cost incurred after a sale. This is valid, because one can always express price as the increment above cost. Thus, the expressions "revenue" and "profit" will be used interchangeably.

(iii) There are no cancellations; if a booking for a particular duration is accepted, then the customer will show up and pay with probability almost surely.

(iv) Two types of customers are considered - because there is no time variation, demand intensities can be set to fixed values for the entire time horizon. Each set of parameters is carefully chosen to distinguish between the different customers types:

- Low-paying customers

$$B_{Leis} \sim \begin{cases} \lambda_b = 5 \\ \lambda_a = 1/14 \\ \lambda_s = 1/7 \end{cases} \tag{9}$$

These customers book early in advance to take advantage of any discounts or promotions, they require a space for long periods and usually these represent leisure customers.

- High-paying customers

$$B_{Busi} \sim \begin{cases} \lambda_b = 25 \\ \lambda_a = 1/3 \\ \lambda_s = 1 \end{cases} \tag{10}$$

These customers book just before or on arrival. High-paying customers are usually business customers who are not flexible within dates, and thus they are willing to pay full prices for just a short period of time.

These two sets of customers are combined according to the time the bookings are made, so that the system makes a decision about bookings in the order they arrive and do not know which booking set they come from.

2.2.2 Rejection Policy and Expected Values

The manager of the carpark can reject a request for a space if they so choose. If the booking is rejected then the customer cannot change their length of stay to be accepted, the potential revenue for each period of stay is lost. As such the bookings will be called *group* bookings over different days and hence different products. All decisions must be based on current information and without knowledge of future events, making it a non-anticipating policy. We call a policy that satisfies this criteria an admissible policy, denoted by π.

Thus, let $V(C,Q,t;k)$ to denote the expected value of the carpark of total capacity C with Q spaces remaining at time t until the space is used at time period T^k. By equation (5), this is

$$V(C,Q,t;k) = E\left[\sum_{i=i^*} f(B^i,k)\Psi(\xi^i)\right]$$

where i^* indicates the next booking made after t.
Since the goal is to maximize expected revenues, we can write the problem as,

Maximize:

$$V(C,Q,t;k) = \max_{\pi}\left\{E\left[\sum_{i=i^*} f(B^i,k)\Psi(\xi^i)\right]\right\} \quad (11)$$

subject to:

$$0 \le C^k \le C, \quad \forall k = 0,1,\ldots,K, \quad (12)$$

where $\pi \in \Pi$ is any policy from the set of all admissible rejection policies. The capacity constraint in (12) requires that the number of cars present in any period should never exceed the total capacity of the carpark.

The quantity $V(C,Q,t;k) - V(C,Q-1,t;k)$, is the opportunity cost lost, incurred when we move from a carpark of total capacity C with Q spaces remaining at time t to one with $Q-1$ spaces left. This quantity suggests how much the q^{th} unit of space is expected to be worth at time t, denoted as the *Expected Added Value of the space at time t*.

2.3 Rejection Algorithm

Our rejection algorithm is based on (Gallego and van Ryzin, 1994) and Littlewood's Rule ((Littlewood, 1972)), and suggests that a booking (at t_b) will be rejected if the total revenue generated is lower than the expected revenue of all potential future bookings that the car will displace over all periods it is present in the carpark. In other words, it makes sense to accept the booking i, only if the price satisfies:

$$\Psi(\xi^i) > V(C,Q,t_b;k) - V(C,Q-1,t_b;k) \quad (13)$$

However, the booking decision should be taken according to the total length of stay and not for each day period individually; so for the ith booking made within the period t^n we find it convenient to introduce the *Added Value across all periods* during which the car is present to be:

$$A = \sum_k f(B^i,k)[\Psi(\xi^i) - (V(C,Q,t^n;k) - V(C,Q-1,t^n;k))], \quad (14)$$

with $n \le k$. Then, the rule is

Accept if: $A \ge 0$
Reject if: $A < 0$.

That demand is time invariant along with the no discounting assumption enable us to calculate the expected value V going backward and forwards in time at the same time. Therefore, the expected value of the carpark of total capacity C with Q_j spaces remaining at t^n for the space to be used at t^k is

$$V(C,Q_j,t^n;k) = V(Q_j,Q_j,0;k-n) = v_{k-n,j}.$$

The resulting 2D-matrix v will then be used to determine the rejection policy π in the following algorithm;

2.3.1 Optimal Rejection Policy Algorithm

1. Choose a booking horizon T with K periods sufficiently large to capture nearly all of bookings in each set and a maximum capacity for the carpark C.
2. Initialize the rejection matrix $v^0 = 0$ so the value of a space is zero, where v^q is the qth guess at the solution of the rejection algorithm v.
3. Use Monte-Carlo to generate booking sets in within the time interval $[0,T]$.
4. Evaluate the expected value of the carpark (of total capacity C) at time t for all time periods t^k and

Table 1: Parameters Used.

Maximum Capacity	$C_{max} = 100$
Time Horizon	$[0,T]$, where $T = 30$ days
Pricing function	$\psi_1 = 5$, $\psi_2 = 10$, $\mu = 2/11$

all possible capacities $0 < Q_j < C$ to generate the matrix

$$v_{k,j}^{q+1} = E\left[\sum_i f(B^i,k)\Psi(\xi^i)\right]$$

given that for the ith booking in the period t^n the added value is

$$A = \sum_k f(B^i,k)[\Psi(\xi^i) - \left(v_{k-n,j}^q - v_{k-n,j-1}^q\right)]. \quad (15)$$

5. Go to step 3 and repeat until $||v^{q+1} - v^q|| < \varepsilon$.

The parameters we use to derive our results are listed in table 1.

2.4 Probability Distributions

Previously, we described how a set of bookings can be characterized by intensity parameters denoted λ_b, λ_a and λ_s, and assumed that the number of bookings made in a period of time follows a Poisson distribution.

Next, denote the parameters that correspond to the leisure booking set and those that correspond to the business booking set by the subscripts 1 and 2, respectively. Thus, the probability that a customer arrives at the carpark η $(= t_a - t_b)$ days after the booking has been made, follows an Exponential distribution, namely

$$\rho_a(\eta) = \sum_i \alpha_i \rho_{a_i}(\eta) = \sum_i \alpha_i \lambda_{a_i} e^{-\lambda_{a_i}\eta} \text{ for } i = 1,2 \quad (16)$$

and, similarly, the probability that a customer stays in the carpark for ξ $(= t_d - t_a)$ days is given by

$$\rho_s(\xi) = \sum_i \alpha_i \lambda_{s_i} e^{-\lambda_{s_i}\xi} \text{ for } i = 1,2, \quad (17)$$

where the weight

$$\alpha_i = \frac{\lambda_{b_i}}{\sum_j \lambda_{b_j}}$$

is the probability of the next booking to be from booking set i.

These are the quantities used to generate a booking set in the Monte-Carlo approach. Given these quantities, we may denote the cumulative probability that a customer arrives not more than η days after booking as

$$P_a(\eta) = \sum_i \alpha_i P_{a_i}(t) = \sum_i \alpha_i \int_0^\eta \rho_{a_i}(t) = 1 - \sum_i \alpha_i e^{-\lambda_{a_i}\eta}, \quad (18)$$

and the cumulative probability of staying not more than ξ days as

$$P_s(\xi) = 1 - \sum_i \alpha_i e^{-\lambda_{s_i}\xi} \quad (19)$$

Now, let us consider the probability that a customer departs from the carpark exactly z days after the booking has been made. If we denote this by $\rho_d(z)$ we then may write:

$$\rho_d(z) = \sum_i \alpha_i \int_0^z \rho_{a_i}(t)\rho_{s_i}(z-t)\,dt. \quad (20)$$

Thus, $\rho_d(z)$ can be found by integrating over all possible combinations of arrival time and length of stay that if added together they give exactly z days. Or in other words we sum over all instances where the length of stay plus the arrival time is equal to z.

Then, by defining the corresponding cumulative probability as,

$$P_d(z) = \int_0^z \rho_d(t)\,dt \quad (21)$$

and using (18), it can be proved that the probability of a customer being present z days after the booking, $g(z)$, may be written as,

$$g(z) = P_a(z) - P_d(z) \quad (22)$$

Next, using conditional probabilities, we can show that the expected distribution of stay given that the customer booking z days in advance will be present and stay ξ days is

$$\rho_s(\xi|z) = \sum_i \alpha_i \frac{\rho_{s_i}(\xi)\,[P_{a_i}(z) - P_{a_i}(\max\{z-\xi,0\})]}{g(z)}. \quad (23)$$

This probability is the most important function we deal with, as it will tell us the distribution of which type of customers (characterized by their length of stay) are present on a particular day. Therefore, the

cumulative probability of a customer staying at most ξ days given that he is present z days after the booking, is given by

$$P_s(\xi|z) = \int_0^{\xi} \rho_s(s|z)\,ds \quad \text{for } i = 1,2. \qquad (24)$$

This cumulative probability will be vital for constructing the rejection policy.

3 LINKING TO THE PDE

We can now consider the continuous dynamic formulation for the revenue generated in the carpark at the instant $t = T$. The aim is to derive a PDE, going backwards in time, such that the Monte Carlo simulations will converge to its solution as the time interval goes to zero, $\Delta t \to 0$. The PDE method implies that the problem is solved for each period of time independently, because the states of the carpark before or after the period in consideration do not contribute to the decision being made. This implies that there are no group bookings and, therefore, the expected added values of the spaces should be slightly higher.

Let $V = V(Q,t;T)$, be the instantaneous rate at which revenue is generated at time t for cars present over the instant T.

Assume that bookings present at time T arrive according to a Poisson distribution with time varying intensity derived by the function $f(t;T)$. This can be written as,

$$\bar{f}(t;T) = \left(\sum_i \lambda_{b_i}\right) g(T-t).$$

Let $Q(t;T)$ to express the number of carparking spaces remaining at time t for the instant T. We can then consider what happens during an infinitesimal period dt (see (Gallego and van Ryzin, 1994)); we sell one space ($dQ = -1$) with probability $\bar{f}(t;T)dt + o(dt)$, we do not sell any space ($dQ = 0$) with probability $1 - \bar{f}(t;T)dt - o(dt)$ and we sell more than one spaces ($dQ > 1$) with probability $o(dt)$. Taking $dt \to 0$, we obtain that $E[dQ] = -\bar{f}(t;T)dt$.

Noting that Q is a discrete jump process, as only entire spaces can be sold and not parts of them, we could write that the change in the value over an instant t is,

$$dV = \frac{\partial V}{\partial t}dt - \bar{f}(t;T)dt[V(Q,t;T) - V(Q-1,t;T)]. \qquad (25)$$

Equation (25) gives us an indication on the probability of a customer arriving but, it does not capture the booking's length of stay nor the price he has to pay for that period.

Define $\rho_s(\xi|t;T)$ as the conditional probability of a customer booking at t to stay for ξ days given that he is present at time T. Then, the instantaneous cashflow at time t for customers present at T can be expressed as the total number of customers booking multiplied by the average price paid, namely

$$dV = -\bar{f}(t;T)dt \int_0^{\infty} \rho_s(\xi|t;T)\Psi(\xi)\,d\xi. \qquad (26)$$

3.1 Rejection Policy

Since in our current setting, intensities are time invariant and the solution of V at T is independent from all other T, we can use a change of variables to write $\tau = T - t$.

We accept a booking only if its corresponding price Ψ is greater than some optimal (minimum) price $\Psi^*(Q,\tau)$. Because there is one-to-one correspondence between price and length of stay (see equation (4)), with the price to be monotonically decreasing in duration of stay, we accept a booking only if its corresponding duration of stay ξ is less than some optimal (maximum) duration $\xi^*(Q,\tau)$. Thus, given some optimal maximum duration of stay ξ^* we may find that the instantaneous booking arrival rate for customers booking τ days before T to be present at T is

$$P_s(\xi^*|\tau)\bar{f}(\tau). \qquad (27)$$

Similarly, we can show that the resulting instantaneous cashflow rate is

$$\bar{f}(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\Psi(\xi)\,d\xi. \qquad (28)$$

Therefore, we may equate (25) and (26) and use (27) and (28), to obtain

$$\frac{\partial V}{\partial \tau} + P_s(\xi^*|\tau)\bar{f}(\tau)[V(Q,\tau) - V(Q-1,\tau)] = \bar{f}(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\Psi(\xi)\,d\xi. \qquad (29)$$

Since the objective is to maximize V by controlling ξ^*, we may write the problem as,

$$\frac{\partial V}{\partial \tau} = \max_{\xi^*}[P_s(\xi^*|\tau)\bar{f}(\tau)(V(Q-1,\tau) - V(Q,\tau)) + \bar{f}(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\Psi(\xi)\,d\xi], \qquad (30)$$

with the boundary conditions

$$V = 0 \quad \text{when} \quad \tau = 0 \tag{31}$$

$$V = 0 \quad \text{when} \quad Q = 0. \tag{32}$$

The solution to the optimization problem in equation (30) is the optimal value $V(Q,\tau)$ and the values $\Psi(\xi^*)$, with $\xi^* = \xi^*(Q,\tau)$, that achieve the supremum form the optimal rejection policy.

In order to check if our optimal solution is consistent with that derived in the discrete-time case, we can differentiate (30) with respect to the control ξ^* to obtain,

$$V(Q,\tau) - V(Q-1,\tau) = \Psi(\xi^*) \tag{33}$$

Equation (33) indicates that the marginal value should always be equal to the revenue generated by rejection at the optimal level $\xi^* = \xi^*(Q,\tau)$, meaning that a booking is accepted only if $\xi \leq \xi^*$.

4 NUMERICAL RESULTS

In figure 1, we illustrate the performance of our rejection algorithm by comparing it with three different carparks which operate on a first-come-first-served basis; for this we assume day intervals ($\Delta t = 1$). Figure 1 shows the expected revenues generated in the carparks with varying capacities on day $t = T$. One can see, that the revenue increases with capacity.

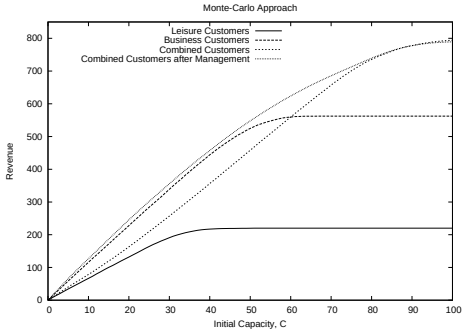

Figure 1: Performance of the rejection algorithm after iterations.

However, as soon as demand is exhausted, no more revenue can be generated. In particular, a carpark that accepts bookings only from the leisure (business) set does not need more than 40 (60) spaces to meet the demand. Under the particular set of parameters, when capacity is large enough, the revenue generated from the business set (560) is significantly higher than that from the leisure set (220). Looking at the combined set we can see that, when capacity is large, the revenue generated equals the sum of the revenues from the leisure set and the business set (780). This, only occurs when capacity is more than 100. However, when capacity is less than 60, accepting only business customers results in greater revenue. The upper line ('Combined Customers after management') presents the expected revenues generated after imposing the proposed rejection algorithm and making optimal decisions based on expected revenues. It is clear that our algorithm outperforms all other carparks for all capacities.

4.1 Monte Carlo Convergence

So far, we have considered a discrete-time model with the timestep being equal to a day, i.e. $\Delta t = 1$. In other words, if there was a booking request to arrive on Wednesday at 22:00 pm and leave on Thursday morning at 08:00 am, the system would reserve a space for the whole day of Wednesday and Thursday and require the customer to pay the daily price for two days, even though the stay would only lasted 10 hours. In the real world, however, a customer might require a space for two and a half days, for ten hours or even for thirty minutes and he would expect to pay the corresponding price.

Thus, our rejection algorithm has to decide whether to accept the booking, according to the availability of spaces for only the particular hours requested and not for the whole day period. This effect can be captured by reducing the time interval in consideration.

Figure 2 shows the convergence of the Monte Carlo for day T with varying time intervals. The maximum daily revenue seems to have converged to around 497. Nonetheless, this convergence is slow, as it is of order $O(\Delta t)$.

The optimal rejection policy (the 2D-matrix $v_{k,j} - v_{k,j-1}$), when $\Delta t = 1/192$, can be seen in figure 3. The figure is smoothed out using 10 iterations of 2000 runs each. There is an increasing pattern in the spaces values, as capacity remaining goes to zero; when capacity is large ($Q > 60$), spaces become worthless for all times, indicating that bookings are accepted irrespective of their length of stay. Furthermore, when there are only few spaces remaining, spaces become more valuable with a maximum added value of around 12 monetary units. By reducing the time interval Δt, we managed to 'squash in' more customers, meaning that we accept more bookings than before; however, such an approach reduces the averaged added values of the spaces.

4.2 PDE Results

Next, we present our results from the continuous-time

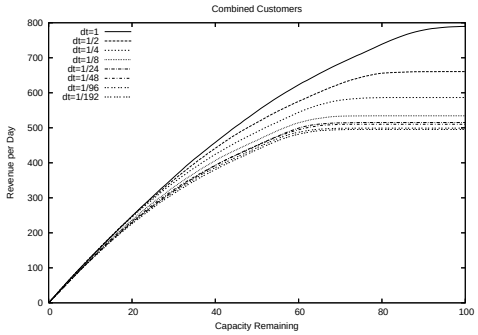

Figure 2: Monte-Carlo Convergence.

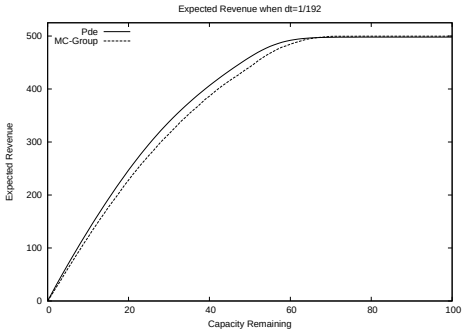

Figure 4: Expected Revenues from PDE and Monte Carlo computations with $\Delta t = 1/192$.

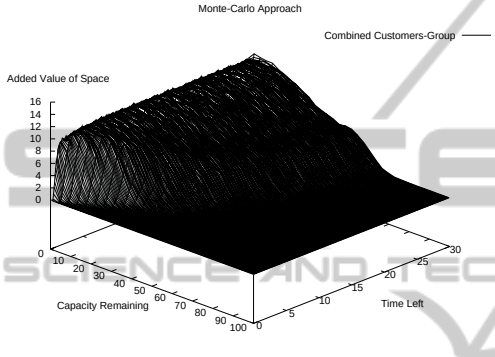

Figure 3: Monte-Carlo Optimal Rejection Policy.

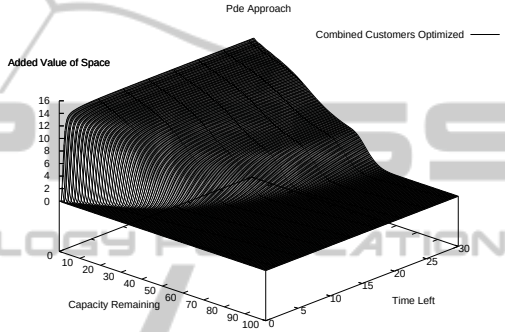

Figure 5: PDE Optimal Rejection Policy.

PDE model and compare them with the Monte-Carlo discrete-time model. In figure 4 we compare the expected revenues generated from the PDE model with those from the discrete model, for day T. When capacity is adequate to meeting the demand from both customer sets, the Monte Carlo simulations seem to have converged to the continuous values. However, for carparks with less than 60 spaces, Monte Carlo approach generates slightly lower revenues. A possible explanation to this lies on the way optimal decisions are made using the rejection algorithm. The decision was made depending upon the total duration of stay (added value of all periods) and not as an individual decision at every period. If the algorithm could treat each period independently we would expect to be able to reject a request and reserve the space for another potentially higher-paying customer. Nevertheless, in our *group*-decision algorithm it happens that a space might be used by a booking that comes and stays for ten time periods because the added value of the nine periods is too much to miss out, that we end up filling up the tenth period too, even though that was not the optimal decision for that period. Therefore, this practice slightly reduces the expected revenues since we tend to slightly accept more long-stay bookings.

The question, however, is to be able to derive optimal rejection policies for the case where group bookings are allowed.

On the one hand, calculating the correct optimal rejection policy using the Monte Carlo approach is computationally intensive as the simulations are based on the number of paths and iterations taken; as a result, the value surfaces produced are not smooth enough (see figure 3) and it takes too long to be found. If we knew the correct value surfaces, derived by avoiding any extensive calculations, we would be able to run the MC-Group with hundreds of thousand paths and based on these surfaces we could make optimal decisions and determine the optimal revenues.

On the other hand, the nature of the continuous PDE model, generates the values quicker and the optimal surfaces are much smoother (see figure 5). This is a fortunate outcome for us, but still we cannot just replace the Monte Carlo approach with the PDE approach, since the latter does not solve the 'full' problem of dealing with group bookings.
Therefore, what would we aim to do now is an attempt to generate optimal revenues using the MC-Group by exploiting the information coming from the PDE.

4.2.1 Proposed Approach

We present the steps to be taken, as a possible approach to the problem:

(i) Solve the PDE and derive the values $V(Q,\tau)$ for all states (Q,τ).

(ii) Find the optimal rejection policy by evaluating the quantities $V(Q,\tau)-V(Q-1,\tau)$.

(iii) Choose a value for Δt and use this policy in the Monte Carlo to make decisions and to optimally manage the group bookings.

(iv) Calculate the effect on the resulting revenues and compare with the revenues obtained without using the PDE policy.

(v) Based on these values, derive the new "updated" rejection policy.

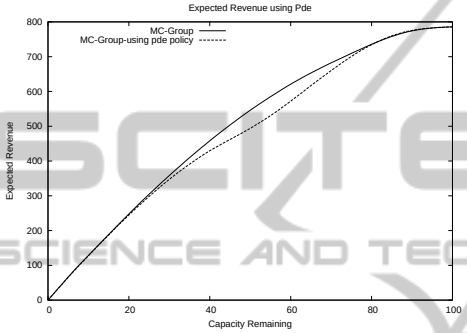

Figure 6: Effect of the PDE rejection policy on Expected Revenues with $dt = 1$.

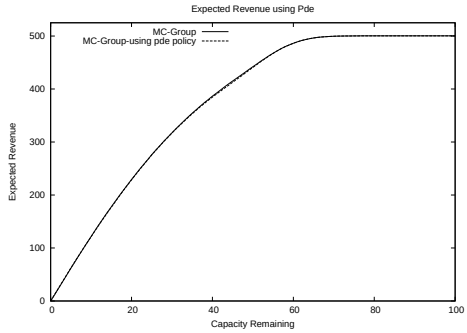

Figure 7: Effect of the PDE rejection policy on Expected Revenues with $\Delta t = 1/192$.

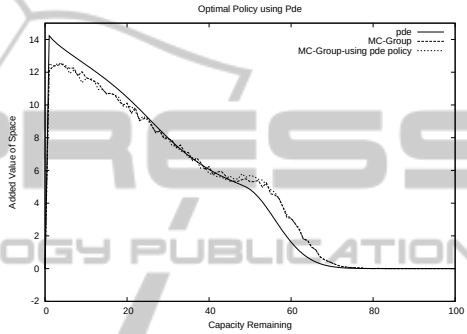

Figure 8: Effect of using the PDE Policy on the MC Policy with $\Delta t = 1/192$.

Figure 6 shows the effect on the expected revenues when the PDE policy is used in the Monte Carlo with the time intervals to be days (i.e. $\Delta t = 1$). The solid line ('MC-Group') shows the expected revenues using the MC-Group algorithm as in figure 2. These values are the optimal ones when $\tau = 30$. Thus, the objective is to minimize the distance between the optimal and the approximated lines. The dashed line ('MC-Group-using PDE policy') shows the expected revenues generated when the intensive simulations are replaced by the procedure described above. The expected revenues with and without using the PDE rejection policy are close to each other. In particular, we find that their difference is always less than 8% at all capacities. However, we believe that the proposed approach will work even better for small Δt, as the PDE policy is derived using the continuous-time model. Figure 7 shows the effect on the expected revenues when the PDE policy is used in the Monte Carlo with the time interval $\Delta t = 1/192$. Clearly, the expected revenues generated are much closer and they are always within 4% of each other.

Figure 8 shows how the rejection policies are formed, before and after using the PDE policy. The dashed line represents ('MC-Group') the optimal rejection policy, the solid line ('pde') is the PDE policy we input in the algorithm, and the dotted line ('MC-Group-using pde policy') is the resulting "updated" rejection policy. We observe that the proposed approach generates a rejection policy that is significantly close to the original optimal one whilst achieving a reduction in the computation time; in particular, the relative difference between the policies never exceed 8%.

Our results imply that even though we have used the PDE optimal policy we could still generate near-optimal revenues. This validates the use of the PDE method for deriving the optimal policy and the system could then use the Monte Carlo approach to make decisions about the 'group' bookings.

5 CONCLUSIONS

The approach followed to solve the discrete-time model required the use of a Monte Carlo scheme. The rejection algorithm was developed to manage bookings according to the future expected revenues in the carpark. Decisions were made optimally and the algorithm has been proven to work well, as it produced greater expected revenues than all unmanaged carparks in consideration. However, the large number of paths and iterations used, slowed down the com-

putation process, and produced an optimal policy that was not sufficiently smooth.

The fact that each state in the carpark can be solved without information or dependence on any other states, has given rise to an equivalent continuous-time PDE approach. The model was developed, based on the probability distributions of the bookings, inter-arrival times as well as the duration of stay. The rejection algorithm was considered not to solve the 'full' problem and, as a result, the expected revenues were slightly different (higher) than the obtained values using the Monte Carlo approach; however, a solution to the problem could be found faster and the resulting rejection policy was much smoother.

Having this in mind, we developed an approach so that the smooth rejection policy from the PDE could be used in the Monte Carlo approach to solve the 'full' problem. Our results are promising, since we managed to replicate the optimal expected revenues with a tolerance of around 8%; a possible further improvement could still be achieved.

One natural extension to the model is to convert it to a dynamic pricing model where demand intensity may be uncertain; it may be time varying ($\lambda = \lambda(t)$) or, to also depend on the pricing function ($\lambda = \lambda(t, \Psi(p))$) - the idea could then be extended to multiple carparks.

REFERENCES

Arnott, R. and Rowse, J. (1999). Modeling parking. *Journal of Urban Economics*, 45(1):97 – 124.

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

Littlewood, K. (1972). Forecasting and control of passenger bookings. *12th Sympos. Proc.*, pages 95–128.

Onieva, L., Muñuzuri, J., Guadix, J., and Cortes, P. (2011). An overview of revenue management in service industries: an application to car parks. *The Service Industries Journal*, 31(1):pp.91–105.

Teodorović, D. and Lučić, P. (2006). Intelligent parking systems. *European Journal of Operational Research*, 175(3):1666 – 1681.

Teodorović, D. and Vukadinović, K. (1998). *Traffic control and Trasport Planning: A Fuzzy Sets and Neural Netweorks Approach.* Kluwer Academic Publishers, Boston.

Verhoef, E., Nijkamp, P., and Rietveld, P. (1995). The economics of regulatory parking policies: The (im)possibilities of parking policies in traffic regulation. *Transportation Research Part A: Policy and Practice*, 29(2):141 – 156.

Vickrey, W. S. (1969). Congestion theory and transport investment. *The American Economic Review*, 59(2):pp. 251–260.

Vikrey, W. S. (1994). Statement to the joint committee on washington, dc, metropolitan problems (with a foreword by richard arnott and marvin kraus). *Journal of Urban Economics*, 36(1):42 – 65.

Wang, L. X. and Mendel, J. (1992). Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6):1414 –1427.

Young, W., Thompson, R. G., and Taylor, M. A. (1991). A review of urban car parking models. *Transport Reviews*, 11(1):63–84.

Zhao, Y., Triantis, K., Teodorovic, D., and Edara, P. (2010). A travel demand management strategy: The downtown space reservation system. *European Journal of Operational Research*, 205(3):584 – 594.