# ON REVENUE MANAGEMENT TECHNIQUES: A CONTINUOUS-TIME APPLICATION TO AIRPORT CARPARKS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2013

By
**Andreas Papayiannis**
School of Mathematics

# Contents

**Abstract** **16**

**Declaration** **17**

**Copyright** **18**

**Acknowledgements** **19**

**Dedication** **21**

**1 Introduction** **22**

1.1 Overview . . . . . 22
1.2 Motivation . . . . . 23
1.3 Outline . . . . . 26

**2 Background theory** **28**

2.1 Stochastic processes . . . . . 28
2.1.1 Point process . . . . . 29
2.1.2 Poisson process . . . . . 30
2.1.3 Wiener process . . . . . 33
2.2 Discrete-time optimisation . . . . . 34
2.2.1 Linear programming . . . . . 34
2.2.2 Duality and complementary slackness . . . . . 36
2.2.3 Integer programming . . . . . 37
2.2.4 Dynamic programming and the Bellman equation . . . . . 38
2.3 Continuous-time optimisation . . . . . 40
2.3.1 Calculus of variations . . . . . 40
2.3.2 Relationship to optimal control theory . . . . . 41
2.3.3 Stochastic optimal control . . . . . 45

**3 RM: Literature review** **49**
3.1 Origins of revenue management and its evolution through time . . 49
3.2 The revenue management system . . . . . . . . . . . . . . . . . . 51
3.2.1 Data collection . . . . . . . . . . . . . . . . . . . . . . . . 51
3.2.2 Estimation and forecasting . . . . . . . . . . . . . . . . . . 51
3.2.3 Optimisation . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.2.4 Booking control . . . . . . . . . . . . . . . . . . . . . . . . 52
3.3 Quantity-based vs price-based RM . . . . . . . . . . . . . . . . . . 53
3.4 Single-resource RM . . . . . . . . . . . . . . . . . . . . . . . . . . 54
3.4.1 Capacity control . . . . . . . . . . . . . . . . . . . . . . . 55
3.4.2 Dynamic pricing . . . . . . . . . . . . . . . . . . . . . . . 60
3.5 Network RM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
3.5.1 Dynamic pricing on the network . . . . . . . . . . . . . . . 65
3.5.2 Network capacity control . . . . . . . . . . . . . . . . . . . 67
3.6 Main approximation methods and related literature . . . . . . . . 70
3.6.1 Approximations based on decomposition . . . . . . . . . . 71
3.6.2 Approximations based on simplified network models . . . . 72
3.6.3 Optimal control in practice . . . . . . . . . . . . . . . . . . 74
3.6.4 Network models in the hotels . . . . . . . . . . . . . . . . 74
3.7 Optimisation strategies for carparks . . . . . . . . . . . . . . . . . 75

**4 Network setting in the jetpark** **78**
4.1 Problem setting . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
4.2 Model assumptions . . . . . . . . . . . . . . . . . . . . . . . . . . 79
4.3 The model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
4.3.1 Structure of the bookings . . . . . . . . . . . . . . . . . . . 82
4.3.2 Generating bookings . . . . . . . . . . . . . . . . . . . . . 83
4.3.3 Bookings and carpark spaces . . . . . . . . . . . . . . . . . 84
4.3.4 Pricing function . . . . . . . . . . . . . . . . . . . . . . . . 85
4.3.5 Carpark revenue . . . . . . . . . . . . . . . . . . . . . . . 87
4.3.6 Booking rejection policy . . . . . . . . . . . . . . . . . . . 88
4.3.7 Rejection policy based on opportunity costs . . . . . . . . 89
4.3.8 Calculating and implementing the opportunity costs . . . . 90
4.3.9 Measuring the effectiveness of capacity allocation . . . . . 92

**5 Developed approximation methods** **94**
5.1 A discrete-time model based on Monte Carlo (MC) . . . . . . . . 97
5.1.1 Numerical scheme . . . . . . . . . . . . . . . . . . . . . . . . 101
5.1.2 Rejection policy algorithm . . . . . . . . . . . . . . . . . . . 101
5.1.3 Successive under-relaxation rejection policy algorithm . . . 103
5.2 Probability Distributions . . . . . . . . . . . . . . . . . . . . . . . . 106
5.2.1 Non-stationary derivations of the distributions . . . . . . . 106
5.2.2 Adjusting for a time-invariant setting . . . . . . . . . . . . 109
5.3 A stochastic PDE model . . . . . . . . . . . . . . . . . . . . . . . . 113
5.3.1 A time-invariant model . . . . . . . . . . . . . . . . . . . . . 116
5.3.2 Rejection policy . . . . . . . . . . . . . . . . . . . . . . . . . 117
5.3.3 Numerical scheme . . . . . . . . . . . . . . . . . . . . . . . . 119
5.3.4 Exhaustive PDE algorithm . . . . . . . . . . . . . . . . . . . 120
5.3.5 Value-policy PDE algorithm . . . . . . . . . . . . . . . . . . 121
5.4 A deterministic model based on Pontryagins maximum principle . 122
5.4.1 Applying the maximum principle . . . . . . . . . . . . . . . 124
5.4.2 Solution of the deterministic problem under the two cases . 125

**6 Numerical Results** **128**
6.1 Monte Carlo Results . . . . . . . . . . . . . . . . . . . . . . . . . . 128
6.1.1 A First-Come-First-Served (FCFS) Policy . . . . . . . . . . 129
6.1.2 Implementing our Monte Carlo (MC) rejection policy . . . 133
6.1.3 Convergence of the MC method . . . . . . . . . . . . . . . . 136
6.2 PDE Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
6.2.1 No optimisation . . . . . . . . . . . . . . . . . . . . . . . . . 144
6.2.2 Optimal rejection policy implemented . . . . . . . . . . . . 145
6.2.3 Further investigation on the rejection policy . . . . . . . . 148
6.2.4 Solution analysis: numerical integrity . . . . . . . . . . . . 154
6.3 Pontryagins deterministic results . . . . . . . . . . . . . . . . . . . 156
6.3.1 Pontryagins vs PDE . . . . . . . . . . . . . . . . . . . . . . . 159
6.3.2 Calculating the expected marginal values from Pontryagins solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
6.4 Performance of the three methods when approximating the network model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
6.4.1 Comparison under false estimates of the booking intensities 170
6.5 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174

**7 PDE model for a finite time interval** **176**
7.1 Model formulation . . . 177
7.2 Numerical scheme . . . 185
7.3 Numerical results . . . 187
7.3.1 Results on the optimal solution of the adjusted PDE . . . 188
7.3.2 Performance of the adjusted PDE in the Network model . 189
7.4 Closing remarks . . . 201

**8 Extensions of the PDE model** **206**
8.1 Hybrid PDE model that adjusts for the network solution. . . . 207
8.1.1 Model formulation . . . 207
8.1.2 Results . . . 213
8.2 Second-order PDE . . . 222
8.2.1 Brownian motion with drift as an approximation to the jump process . . . 223
8.2.2 Optimal rejection condition . . . 225
8.2.3 Numerical scheme . . . 227
8.2.4 Results . . . 230

**9 Conclusions** **234**

**Case study: RM in car supermarkets** **239**
Model formulation . . . 240
Defining the price . . . 240
Defining the time unit . . . 241
Formatting the data . . . 241
Modelling the demand . . . 241
Actual sales and revenue . . . 244
Estimated sales and revenue . . . 245
Numerical results on car model $XXX$ . . . 246
Investigating sensitivity of prices . . . 248
Out-of-sample forecasts . . . 250
Concluding remarks . . . 250

**A Analytical form for $\rho_s(\xi|z)$** **261**

**B Analytical form for $P_s(\xi|z)$** **263**

**C Calculation of $p$ and $r$** **265**

Word Count: 60,032

# List of Tables


6.1 Convergence of the MC approach. Computation times (in seconds). . . . 138
6.2 PDE optimal solution. First order convergence in time dimension for the two numerical schemes. . . . 154
6.3 PDE optimal solution. Computation times (in seconds) of the two numerical schemes for different grids. . . . 155
6.4 Pre-calculation of matrices $p$ and $r$. Computation times (in seconds) for different grids. . . . 155
6.5 Effect on the solution of changing the size of the control step $\Delta\xi$. 156
6.6 Optimal fixed lengths of stays $\xi^*$ for the Pontryagins method, under two numerical schemes. . . . 158
6.7 Calculate the optimal fixed length of stays $\xi^*$. Computation times (in seconds) for the two numerical schemes. . . . 159
6.8 Expected perpetual revenues for a carpark of size C. . . . 168
6.9 Combinations of average booking intensity estimates used to derive the PDE and Pontryagins rejection policies. . . . 171
6.10 Expected perpetual revenues using the PDE rejection policy with wrong initial estimates for the average booking intensities. . . . 172
6.11 Expected perpetual revenues using the Pontryagins rejection policy with wrong initial estimates for the average booking intensities. . 173

7.1 Model parameters . . . . 187
7.2 Adjusted PDE solution. Computation times (in seconds) for different choices of $\Delta T$. . . . 188
7.3 Effect of using discrete time-slots on the expected total demand. . 190
7.4 Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 1$. 202
7.5 Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.5$.203
7.6 Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.25$.204

7.7 Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.125$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205

8.1 Model parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
8.2 Expected perpetual revenues for a carpark of size C. . . . . . . . . . 220
8.3 Second-order PDE solution. First order convergence in time for the implicit scheme and computation times (in seconds) for different grids. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 231
8.4 Second-order PDE solution. Second order convergence in space for the implicit scheme and computation times (in seconds) for different grids. . . . . . . . . . . . . . . . . . . . . . . . . . . . 231

CS1 Sample data example . . . . . . . . . . . . . . . . . . . . . . . . . 242
CS2 Results on estimating the total number of sales . . . . . . . . . . . 248
CS3 Results on estimating the total revenue . . . . . . . . . . . . . . . . 248
CS4 Revenue and sales sensitivity on prices. . . . . . . . . . . . . . . . . 249
CS5 Forecast sales and revenue for October 2011. . . . . . . . . . . . . . 250

# List of Figures


3.1 Schematic overview of a typical revenue management system. . . . 53

4.1 Exponential price-rate function. . . . . . . . . . . . . . . . . . . . . . . . 81
4.2 Booking requests examples for which $f(B^i, k) = 1$. . . . . . . . . . 85
4.3 Price rate per day as a discrete step function. . . . . . . . . . . . 86
4.4 A carpark setting example with three booking requests. The brown customer is allocated one day but the blue and red customers are allocated two days. . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
4.5 Schematic overview of the program flow . . . . . . . . . . . . . . . 91
4.6 Illustration of computing the steady-state equilibrium position. . . 93

5.1 An empty carpark with size $C = 1$. . . . . . . . . . . . . . . . . . . . 100
5.2 A partially filled carpark with size $C = 2$. . . . . . . . . . . . . . . 100
5.3 A semi-filled carpark with size $C = 2$ but only $Q = 1$ spaces available for all days. . . . . . . . . . . . . . . . . . . . . . . . . . . 100
5.4 Schematic overview of the MC procedure . . . . . . . . . . . . . . 105
5.5 Breakdown of bookings for a particular day $t$, with respect to different durations of stay, $\xi$, and various pre-bookings times, $\eta$. . . . 108
5.6 Joint density $\phi(\eta, \xi)$ as a function of the pre-booking time $\eta$ and the length-of-stay $\xi$. . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
5.7 $P_a(z, t), P_d(z, t)$ and $g(z, t)$ for a given day $t$. . . . . . . . . . . . . 112
5.8 Conditional probability and cumulative density of stay, $\rho_s(\xi|z)$ and $P_s(\xi|z)$ on a given day $t$. . . . . . . . . . . . . . . . . . . . . . . . . 114
5.9 $2D$ matrices $p$ and $r$, as functions of the length-of-stay $\xi$ and time left $\tau$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121

6.1 Expected revenue for a carpark on day $T$ with $\tau$ days remaining until $T$, using the FCFS policy. . . . . . . . . . . . . . . . . . . . . . . 130

6.2 Expected revenue for a combined carpark on day $T$ when there are $\tau$ days remaining until $T$, using the FCFS policy. . . . . . . . . . 131
6.3 Expected marginal value of the space on day $T$ for the combined carpark, as a function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), using the FCFS policy. . . . . . . 132
6.4 Expected marginal values for the FCFS policy, as a function of the capacity remaining $Q$ and the time left $\tau$. . . . . . . . . . . . . . . 132
6.5 Expected revenue on day $T$ for carparks operating under a FCFS policy, as a function of the time remaining $\tau$ (upper figures) and capacity remaining $Q$ (lower figures). . . . . . . . . . . . . . . . . . 133
6.6 Expected revenue for the combined carpark on day $T$ with $\tau$ days remaining until $T$, using the MC policy. . . . . . . . . . . . . . . . 134
6.7 Expected marginal value of the space for the combined carpark on day $T$ as function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), using the MC policy. . . . . . . . 135
6.8 Bid-price table with expected marginal values for the MC policy, as a function of capacity remaining $Q$ and time left $\tau$. . . . . . . . 135
6.9 Expected revenue for different carparks on day $T$ as a function of the time remaining $\tau$ (upper figures) and capacity remaining $Q$ (lower figures). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
6.10 Convergence in $\Delta T$ of the expected revenue generated by the MC method. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
6.11 Expected revenues for the MC method as functions of the time remaining $\tau$, after setting $\Delta T \to 0.0125$. . . . . . . . . . . . . . . 139
6.12 Resulting bid-price table with expected marginal values for the MC policy after setting $\Delta T \to 0.0125$. . . . . . . . . . . . . . . . . . . . 140
6.13 Expected marginal values with 50 days left as functions of the capacity remaining $Q$ for different sizes $\Delta T$. . . . . . . . . . . . . 141
6.14 Three carparks that price according to day ($\Delta T = 1$), half-day ($\Delta T = 0.5$) and quarter-of-day ($\Delta T = 0.25$) intervals. The figures show different cases of bookings that require to stay for a length less than $\Delta T$. According to when they arrive, the blue bookings are allocated one period of stay whereas the purple are allocated two. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143

6.15 Expected revenue for a carpark on day $T$ when there are $\tau$ days remaining until $T$, using no optimisation. . . . . . . . . . . . . . . 144
6.16 Expected marginal value of the space for a carpark on day $T$ as function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), with no optimisation in place. . . . . . . . . . 145
6.17 Bid-price table with expected marginal values without optimisation, as a function of capacity remaining $Q$ and time left $\tau$. . . . . 146
6.18 Expected revenues on day $T$ as functions of the time left $\tau$ after implementing the optimal PDE policy. . . . . . . . . . . . . . . . 146
6.19 Expected marginal value of the space on day $T$ as a function of the capacity remaining $Q$ and time left $\tau$, using the optimal PDE policy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
6.20 Bid-price table with expected marginal values from the optimal PDE policy, as a function of capacity remaining $Q$ and time left $\tau$. 147
6.21 Optimal expected marginal values as functions of the capacity remaining $Q$ and time left $\tau$. The left figure corresponds to a carpark with customers from only the leisure class while the right figure to a carpark with customers from only the business class. . . . . . . 149
6.22 Optimal expected marginal values with 50 days left, as functions of the capacity remaining $Q$, for two carparks operating with only leisure customers (dashed line) and only business customers (solid line). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
6.23 Effect on the expected marginal values of varying the staying intensity $\lambda_s$. We consider customers from a single class only, with $\lambda_b = 5$ and $\lambda_a = 1/14$. . . . . . . . . . . . . . . . . . . . . . . . 150
6.24 Optimal expected marginal values for a carpark with customers from both customer classes. We also plot the corresponding business class and the leisure class adjusted 25 units towards the right. 151
6.25 Effect on the expected marginal values of changing the expected total demand. . . . . . . . . . . . . . . . . . . . . . . . . . . 152
6.26 Effect on the expected marginal values of changing the decaying coefficient $\mu$ of the price-rate function. . . . . . . . . . . . . . . . 153
6.27 Effect on the expected marginal values of changing the ratio of the booking intensities of the two customer classes. . . . . . . . . . . 153
6.28 Non-optimised deterministic trajectories for the selling of the spaces. 157

6.29 Optimal deterministic trajectories for the selling of the spaces. . . 157
6.30 Comparison of the expected revenues of the PDE and the Pontryagins method. These are the expected revenues generated in a carpark with $\tau = 50$ days left, plotted as functions of the carpark capacity. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
6.31 Calculation of marginal values through Pontryagins solution. . . . 161
6.32 Bid-price table with expected marginal values derived from the Pontryagins solution. . . . . . . . . . . . . . . . . . . . . . . . . . 162
6.33 Expected perpetual revenue per day to be generated under the rejection policies of the four methods: MC, FCFS, PDE and Pontryagins. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
6.34 Relative comparison of the expected perpetual revenues for the three methods (against the MC). . . . . . . . . . . . . . . . . . . . 165
6.35 Expected marginal values for the three methods, as functions of capacity remaining $Q$, in the perpetual sense. . . . . . . . . . . . 166
6.36 A state of a carpark of size $C = 11$, with a highly congested resource (peak) on day $T$. The brown boxes indicate spaces that have been already sold and the white boxes indicate the remaining available spaces. . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
6.37 Average occupancy rate as function of the carpark size $C$, under the four rejection policies. . . . . . . . . . . . . . . . . . . . . . 170

7.1 Extended region covered by $\tilde{g}(z, t; \Delta T)$ for bookings made at time $t$.178
7.2 The adjusted occupancy probability $\tilde{g}(z; \Delta T)$, for different choices of $\Delta T$. $g(z)$ is also shown. . . . . . . . . . . . . . . . . . . . . . 179
7.3 The adjusted conditional probability density $\tilde{\rho}_s(\xi|z = 2; \Delta T)$ for different choices of $\Delta T$. $\rho_s(\xi|z = 2)$ is also shown. . . . . . . . . 180
7.4 Calculate the expected number of stay periods, $\mathrm{E}[D]$, of the customers that are present between $[z,\, z + \Delta T]$ time after the booking and have length of stay $\xi$. We use a uniform distribution $u(0,\, \Delta T + \xi)$ for the customers conditional arrival times given the specified length of stay $\xi$. . . . . . . . . . . . . . . . . . . . . . . . 181
7.5 Relationship between required length of stay, $\xi$, and expected length of stay, $\mathrm{E}[D]\, \Delta T$, for different choices of $\Delta T$. . . . . . . . . . . . . 183

7.6 Plots of the adjusted price rate function used in the reformulated PDE and the discrete-jump price function used in the MC, for varying choices of $\Delta T$. The continuous pricing function (4.3) is shown as well. . . . . 184
7.7 Bid-price tables with expected marginal values $\Delta\tilde{V}(Q,\tau)$ from the adjusted PDE, for varying choices of $\Delta T$, as functions of the time left $\tau$ and the capacity remaining Q. . . . . 190
7.8 Expected marginal values from the adjusted PDE with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$. . . . . 191
7.9 Expected marginal values from the MC with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$. . . . . 192
7.10 Expected marginal values from the adjusted Pontryagins with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$. . . . . 193
7.11 Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 1$ (against the MC). . . . . 193
7.12 Expected marginal values for the three methods with $\Delta T = 1$, as functions of the capacity remaining $Q$ in the perpetual sense. . . . 194
7.13 Average occupancy rate as function of the carpark size $C$, under the four rejection policies for $\Delta T = 1$. . . . . 197
7.14 Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.5$ (against the MC). . . . . 197
7.15 Expected marginal values for the three methods with $\Delta T = 0.5$, as functions of capacity remaining $Q$ in the perpetual sense. . . . 198
7.16 Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.25$ (against the MC). . . . . 199
7.17 Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.125$ (against the MC). . . . . 199
7.18 Expected marginal values for the three methods with $\Delta T = 0.25$, as functions of capacity remaining $Q$ in the perpetual sense. . . . 200
7.19 Expected marginal values for the three methods with $\Delta T = 0.125$, as functions of capacity remaining $Q$ in the perpetual sense. . . . 200

8.1 Illustration of the days . . . . . 208

8.2 Probability $\hat{g}(z|\tau, t)$ with fixed $\tau$ and $t$, for the two cases $0 \leq z < \tau$ (bottom figure) and $\tau \leq z < \infty$ (top figure). . . . . . . . . . . . . 209
8.3 Possible shape of $\hat{g}(z|\tau, t)$ as function of $z$, for fixed $\tau$ and $t$. . . . 210
8.4 Possible shape of the weighting function $W(z|\tau, t)$ as function of $z$, for fixed $\tau$ and $t$. . . . . . . . . . . . . . . . . . . . . . . . . . . 210
8.5 Effect of choosing an arbitrary optimal sales trajectory, on the resulting state of the carpark around time $\tau$. The top and centre figures show a sub-optimally chosen trajectory that results in relatively higher (top figure) and lower (centre figure) neighboring states. The bottom shape shows the correct optimal sales trajectory that has to implemented. . . . . . . . . . . . . . . . . . . . . . . 214
8.6 Conditional distribution $\hat{g}(z|\tau)$ as function of $z$, for different choices of $\tau$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 215
8.7 Weighting function $W(z|\tau)$ as function of $z$, for different choices of $\tau$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 215
8.8 Bid-price table of the original PDE (upper figure) and of the weighted PDE scheme (lower figure), as functions of the capacity remaining $Q$ and time left $\tau$. . . . . . . . . . . . . . . . . . . . . . . . 217
8.9 Expected marginal values of the weighted PDE for fixed $Q$, as functions of the time left. The different curves represent different choices of $Q$, 1, 5, 10, top to bottom. The marginal values of the original PDE are plotted as well. . . . . . . . . . . . . . . . . . 218
8.10 Relative comparison of the expected perpetual revenues for the original and weighted PDE against the MC, as functions of carpark size $C$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
8.11 Average occupancy rate under the original PDE, the weighted PDE and the MC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
8.12 Relative comparison of the expected perpetual revenues for the original and weighted PDE against the MC, as functions of carpark size $C$. The booking intensity for the leisure class is fixed to $\lambda_{b_1} = 5$. The booking intensity for the business class is lower to $\lambda_{b_2} = 5$ and $\lambda_{b_2} = 2$ in the left and right figures, respectively. . . . . . . . 222
8.13 A Poisson process path with intensity $f(\tau)$ (left figure) and a Brownian motion path with drift term $f(\tau)$ and volatility $\sqrt{f(\tau)}$ (right figure). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 226

8.14 Expected revenues for the original and the second order PDE scheme with 50 days left, as a function of the remaining capacity $Q$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 233
8.15 Comparison of the optimal condition $\Psi(\xi^*)$ as a function of the remaining capacity $Q$, with 50 days left. . . . . . . . . . . . . . . . 233

9.1 States of a carpark with size $C = 5$ under extreme scenarios. The left figure shows a carpark being highly congested on day $T$ but with abundant capacity on its surrounding days, whereas the right figure shows a carpark almost empty on day $T$ but sold out on the neighbouring days. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 237
CS1 Estimating the price-demand function $\lambda(p)$ . . . . . . . . . . . . . . 244
CS2 Car model $XXX$ sales over a two-year period. . . . . . . . . . . . 246
CS3 Average sales intensities (# sales/week) per price-bin $p^k$ for the period 14/12/2009 to 12/12/2011. . . . . . . . . . . . . . . . . . . . 247
CS4 Expected % change in total revenue by shifting the mean of the stock distribution. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 248

# Abstract

This thesis investigates the revenue management (RM) problem encountered in an airport carpark of finite capacity, where the available parking spaces should be sold optimally in advance in order to maximise the revenues on a given day. Customer demand is stochastic, where random pre-booking times and stay lengths overlap with each other, a setting that generates strong inter-dependence among consecutive days and hence leads to a complex network optimisation problem.
Several mathematical models are introduced to approximate the problem; a model based on a discrete-time formulation which is solved using Monte Carlo (MC) simulations and two single-resource models, the first based on a stochastic process and the other on a deterministic one, both developed in continuous-time that lead to a partial differential equation (PDE). The optimisation for the spaces is based on the *expected displacement costs* which are then used in a *bid-price control* mechanism to optimise the value of the carpark.
Numerical tests are conducted to examine the methods' performance under the network setting. Taking into account the methods' efficiency, the computation times and the resulting expected revenues, the stochastic PDE approach is shown to be the preferable method.
Since the pricing structure among operators varies, an adjusted model based on the stochastic PDE is derived in order to facilitate the solution applicable in all settings. Further, for large carparks facing high demand levels, an alternative second-order PDE model is proposed.
Finally, an attempt to incorporate more information about the network structure and the inter-dependence between consecutive days leads to a weighted PDE scheme. Given a customer staying on day $T$, a weighting kernel is introduced to evaluate the conditional probability of stay on a neighbouring day. Then a weighted average is applied on the expected marginal values over all neighbouring days. The weighted PDE scheme shows significant improvement in revenue for small-size carparks. The use of the weighted PDE opens the possibility for new ways to approximate network RM problems and thus motivates further research in this direction.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright

# Acknowledgements

*"As you set out for Ithaka hope the voyage is a long one, full of adventure, full of discovery."*

*Constantinos C. Cavafy, Poet*

Obtaining a PhD is fascinating. Success, however, is defined by your encounters along the entire journey; and mine, although not an easy endeavour, has been absolutely worth it.

As this journey was never truly solo, I would like to express my appreciation to those who have helped reach the destination.

On the substance of the thesis, I firstly need to express my deepest gratitude to my first supervisor Prof. Peter Duck who supported me throughout with his knowledge and invaluable assistance. Most heartfelt thanks go to my second supervisor Dr. Paul Johnson for his outstanding supervision; without his timely advice, guidance and humour, this thesis might never have been completed.

I am also grateful to Dr. Dmitry Yumashev for his significant contribution to the knowledge I gained and the long hours spent discussing several aspects of the problem. I must also thank Prof. Syd Howell and Dr. Nathan Proudlove for their invaluable advice when embarking on this project. Special thanks to Dr. Geoff Evatt for the enthusiasm and motivation he brought to our research group as well as the group outings he organised.

Further, my thanks are extended to my friends: my office mates Mishari Al-Forail, Mingliang Cheng, Wang Li, Hugo Ramirez, Javier Hernández Avalos, James Blair and David Liu for the good times we spent in and out of the office and the priceless relationship established between us, my Cypriot friends in Manchester for the great nights out and especially my flatmate Klitos for his great friendship and help whenever technology failed me.

A person who I am definitely indebted to is my fantastic girlfriend Georgia; I could not have asked for a better companion to make this long road less lonely. I cannot imagine how this thesis would have turned out without her constant love, understanding, faith in me and precious care. Thank you for your patience particularly during the write-up phase when I would retreat to my computer for days.

My boundless appreciation goes to my wonderful parents, Lambros and Maria, for their love, continuous encouragement and unwavering support. Mum and Dad, you are awesome parents, I am forever grateful for everything you have done for me and it only seems right that I dedicate this thesis to you.

# Dedication

To my parents

# Chapter 1

# Introduction

*"Yield management is not a systematic approach to abusing customers. Think of it as a game in which all players can win."*

*Walter J. Relihan III, 1989*

## 1.1 Overview

All businesses, from retailing and grocery stores to hotels and airlines, struggle with the same issues, these are: the increase of competition, increase of special offers and promotions, increase of distribution channels, increase of price transparency and decrease in time to respond and to react. As these factors have a direct effect on a firm's revenues and profits, managers have always been searching for strategies to maximise the profits of their business. This has given rise to *Revenue Management* (RM).

RM addresses three basic categories of demand management decisions; structural, price and quantity decisions (Talluri and van Ryzin, 2005). Structural decisions are responsible for the selling format in use; for instance, whether prices are posted or determined through negotiations and/or auctions. They also determine which segmentation mechanisms are implemented, as well as whether cancellation and refund options are available. Price decisions, on the other hand, are about setting the prices appropriately. Determining factors include time, capacity remaining, the product's lifetime and the type of segment. Finally, quantity decisions regard the allocation of the available capacity to the different segments, products or distribution channels. These decisions are also responsible for a product being withheld from the market and released at a later time.

The theory of demand management decisions and RM in general is not a new idea. In fact, every seller in history has had to make such decisions. However, what is actually new about RM is *the manner* in which these decisions are made nowadays; this involves technologically sophisticated and detailed operational techniques. As such, a vital role in the development of these methods has been the progress in mathematical sciences, economics and statistics as well as the evolution in information technology (Talluri and van Ryzin, 2005).

More precisely, the advances in mathematical sciences enabled operators to model the customer demand, to make an accurate forecast on the market response and to evaluate the uncertainties associated with the decision-making. The progress in information technology enables organisations to compute optimal solutions using complex algorithms to models with many parameters. The new RM systems are not only fast and accurate, they are capable of holding vast amounts of data.

Consequently, modern RM has improved in both the quality and speed of demand management decisions, as the models implemented now are of greater scale (they cover more situations and uncertainties), more accurate (better forecasting techniques) while still being computable in reasonable time.

Queenan et al. (2007) state that RM can potentially improve revenues 3-7% in the airline, hotel and car rental industries. Great examples include American airlines who recorded an additional profit of around $1.4 billion over a two year period, 1989-1991, Marriot hotels who improved their annual revenue in 1991 by $25-30 million, while National Car Rental benefited by around $56 million after one year of implementing a RM system (Baker and Collier, 2003).

## 1.2 Motivation

Over the last twenty years, cars have formed the main transportation system for people worldwide, especially in developed countries. At the same time, large cities have become overcrowded. As a result, parking facilities are an essential component for building up a sustainable transportation system. Given the limited land-use for parking spaces in modern cities, optimal parking pricing can be a flexible and desirable tool for most system planners and regulators (Qian and Rajagopal, 2013). For instance, an appropriate pricing scheme could potentially persuade commuters to use different means of transport, such as buses or trains, effectively reducing traffic congestion during peak hours.

However, in an attempt to bring parking pricing into an alignment with actual market forces, this creates an opportunity for carparking owners to exploit the increased demand to maximise their turnover.

In order for RM techniques to be efficient, carparks need to satisfy a set of conditions (Kimes, 2000):

1. The company provides service based on a fixed capacity. Fixed capacity means that no more customers can be accepted and hence no more revenue can be generated when there are no parking spaces left to sell.

2. Customers can be segmented into distinct groups according to specific characteristics. For example, customers can be differentiated according to their willingness to pay, response to price changes, length of stay, the service required or the pre-booking time. Put simply, when people do not value the product identically, then extra revenue can be made by charging different prices for different customers. The more variation there is among the customers' willingness to pay, the more RM is likely to increase revenue.

3. The inventory is perishable and it can be sold in advance. A perishable inventory has a constant utility up until an expiration date, at which point the utility drops to zero. In carparks, the available parking slots for a given future day $T$ (the day when the slots are to be consumed) have some associated value up until $T$ and become worthless after that day. This time limit of selling the inventory implies that as time passes, the revenue opportunity of an empty parking slot is lost forever; unsold slots on day $T$ cannot be carried out to sell at a later time $t > T$. This naturally leads to selling the inventory *in advance*, which also allows the carpark operators to plan ahead and operate more steadily.

4. The demand fluctuates between different times of the year. For example, the demand for flights during Christmas or summer periods is higher than for the rest of the year; similarly, demand in carparks, located near an airport, is affected accordingly.

5. The company is structured in a way that it faces relatively high fixed costs and low variable costs. In the carparking industry, increasing the size of the carpark requires a huge amount of investment, as opposed to the cost

incurred by completing a sale (accept a request, allocate a space to the customer, update the system).

In this thesis we consider optimising the revenue of an *airport carpark*.

Airport carparks or *jetparks* are by definition located near the airports and operate with customers who plan to fly and thus look for a place to park their car until they return. Most airport carparks maintain an internet pre-booking system where customers can choose to book among different types of jetparks. The two main types are the *premium* and the *long-stay* jetparks. The premium ones are located within a walking distance from the terminals and thus they are the most expensive. On the other hand, people parking at the long-stay jetparks are often required to take the shuttle buses in order to reach to the airport, and therefore these jetparks are less expensive. These long-stay jetparks are usually very large and rarely sell out. In contrast, the premium jetparks are relatively smaller (as they are built close to the airport where the land-space is limited) and managers usually have to re-direct customers to the long stay carparks due to excess demand over scarce capacity. Then, it is precisely in this type of carpark that RM techniques may prove to be beneficial and rewarding.

While parking lot operations share common traits with other prominent RM problems, the academic literature has paid surprisingly little attention to the parking industry. Taking into account the size of big international airports as well as the success stories of American airlines (Smith et al., 1992) and Marriot hotels (Cross, 1997), we believe that RM in jetparks can achieve similar performance. Therefore, the environment we consider is premium jetparks, although the techniques developed could potentially be implemented in different industries also.

Two streams of customers are assumed: The price-sensitive (low-paying) and the time-sensitive (high-paying) customers. The former set of customers is the group of people who tend to book early in advance to take advantage of any promotional discounts or offers. These customers usually require to stay for longer periods that might coincide with a planned holiday trip and thus they will be referred to as the *Leisure* class. The latter set comprises of the customers that are not flexible with times and they are willing to pay a premium in order to receive service. These customers, however, are described by shorter stays on average and thus they will be referred to as the *Business* class. Note that customers from both classes pre-book and arrive in the carpark in random order and can stay for any length of time which is a again random.

Usually, the required length of stay is measured in integer number of days (as opposed to urban city carparks which it is measured in hours or minutes) and thus the corresponding price is quoted as *price rate per day*. Each customer is quoted a price per day depending on the number of days they are staying for. Mathematically, this is expressed through the *pricing function* which relates the quoted daily price rate to the required length of stay.

One may notice that among the range of industries for which RM is applied to, airport carparks are most related to hotels. In particular, the main product that is rented in both cases is a "space" in the facility (room in the hotel, parking spot in the carpark) for a day, although additional revenue opportunities could be achieved in the hotel through other services. Therefore, the major challenges are similar, as the manager needs to address the problem of *multi-day* stays. For instance, what price should be quoted to a customer who wants to arrive on a low-demand day and stay through several high-demand days? To this end, studying some of the main characteristics of hotel RM (regarding the optimisation stage) is important in understanding how we should attempt to tackle our problem.

Clearly the uncertainty in customer demand along with the complexity in dealing with the inter-dependence within the days results in an intellectually stimulating and challenging problem that we aim to address in this thesis.

We develop several mathematical models to approximate the problem. The optimisation for the spaces is based on the *expected displacement costs* - the expected revenue loss incurred by removing one unit from the capacity. Each approach will generate a set of *expected marginal values* which are then used in a *bid-price control* mechanism to optimise the carpark.

## 1.3 Outline

The remainder of this report is organised as follows. Chapter 2 introduces the reader to the main mathematical tools that are used in the academic literature and throughout the thesis. A review of the relevant literature and current RM practices may be seen in chapter 3. We have divided the academic literature into two main categories, the *single-resource* and the *multiple-resource* or *network* RM. In each category, the main models are shown for under both a deterministic and a stochastic setting.

In chapter 4, we set the scene, list the set of assumptions and define the problem we aim to solve. This is a stochastic network setting whereby customers book to stay for any length of time and as such the optimisation of the spaces on one day is inter-connected to the neighbouring days.

The complex structure of the problem necessities the derivation of mathematical methods that aim to approximate the problem. In particular, three methods are developed, a discrete-time model based on Monte-Carlo (MC) simulations and two continuous-time models that lead to a Hamilton-Jacobi-Bellman-type (HJB) partial differential equation (PDE). These methods are either solutions to a simplified network problem or to a decomposed single-resource problem. The methodology underlying these approaches is presented in chapter 5.

Extensive numerical results for each approach, along with performance comparison among them is found in chapter 6. Because of the continuous-time nature of two of the models, the comparison is performed in the limit, i.e. as the time interval goes to zero. Thus, in chapter 7, we present an adjustment to the main PDE model that allows us to obtain solution for any finite time interval considered.

Based on the original PDE, two further extension models are presented in chapter 8. The first suggests to apply some kind of a weighting scheme on the solution of the PDE in order to incorporate more information about the interdependence of the days. The second model assumes continuous capacity and under some assumptions on the demand process it leads to a second order PDE. Results from both schemes are presented and discussed.

Concluding this study, chapter 9 presents an overall summary of the developed mathematical methods as well as a discussion on the main contributions achieved.

As a final note we present a small case study undertaken in collaboration with the industry, as part of a three-month internship during my studies.

*Note: All numerical results and computation times presented, are obtained using an Intel Xeon(R) E450 @ 3.00 GHz processor with 16GB of RAM.*

# Chapter 2

# Background theory

## 2.1 Stochastic processes

For the following definitions in this section we follow Zukerman (2012). For a given sample space $\Omega$, we define the $\sigma$-algebra $\mathcal{F}$ on $\Omega$ as a family of subsets of $\Omega$ with the following characteristics:

- $\emptyset \in \mathcal{F}$
- $F \in \mathcal{F} \implies F^C \in \mathcal{F}$, where $F^C = \Omega \setminus F$ is the complement of $F$ in $\Omega$
- $A_1, A_2, \dots \in \mathcal{F} \implies A = \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$

The pair $(\Omega, \mathcal{F})$ is called a *measurable space*. A probability measure $\mathcal{P}$ on $(\Omega, \mathcal{F})$ is a function $\mathcal{P} : \mathcal{F} \to [0, 1]$ such that

- $\mathcal{P}\ (\emptyset) = 0, \quad \mathcal{P}\ (\Omega) = 1$
- if $A_1, A_2, \dots \in \mathcal{F}$ and $\{A_i\}_{i=1}^{\infty}$ are disjoint subsets then

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

The triple $(\Omega, \mathcal{F}, \mathcal{P})$ is called a *probability space*.

We define a stochastic process $X(t)$, $t \in T$ on a given probability space $(\Omega, \mathcal{F}, \mathcal{P})$ as a collection of random variables in correspondence to the given index set $T$. If the index set $T$ is countable then the process is called *discrete-time* stochastic process, while if it is not countable it is said to be a *continuous-time* stochastic process. Moreover, the stochastic process $X(t)$, $t \in T$ can be *discrete-space* or *continuous-space* according to whether the sample space $\Omega$ is discrete or continuous.

### 2.1.1 Point process

A special class of stochastic processes is called *Point* processes. A Point process is a sequence of events[1] that occur at random points of time $t_i$, $i = 1, 2, \ldots$, with $t_{i+1} > t_i$. The index set $T$ where the $t_i$'s get their values from is usually considered to be the positive real line. A Point process can be defined by its corresponding *Counting Process* $N(t), t \geq 0$, where $N(t)$ is the number of events occurred within $[0, t)$. The following properties hold for the counting process $N(t)$:

1. $N(0) = 0$
2. $N(t) \in \mathrm{Z}$
3. If $s > t$, then $N(s) \geq N(t)$ and $N(s) - N(t)$ is the number of occurrences within the period $(t, s]$.
4. Orderliness:
   The probability that two or more arrivals happen at once is negligible. In other words, we require that
   $$\lim_{\Delta t \to 0} \Pr(N(t + \Delta t) - N(t) > 1 | N(t + \Delta t) - N(t) \geq 1) = 0.$$
5. Markov Property:
   The future evolution of the process is statistically independent from its past.

   Another way to define the point process is through the stochastic process of the interarrival times $\Delta_i = t_{i+1} - t_i$. If we then think of $Y$ as the waiting time between successive events the memoryless property may be expressed as
   $$\Pr(Y > t + s | Y > t) = P(Y > s).$$

Two well known processes that belong to the class of discrete-space Point processes are the discrete-time *Bernoulli process* and the continuous-time *Poisson process.*

[1] In queueing theory these are called arrivals, as they correspond to time points when arrivals join the queue.

### 2.1.2 Poisson process

#### 2.1.2.1 Homogeneous Poisson process

In addition to its basic properties, in order for a counting process $N(t)$ to be defined as a *Poisson process with rate* $\lambda$ we further require that:

1. The number of events in any interval of length $t$ follows a Poisson distribution with rate $\lambda t$. Therefore, the number of events in $[s,t)$, $N(t)-N(s)$ satisfies:
$$\Pr\big[N(t)-N(s)=n\big]=\frac{\lambda^n\,(t-s)^n\,e^{-\lambda\,(t-s)}}{n!}, \tag{2.1}$$
where $\lambda$ is the average number of events in a unit of time.

2. Independent Increments:
The number of events in any disjoint time intervals $[t_1,t_2]$ and $[t_3,t_4]$ are independent random variables.

3. Stationary Increments:
The number of events within the time intervals $[s,t+s]$ and $[0,t]$ follow the same distribution (stationary), i.e.
$N(t+s)-N(s)\overset{\mathrm{D}}{=}N(t)-N(0)$ for $s,t\geq 0$.

4. The inter-arrival times of events are exponentially distributed with rate parameter $\lambda$, such that the average inter-arrival time is $1/\lambda$.

The orderliness property of the Poisson process leads to the following *small interval conditions*:

1. $\Pr\big[N(t+\Delta t)-N(t)=0\big]=e^{-\lambda\Delta t}\approx 1-\lambda\Delta t+o(\Delta t)$
2. $\Pr\big[N(t+\Delta t)-N(t)=1\big]=\lambda\,\Delta t\,e^{-\lambda\Delta t}\approx\lambda\Delta t+o(\Delta t)$
3. $\Pr\big[N(t+\Delta t)-N(t)\geq 2\big]=o(\Delta t)$,

where any function $g(.)$ is $o(\Delta t)$ if

$$\lim_{\Delta t\to 0}\frac{g(\Delta t)}{\Delta t}=0.$$

Note that the probability of an event happening within an interval is proportional to the length of the interval. In particular, any process that satisfies the stationary, independence and small interval conditions together with $N(0) = 0$ *is* a Poisson process.

Alternatively, one may write the Poisson process in its differential form, that is

$$dN = \lim_{\Delta t \to dt} N(t + \Delta t) - N(t), \tag{2.2}$$

and thus express the state dynamics as

$$dN = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt. \end{cases} \tag{2.3}$$

#### 2.1.2.2 Non-homogeneous Poisson process

A non-homogeneous Poisson process is a Poisson process with time varying intensity rate, $\lambda(t)$, satisfying the independent increment property but *not* the stationary increment property. Therefore, the number of events in $[s, t)$, satisfies:

$$\Pr\big[N(t) - N(s) = n\big] = \frac{(\Lambda(t) - \Lambda(s))^n \; e^{-(\Lambda(t) - \Lambda(s))}}{n!}, \tag{2.4}$$

where

$$\Lambda(t) = \int_0^t \lambda(x)\, dx \tag{2.5}$$

is the number of events expected to happen within $[0, t)$.

Then, the orderliness property of the non-homogeneous Poisson process leads to the following *small interval conditions*:

1. $\Pr\big[N(t + \Delta t) - N(t) = 0\big] \approx 1 - \lambda(t)\Delta t + o(\Delta t)$
2. $\Pr\big[N(t + \Delta t) - N(t) = 1\big] \approx \lambda(t)\Delta t + o(\Delta t)$
3. $\Pr\big[N(t + \Delta t) - N(t) \geq 2\big] = o(\Delta t)$.

Note that the homogeneous Poisson process with constant rate $\lambda$ is a special case of the inhomogeneous Poisson process, where $\Lambda(t)$ reduces to $\lambda t$.

Alternatively, according to Gallager (1996) we may consider a non-homogeneous Poisson process as a homogeneous Poisson process *over a non-linear time scale.* In particular, if $\tilde{N}(t)$ is a homogeneous Poisson process with rate 1 then the inhomogeneous Poisson process may be expressed as $N(t) = \tilde{N}(\Lambda(t))$.

#### 2.1.2.3 Properties of Poisson processes

- Superposition
  The superposition of two inhomogeneous processes with intensities $\lambda_1(t)$ and $\lambda_2(t)$ results in an inhomogeneous Poison process with intensity $\lambda(t) = \lambda_1(t) + \lambda_2(t)$.

- Random selection
  A random selection of events from an inhomogeneous Poisson process with intensity $\lambda(t)$, such that each event is independently selected with probability $p(t)$, leads to an inhomogeneous process with intensity $p(t)\,\lambda(t)$.

- Subdivision
  If an inhomogeneous Poisson process with intensity $\lambda(t)$ is subdivided into two processes with probability $p_1(t)$ and $p_2(t)$ respectively, then the resulting subprocesses are independent inhomogeneous Poisson processes with intensities $p_1(t)\,\lambda(t)$ and $p_2(t)\,\lambda(t)$, respectively.

- Conditioning on the number
  If the total number of arrivals in the interval $[0, t)$ is $n$, then the event times are distributed independently in this interval according to the density function $\lambda(t)/\int_0^t \lambda(x)dx$.

#### 2.1.2.4 Generating the Poisson process

We first begin by describing the way to generate event times of a homogeneous Poisson process with rate $\lambda$. The approach used is based on the *Inverse Transformation* method (see Çinlar, 1975). From the properties of the Poisson process we know that the time between events is exponentially distributed with rate parameter $\lambda$. Then, the cumulative density function (cdf) of the waiting time is given by

$$H_x(t) = \int_0^t \lambda\, e^{-\lambda s}\, ds = 1 - e^{-\lambda t}, \tag{2.6}$$

and the inverse cdf by

$$H^{-1}(u) = -\frac{\log(1-u)}{\lambda}, \qquad 0 \le u < 1. \tag{2.7}$$

Therefore, if $u \sim U(0, 1)$ is a uniform random variable in $[0, 1)$, the next event occurs at time

$$t = -\frac{\log(1-u)}{\lambda} = -\frac{\log(u)}{\lambda}. \tag{2.8}$$

For the case of generating event times of a non-homogeneous process with intensity $\lambda(t)$ the procedure is slightly more complex. The available methods can be grouped into three main categories: inversion methods, order-statistic methods and acceptance-rejection methods. Here, we only describe the method we have implemented for our problem; this falls into the first category and it is named *Çinlars* method.

According to Çinlar (1975) we first generate event times from a homogeneous Poisson process of rate 1 and then use $\Lambda(\cdot)$ to obtain the event times for the required non-homogeneous Poisson process. More precisely, the procedure to find the next event time $t$ is

1. Draw a random number $u \sim U(0,1)$
2. Generate an event from the homogeneous Poisson process with $\lambda = 1$ by

$$s = -\log(u).$$

3. Generate $t$ according to,

$$t = \inf\{v : \Lambda(v) \geq s\},$$

or

$$t = \Lambda^{-1}(s),$$

if the function $\Lambda(\cdot)$ is invertible.

### 2.1.3 Wiener process

Another class of stochastic processes is called a *Wiener process* or so-called *Brownian motion*. The Brownian motion $W(t)$ is a continuous-time continuous-space stochastic process and has the following properties:

1. $W(0) = 0$
2. It has stationary increments, i.e.
   $W(t+s) - W(s) \overset{\mathrm{D}}{=} W(t) - W(0)$ for $s, t \geq 0$.

3. It has independent increments, i.e.
   for $t_1 < t_2 < \cdots < t_n$ the random variables $W(t_n) - W(t_{n-1}), \ldots, W(t_1) - W(t_0)$ are all independent.

4. The map $t \to W(t)$ is continuous

5. $W(t) - W(s) \sim N(0, t-s)$, where $N(\cdot)$ stands for the Normal distribution.

6. It satisfies the Markov Property.

The differential of a Brownian motion $dW$ can be defined as

$$dW = \lim_{\Delta t \to dt} W(t + \Delta t) - W(t). \tag{2.9}$$

In particular, $dW$ is normally distributed with mean 0 and variance $dt$, and it is often expressed as

$$dW = \epsilon\sqrt{dt},$$

where $\epsilon \sim N(0, 1)$.

## 2.2 Discrete-time optimisation

Optimisation focuses on determining a maximum or a minimum of a given function over a region. Since this thesis refers to revenue optimisation and maximising profitability, we focus on finding the value that maximises our objective function. If there are constraints imposed regarding the possible values of certain variables of the function, the problem becomes a *constrained* optimisation problem; if no constraints are in place the problem is an *unconstrained* optimisation problem.

In RM most practitioners use a class of methods that falls into the broad area of *Mathematical Programming*. Usually, the function's variables represent the different products offered by the firm and the objective is to maximise the profit generated by selling these products subject to some capacity constraints. One such method, which is perhaps the most popular and forms the basis for the other mathematical programs, is called *Linear Programming*.

### 2.2.1 Linear programming

As the name indicates, Linear Programming (LP) applies to linear objective functions. The objective function is a linear combination of the $n$ variables subject to $m$ linear constraints that set an upper or a lower bound on a linear combination of the $n$ variables. LP is widely used in practice, mainly because it is intuitively appealing and computationally efficient (simple).

For illustrative purposes, let us consider a firm with $n$ products $y_1, \ldots, y_n$ that are priced at $c_1, \ldots, c_n$ respectively. Each product $y_j$ is made of a linear combination of the $m$ materials (resources). Each material $i$, $i = 1, \ldots, m$ has available capacity $x_i$. The objective is to decide how many units of each product to build so that the total revenue is maximised. A restriction however applies, as the capacity of each resource is finite.

Let $a_{ij}$ indicate the number of units of material $i$ used to build product $j$. Then, the linear program can be expressed as:

$$\max_{y_1,\ldots,y_n} \sum_{j=1}^{n} c_j y_j,$$

subject to

$$\sum_{i=1}^{n} a_{ij} y_j \leq x_i \quad \text{for } i = 1, 2, ..., m.$$

$$y_j \geq 0 \qquad \text{for } j = 1, 2, ..., n.$$

Note that we have $m$ inequality constraints, each associated with one resource. Also, we impose the restriction that the $y_j$ is either 0 (no units of product $j$ are built) or a positive number.

More conveniently, we may rewrite the LP in its *canonical* form, that is

$$\max_{\mathbf{y}} \mathbf{c}^\top \mathbf{y}, \tag{2.10}$$

subject to

$$\mathbf{A}\mathbf{y} \leq \mathbf{x},$$

$$\mathbf{y} \geq 0,$$

where $\mathbf{c}^\top = (c_1, c_2, \ldots, c_n)$, $\mathbf{y}^\top = (y_1, y_2, \ldots, y_n)$, $\mathbf{x}^\top = (x_1, x_2, \ldots x_m)$ and $\mathbf{A} = [a_{ij}]_{m \times n}$ is the matrix consisting of all the elements $a_{ij}$.

In general, when formulating a problem as an LP we are free to choose between minimising or maximising the objective function, or to use any of $\leq, =, \geq$ to describe the constraints for the resources or the variables values. However, it is always best to write the LP in *standard* form, where the constraints are all equalities while the variables are all non-negative.

### 2.2.2 Duality and complementary slackness

The linear program given in (2.10) is often referred to as the *primal* problem. For each primal problem there is an associated *dual* problem, which is another linear program that takes the form:

$$\min_{\mathbf{b}} \mathbf{x}^\top \mathbf{b}, \tag{2.11}$$

subject to

$$\mathbf{A}^\top \mathbf{b} \geq \mathbf{c},$$

$$\mathbf{b} \geq 0,$$

where $\mathbf{b} = (b_1, b_2 \ldots b_m)$ is the vector of *dual* variables. We note that the dual problem is a *minimisation* problem with a number of variables equal to the number of constraints in the primal (2.10); there is a 1-1 correspondence between the dual variables and the primal constraints.

The dual value $bi$ at the optimum measures the dependence of the primal objective function to the constraint $i$. In particular, $b_i$ is the rate of change of the objective function with respect to the constraint (or resource) $i$ when all remaining constraints (or resources) are kept fixed.

The *Complementary Slackness condition* formalises the above (Phillips, 2005):

*The incremental change in the objective function from an incremental change in constraint $i$ in (2.10) will be (approximately) $b_i$. If constraint $i$ is* non-binding[2]*, then $b_i = 0$ and relaxing the constraint will not change the value of the objective function; if $b_i > 0$, then constraint $i$ in (2.10) is* binding.

In the above example, if $b_i = 2$ for instance, this implies that an extra unit of material $i$ will produce approximately an additional 2 units in revenue, as long as the optimal solution remains unchanged; similarly, $b_i = 0$ implies that increasing the capacity of material $i$ will not bring any additional revenue.

In the RM context, knowing the solution to the corresponding dual problem is vital for decision making, the reason being that each of the resulting dual variable

[2]A constraint $i$ (can be either an inequality or an equation) is said to be binding if, at the optimal solution, it is satisfied by an equation between the variables of the LHS and the resource capacity $x_i$. Otherwise, the constraint is non-binding.

is interpreted as the *marginal value* of the corresponding material. A material that consistently has a high marginal value would be an excellent candidate to be considered for ordering it in bigger volumes whereas one with consistently zero marginal values should be ordered in lower volumes.

Fortunately enough, the complementary slackness conditions enable us to compute the dual values out of the primal solution directly, without actually having to solve the dual problem explicitly.

### 2.2.3 Integer programming

In a typical revenue optimisation problem, most of the variables of interest are discrete rather than continuous; for example, the number of bookings, no-shows and demand are all integers. This implies that we have to use *Integer Programming* (IP); we need to find all values $y$ of the vector in (2.10) but requiring that these values have to be integers.

The restriction that the feasible region consists of only integers renders the problem relatively more challenging than a problem where the feasible region is continuous. Thus, researchers often look for ways to approximate the IP problem rather than solving it directly. The most common approximation method is to solve the LP problem that results by relaxing the integrality constraint. This LP-relaxation technique transforms the *NP-hard*[3] optimisation problem to one that can be solved in a *polynomial time*[4]. Therefore, researchers ususally solve the relaxed LP problem and then based on the solution they design algorithms to approximate the IP problem. The two most popular algorithms used in practice are the *Cutting Plane (Gomori) method* and the *Branch and Bound method*.

For a detailed study of linear and integer programming techniques the reader is referred to Luenberger (2003) and Schrijver (1998).

[3]NP-hard stands for the *non-deterministic polynomial-time hard* and, in brief, it refers to a class of problems that can be solved in a polynomial time only by a non-deterministic machine. In a polynomial time, a deterministic machine, in contrast, can only *verify* whether the solution is correct. Further information on NP-hard problems can be found in Hochbaum (1996).

[4]A problem is said to be solvable in polynomial time if the time it takes to be solved is upper bounded by a polynomial expression in the size of the input of the problem, i.e. $T(k) = O(k^r)$, where $r$ is a non-negative integer.

### 2.2.4 Dynamic programming and the Bellman equation

Another technique that is employed in practice is *Dynamic programming* (DP). Dynamic Programming is a technique used to obtain optimal decisions to a certain problem, when these are made in stages. The objective is to maximise any desirable outcome; mathematically, this is expressed through the *reward function*. A key element of these decisions is that they cannot be viewed in isolation; in fact, one "must balance the desire of high present rewards with the undesirability of low future rewards". Thus, the idea is to break the decision sequence in just two parts, the present reward and the expected future reward that encapsulates the decisions from all subsequent stages. Then, the optimal decision is the one that maximises the sum of these two components.

Following Bertsekas (1995) we consider a finite horizon discrete-time system with $T$ stages, $0, 1, \ldots, T-1$. The state of the system at time $t+1$ is given by $x_{t+1}$ and it is a function of the state at $t$, $x_t$, as well as the corresponding action at $t$, $u_t$, that is

$$x_{t+1} = f_t(x_t, u_t) \quad t = 0, 1, \ldots, T-1.$$

In this representation, $x_t$ belongs to the state space $S_t$ and $u_t$ belongs to the control set $U_t(x_t)$ which itself depends on the current state $x_t$. Further, we assume that the process is Markov in the sense that all the information relevant to the determination of the probability distribution of future values is accommodated in the current state $x_t$.

Define $R_t(x_t, u_t)$ as the *current reward* at time $t$ when the state is $x_t$ and the action chosen is $u_t$. We further impose the restriction that the reward function is *additive over time*. In other words, the reward incurred at time $t$, $R_t(x_t, u_t)$ accumulates over time. Let $R_T(x_T)$ to be the *terminal reward* at the end of the process which only depends on the state of the process (independent of any action chosen). Then, the *total reward over all times* is given by,

$$R_T(x_T) + \sum_{t=0}^{T-1} R_t(x_t, u_t). \tag{2.12}$$

Let the control law $\pi$ to be the sequence of functions, $\pi = (\mu_0, \mu_1, \ldots, \mu_{T-1})$, such that $\mu_t$ maps state $x_t$ onto the control $u_t = \mu_t(x_t)$ and such that $\mu_t(x_t) \in U_t(x_t)$. Then, we may introduce $J_t^\pi(x_t)$ as the *total reward at time t over the* *remaining stages $t, t+1, \ldots, T$ given the control law $\pi$.* In fact, this is

$$J_t^\pi(x_t) = R_T(x_T) + \sum_{k=t}^{T-1} R_k(x_k, u_k) \quad \forall\, t = 0, 1, \ldots, T-1. \tag{2.13}$$

Finally, the *value function* $V_t(x_t)$ is defined as

$$V_t(x_t) = \max_{\pi \in \Pi} J_t^\pi(x_t), \tag{2.14}$$

with the terminal value

$$V_T(x_T) = R_T(x_T). \tag{2.15}$$

where $\Pi$ is the set of all admissible policies. Consequently, the value function $V_0(x_0)$ is the *maximum total reward over all times* if we are in state $x_0$ initially. The policy that leads to $V_0(x_0)$ is denoted by $\pi^* = (\mu_0^*, \mu_1^*, \ldots, \mu_{T-1}^*)$.

The *Bellman Principle* or the *Principle of Optimality* can be expressed as:

*Let $\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{T-1}^*\}$ be an optimal policy for the basic problem and assume that when using $\pi^*$, a given state $x_t$ occurs at time $t$ with positive probability. Consider the subproblem that starts at state $x_t$ at time $t$ and we wish to maximise the total reward from time $t$ to $T$. Then, the truncated policy $\pi^* = \{\mu_t^*, \mu_{t+1}^* \ldots, \mu_{T-1}^*\}$ is the optimal policy for the subproblem.*

In other words, an optimal policy has the property that, whatever the initial action, the remaining choices constitute an optimal policy with respect to the subproblem starting at the state that results from the initial actions.

Thus, the Bellman principle enables us to compute the rewards *recursively.* In particular, we can show that the value function satisfies

$$V_t(x_t) = \max_{u_t \in U_t(x_t)} \left\{ R_t(x_t, u_t) + V_{t+1}(f_t(x_t, u_t)) \right\} \quad \forall\, t = 0, 1, \ldots, T-1. \tag{2.16}$$

Equation (2.16) is solved backwards in time, with the final condition

$$V_T(x_T) = R_T(x_T). \tag{2.17}$$

Such computations are highly efficient because at any time point along the recursion one needs to store only two values, the current reward and the remaining value function, as opposed to the entire set of values from all subsequent times.

## 2.3 Continuous-time optimisation

We can now consider maximising a continuous, differentiable function $f(x)$ over a specified region $a \leq x \leq b$. We note that $x$ can be scalar or even a vector. In general, a function might have several local maxima within a region. We assume, however, that the function in consideration is unimodal i.e. it has the property that any local maximum is a global maximum as well. Continuous optimisation suggests that we start from a particular point in the region and look for another point that gives a higher objective function value. The process continues until we find a point where no further improvement can be found. Such a point is called the *global maximiser*. Writing the above statement in a formal manner we have that for a point $x^*$ to be a maximiser, all partial derivatives of the function at that point are zero, i.e. $\partial f(x^*)/\partial x_i^* = 0 \quad \forall i$. If, however, $\partial f(x^*)/\partial x_i^* > 0$ or $\partial f(x^*)/\partial x_i^* < 0$ for some $i$, then the function $f$ could be increased by increasing $x_i^*$ or decreasing $x_i^*$, respectively.

We can extend the nature of $x : [0, T] \to \mathbb{R}$ to be a function in its own right. Then, the objective becomes to optimise a *functional* as opposed to a function. There are three major approaches to solve such continuous optimisation problems, these are the *calculus of variations*, the *optimal control theory* and the *dynamic programming*. The notes below are based on Chiang (1992).

### 2.3.1 Calculus of variations

In calculus of variations we seek to maximise an objective functional $J(x(t))$, where $x : [0, T] \to \mathbb{R}$. In other words, we seek to find the function $x$ for which $J(x(t))$ is maximised. The fundamental problem in calculus of variations reads,

$$\max \; J(x(t)) = \int_0^T R(x(t), x'(t), t)\, dt \tag{2.18}$$

subject to

$$\begin{aligned} x(0) &= x_0 \\ x(T) &= x_T \end{aligned} \tag{2.19}$$

where $x_0, x_T \in \mathbb{R}$. The integrand $R$ is called the *reward rate* which depends on the time $t \in [0, T]$, the state $x$ and the direction of path $x'(t)$. Further, it is assumed that all the functions that appear in the problem are continuous and continuously differentiable. Note that in the fundamental problem, the initial and terminal points are completely specified. Often, there is a terminal point criterion $h(x(T))$ that sets the value of the objective functional at time $T$. Thus, this term is usually added to equation (2.18).

One can show that the optimal solution to the above problem satisfies the *Euler-Lagrange* equation, namely

$$\frac{\partial R}{\partial x} - \frac{d}{dt}\left(\frac{\partial R}{\partial x'}\right) = 0. \tag{2.20}$$

which in general is a second order non-linear differential equation[5].

Note that equation (2.20) is a necessary condition for finding the optimum $x(t)$ but it does not tell us whether it maximises or minimises $J$.

Despite sitting at the core of functional analysis, calculus of variations has two major limitations that led researchers studying methods to overcome them; it only deals with differentiable functions and interior solutions. One field that overcomes these is *Optimal Control theory*, which is, in fact, the main tool implemented throughout this thesis.

### 2.3.2 Relationship to optimal control theory

As in the calculus of variations, the objective is to maximise the reward functional, but in optimal control theory we achieve this by the use of a control variable $u(t) \in U$, where $U \subset \mathbb{R}$ is the control set. This control variable is measurable in $[0, T]$ and is directly related to the state variable $x$ through the state dynamics.

In particular, the fundamental problem in optimal control theory reads

$$\max_{u \in U} J = h(x(T)) + \int_0^T R(x(t), u(t), t)\, dt \tag{2.21}$$

[5]This may be seen after expanding the second term to obtain $\frac{d}{dt}\left(\frac{\partial R}{\partial x'}\right) = \frac{\partial}{\partial t}\left(\frac{\partial R}{\partial x'}\right) + \frac{\partial}{\partial x}\left(\frac{\partial R}{\partial x}\right)\frac{dx}{dt} + \frac{\partial}{\partial x'}\left(\frac{\partial R}{\partial x'}\right)\frac{dx'}{dt}$.

subject to

$$\begin{aligned} x'(t) &= f(x(t), u(t), t), \\ x(0) &= x_0, \\ x(T) &= x_T, \end{aligned} \tag{2.22}$$

where $x_0, x_T \in \mathbb{R}$, and $f : \mathbb{R} \times U \times [0, T] \to \mathbb{R}$. We notice that $x'(t)$ has been replaced by the control variable $u(t)$ inside the functional. The presence of the control variable $u$ necessitates a linkage between $u$ and $x$, to describe the effect of $u$ on the movement of $x$; this relationship is given through the constraint equation. Also note that when we set $u = x'(t)$, optimal control theory reduces precisely to the calculus of variations.

The advantage of optimal control theory is that the state path does have to be continuous but only piecewise differentiable. At the same time the optimal control path only needs to be piecewise continuous allowing for jump discontinuities. Another advantage is that the control set $U$ can be restricted, which will naturally admits corner (boundary) solutions.

Now, consider $s \in [0, T]$ and define $J(x(s), u(s), s)$ as the *total reward remaining from times* $[s, T]$ when the control policy is $u$, where $u$ is measurable in $[s, T]$. This may be written as,

$$J(x(s), u(s), s) = h(x(T)) + \int_s^T R(x(t), u(t), t)\, dt. \tag{2.23}$$

Then we can define the value function $V(x(s), s)$ as the *maximum total reward remaining from times* $[s, T]$, i.e.

$$V(x(s), s) = \max_{u \in U} J(x(s), u(s), s). \tag{2.24}$$

There are two approaches to solve this deterministic optimal control problem; dynamic programming and the Pontryagins maximum principle.

**Dynamic programming** We begin by illustrating the dynamic programming approach which is in fact a direct extension of the dynamic programming in the discrete setting.

Consider a "small" increment of size $\Delta t$. Then, we can apply the Bellman's principle of optimality on (2.24) to obtain the following recurrence relation:

$$V(x(t),t) = \max_{u \in U} \left\{ R(x(t),u(t),t)\,\Delta t + V(x(t)+\Delta x(t), t+\Delta t) \right\} + o(\Delta t). \tag{2.25}$$

Applying a Taylor series expansion to $V(x(t)+\Delta x(t), t+\Delta t)$ we obtain,

$$\begin{aligned} & V(x(t)+\Delta x(t), t+\Delta t) \\ &\qquad = V(x(t),t) + \frac{\partial V(x(t),t)}{\partial t}\Delta t + \frac{\partial V(x(t),t)}{\partial x}\Delta x(t) + o(\Delta t) \\ &\qquad \approx V(x(t),t) + \frac{\partial V(x(t),t)}{\partial t}\Delta t + \frac{\partial V(x(t),t)}{\partial x} f(x(t),u(t),t)\Delta t. \end{aligned} \tag{2.26}$$

Substituting (2.26) into (2.25), dividing by $\Delta t$ and then taking the limit as $\Delta t \to 0$ we obtain a partial differential equation (PDE), the *Hamilton-Jacobi-Bellman (HJB)* equation,

$$-\frac{\partial V(x(t),t)}{\partial t} = \max_{u \in U} \left\{ R(x(t),u(t),t) + \frac{\partial V(x(t),t)}{\partial x} f(x(t),u(t),t) \right\}, \tag{2.27}$$

solved backwards in time, starting with the final condition

$$V(x(T),T) = h(x(T)). \tag{2.28}$$

The HJB equation is in general a non-linear PDE due to the min/max operator with respect to the value function $V$. Note that in order to solve this PDE we need to make some assumptions mainly on the behaviour of the value function $V$; that is a smooth function and continuously differentiable in $t$ and $x$. If so, how do we the know that the solution of the HJB is the original value function? It turns out that if we can solve the HJB equation then we can obtain an optimal control policy that maximises the right-hand side. Then the solution to the HJB is precisely the value function in the original problem (2.24). This idea is more formally expressed as the *Verification* theorem (see Bertsekas, 1995).

However, in some cases the smoothness assumptions might not hold and thus it is not possible to find classical solutions to satisfy the HJB equation everywhere. Then, the optimal value function is only a weak solution to the HJB. Thus, to solve the optimal control problem one might have to refer to non-smooth analysis and the notion of *viscosity solutions*. This research area is out of the scope of our study, however, the interested reader is referred to Øksendal and Sulem (2007).

**Pontryagins maximum principle** As pointed out before, the objective is to find the optimal control trajectory $u^*(t)$ (and hence the optimal path $x^*(t)$) that maximises $J$. What we then need is a way to compare the different trajectories of alternative controls. This is achieved by the use of the Hamiltonian $\mathcal{H}$.

Let us define the Hamiltonian $\mathcal{H}$ as

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = R(x(t), u(t), t) + \lambda(t) f(x(t), u(t), t), \tag{2.29}$$

where $\lambda(t)$ is the *costate variable* defined by

$$\frac{d\lambda}{dt} = -\frac{\partial \mathcal{H}}{\partial x}. \tag{2.30}$$

Equation (2.30) is referred to as the *equation of motion for* $\lambda$. Similarly, from (2.29) we also obtain that

$$\frac{dx}{dt} = \frac{\partial \mathcal{H}}{\partial \lambda}, \tag{2.31}$$

which is defined as the *equation of motion for* $x$.

Pontryagins maximum principle states that if the optimal control trajectory $u^*(t)$ maximises $J$ (i.e gives rise to the value function $V$) then it also maximises $\mathcal{H}$.

The variable $\lambda$ is also called the *Lagrange multiplier* of the constraint $dx(t) = f(x(t), u(t), t)\, dt$. In fact, it satisfies

$$\lambda = \frac{\partial V(x(t), t)}{\partial x} \tag{2.32}$$

and, therefore, the Hamiltonian is just the expression inside the brackets of equation (2.27).

Finally, the problem requires solving the following set of differential equations

$$\begin{aligned} u(t) &= \arg\max_u \mathcal{H}(x(t), u(t), \lambda(t), t) \\ \frac{d\lambda}{dt} &= -\frac{\partial \mathcal{H}(x(t), u(t), \lambda(t), t)}{\partial x} \\ \frac{dx}{dt} &= \frac{\partial \mathcal{H}(x(t), u(t), \lambda(t), t)}{\partial \lambda}. \end{aligned}$$

Depending on the problem we attempt to solve, an additional condition might be needed. This condition sets the value of $\lambda$ at the terminal time $T$, and it is known as the *transversality condition*. The above problem can actually arise with many variations regarding the terminal times and states imposed. Below, we go through the cases that we will encounter in this thesis.

If the problem in question has fixed a terminal time $T$ but *free* terminal state $x(T)$ then the condition

$$\lambda(T) = 0$$

has to be satisfied as well.

For the problem stated in (2.21) the terminal time as well as the terminal state are fixed. In such a case, the transversality condition is replaced by the terminal condition

$$x(T) = x_T.$$

However, our formulation in section 5.4 will naturally lead to an optimal control problem where the terminal time is fixed and the terminal state $x(T)$ is free to vary only subject to $x_T \geq x_{\min}$, where $x_{min}$ is a given minimum permissible level for $x$. In this case, the transversality condition can be shown to satisfy

$$\lambda(T) \geq 0 \quad x_T \geq x_{\min} \quad (x_T - x_{\min})\lambda(T) = 0.$$

Although quite complicated at first sight, in practice one could assume a free terminal state to try $\lambda(T) = 0$ condition first and check whether the resulting $x_T^*$ value satisfies the terminal restriction $x_T \geq x_{\min}$. If not, we can set $x_T^* = x_{\min}$ and instead treat the problem as one with a fixed terminal state.

Finally, in order to apply Pontryagins maximum principle to an optimal control problem we first need to solve the equations of motion for $\lambda$ and $x$ to obtain general solutions and then use the transversality condition to derive the particular solution for the optimal trajectories $\lambda^*$ and $x^*$. Then, we can evaluate the optimal trajectory $u^*$ that maximises $\mathcal{H}$, and thus $J$.

### 2.3.3 Stochastic optimal control

In a stochastic setting the state variable $x$ is a stochastic process. Therefore, the objective becomes to maximise the *expectation* of a reward functional. In other words, $J(x(s), u(s), s)$ is the *total expected reward remaining from times* $[s, T]$, namely

$$J(x(s), u(s), s) = \mathrm{E}\left[h(x(T)) + \int_s^T R(x(t), u(t), t)\, dt\right] \tag{2.33}$$

and, therefore, the value function $V(x(s), s)$ is now defined as the *maximum total expected reward remaining from times* $[s, T]$,

$$V(x(s), s) = \max_{u \in U} J(x(s), u(s), s) \tag{2.34}$$

In order to illustrate this, we consider two types of stochastic processes, in their general form, and summarise some key features that arise. The approach we use is dynamic programming which in continuous-time leads to the HJB equation.

#### 2.3.3.1 Jump process

Suppose that $x(t)$ is a stochastic differential equation (SDE) of the form

$$dx(t) = f(x(t), u(t), t)dt + g(x(t), u(t), t)dQ, \tag{2.35}$$

where $Q$ is a Poisson process with time varying intensity $\lambda(t)$, $f : \mathbb{R} \times \mathrm{U} \times [0, T] \to \mathbb{R}$ and $g : \mathbb{R} \times \mathrm{U} \times [0, T] \to \mathbb{R}$. In particular, in a period $dt$ the Poisson process $Q$ changes by

$$dQ = \begin{cases} 0 & \text{with probability } 1 - \lambda(t)dt \\ H & \text{with probability } \lambda(t)dt \end{cases} \tag{2.36}$$

where $H$ is the size of the jump.

Now, the associated HJB equation is given by

$$\begin{aligned} -\frac{\partial V(x,t)}{\partial t} = \max_u \Bigg\{ & R(x,u,t) + \frac{\partial V(x,t)}{\partial x} f(x,u,t) \\ & + \lambda(t)\Big(V(x + g(x,u,t)H\,, u, t) - V(x,t)\Big)\Bigg\}, \end{aligned} \tag{2.37}$$

with

$$V(x(T), T) = h(x(T)). \tag{2.38}$$

The last term in (2.37) accounts for the expected rate of change of the value function with respect to the jump size $H$, and it results when applying Itô's formula with jumps and taking expectations, i.e.

$$\begin{aligned}
\mathrm{E}[dV(x(t),t)] &= \mathrm{E}\left[\frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,dx\right] \\
&= \mathrm{E}\Bigg[\frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,f(x,u,t)\,dt \\
&\qquad + \Big(V(x+g(x,u,t)H\,,u,t) - V(x,t)\Big)\,dN\Bigg]. \qquad (2.39)
\end{aligned}$$

Add and subtract the term $\lambda(t)\Big(V(x+g(x,u,t)H\,,u,t) - V(x,t)\Big)\,dt$ to obtain

$$\begin{aligned}
\mathrm{E}[dV(x(t),t)] &= \mathrm{E}\Bigg[\frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,f(x,u,t)\,dt \\
&\qquad + \lambda(t)\Big(V(x+g(x,u,t)H\,,u,t) - V(x,t)\Big)\,dt \\
&\qquad + \Big(V(x+g(x,u,t)H\,,u,t) - V(x,t)\Big)\Big(dN - \lambda(t)dt\Big)\Bigg] \\
&= \frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,f(x,u,t)\,dt \\
&\qquad + \lambda(t)\Big(V(x+g(x,u,t)H\,,u,t) - V(x,t)\Big)\,dt, \qquad (2.40)
\end{aligned}$$

as $dN - \lambda(t)dt$ is a martingale with expected value of zero (a *Compensated* Poisson process).

Alternatively, we can compute the expectation of $dN$ directly, using its definition in equation (2.3), and hence deduce the same result.

#### 2.3.3.2 Itô process

Suppose that $x(t)$ is an *Itô Process*, an SDE of the form

$$dx(t) = f(x(t),u(t),t)dt + g(x(t),u(t),t)dW \qquad (2.41)$$

where $dW$ is a Wiener process, $f:\mathbb{R}\times\mathrm{U}\times[0,T]\to\mathbb{R}$ and $g:\mathbb{R}\times\mathrm{U}\times[0,T]\to\mathbb{R}$. The first term describes the deterministic change of $x$ with time and it is called the *drift term*, whereas the second term incorporates the stochastic behaviour of $x$, usually referred to as the *variance term*.

Then, the HJB equation is now defined as a second order non-linear PDE which is given by

$$-\frac{\partial V(x,t)}{\partial t} = \max_u \left\{ R(x,u,t) + \frac{\partial V(x,t)}{\partial x} f(x,u,t) + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2} g(x,u,t)^2 \right\}, \tag{2.42}$$

with

$$V(x(T),T) = h(x(T)), \tag{2.43}$$

where we suppress the arguments of the functions $x(t)$ and $u(t)$ for clarity.

The presence of a second order term in (2.42) is a direct consequence of applying Itô's formula on $V(x(t),t)$. In particular,

$$\begin{aligned} \mathrm{E}[dV(x(t),t)] &= \mathrm{E}\left[\frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,dx + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}\,dx^2\right] \\ &= \frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,f(x,u,t)\,dt + \mathrm{E}\left[\frac{\partial V(x,t)}{\partial x}\,g(x,u,t)\,dW\right] \\ &\quad + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}\,g(x,u,t)^2\,dt \\ &= \frac{\partial V(x,t)}{\partial t}\,dt + \frac{\partial V(x,t)}{\partial x}\,f(x,u,t)\,dt + \frac{1}{2}\frac{\partial^2 V(x,t)}{\partial x^2}\,g(x,u,t)^2\,dt, \end{aligned} \tag{2.44}$$

where the last equation is derived when noting that the expected value of the Wiener process $dW$ is zero. For an in-depth analysis on stochastic differential equations the reader is referred to Øksendal (1995) and Øksendal and Sulem (2007).

As the problem we consider in this thesis is stochastic, equations (2.37) and (2.42) will be used to derive two models: the main PDE model in section 5.3 which is based on the Poisson process and one of the extension models in section 8.2 based on the Wiener process.

# Chapter 3

# RM: Literature review

Some of the contents of this chapter are based on excerpts from Talluri and van Ryzin (2005) and Phillips (2005), which offer a comprehensive overview of the main models developed for RM.

## 3.1 Origins of revenue management and its evolution through time

Revenue management techniques were first conceived by American Airlines in around 1960. According to McGill and van Ryzin (1999), during these early stages, passenger cancellations and no-shows had been the first task that airline managers had to deal with. Thus, almost all research on reservations control focused on the so-called overbooking control. By this, we mean the practice to increase the total volume of sales taking into account cancellations and no-shows. This technique, however, comes at its own cost; it runs the risk of denying service to customers and facing the resulting legal and regulatory issues. Since overbooking control was based only on predictions about the probability distributions of passengers who did turn-up to travel, an accurate forecast on the number of cancellations and no-shows was needed. As a result, overbooking control on the one hand and forecasting on the other were the primary areas of research within the decade 1970-80. Some of the early work belongs to Rothstein (1971) who presented a model for determining airline overbooking policies.

Shortly after 1970, discount fares were offered for the first time. However, these fares were identical for all carriers and set by the Civil Aviation Board.

The discount fares had the restriction of reserving travel at least three weeks in advance, in order to drive in customers with low willingness to pay. This new strategy can be characterised as the first attempt to segment the market according to price. The problem now was to decide the optimal number of seats to be offered at the discount rate, as too many low-fare seats imply less full-fare available seats, and thus, less revenue.

In 1972, Littlewood (1972) proposed a solution method for a two-fare seat-inventory control problem. His scheme suggests equating the marginal revenues in each of the two fare classes and rejecting a low fare customer if the resulting revenue from the sale is less than the expected revenue of selling that same seat at the higher fare. McGill and van Ryzin (1999) in their review paper state that Littlewood's seat inventory rule can be considered as the beginning of a new approach, so-called *Yield management*[1].

Yield management, however, faced its greatest growth after the Airline Deregulation Act of 1978 (see Chiang et al., 2007). More precisely, the pricing restrictions imposed by the Civil Aviation Board on airlines were then reduced and, as a consequence, this pricing freedom led to a rapid development of new, more sophisticated models.

During the decade 1990-2000 the increasing interest in revenue management became evident in many other industries which operate with perishable assets. As a result, Weatherford and Bodily (1992) introduced the term *Perishable Asset Revenue Management* (PARM) which aimed to generalise the science of yield management to other industries (such as hotels, car rentals). Within this decade dynamic pricing strategies soon gained attention by many researchers (such as Gallego and van Ryzin, 1994; Bitran and Mondschein, 1997). What is more, models soon became more sophisticated and capable of modelling larger problems with multiple-resources, the so-called network models.

Since 2000, another important aspect of the problem has been studied, this is the risk associated with the decisions which until then has not been addressed. Relevant work is found in Lai and Ng (2005); Liu et al. (2006). Today most research focuses upon customer-choice models, where customers are assumed to act strategically when purchasing a product. A detailed review on customer-choice modelling can be found in Shen and Su (2007).

[1]This particular name was given as an indication that the goal of managing bookings in these early stage was to increase *yield*- the average amount one passenger pays to fly one mile.

Today, this field of research is generally known as *Revenue Management* (RM) and it may applied to any type of industry. According to Chiang et al. (2007) a potential path for RM in the future is to be applied in non-traditional industries such as internet providers, IT services or cellular network services. Finally, of great importance is also the study of the impact of competition and collaboration into a business.

Several review papers on the evolution of RM techniques are available. Perhaps the earliest one belongs to Weatherford and Bodily (1992), while more recent reviews include McGill and van Ryzin (1999), Pak and Piersma (2002), Bitran and Caldentey (2003) and Chiang et al. (2007).

## 3.2 The revenue management system

There are four steps that a RM system has to follow:

### 3.2.1 Data collection

Collect and store relevant historical data such as demand, prices, costs, cancellations or no-shows etc. The primary source of data in most RM systems is transactional databases. Reservation databases are the most widely used transactional databases in the hotel and airline industry, and they appear in two formats: either as a total number of bookings in a class or as a customer booking record which is information about each individual booking.

### 3.2.2 Estimation and forecasting

Estimate the parameters of the demand model, forecast the demand based on the estimated parameters, forecast the proportion of cancellations and no-shows; the role of these processes is critical in every RM system. The extent to which this can be realised is investigated by Pölt (1998); he estimates that a revenue management system could produce a 1% additional increase in revenue if the forecasting error is reduced by 20%. Below are the main types of forecasting methods:

1. Ad-hoc methods (moving averages, exponential smoothing)

2. "Pick-up" methods (additive or multiplicative)

3. Time-series and regression models
4. Bayesian methods and machine learning
5. Combined methods (weighted average of historical and advanced booking forecasts)

For an in-depth review on forecasting methods the reader is referred to Weatherford and Kimes (2003).

Often, demand is modelled by a Poisson process because of the memoryless property that characterises the inter-arrival times. However, improved models suggest the use of Compound Poisson processes that reflect real world data more accurately. Such models can be seen in Gallego and van Ryzin (1994), Lautenbacher and Stidham (1999) and Zhao (1999).

### 3.2.3 Optimisation

When the estimation process and modelling has been completed, the task is to determine the optimal set of controls to be applied, until the next re-optimisation. These controls usually deal with setting the optimal prices (markdowns, discounts) or specifying the optimal inventory allocations to each product. The main techniques used are dynamic programming or mathematical programming techniques.

### 3.2.4 Booking control

The last component of the RM system is to implement the optimised control in order to control the sale of the inventory. Usually, the optimisation task is performed on a slightly more simplified version of the problem, in order to increase computation speed. Therefore, the optimisation task can only form the basis for the design of the booking control, rather than simply being used directly in the decision making. In this way, more complicated controls are constructed which can deal with the real problem more accurately.

Figure 3.1 illustrates a schematic overview of a typical RM system with all the relevant stages. The data is collected and is sent to the estimation and forecasting department for analysis. Once the forecasts are ready, the optimisation task is

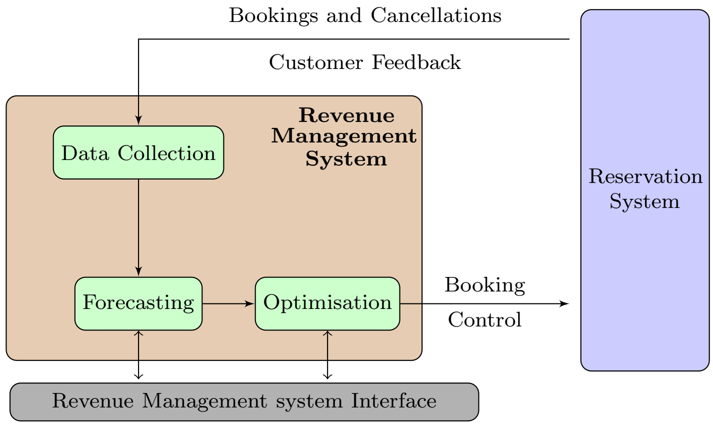



Figure 3.1: Schematic overview of a typical revenue management system.

performed based on these forecasts, which guides the reservation system in making an accept/deny decision on the booking.

In this thesis we focus on the Optimisation and Booking control stage. In other words, we assume that the estimation/forecasting stage would have been carried out and all relevant probability distributions for the customer demand are available.

## 3.3 Quantity-based vs price-based RM

There are two types of RM techniques; *Quantity*-Based and *Price*-Based.

Quantity-based RM uses capacity-allocation decisions as the primary tool to manage demand. In particular, it decides which products should be on offer and which to be closed during the booking process in such a way that revenues are maximised.

In price-based RM, the control of customer demand is achieved by varying prices appropriately over time. In other words, the prices are not pre-determined and thus a change in the posted price has a direct effect on the customer demand. Specifically, an increase in price reduces the volume of customers that will be willing to buy. There is not a single correct approach between the two as the choice of RM model depends upon the nature of the product on offer. For example, airlines might prefer pre-fixed prices for the different fares and focus on the optimal seat allocation, while in retailing the price becomes the main control variable for the demand.

On the one hand, Gallego and van Ryzin (1997) argue that price-based RM might be preferred, as rising the prices a firm can achieve the same reduction in sales but increase revenues at the same time. On the other hand, such price changes might be costly in practice and also might have a negative effect on customer goodwill.

As stated in Talluri and van Ryzin (2005) it is sometimes not clear how to distinguish between the two categories. A customer is "turned away" if either the price of the product is higher than his *reservation price*[2] - the maximum amount he is willing to spend for the product - in which case he decides not to buy, or if the product is withdrawn from sale in which case he does not get the opportunity to buy. The first scenario falls under the price-based where the second scenario under the quantity-based RM.

Nonetheless, whichever type of RM technique is chosen the available models can be divided into two main categories, single-resource or network models, which are explained in the following sections.

## 3.4 Single-resource RM

*Single-resource* refers to the task of optimally allocating capacity of a single resource to different classes of demand. For example, in hotels this refers to controlling the allocation of a single type of room for a given date among the different rate classes and in airlines it refers to controlling the allocation of a single type of seat on a given flight-leg among the different fare classes.

Despite the fact that in reality most RM problems (including ours) are *network* problems in which customers would require a bundle of resources (i.e. a number of consecutive nights in a hotel or two connecting flights), we find it useful to study the single-resource case, since it is often convenient to regard the network problem as a collection of single-resource problems. In fact, when there is one-to-one correspondence between capacity and final product and the demands for products are independent, then, the network problem reduces to a set of disconnected single-resource problems (to be explained in later sections).

[2]A reservation price is the maximum amount of money a customer is willing to pay for a particular product. Customers buy the product only if their reservation price is greater than the product's price. The reservation price is usually modelled by a continuous probability distribution over the population of potential customers (see Bitran and Mondschein, 1997).

### 3.4.1 Capacity control

Let us assume that a firm operates by selling its capacity to $N$ distinct classes with class 1 to be the highest, 2 to be the second highest and class $N$ the lowest that require the same resource. For example, if the firm is a hotel then the classes refer to different discount room rates that are accompanied with different restrictions. Further, we assume that each customer demands a single unit out of the resource's capacity. Thus, the problem is to optimally allocate the capacity of the resource to the $N$ classes.

#### 3.4.1.1 Types of capacity control

There are three types of capacity controls mainly employed in practice:

(1) Booking limits

These are controls that limit the amount of products that can be sold to a given class. In particular, the lowest (highest) booking limit corresponds to the lowest (highest) customer class. Booking limits can either allocate the available capacity into distinct blocks such that demand from one class has only access to its allocated capacity (partitioned allocation), or allow for overlapping between classes whereby the higher classes have access to all the capacity reserved for lower classes (nested allocation). In reality when demand is stochastic, nested booking limits are often preferred over the partitioned ones. The limitation of partitioning the capacity into separate allocations is that excess demand of a high class is thrown away when the allocated capacity of that class is sold out, even though there might still be available capacity for lower classes.

(2) Protection levels

These are controls that set the amount of capacity to be reserved for a particular class or set of classes. In particular, a protection level for for the $n^{th}$ class indicates the amount of capacity to be reserved for class $n, n-1, n-2, \ldots 1$. The protection levels may also be either partitioned or nested. In particular, for the nested case, the booking limit for class $n$, $b_n$, is related to the protection level for class $n-1$, $y_{n-1}$, by

$$b_n = C - y_{n-1} \quad \forall n = 2, 3, \ldots, N \,,$$

where $C$ is the initial capacity.

(3) Bid prices
The key difference between these controls over the former ones is that they are revenue-based rather than class-based. More precisely, bid-price controls set a threshold price (that usually depends on capacity or time), according to which only requests whose revenues exceed this price are accepted. In general, bid prices are simpler to implement as they only require storage of a single threshold value at each point rather than a set of booking limits or protection levels for each class.

#### 3.4.1.2 Displacement cost intuition

Optimal capacity controls usually rely on complex mathematical procedures and scientific algorithms. However, the overall idea is simple and intuitive - the manager should accept a request, and thus give out one unit of capacity, only if the revenue received is greater than or equal to the value of the capacity itself. If we know the exact value of the capacity then the decision becomes trivial. But how do we measure the value of the capacity? Well, this is captured through its (expected) *displacement cost* - the expected revenue we would lose if we had one less unit of capacity available. This concept naturally leads to the idea of the *value function* $V(x)$ which accounts for the optimal expected revenue to be generated with $x$ units of capacity available to sell. Then, the displacement cost is just the difference $V(x) - V(x-1)$.

The displacement cost idea plays a central role in our study and therefore it will be examined in detail in the later chapters. Usually alternative[3] names might be used such as *opportunity cost* or *expected marginal values* and so for our purposes we will be using these interchangeably.

#### 3.4.1.3 Static models

In single-resource capacity control there are two solution methods, static and dynamic. Static models assume that the demand arrives in non-overlapping intervals, starting from the class that corresponds to the lowest price. They further assume that demands for different classes are independent random variables, and

[3]To be precise these terms are not exactly equal to one another. However, they all encapsulate the main idea of where bid-prices should be based upon.

they do not depend on any capacity controls, or on the availability of other classes. Moreover, in the static model, demand arrives in an *aggregate* amount and the task is to decide how much of this amount to accept[4]. Although static models ignore the stochastic evolution of the booking process, customers often exhibit a buying behaviour that justifies this simplification. For instance, in the airlines, low paying customers tend to book well in advance whereas business travellers book just before the departure of the flight; a further assumption that these time periods do not overlap enable the static formulation for the problem.

As mentioned earlier, the first static single-resource model dates back to the work of Littlewood (1972). The proposed scheme assumes two fare classes with prices $p_1 > p_2$. The demand for class $n$ is denoted by $\lambda_n$. Demand for class 2 is realised first, and thus the question is how much demand of class 2 we should accept before knowing the demand for class 1. Suppose that we have $x$ units of capacity remaining and a request from class 2 arrives. The firm is faced with two options: either accept the request and receive $p_2$ amount of money or reject the request and hope that the demand for class 1 will fill up the place, i.e. $\lambda_1 \geq x$. In this case, the expected marginal value is $p_1 P(\lambda_1 \geq x)$. Therefore, it makes sense to accept the request from class 2 if

$$p_2 \geq p_1 P(\lambda_1 \geq x). \tag{3.1}$$

Note that *Littelwood's rule* is an optimal condition for the two-fare problem.

During the next decade, Belobaba (1987) and Belobaba (1989) extended Littlewood's rule to multiple nested fare classes, out of which the well known heuristics *EMSR-a* and *EMSR-b* have been developed. However, when multiple classes are considered, Littlewood's rule is no longer optimal. Fortunately, optimal policies for the multiple fare classes have been derived in Curry (1990), Wollmer (1992), Brumelle and McGill (1993) and Robinson (1995), even though many airlines still prefer the heuristic approximations due to their simplicity.

#### 3.4.1.4 A general dynamic model

Unlike static models, in dynamic models the demand for different classes can arrive in a random order while all remaining assumptions are retained. For a

[4] In real life, demand arrives sequentially over time and the control decision has to be taken based only on the observable demand by that time.

discrete-time formulation we need to impose the extra assumption that arrivals should be Markovian and that at most one arrival occurs per period to make them tractable. For a continuous-time formulation the most common way is to assume that demand follows a Poisson process.

Building upon the previous setting, suppose that the $N$ classes have associated prices $p_1 \geq p_2 \geq ... \geq p_N$. Assuming a discrete-time setting we consider $T$ total time periods and in each period $t$ demand is permitted to arrive from any class $n$. Now let $\lambda_n(t)$ denote the probability of an arrival of class $n$ in period $t$. The time interval between two periods is taken to be sufficiently small such that the orderliness property (at most one arrival per period) is maintained, that is

$$\sum_{n=1}^{N} \lambda_n(t) \leq 1.$$

Let $R(t)$ be a random variable satisfying the following;

$$R(t) = \begin{cases} p_n & \text{if demand for class } n \text{ arrives in period } t \\ 0 & \text{if otherwise.} \end{cases} \tag{3.2}$$

Then, we have that $P(R(t) = p_n) = \lambda_n(t)$. Next define the control variable $u$ to be 1 if we accept the booking and 0 if we do not. Finally, define $V_t(x)$ the value function with $x$ units of capacity as of time $t$.

The optimal value of $u$ should maximise the current revenue plus the revenue to go, that is

$$R(t)u + V_{t+1}(x-u).$$

As a result we obtain the following Bellman[5] equation

$$\begin{aligned} V_t(x) &= E\left[\max_{u\in\{0,1\}} \Big\{R(t)u + V_{t+1}(x-u)\Big\}\right] \\ &= E\left[\max_{u\in\{0,1\}} \Big\{R(t)u + V_{t+1}(x-u) + V_{t+1}(x) - V_{t+1}(x)\Big\}\right] \\ &= V_{t+1}(x) + E\left[\max_{u\in\{0,1\}} \Big\{\left(R(t) - \Delta V_{t+1}(x)\right)u\Big\}\right], \end{aligned} \tag{3.3}$$

[5]Note that the maximisation term appears inside the expectation, unlike common dynamic programs where the order is reversed. This is because we assume that the random variable $R(t)$ is realised *before* we need to make the optimal decision.

where

$$\Delta V_{t+1}(x) = V_{t+1}(x) - V_{t+1}(x-1), \quad \forall x > 0$$

is the expected marginal value of the $x^{th}$ unit of space in period $t+1$ and the boundary conditions are given by,

$$V_{T+1}(x) = 0 \quad \forall\, x = 0, 1, .., C\,,$$

with

$$V_t(0) = 0 \quad \forall\, t = 1, .., T.$$

**Optimal policy** For every capacity remaining $x$ and time period $t$ the following two properties hold for the value function:

(a) Inventory Monotonicity: $\Delta V_t(x+1) \leq \Delta V_t(x)$,
meaning that at any given point in time the expected marginal value decreases as the capacity remaining increases.

(b) Time Monotonicity: $\Delta V_{t+1}(x) \leq \Delta V_t(x)$,
meaning that at a given capacity level the expected marginal value decreases as time passes.

**Optimal control** Finally, when a request for the fare class $n$ arrives at $t$, the optimal control is to accept it only if:

$$p_n \geq \Delta V_{t+1}(x). \tag{3.4}$$

The optimal control can be implemented in three ways: as time-dependent nested protection levels

$$y_n^*(t) = \max\Big\{x : p_{n+1} < \Delta V_{t+1}(x)\Big\},$$

as time-dependent nested booking limits

$$b_n^*(t) = C - y_{n-1}^*(t),$$

or as a bid-price control $\pi(t, x)$ with the bid price equal to the marginal value

$$\pi(t, x) = \Delta V_t(x).$$

The dynamic model presented here formed the basis for most researchers when solving the stochastic single-resource capacity-allocation problem. For the airlines, Lee and Hersh (1993) prove both the monotonicity properties of the value function. Further, they allow for multiple-seat bookings and show that only the time monotonicity is still satisfied. Subramanian et al. (1999) extends the work of Lee and Hersh (1993) to include overbooking, cancellations and no-shows. In the hotel industry, Bitran and Mondschein (1995) worked under the above setting and they too derived the monotonicity properties of the value function. Similarly, Bitran and Gilbert (1996) extended the problem by incorporating no-shows and cancellations, while Koide and Ishii (2005) also account for overbooking and early discount but not no-shows.

Although, most researchers worked under the discrete-time environment we notice the work of Bitran and Gilbert (1996), Liang (1999), Zhao and Zheng (2001) and Feng and Xiao (2001) who studied the problem in continuous-time. In particular, Liang (1999) reformulates and solves the problem proposed by Lee and Hersh (1993) and uses heavyside functions to express the optimal control, while Zhao and Zheng (2001) modifies the problem such that fares may be closed down but if so they cannot reopen.

Finally, a detailed comparison between dynamic and static models can be found in Lautenbacher and Stidham (1999). By showing that both models can be expressed as dynamic programs, the authors present a unified framework as a Markov decision process to link the two. Properties of the value function are then derived based on results from the queueing-control theory.

### 3.4.2 Dynamic pricing

We now describe some models that explicitly use the price as the main tool to influence the customer demand; this method requires an explicit relationhip between quoted price and customer demand.

#### 3.4.2.1 Demand function

*Demand functions* or *price-demand functions* describe the relationship between the price of a product and the corresponding demand for that price. In other words, for a given price, demand functions show how many customers are willing to purchase the product.

If we define $p$ to be the price of the product then $\lambda(p)$ represents the demand that corresponds to this price. If we denote the set of feasible prices the product can take by $\Omega_p$, then the domain of the demand function is $\Omega_p \in (0, +\infty)$. We further impose some regularity assumptions in the demand function. According to Talluri and van Ryzin (2005) we require:

- The demand function is continuously differentiable on $\Omega_p$.
- The demand function is strictly decreasing, i.e. $\lambda'(p) < 0$ on $\Omega_p$.
- The demand function is bounded above and below,

$$0 \leq \lambda(p) < \infty \quad \forall p \in \Omega_p.$$

- For high prices, the demand function tends to zero, i.e. $\inf\limits_{p \in \Omega_p} \lambda(p) = 0$.

#### 3.4.2.2 Deterministic dynamic pricing

In this subsection, we present a general single resource dynamic pricing problem introduced in Gallego and van Ryzin (1994). The problem is formulated in continuous-time to match up with our current study. For the deterministic model presented here, we assume that the demand intensity $\lambda(p, t)$ depends both on time and price. Alternatively, we may treat the intensity as the control variable and thus the instantaneous price rate as a function of the demand, that is $p(t, \lambda(t))$. We can then consider a firm which sells $x$ units of a product over a finite time horizon [0, $T$], such that the salvage value from any remaining inventory at time $T$ is zero. Then, the problem is to maximise revenues by monitoring the sequence of prices (or equivalently the flow of demand) throughout the entire selling horizon, i.e.

$$\max_{\lambda(t)} \int_0^T \lambda(t)\, p(t, \lambda(t))\, dt \tag{3.5}$$

such that

$$\int_0^T \lambda(t)\, dt \leq x \tag{3.6}$$

$$\lambda(t) \in \Omega_\lambda, \tag{3.7}$$

where $\Omega_\lambda$ is the set of allowable demand rates.

When demand is time-homogeneous ($\lambda(p,t) = \lambda(p)$) one can show that the optimal price $p^*$ to charge (or equivalently the optimal demand intensity) is a *fixed price*. To characterise this solution let $p^r = \arg\max\{p\,\lambda(p) : p \geq 0\}$ to be the price that maximises the revenue rate, that is the *revenue maximising* price and $p^0$ the price at which we sell exactly $x$ units by time $T$, the *run-out* price. Then one can show that $p^*$ satisfies:

$$p^* = \begin{cases} p^r & \text{if } \lambda(p^r)\,T \leq x \text{ (abundant capacity)} \\ p^0 & \text{if } \lambda(p^r)\,T > x \text{ (scarce capacity).} \end{cases} \tag{3.8}$$

Often, researchers prefer to work with deterministic models as they are simple and easy to analyse. However the importance of these models is broader. In particular, they are used to derive bounds and to provide insights on how different parameters influence optimal pricing policies. In fact, they have been shown to be asymptotically optimal for the stochastic problem (see Gallego and van Ryzin, 1994).

In the context of deterministic models Smith and Achabal (1998) extend the demand intensity to depend also on the available capacity, while Bass (1969) assumes that demand is affected by the amount sold by that time and also by the population size.

#### 3.4.2.3 Stochastic dynamic pricing

For the stochastic formulation of these problems most researchers assume Markovian arrivals for the demand process and they assume that an estimate of the arrival process is available, often called the *booking curve*. In particular, the demand is assumed to follow a Poisson process with intensity $\lambda$ such that in a sufficiently small time interval $\Delta t$ there is at most one sale that happens with probability $\lambda\Delta t$. Let $N(t)$ to be the number of items sold up to time $t$ and note that demand is realised at time $t$ if $dN(t) = 1$. Then the problem of maximising the total expected revenues over $[0,\, T]$ becomes,

$$V(x,t) = \max_u E_u \left[ \int_t^T p(t, \lambda(t))\, dN(t) \right] \tag{3.9}$$

such that

$$\int_0^T dN(t) \leq x \tag{3.10}$$

$$\lambda(t) \in \Omega_\lambda. \tag{3.11}$$

where $u$ is any pricing policy that belongs to the set of feasible pricing policies $U$.

Dynamic pricing strategies may be found in the early work of Kincaid and Darling (1963) who introduced the continuous-time setting where customers reservations prices change over time. Similar assumptions for the reservation price distribution have been studied by Bitran and Mondschein (1997), Bitran et al. (1998) and Zhao and Zheng (2000).

Gallego and van Ryzin (1994) study the stochastic problem when demand intensities are time-stationary and depend solely on the price, i.e. $\lambda = \lambda(p)$, while the reservation price distribution does not change over time.

Using the Bellman principle they show that (3.9) is the solution to the following Hamilton-Jacobi-Bellman (HJB) equation,

$$-\frac{\partial V(x,t)}{\partial t} = \sup_{\lambda} \left\{ \lambda \Big( p(\lambda) - (V(x,t) - V(x-1,t)) \Big) \right\}, \tag{3.12}$$

with the final and boundary conditions

$$V(x,T) = 0 \quad \forall x,$$
$$V(0,t) = 0 \quad \forall t.$$

According to (3.12) it is easy to notice that the optimal demand intensity satisfies $\lambda^* = \max\{\lambda : p(\lambda) \geq V(x,t) - V(x-1,t)\}$ which implies that the optimal price is given by

$$p^* \geq V(x,t) - V(x-1,t).$$

The quantity $\Delta V(x,t) = V(x,t) - V(x-1,t)$ represents the opportunity cost of selling a unit of capacity at time $t$ when the available inventory is $x$.

Further, Gallego and van Ryzin (1994) prove that the value function is increasing and concave in both the capacity and the remaining time $\tau = T - t$. The monotonicity of the value function can be summarised in the following two properties

(1) Inventory-monotonicity property: $p_{x,t} \geq p_{x+1,t}$
at a given point in time the optimal price decreases as the inventory increases, and

(2) Time-monotonicity property: $\partial p/\partial t \leq 0$
at a given inventory level the optimal price decreases as the time remaining decreases (as we move towards $T$).

In a similar framework, Bitran and Mondschein (1997) study optimal strategies in retailing. For the case of time-invariant reservation price distributions the authors derive identical properties for the optimal price to that of Gallego and van Ryzin (1994). However, when reservations prices are also allowed to vary over time, they show that the time-monotonicity property might be violated. Therefore, Zhao and Zheng (2000) derived a sufficient condition under which the time monotonicity property holds. This condition requires that the conditional probability that a customer would buy at a higher price given that he/she is willing to buy at a lower price, is decreasing over time. In other words, the willingness of a customer to pay a premium for a product decreases over time.

Other extensions to the basic model include the work of Feng and Xiao (2000a), Feng and Xiao (2000b) and Feng and Gallego (2000) who restrict the optimal prices to be chosen from a finite set. Moreover, Bitran and Mondschein (1997) studied markdown price policies that are allowed to take effect only at a finite set of decision times. The problem is further generalised to the case of a retail chain[6] by Bitran et al. (1998). In a similar setting, Feng and Gallego (1995) studied the optimal time to switch from an initial price $p_1$ to another greater or lower price $p_2$ (in other words they restrict the price changes to only happen once within the horizon). They show that the optimal switching policy is of a threshold type; i.e. there exist sequences of time thresholds that tell us to switch as soon as the time has crossed the time threshold that corresponds to the number of yet unsold items.

[6]In their setting, the product is sold in different locations, each of with its own Poisson demand process.

## 3.5 Network RM

Having illustrated the single-resource problem we can move on to the more realistic *multiple-resource* or *network* problem where customers might require a bundle of resources. In RM literature a subcollection of the resources constitutes a specific *product*. For example, in airlines a journey that requires a set of connecting flights defines a product for which one unit of capacity (a seat) out of each flight-leg (resource) is required. Likewise, in the hotel industry a stay from Monday to Friday accounts for a product that is build out of one unit of capacity (a room) from Monday (the first resource), one unit of capacity from Tuesday, all the way up to one unit of capacity from Friday. Clearly, the hotel situation is the one we encounter in our airport carpark setting as well, with the only difference that now customers are represented by cars and capacities on each day refer to the parking spaces.

Based on the Talluri and van Ryzin (2005) production model we assume a network of $n$ final products and $m$ types of resources. One unit of product $j$, requires $a_{ij}$ units of resource $i$. We restrict the domain of $a_{ij}$ to consist only of 0's and 1's (i.e. products require exactly 1 unit of capacity from each resource involved in making the product)[7]. We define the *incidence matrix* or the *bill of materials matrix* $\mathbf{A}^{m\times n} = [\{a_{ij}\}]$ and let $\mathbf{A}_j$ to denote the set (vector) of resources that are used by product $j$. The initial capacity of each resource $i$ is denoted by $x^i$ so that $\mathbf{x} = (x^1, \ldots, x^m)$ is the corresponding vector of capacities. Once product $j$ is sold the remaining capacity vector becomes $\mathbf{x} - \mathbf{A}_j$.

### 3.5.1 Dynamic pricing on the network

Starting from the dynamic pricing models, we denote the $n$-dimensional price vector $\mathbf{p} = (p_1, \ldots, p_n)$ and $\boldsymbol{\lambda}(\mathbf{p}) = (\lambda_1(\mathbf{p}), \ldots, \lambda_n(\mathbf{p}))$ the associated (time-homogeneous) vector of demand intensities. Define the revenue function $r(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{p}(\boldsymbol{\lambda})$ and let $\boldsymbol{\lambda}^r$ to be the demand vector that maximises the revenue.

When the demand is assumed to be deterministic, Bitran and Caldentey (2003) show similar results to the single-resource case, i.e. that the optimal

[7]This is a common assumption in the literature, however, Gallego and van Ryzin (1997) extend the domain of $a_{ij}$ to any possible value.

price is a fixed price vector, namely

$$\mathbf{p}^* = \begin{cases} \mathbf{p}(\boldsymbol{\lambda}^r) & \text{if } T\,A\,\boldsymbol{\lambda}^r \leq \mathbf{x} \text{ (abundant capacity)} \\ \mathbf{p}(\boldsymbol{\lambda}^0) & \text{if } T\,A\,\boldsymbol{\lambda}^r > \mathbf{x} \text{ (scarce capacity),} \end{cases} \tag{3.13}$$

where $\boldsymbol{\lambda}^0$ is the solution to the following KKT conditions

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}} \left[\boldsymbol{\lambda}^\top \mathbf{p}(\boldsymbol{\lambda})\right] - A^\top \pi = 0 \\ \pi^\top \left[T\,A\,\boldsymbol{\lambda} - \mathbf{x}\right] = 0 \\ \boldsymbol{\lambda} \geq 0 \\ \pi \geq 0 \end{aligned}$$

and $\pi$ is the $m$-dimensional vector of Lagrange multipliers.
Also, these authors show that the optimal price in the network problem can *increase* with the level of capacity, as opposed to the single resource case where it is decreasing. Further, they determine under which condition the inventory monotonicity property can be sustained.

Of great importance in the stochastic setting is the study of Gallego and van Ryzin (1997). The authors study the problem in continuous-time by modelling the demand as a Poisson process with intensity $\lambda(p,t)$, and derive the HJB equation for the expected revenue similar to (3.12). This is,

$$-\frac{\partial V(\mathbf{x},t)}{\partial t} = \sup_{\mathbf{p}} \left\{ \sum_{j=1}^{n} \lambda_j(\mathbf{p}) \Big( p_j - (V(\mathbf{x},t) - V(\mathbf{x}-\mathbf{A}_j,t)) \Big) \right\}, \tag{3.14}$$

with final and boundary conditions given by

$$\begin{aligned} V(\mathbf{x},T) &= 0 \quad \forall \mathbf{x} \\ V(\mathbf{0},t) &= 0 \quad \forall t. \end{aligned}$$

The quantity $V(\mathbf{x},t) - V(\mathbf{x}-\mathbf{A}_j,t)$ is the opportunity cost of selling a product $j$ at time $t$. In particular it is easy to see that the optimal price satisfies

$$p_j^* \geq V(\mathbf{x},t) - V(\mathbf{x}-\mathbf{A}_j,t).$$

However, this is a set of differential equations which is difficult to solve and for which a closed-from solution is rarely available. Finally, dynamic pricing on a network has also been studied by Zhang and Weatherford (2012).

### 3.5.2 Network capacity control

Most research on network RM deals with a static-view on prices. In other words, the prices of the products are pre-determined and the manager's problem is to decide which requests to admit and which to reject. As a result, capacity-control models are widely available and commonly used in practice.

#### 3.5.2.1 Virtual nesting

As with single-resource problems, in network problems capacity control may be achieved in several ways. The ability of nested controls to dynamically share the capacity among the classes is a great feature over the partitioned controls. However, how to directly extend them in the network setting is not clear. Therefore, researchers developed a new type of control named *virtual nesting* which is a hybrid method of network and single-resource controls that maintains the nested-allocation structure.

Similar to section 3.4, virtual nesting applies single-resource nested-allocation controls to each resource in the network. The only difference is that the classes used in the nested allocations are *virtual classes*. In particular, through a process known as *indexing*, products that share a given resource are clustered into the virtual classes of that resource based on their network value. Then, nested booking limits are computed based on these virtual classes. Finally, a request is accepted only if *all* the virtual classes on *all* the related resources are available.

Virtual nesting was originally developed by American airlines (see Smith and Penn, 1988) and until now it is the main form of control in the airline industry. In general, these controls do not deviate much from the basic structure of the nested booking controls, but they can be highly complex and computationally intensive.

#### 3.5.2.2 Bid-price control

While, in general, extending the single-resource nested policies to the network environment is a difficult task, for bid-price controls this is straightforward. In particular, a bid-price control in the network setting specifies a set of bid prices for each resource. The bid prices now account for the marginal cost to the network of consuming the next incremental unit of a resource's capacity. In this way, a request for a particular product is accepted only if there is available capacity and the overall price exceeds the *sum* of the bid prices for all resources used to build up the product.

Network bid-price controls remain simple and fast as they only require a single threshold value to be stored, one for each resource (not each product). In addition, they retain their intuitive nature as the bid-price of a given resource can be interpreted economically as the marginal value of that resource to the network.

Network bid-price controls were initially developed by Simpson (1989) and then studied by Williamson (1992) in her PhD Thesis, who argues that despite its simplicity a bid-price control allows for "implicit" nesting, as when a product is open for sale it has access to the entire capacity of the resources involved. However, she emphasises that to be effective bid-prices must be revised and updated frequently.

Today, bid-price controls are the dominant form of control used in the hotel industry. In fact, this is the means of control we implement in our models and for this reason we take some time to examine it in rather more detail.

#### 3.5.2.3 Optimal capacity control

To see the strong connection between the bid prices and the opportunity costs we adopt the network setting described in Talluri and van Ryzin (2005).

Network capacity control is often formulated in a discrete-time setting where $T$ denotes the time of service and we start the problem at time 0 which is $T$ number of periods in advance.

The price vector at time $t$, $\mathbf{p}(t) = (p_1(t), \ldots, p_n(t))$ is now pre-determined (fixed) and so is the demand vector $\boldsymbol{\lambda}(t)$. Further assume that there is at most one arrival per time period.

Next, we define the $n$-dimensional random vector[8] $\mathbf{R}(t)$. This vector has *only* one non-zero entry at a time. For example, if we have $\mathbf{R}(t) = (0, p_2(t), 0, \ldots, 0)$ it means that at time $t$ demand has arrived for product 2 with associated price $p_2(t)$.

The *control vector* for all products at time $t$ is defined as $\mathbf{u}(t) = (u_1(t), \ldots, u_n(t))$. If $u_j(t) = 1$ then the request for product $j$ at time $t$ is accepted and if $u_j(t) = 0$

[8]This definition is the direct network extension to that in 3.2.

then the request is rejected. In general, the decision vector depends on the available capacity $\mathbf{x}$ the time $t$ and the set of prices $\mathbf{p}$, namely $\mathbf{u}(t) = \mathbf{u}(t, \mathbf{x}, \mathbf{p})$.

Finally, by defining $V_t(\mathbf{x})$ as the maximum expected revenue that is generated from time periods $t, t+1, \ldots, T$ when the available capacity vector at time $t$ is $\mathbf{x}$, we construct the following Bellman equation,

$$V_t(\mathbf{x}) = \mathrm{E}\left[\max_u \left\{\mathbf{R}(t)^\top \mathbf{u}(t, \mathbf{x}, \mathbf{p}) + V_{t+1}(\mathbf{x} - \mathbf{A}\mathbf{u})\right\}\right], \tag{3.15}$$

with the boundary condition

$$V_{T+1}(\mathbf{x}) = 0 \quad \forall\, \mathbf{x}. \tag{3.16}$$

The optimal control $\mathbf{u}^*(t, \mathbf{x}, p)$ is then defined as

$$\mathbf{u}_j^*(t, \mathbf{x}, p_j) = \begin{cases} 1 & \text{if } p_j \geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - \mathbf{A}_j) \text{ and } \mathbf{A}_j \leq \mathbf{x} \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

The optimal control policy for the above problem says that: we accept a request for product $j$ at time $t$ (with price $p_j$) if there is sufficient capacity remaining $\mathbf{x}$ and the product's price exceeds the *opportunity cost* that is incurred when removing the resources that constitute the product, i.e.

$$p_j \geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - \mathbf{A}_j). \tag{3.18}$$

Therefore, based on (3.18) we can construct a bid-price control (assuming differentiability of the value function $V$) as

$$\begin{aligned} p_j &\geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - \mathbf{A}_j) \\ &\approx \nabla V_{t+1}^\top(\mathbf{x}) \mathbf{A}_j \\ &= \sum_{i \in \mathbf{A}_j} \frac{\partial}{\partial x_i} V_{t+1}(\mathbf{x}) \\ &:= \sum_{i \in \mathbf{A}_j} \pi_i(t, \mathbf{x}), \end{aligned} \tag{3.19}$$

where $\nabla V_{t+1}^\top(\mathbf{x})$ is the gradient of the value function and the bid price of resource $i$ is defined by $\pi_i(t, \mathbf{x}) = \frac{\partial}{\partial x_i} V_{t+1}(\mathbf{x})$. Mathematically, a bid-price control $\mathbf{u}(t, \mathbf{x}, \mathbf{p})$

reads

$$\mathbf{u}_j(t, \mathbf{x}, p_j) = \begin{cases} 1 & \text{if } p_j \geq \sum_{i \in \mathbf{A}_j} \pi_i(t, \mathbf{x}) \text{ and } \mathbf{A}_j \leq \mathbf{x} \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

Talluri and van Ryzin (1998) presents a detailed study on the existence and optimality of such bid-price controls. In particular, they argue that bid-price controls are *not* optimal in general as they fail to capture the true opportunity costs. In fact, the authors prove that bid-price controls are optimal only if the opportunity cost of selling a product is equal to the sum of the opportunity costs incurred by consuming each of the resources (used by the product) separately, that is when

$$V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - \mathbf{A}_j) = \sum_{i \in \mathbf{A}_j} V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - \mathbf{e}_i) \text{ and } \mathbf{A}_j \leq \mathbf{x}. \tag{3.21}$$

where $\mathbf{e}_i$ is the $m$-dimensional vector with 1's in the $i^{th}$ entry and 0's in all the remaining entries. The main reason that causes this sub-optimality is the fact that this bid-price *additive* property does not hold, in general (for a comparison between additive and non-additive bid prices see Bertsimas and Popescu, 2003). Large *relative changes* in capacity on several resources are not expected to produce the same revenue as the sum of the individual changes (see equation (3.19)).

Fortunately however, Talluri and van Ryzin (1998) show that bid-price controls are asymptotically optimal as the scale of the problem increases. This result is also verified in Cooper (2002) and it is of great importance because employing bid-price controls is not just appealing (simple and intuitive structure) but now justified as well. Analysis of network bid-price controls has also been studied by Günther et al. (1999).

## 3.6 Main approximation methods and related literature

Calculating the optimal solution to a network problem is computationally intensive with a geometrically increasing complexity. As Talluri and van Ryzin (2005) explain, a network problem involving $m$ resources, all with initial capacities $C$, has $C^m$ states. Therefore, researchers are confined in searching for reasonable approximations, which are solutions to a simplified version of the original problem.

### 3.6.1 Approximations based on decomposition

One type of approximation method is to decompose the $m$-resource network problem to a set of $m$ single-resource problems that include some network information in them but they are essentially solved *independently* using single-resource methods. Then they approximate the network problem by adding up the solutions to these single-resource problems. In other words, the value function is approximated as

$$V_t(x) \approx \sum_{i=1}^{m} V_t^i(x^i), \tag{3.22}$$

where $x^i$ and $V_t^i(x^i)$ denote the capacity and the single-resource value-function solution for resource $i$.

The most popular network decomposition methods are the *Prorated Expected-Marginal-Seat-Revenue (EMSR)* (see Williamson, 1992) and the *displacement-adjusted virtual nesting (DAVN)* (see Smith and Penn, 1988), both originally developed for the airlines.

Prorated EMSR has been proposed by Williamson (1992) and allocates a portion of product's $j$ revenue to each of the resources $i \in \mathbf{A}_j$ that contributed in building the product $j$. In this way one obtains some measure of the contribution of the entire product to each of the resources used. Then the approach is to solve a single-resource problem using an EMSR heuristic, one for each resource $i$ used, and treat the resulting marginal values $\Delta V_t^i(x^i)$ as the corresponding bid prices.

DAVN is a virtual nesting approach that uses *adjusted* product prices instead. Similar to the Prorated EMSR approach, the idea is to measure the revenue benefit of accepting product $j$ to each of its resource components $i \in \mathbf{A}_j$. The adjusted price on resource $i$ of product $j$ is computed by subtracting the bid-price values of all the remaining resources $h \in \mathbf{A}_j, h \neq i$ from the product's price. These adjusted prices are then used to derive the virtual classes for each resource $i$. Based on these virtual classes we solve a single-resource model and compute $c$ protection levels for each resource $i$.

DAVN approaches are appropriate when the objective is to construct virtual nesting controls. In contrast, if one is using bid-price controls it might be more appropriate to use the displacement-adjusted revenues to solve $m$ dynamic programs, one for each resource, and calculate the bid prices of each resource by their corresponding marginal values.

While network decomposition methods are adopted when solving the network airline problem (see Bertsimas and de Boer, 2005; van Ryzin and Vulcano, 2008), for hotels, virtual nesting is not the preferable method. As opposed to the airlines where the virtual classes for each flight-leg are often less than ten, in the hotels this number is large, as virtual classes have to be maintained for every single day (see Vinod, 2004). Therefore, a different set of approximation methods dominate in the hotel industry, which falls under the category shown below.

### 3.6.2 Approximations based on simplified network models

These techniques retain the full network structure of the original problem but some further assumptions are placed in order to simplify it, the main being that demand is realised as an aggregate amount in each time which effectively reduces the dynamic problem to a static one. Consequently, this simplification naturally leads to using mathematical programs (linear or integer programs as described in section 2.2) to formulate and solve the problem.

One such method, originally developed by Simpson (1989) and Williamson (1992), is the deterministic linear programming (DLP) which states the problem as

$$V_t^{LP}(\mathbf{x}) = \max \mathbf{p}^\top \mathbf{y} \tag{3.23}$$

subject to

$$\mathbf{A}\mathbf{y} \leq \mathbf{x} \tag{3.24}$$

$$0 \leq \mathbf{y} \leq \mathrm{E}[\mathbf{D}]. \tag{3.25}$$

In this representation the $\mathbf{y} = (y_1, \ldots, y_n)$ corresponds to the vector of $n$ products offered and $\mathbf{D} = (D_1, \ldots, D_n)$ the aggregate demand vector for each of these products. The set of constraints in (3.24) accounts for the capacity restriction on the resources whereas (3.25) represents the demand constraints and bounds the allocations by the expected demand. Consequently the problem reduces to finding the optimal combination of products that maximises (3.23).

Such models are simple and computationally fast which led many researchers to study them in various industries, among them Williamson (1992), de Boer et al. (2002), Bertsimas and Popescu (2003) for airlines and Weatherford (1995), Baker and Collier (1999) and Goldman et al. (2002) for hotels.

However, the DLP has a serious drawback; that is, it only considers expected demand and ignores all other distributional information. Consequently, when demand is highly variable the solution will lead to poor management decisions. Therefore, the necessity of incorporating uncertainty more accurately into the model led researchers to two main variates.

The first has been developed by Wollmer (1986) and further studied by de Boer et al. (2002) and suggests replacing each $y_j$ in the objective function (3.23) by the expected sales of the product $j$ under the partitioned allocation $y_j$, i.e. by the term $\mathrm{E}[\min\{D_j, y_j\}] \quad \forall j = 1, \ldots, N$. Indeed, this approximation resolves the problem seen in the DLP method. However, it comes with a downside since not only does the resulting program become nonlinear[9] and more difficult to solve but the resulting revenues are consistently lower than those of the DLP.

A second approach proposed by the independent studies of Smith and Penn (1988) and Talluri and Van Ryzin (1999) is based on simulating a sequence of demand realisations and solving a DLP problem, one for each. In particular, they propose to replace the expectation vector $\mathrm{E}[\mathbf{D}]$ in the demand constraints (3.25) by an actual realisation vector and proceed as in the DLP. Then taking the expectation of the DLP solution gives an estimate for the value function. This probabilistic method is in fact simpler than the former, as it is just a simple modification to the original DLP model but its performance strongly depends on the number of samples used.

In a recent study, Maglaras and Meissner (2006) consider a single-resource multi-product framework and formulate both a dynamic-pricing model and a dynamic capacity control problem. Interestingly, the authors show that the multi-dimensional problem can be written in a simpler format such that the firm only controls the aggregate rate at which the resource is jointly consumed by the products.

[9]In general, the program is non-linear but under certain conditions it can be expressed as a linear program (see de Boer et al., 2002; Talluri and van Ryzin, 2005)

### 3.6.3 Optimal control in practice

Whichever techniques are used (discrete/continuous-time and/or quantity/price-based) the optimal control essentially reduces to calculating the opportunity cost

$$V(\mathbf{x}) - V(\mathbf{x} - \mathbf{A}_j, t),$$

which is often approximated by the corresponding bid prices. Then, the bid-prices are either used directly in a bid-price control mechanism or indirectly to set up the DAVN decomposition.

The simplest way to achieve this is through the solution of mathematical programs. More specifically, if we define a given mathematical program by $MP$ then the opportunity cost is approximated by

$$(\nabla V_t^{MP}(\mathbf{x}))^\top \mathbf{A}_j,$$

which can easily be found as the dual variables associated with the resources (see section 2.2.1 on linear programming).

Alternatively, Bertsimas and Popescu (2003) propose approximating the opportunity costs by

$$V_t^{MP}(\mathbf{x}) - V_t^{MP}(\mathbf{x} - \mathbf{A}_j),$$

which effectively takes out some of the limitations of the bid prices. However, this comes at its own cost as one should solve the mathematical program twice for each product, a procedure that increases the computational effort.

### 3.6.4 Network models in the hotels

As pointed out already, the network formulation in hotels can be seen as one where resources are represented by the different days. Then multi-day stays are analogous to the multi-leg itineraries of an airline network. However, some key differences are that in the hotels it is not clear how to define the end of the horizon. For this reason, researchers use the so called rolling window procedure where the time horizon moves along as time progresses. Details on this techniques can be found in Goldman et al. (2002). Second, according to Zhang and Weatherford (2012), the network structure is more pronounced as it is not uncommon for customers to stay for a week, which is analogous to a seven-leg itinerary in the airlines.

Each product that goes into the mathematical program now referred to as a triple $(a, L, k)$ i.e. the arrival day $a$, the length of stay $L$ and the price class $k$. Thus, for an optimisation period of, say, 30 days with maximum length of stay of one week and 10 price classes we have as many as 2100 different products to optimise over.

Weatherford (1995) was among the first to study a DLP model to formulate the network model in the hotel industry and showed significant improvement over single-resource models. Then, it was Baker and Collier (1999) who presented a comparative analysis of five different booking control policies.

Stochastic mathematical programs are based on the linear version of the non-linear probabilistic model described in 3.6.2. The linearisation requires setting up a set of demand scenarios for each type of stay and form the model based on the concept of expected marginal revenue (as in Littlewood, 1972). These models have been studied in Goldman et al. (2002) and de Boer et al. (2002), while Lai and Ng (2005), Liu et al. (2006) and Liu et al. (2008) extend the model to account for the risk in revenue from the random demand in several scenarios.

However, some disadvantages arise when using the mathematical programs. First, the problem is static by definition which means that the resulting bid prices will be also static. Thus, unless we frequently re-solve and update the bid prices, the resulting control will lead to poor decisions. Second, implementing uncertainty into the mathematical program is not easy and in fact renders the model nonlinear. Linearisiation techniques could ofcourse be used to simplify it but will result in a large number of variables to monitor. Third, the number of products to model can increase rapidly when customers are allowed to stay for longer than one week.

For a critical review on hotel RM the interested reader is referred to Ivanov and Zhechev (2012).

## 3.7 Optimisation strategies for carparks

Parking plays a vital role in the transportation industry and in the customer satisfaction of travellers. Parking pricing, availability and accessibility are the three major components that influence travellers' decisions about which type of transport to use, where and when to park (Qian and Rajagopal, 2013). Thus, in order to design a reliable and effective transportation system all these components have to be optimised simultaneously.

As such, there has been an increasing interest in carparking problems within the last two decades. Most researchers have studied traffic congestion[10] problems; among them are Verhoef et al. (1995), Teodorović and Vukadinović (1998), Arnott and Rowse (1999), D'Acierno et al. (2006), Zhao et al. (2010), to list just a few. Todd (2006) investigates the main problems with the parking planning practices and evaluates several strategies that help to increase parking efficiency. For further descriptive information and qualitative guidelines the reader is referred to the parking management implementation guide found in Todd (2010, 2012).

Although parking pricing strategies are efficient and widely used in travel demand management policies (D'Acierno et al., 2006), there is not much literature in RM being applied to carparks; however, we do discuss a few articles that seem to have studied parking inventory control for revenue maximisation, rather than for traffic congestion control.

Teodorović and Lučić (2006) propose an "intelligent" parking space inventory control system, based on fuzzy logic and integer programming techniques. They study the problem of applying an optimal accept/deny decision on the bookings in the order these arrive, assuming that multiple customer classes with customer arrival and departure times are both stochastic. First, they simulate realisations of these processes in order to derive an upper bound solution using integer programming (perfect future prediction), with the procedure to be repeated for every scenario simulated. Then, by analysing the results of this deterministic optimal solution, it is possible that at any time a request arrives to know the availability of the parking capacity ($X_1$) and the relative requests' revenue ($X_2$), as well as the percentage of all future requests that make less relative revenue than the current request ($Y$). By treating the first two quantities as the *antecedents* and the last one as the *consequence*, fuzzy rules are generated (for detailed information in creating fuzzy rules, the reader is referred to the work of Wang and Mendel, 1992). The outcome of the fuzzy rule is the value of $Y$; if this is less than or equal to a particular constant, chosen by the analyst, the booking is rejected. The results show that the relative difference (between the proposed algorithm and the optimal upper bound) never exceeds 10% for any simulated scenario.

[10]Traffic congestion occurs when traffic demand on road networks is sufficiently high and is characterised by slower driving speeds, longer destination times and increased vehicular queueing.

Guadix et al. (2008, 2011) consider the presence of a group of *subscribers* along with the individual customers. The main difference with subscribers over individual customers is that they have a space rented for the same period of time during the same hours, and as such they should be treated differently. They formulate the problem as an integer program and solve it under both a deterministic and a stochastic environment. Then, three different algorithms for capacity allocation are tested: a first-come-first-served, distinct and nested method. By comparing against a perfect information model they show that a stochastic model using nested allocation provides revenues that are closest to the optimal values.

Rojas (2006) in his master's thesis assumes up to four customer classes and implements the $EMRS$-*a* and $EMSR$-*b* (see Belobaba, 1989) heuristics to determine the number of parking spaces to be reserved for the higher classes.

In a similar study, Van Den Eijnden (2009) also uses these heuristics to compute the booking limits for each class. However, as explained in section 3.4 these models only solve the simpler single-stay problem whereby effects from neighbouring days are ignored. Therefore, he investigates an alternative means to compute the booking limits so that the inter-dependence within the days is accounted for. In particular, he proposes to calculate the expected average daily rate a class-$i$ customer would pay, as a weighted average scheme over the rates of the neighbouring days the customer is likely to be staying for. According to this quantity he could then decide on whether class $i$ should remain open or should close. His methodology, although not optimal, has improved occupancy levels significantly.

In conclusion, our work may be distinguished from all previous studies in the parking industry as airport carparks possess some unique features; they operate using an online pre-booking reservation system where customers can book early in advance and the price rates can vary with the length of stay. Note that we also aim to solve the multi-day problem. Nonetheless, our approach is to compute *dynamic* (time and state dependent) bid-price controls using the expected marginal values of the parking spaces and decide whether to accept/reject a request based on the sum of bid prices over the days the customer is staying for. The bid prices are dynamic in the sense that our solution specifies a distinct threshold value for *every* combination of capacity remaining $Q$ and time left $\tau$ until arrival. Finally, the problem is formulated in continuous-time and thus to solve it we use alternatives to mathematical programming techniques. This is considered in detail over the following chapters.

# Chapter 4

# A network setting for the airport carpark

## 4.1 Problem setting

Our problem is as follows:
Suppose that an airport operator utilises an internet pre-booking system to sell carparking spaces within the premium jetpark, and that this carpark has fixed capacity $C$. Bookings are recorded in the system on a continuous-time basis. Each booking specifies the exact time the customer requires to arrive at the carpark and the exact time they want to leave. The bookings arrive in order and may require to stay for multiple days or even hours. Suppose that we are on day $t$ and a booking request arrives requiring a slot on a future day $T$. The carpark manager[1] may decide to either accept or reject the booking. If he decides to accept the request, he instantly realises the corresponding revenue from the sale and in return he gives out a space of the carpark on that day. However, by doing so, he loses out on the opportunity of *waiting* for any potential future requests that might have been more profitable. The question that naturally arises is: how should the manager act? The stochastic nature of the customer demand and because such managerial decisions are irreversible, results in a challenging problem which we aim to answer in this thesis.

[1]The term "carpark manager" is a loose term that describes the set of decision rules put in place by the airport operator and it might be used frequently for the remainder of this thesis.

**Objective** Our task is to maximise the expected revenue in the carpark for the future day $T$ by applying *dynamic capacity control* techniques to the booking requests. In other words, we want to apply optimal admission/rejection decision for each booking, in the order they arrive in such a way that the expected revenues are maximised.

**Challenges** There are two main challenges in our problem; first, the stochastic nature of the customer demand means that decisions today for the future should be made under uncertainty about the final result. Second, the *network* structure of the problem means that customers with different arrival times and/or lengths of stays overlap and as such every day feeds into the next in a nonlinear manner. Thus, the decision on a day $T$ should be based not only on that day but on the state of the carpark in the neighbouring days as well. Clearly, this problem becomes very large and even when optimal solutions exist, these are unrealistically time consuming if not impossible to solve. Therefore, we will be studying different methods of calculating slightly sub-optimal rejection policies of reasonable performance.

Let us first describe the overall network setting and the varying assumptions imposed.

## 4.2 Model assumptions

In particular, we assume the following:

(1) There is only one type of parking space in the carpark, for simplicity.

(2) There are two streams of customers:

- Low-paying customer class, $L$

  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 customer class, $H$

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

Since these two sets of customers describe different types of customer behaviour they will have different average daily booking intensities $\lambda_b$, different average times between booking and arrival, $\bar{\eta} = {}^1/_{\lambda_a}$ and different average length of stays $\bar{\xi} = {}^1/_{\lambda_s}$. Therefore, the $n^{th}$ customer class could be expressed as

$$\mathcal{B}_n \sim \begin{pmatrix} \lambda_{b_n} \\ \lambda_{a_n} \\ \lambda_{s_n} \end{pmatrix}. \tag{4.1}$$

Under this representation there are three sources of uncertainty as the number of bookings per day from the $n^{th}$ class, the advance-times and length-of-stay are all stochastic variables.

Using equation (4.1) we can define the total booking profile as a linear combination of the single booking classes, namely

$$\mathcal{B} = \sum_n \mathcal{B}_n. \tag{4.2}$$

In our model only two classes are assumed and thus we use subscripts 1 and 2 for the leisure and business classes, respectively.

(3) Models like Liu et al. (2008) and Teodorović and Lučić (2006) assume that the price of a parking spot is linearly proportional to the length of stay. However, in airport carparks it is often the case that the *longer* one stays in the carpark the *less* price they pay per day (although the total price of the booking is higher). In other words, the length of stay becomes a variable that affects the quoted total price in a nonlinear manner.

Therefore, we define the pricing function[2] $\Psi(\xi)$ that calculates the (average) *price rate to be paid per day.* The pricing function relates the daily price rate to the length of stay $\xi$ and it is applied to *all* customers irrespective of the class they "belong" to.

An examination on real carparks pricing strategies indicates that a pricing function which is common in practice is of an exponential form, i.e. the

[2]This should be distinguished from the price-demand function seen in section 3.4.2.2, which explicitly relates the price to the demand.

average price rate per day $\Psi$ decreases monotonically in $\xi$ according to

$$\Psi(\xi) = \psi_1 + \psi_2 e^{-\mu\xi}, \tag{4.3}$$

where $\psi_1$, $\psi_2$ and $\mu$ are positive constants.

The *total price* to be paid by a customer staying for $\xi$ days may then be calculated by

$$\xi\,\Psi(\xi).$$

Figure 4.1 shows a typical pricing function of this form. The parameter $\psi_1$ indicates the lowest price rate which may be charged per day; this is asymptotically achieved in the limit when the length of stay grows to infinity.

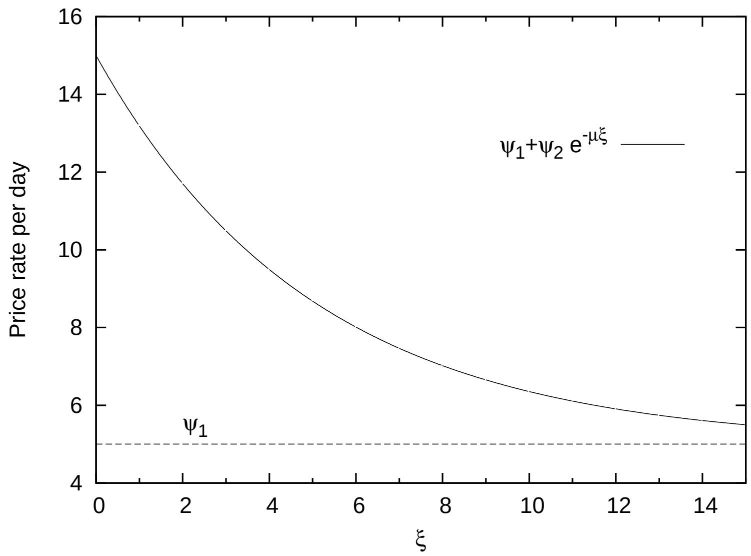



Figure 4.1: Exponential price-rate function, with $\psi_1 = 5$, $\psi_2 = 10$ and $\mu = 0.2$.

Combining the above with the characteristics of the two customer classes we notice that leisure customers will be quoted, on average, lower price rates per day, although all customers are treated under the same pricing scheme. This is perhaps a unique feature encountered in the airport carpark and as far as we know there are no existing models in the literature that explicitly deal with this situation.

We note that we are not concerned in deriving the optimal pricing-function parameters $(\psi_1, \psi_2, \mu)$. Particularly, we consider that these have been derived or pre-set by the management and they are kept *fixed* throughout. However, it is important to understand that although the pricing function is fixed, the price rate per day charged to a customer *does* vary according to his/her required duration of stay.

(4) No discounting on prices takes place, for simplicity. We do not consider the time-value of money as our objective is to examine the performance of the rejection algorithm.

(5) There is no marginal cost incurred after a sale. This is a reasonable assumption, because one can always express price as the increment above cost (perhaps with a slight variable transformation). Thus, the expressions "revenue" and "profit" will be used interchangeably.

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

(7) The network environment is originally formulated in discrete-time meaning that spaces are sold for integer number of time-periods. In other words, if we assume day interval-periods, the customers although being able to book and arrive at any time and ask to stay for any length of stay $\xi$ (continuous quantity), they will be priced according to the *number of periods* they will be staying for. This point should be made clear later in section 4.3.4.

# 4.3 The model

## 4.3.1 Structure of the bookings

Each booking consists of three time parameters, 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

$$B^i = \begin{pmatrix} t_b \\ t_a = t_b + \eta^i \\ t_d = t_a + \xi^i \end{pmatrix}, \tag{4.4}$$

where $t_b$ denotes the booking time, $t_a$ the arrival time with $\eta^i$ denoting the pre-booking time and $t_d$ the departure time with $\xi^i$ denoting the required duration of stay.

In particular, we assume that the number of bookings per day from customer class $n$ follows a Poisson process with stationary intensity $\lambda_{b_n}$. Furthermore, we assume that on average customers of the $n^{th}$ class arrive at the carpark $\bar{\eta} = 1/\lambda_{a_n}$ days after their booking and they stay for an average $\bar{\xi} = 1/\lambda_{s_n}$ days. Thus, the times between booking and arriving in the carpark as well as arriving and departing from the carpark may be modelled by exponential distributions with rate parameters $\lambda_{a_n}$ and $\lambda_{s_n}$ respectively.

Note that, 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. In addition, the pre-booking time $\eta$ as well as the duration of stay $\xi$ are both stochastic.

### 4.3.2 Generating bookings

Recall that for the $n^{th}$ customer class, the bookings are assumed to follow a homogeneous Poisson process with intensity[3] $\lambda_b$. Thus, the expected number of bookings per day is $\lambda_b$, while the expected number of bookings in $[0, t]$ is $\lambda_b\, t$. The time between two bookings is thus exponentially distributed with rate parameter $\lambda_b$ (with mean time given by $1/\lambda_b$), namely

$$\rho_b(t) = \lambda_b e^{-\lambda_b t} \tag{4.5}$$

and it can be calculated as (see section 2.1.2.4)

$$t = -\frac{1}{\lambda_b}\log(r), \tag{4.6}$$

where $r$ is a random variable from the uniform distribution $U(0, 1)$. Therefore, to get the time of the next booking $t^i$ we simply have

$$t^i = t^{i-1} - \frac{1}{\lambda_b}\log(r^i), \tag{4.7}$$

where $t^{i-1}$ denotes the time of the last booking.

Therefore, given the last booking $B^{i-1}$ we can simulate the next booking $B^i$

[3] In this section we slightly abuse notation by omitting the subscript indicating the class we refer to as it should be made clear from the text.

as

$$B^i = \begin{pmatrix} t_b = t^{i-1} - (1/\lambda_b)\log(r^i) \\ t_a = t_b - (1/\lambda_a)\log(r^{i+1}) \\ t_d = t_a - (1/\lambda_s)\log(r^{i+2}) \end{pmatrix}, \tag{4.8}$$

where $t^{i-1}$ denotes the booking time of the last booking, $B^{i-1}$.

The simulation procedure in (4.8) begins at time $t = 0$ and it is terminated when the next simulated booking time, $t_b$, is greater than $t = T$. Then, the number of generated bookings constitute one *booking set* from the chosen customer class.

We can, therefore, proceed as above to simulate a set of bookings for each customer class $n$. Then, the resulting booking sets are added together to form a *combined booking set* so that the system does not differentiate between bookings. In this way, the system can only see the bookings in the order these arrive without knowing the customer class these have originated from.

### 4.3.3 Bookings and carpark spaces

Assume that we are on day 0 with an empty carpark that is about to begin its operations. Furthermore, we seek to maximise the expected revenues on a future day $T$. The idea is to consider the situation in the carpark within a time frame $[0, \beta]$ with $\beta > T$.

In particular, we split the interval into $K$ discrete-time periods of length $\Delta T$ (A *day* corresponds to having $\Delta T = 1$, the standard unit of analysis in the report. For $\Delta t < 1$ we have time periods that are fractions of a day, namely half-days, six hours, thirty minutes etc. ) and we define the time points

$$\begin{aligned} &t^0 = 0 \\ &t^k = k\Delta T \quad \text{for} \quad k = 1, 2, \ldots, K-1 \\ &t^K = \beta. \end{aligned} \tag{4.9}$$

Then the $k^{th}$ day, $T^k$, is given by

$$T^k = [t^k, t^{k+1}) \quad \forall\, k = 0, \ldots, K-1. \tag{4.10}$$

Then, if $M^k$ denotes the number of cars present in the car park at any time during the day $T^k$, we have

$$M^k = \sum_i f(B^i, k), \tag{4.11}$$

where

$$f(B^i, k) = \begin{cases} 1 & \text{if} \quad t^k \leq t_a < t^{k+1} \\ 1 & \text{if} \quad t^k < t_d \leq t^{k+1} \\ 1 & \text{if} \quad t_a < t^k \quad \text{and} \quad t_d > t^{k+1} \\ 0 & \quad \text{otherwise} \end{cases} \tag{4.12}$$

denotes whether the $i^{th}$ booking is present within the day $T^k$. Note that we consider cars departing at $t = t^{k+1}$ as being present during the period (this is explained below). Figure 4.2 shows examples for which booking requests satisfy $f(B^i, k) = 1$.

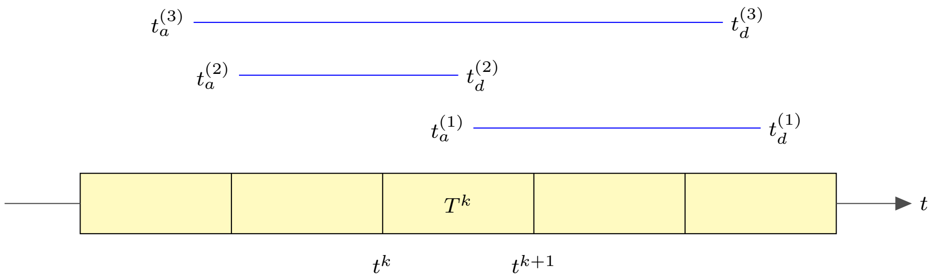



Figure 4.2: Booking requests examples for which $f(B^i, k) = 1$.

### 4.3.4 Pricing function

The *discrete* duration of stay, $D^i$, for each customer is calculated[4] according to the number of days they will be present in the carpark, i.e

$$D^i = \sum_k f(B^i, k) \in \mathbb{Z}. \tag{4.13}$$

Consequently the pricing function (4.3) becomes a step function, with step-size equal to a day.

$$\Psi(D^i) = \psi_1 + \psi_2 e^{-\mu D^i}. \tag{4.14}$$

[4]Although bookings can ask for any duration of stay $\xi$ which is a continuous quantity, in this discrete-time problem the duration is translated as the number of days $D$ the booking happens to fall within. Therefore, $D$ it does not exactly equal $\xi$. Note that when $\Delta T < 1$ then $D$ refers to the *number of time intervals of size* $\Delta T$. Thus, $D$ and $\xi$ become the same in the limit, when the size of the interval $\Delta T \to 0$.

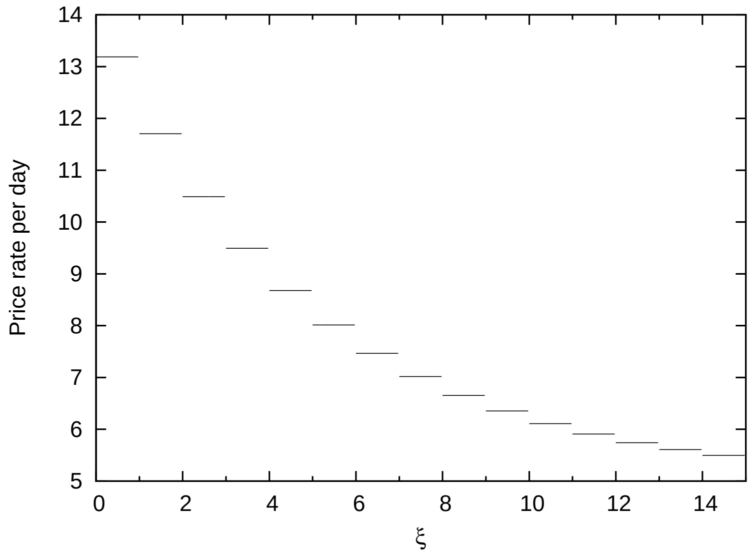



Figure 4.3: Price rate per day as a discrete step function, with $\Delta T = 1$, $\psi_1 = 5$, $\psi_2 = 10$ and $\mu = 0.2$.

Figure 4.3 illustrates the resulting pricing function for the case of day intervals. For a more detailed analysis of the pricing function, figure 4.4 is provided. If a customer makes a booking according to which he wants to arrive and depart on the same day, then he will be charged the price rate for one day stay $\Psi(D = 1)$ (brown customer). If the customer wants to stay for less than a day in total but his booking request spans over two days, he will then have to reserve the space for two days and pay the price rate $\Psi(D = 2)$ per day of stay (red customer). Also, if the customer arrives in sometime within a day and leaves exactly at the beginning of the next stay he will still get charged for two days (blue customer).

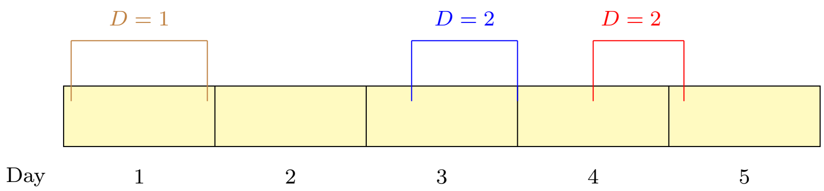



Figure 4.4: A carpark setting example with three booking requests. The brown customer is allocated one day but the blue and red customers are allocated two days.

### 4.3.5 Carpark revenue

Therefore, the revenue generated in $T^k$ over all bookings from all preceding times is expressed as

$$J = \sum_i f(B^i, k)\, \Psi(D^i) \tag{4.15}$$

and the total revenue $TR$ for the carpark becomes

$$TR = \sum_k \sum_i f(B^i, k)\, \Psi(D^i). \tag{4.16}$$

Consider that we are on time $t$, when we have already observed all bookings prior to this time. Then, the *expected revenue remaining* to be generated for the future day $T^k$, $t < t^k$, is given by

$$J(t; k) = \mathrm{E}\left[\sum_{i=i^*} f(B^i, k)\Psi(D^i)\right],$$

where $i^*$ indicates the *first* booking made after time $t^k$.

**Carpark revenue with capacity constraints** Until now, we have assumed that the carpark has unlimited capacity and therefore all bookings have been accepted and thus contributed to the revenue. To make things more realistic, we set a finite capacity $C$ to the carpark. Then, we may define $Q^k$ as the number of spaces remaining in the carpark for day $T^k$. In fact, we must have

$$Q^k = C - M^k \quad \forall\, k.$$

Let us suppose that $M^k|_i$ is the number of cars present in day $T^k$ after the first $i$ bookings have been realised. Then, by imposing capacity restrictions on the carpark implies that the $(i+1)^{th}$ booking can be a candidate for receiving service only if

$$M^k|_i + f(B^{i+1}, k) \le C \quad \forall\, k, \tag{4.17}$$

or equivalently

$$Q^k|_i - f(B^{i+1}, k) \ge 0 \quad \forall\, k. \tag{4.18}$$

### 4.3.6 Booking rejection policy

If the volume of booking requests is greater than the carpark capacity, then some customers will be denied service. This means that the total revenue that can be generated in a period depends upon the carpark capacity which is initially $C$.

Therefore, the expected revenue in day $T^k$ of a carpark of total capacity $C$ but with only $Q$ spaces remaining as observed at time $t < t^k$ is given by

$$J^u(Q,t;k,C) = \mathbb{E}\left[\sum_{i=i^*} f(B^i,k)\, u^i(k)\Psi(D^i)\right], \tag{4.19}$$

where $u \in U$ is any booking rejection policy from the set of admissible policies $U$ that abide by the capacity constraints, i.e.

$$u^i(k) = \begin{cases} 1 & \text{if } f(B^i,k) = 1 \text{ and the booking } i \text{ is accepted} \\ 0 & \text{otherwise.} \end{cases} \tag{4.20}$$

A couple of points to note: First, under any given policy $u$, equation (4.17) (or (4.18)) is a *necessary* condition for a booking to be accepted, but *not always sufficient*. In fact, it forms a sufficient condition only when the rejection policy implemented is a *first-come-first-served* (FCFS) policy. Second, any given policy $u \in U$ ensures that booking requests may either be accepted in "total", i.e. for the full number of days requested by the customer, or for none. Rejected customers cannot alter their preferences to be accepted, rendering each decision made as final. Last, all decisions are made based on the current information without any future knowledge of the demand, rendering it a non-anticipative policy.

**Expected Value Function** The carpark manager wants to maximise the expected revenue generated in the carpark. In order to achieve this, they must determine the rejection policy $u^*$ that gives rise to the value function $V(Q,t;k,C)$; this is the *maximum expected revenue in day $T^k$ of a carpark of total capacity $C$ with $Q$ spaces remaining as observed at time $t$*, expressed by

$$V(Q,t;k,C) = \max_{u\in U} J^u(Q,t;k,C). \tag{4.21}$$

### 4.3.7 Rejection policy based on opportunity costs

Our task is to derive an optimal rejection policy under which bookings will be optimally allowed service so that the expected revenues will be maximised.

By definition, a FCFS policy provides service to customers on a first-come-first-served basis. When a booking request arrives in the system, the FCFS policy checks the available capacity for the days requested and it only denies service when there is no space available for one or some of the days in the booking. It is clear, that such a rejection policy cannot be optimal in general, as it accepts/rejects customers based on capacity availability only, completely ignoring the booking's profile (price value, pre-booking time, length of stay). Consequently, the spaces in the carpark might be filled out from leisure customers early in the booking horizon, while business customers arriving later and willing to pay more are denied service. Therefore, in order to find the optimal rejection policy $u^*$, under which expected revenues are maximised, we need a more sophisticated approach.

Our rejection algorithm suggests that a booking request (at $t_b$) will be rejected if the total revenue generated by this booking 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.

Let us first consider the quantity

$$\Delta V(Q, t; k, C) = V(Q, t; k, C) - V(Q - 1, t; k, C).$$

This quantity is the *opportunity cost* which is incurred when we switch from a carpark (of total capacity $C$) with $Q$ spaces remaining for the day $T^k$ as of time $t$ to one with only $Q - 1$ spaces left. This opportunity cost arises after every sale of a space happens, and therefore it suggests how much the $Q^{th}$ unit of space is expected to be worth as of time $t$; we denote this by the *Expected marginal value of the $Q^{th}$ space as of time $t$.*

Then, it is only sensible to accept the booking $i$ on day $T^k$, if the price we receive is greater than or equal to the corresponding expected marginal value. Therefore, for accepting the booking $i$ made in day $T^m$ for $T^k$, we require:

$$\Psi(D^i) \geq \Delta V(Q, m; k, C). \tag{4.22}$$

However, we seek to solve the network problem which means that the booking decision should be made according to the "total" length of stay (i.e. on the entire set of $T$'s the car requires to be present within) and not for each day period individually. In fact, given that there is capacity available, the total price received from the booking should also be greater than the *sum* of the expected marginal revenues for the days requested by the booking. Therefore, the optimal control for day $T^k$ may be implemented as a dynamic (time- and state-dependent) *bid-price control* where the bid price is equal to the expected marginal value, that is

$$\pi(Q, m; k; C) = \Delta V(Q, m; k, C), \tag{4.23}$$

for every $Q = 0, 1, \ldots, C$ and $m = 0, 1, \ldots, k$.

Consequently, we find it convenient to introduce the *Expected Added Marginal Value across all periods EAMV* during which the car is present to be:

$$EAMV = \sum_k f(B^i, k) \left[\Psi(D^i) - \pi(Q, m; k, C)\right], \tag{4.24}$$

with the decision rule

$$\begin{aligned} \text{Accept if:} \quad & EAMV \geq 0 \\ \text{Reject if:} \quad & EAMV < 0. \end{aligned}$$

As a result, the optimal rejection policy $u^*$ is a bid-price policy and reads

$$(u^*)^i(k) = \begin{cases} 1 & \text{if } f(B^i, k) = 1 \text{ and } EAMV \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.25}$$

We note that the optimal rejection policy in this model preserves the network structure of the problem whereby the decision depends on the full set of days the customer is staying for, such that customers are either accepted for the entire number of days requested or for none. As such this model setting will be denoted as the *Network Model*.

### 4.3.8 Calculating and implementing the opportunity costs

The network model has four dimensions: the carpark capacity $C$, the target day to optimise $T^k$, the current day $T^m$ and the available spaces $Q(k)$ for day $T^k$ as of day $T^m$. In addition, there are $K$ number of such target days to be optimised.

Thus, calculating the full set of optimal expected marginal values

$$\Delta V(Q(k), t; k, C)$$

for each time and state of the system it is prohibitively time consuming if not impossible.

Therefore, we are confined to studying some rather more simplified problems whose solutions will be used to construct a bid-price policy and thus approximate the network model. As discussed in the literature, these problems will either preserve the network structure but scaled down significantly, or they will effectively decompose the network into a single-resource problem.

Once a rejection policy has been computed (by a particular approximation method) it is integrated into the revenue management system in the form of a bid-price table. Then, booking requests start coming to the system requiring multiple lengths of stays and the carpark manager has to make a decision whether to accept or reject the request. The decision is based upon the EAMV of the booking (see section 4.3.7) which is then computed according to the installed rejection policy.

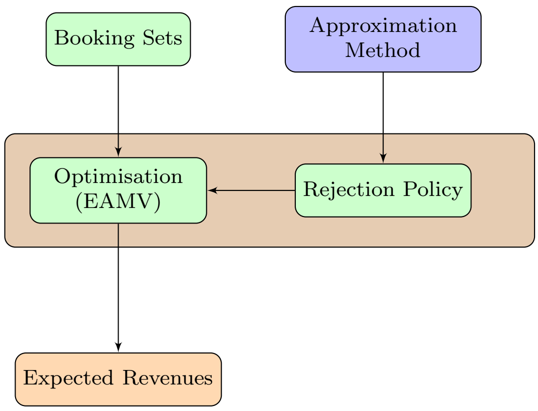



Figure 4.5: Schematic overview of the program flow

Figure 4.5 illustrates the main idea behind the comparison procedure. As we can see, the marginal values of an approximation method are imported to form the bid-price rejection policy. Then booking sets are simulated (see section 4.3.2) and decisions are made according to the EAMV algorithm. Finally, the revenues are averaged over a large number of booking sets to obtain the resulting expected revenues.

### 4.3.9 Measuring the effectiveness of capacity allocation

The aim is to maximise the expected revenue we would make on a given future day $T$ under a given rejection policy. But how should we measure the effectiveness of one method over the other?

To do this we need to define an appropriate metric. In the hotel industry, the management may target to increase the *average occupancy rate* - the fraction of booked rooms over available rooms per night- or the *average revenue rate* - the total revenue rate per night over the number of rooms in the hotel. Each of these metrics could be treated in isolation depending on the manager's objective which can either be to maximise revenue rate or to maximise the occupancy rate. A more representative metric that measures both revenue and capacity utilisation is the *revenue-per-available-room-night (RevPar)* which is defined as the multiplication of the above average daily revenue rate and the occupancy rate (for details see Phillips, 2005).

For our problem, we are only concerned in finding the policy that maximises the expected revenues on a given day without worrying about the utilisation of the resources. In fact, a policy that has resulted in higher revenue is more successful than one that resulted in lower revenue, even if the first policy resulted in lower occupancy rate.

Since our model is time-stationary it makes sense to examine the average revenue rate once the system is at a *steady-state equilibrium* position, i.e. when the initialisation (transient) effects have already been absorbed in the system. This will indicate how the model would perform if it left running forever, i.e. a *perpetual* revenue.

More specifically, we simulate a booking set (a path) and we let the algorithm to run for sufficiently long. Then, we focus on an intermediate 20-day interval somewhere in the future. The average revenue rate per day can thus be computed by averaging over these individual expected revenue rates. An illustration on this is shown in figure 4.6.

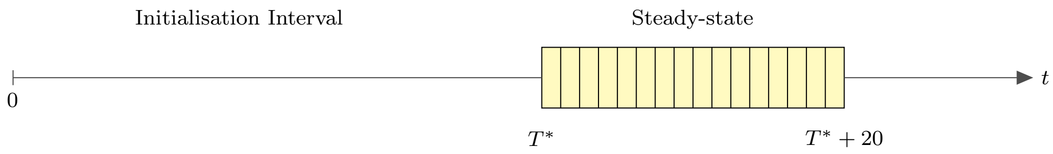



Figure 4.6: Illustration of computing the steady-state equilibrium position. An initialisation period of length $T^*$ is used. The steady-state revenue is computed by averaging over the individual revenues of the 20 days which lie immediately after $T^*$.

This procedure is repeated under a thousand simulated booking sets and an estimate of the perpetual revenues is calculated by taking the average over the number of paths. Finally, the resulting perpetual revenues from one method are compared against the others to examine the methods' overall performance. Note that there is no analytical or full network solution to compare against. This analysis will be dealt in detail later in chapter 6, as next we will present the approximation methods under consideration.

# Chapter 5

# Three approximation methods for the carpark network model

*Part of the work in sections 5.1, 5.2 and 5.3 has been published in ICORES 2012 International conference proceedings.*
*"Continuous-Time Revenue Management in Carparks", which has been awarded with the Best Student Paper Prize, is available online in SciTePress Digital Library at http://www.scitepress.org/DigitalLibrary.*

The network problem described in chapter 4 is a typical problem that could be formulated as a mathematical program, e.g. a linear or integer program. In particular, we can define an optimisation period of a set of $m$ days (i.e. $m$ resources) and construct products by modelling all possible combinations of arrival and length of stay that fall in this interval. As shown in section 3.6.4 these programs are appropriate for the hotel industry especially when there are multiple products and multiple customer classes.

In our problem we do not restrict products to be of less than a certain length of stay. This means that for an optimisation period of $m$ days the number of products are of order $O(m^2)$. Even worse, if the time period $\Delta T$ is taken to be small, in order to optimise over the same interval of $m$ days we need to use $m/\Delta T$ periods (or resources) resulting in an enormous number of products to analyse.

Also, if we incorporate uncertainty in the problem and thus result with a non-linear program, the linearisation methods will generate an even greater number of variables to manage (as discussed in section 3.6.4).

Finally, it is of our interest to construct dynamic bid-price controls, that is controls which depend on the time and state of the system. Although computationally very efficient, the mathematical programs are static in nature (assume aggregate demand for the products) and thus they will generate dual values that can only serve as static bid-prices for the original problem. Then, this will have to be re-updated frequently in order to perform well.

Therefore, this study investigates whether dynamic bid-prices based on alternative formulations are sufficient in approximating the specified network problem. Three main methods are developed:

1. Monte-Carlo (MC)
   This method solves a more simplified network problem with less dimensions. This model is still formed in discrete-time as the network model (although bookings are made in continuous-time) but for the expected marginal values the explicit difference between carpark size $C$ and capacity remaining $Q$ is suppressed. This model is the simplest model intuitively, however it is more computationally intensive as the solution is based upon Monte-Carlo simulations of the booking patterns. In regards to the existing literature our the MC method could be clustered into the second category of approximations methods, that is the methods based on simplified network models.

2. PDE
   This is a continuous-time approach which assumes that days do not interact with each other, enabling decisions to be made for each day individually. In other words, we could think of this as a network decomposition method where the network problem is reduced into a set of independent single-resource dynamic programs. In 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 but it leads to a partial differential equation (PDE) for which appropriate schemes can be solved relatively fast.

3. Pontryagins
   This is again a continuous-time model where the interdependence within days is also ignored. However, we assumed that the parking spaces are a *continuous* quantity, as opposed to integer spaces. This extra assumption results in a “fully” PDE formulation (both time and space) and turns the

problem into a deterministic one. Thus, we use the Pontryagins maximum principle to obtain the optimal solution. This approach is developed to examine how well a deterministic model can perform in a stochastic environment.

From each method we calculate the expected marginal values, which we use to construct a $2D$ bid-price table with respect to the time left $\tau$ and the capacity remaining $Q$ (see section 4.3.7). Then, the coordinate $(Q, \tau)$ would tell us the bid price we should set on the future day that lies $\tau$ days later when the remaining capacity on that day is $Q$. In this way, the bid prices are defined for a set of possible outcomes (any combination of time left and capacity remaining) and thus there is no need to frequently update them. Finally, the bid-price tables are implemented in the algorithm in section 4.3.8 to access the methods' performances.

**DLP revisited** As already discussed in sections 3.6.2 and 3.6.4, perhaps the most popular mathematical program is the DLP. This is in fact the simplest and most intuitive mathematical program. However, being deterministic, i.e. it only considers the mean customer demand for the different products and ignores all other distributional information, it can lead to poor approximations to the true marginal expected values (see Talluri and Van Ryzin, 1999).

Despite this deficiency, a key aspect of the DLP renders it very important when evaluating the performance of different methods. According to de Boer et al. (2002) the optimal value of the DLP, $V^{LP}$, overestimates the expected outcome of implementing its solution as a booking control policy. More precisely, it follows from Jensen's inequality that $V^{LP}$ provides an *upper bound* on the optimal value of the original problem (see Chen et al., 1998; Cooper and Homem-de Mello, 2007, for details). Therefore, the DLP could be used as an effective benchmark model to test the performance of other more sophisticated methods.

Unfortunately, recall that our setting assumes no length restriction on the products and thus the DLP might become too large to be solved in acceptable time. This is in fact the reason for which this intuitive benchmark has not been included in the results to come.

## 5.1 A discrete-time model based on Monte Carlo (MC)

Let us first a take a simple example to address the size of the network problem. Assume that today is day 5 and we want to maximise the amount of cash realised on the $30^{th}$ day. The initial size $C$ of the carpark for each day is 100 but the remaining capacity $Q$ for each day 5 to 30 is different due to customers that have booked and been accepted from times before day 5. Then, for a customer who books today (on day 5) to park on the future day 30, the system checks the price rate he/she will have to pay, $\Psi(\xi)$ (depending on his/her length of stay $\xi$), against the cash threshold of $\Delta V(Q(30), 5; 30, 100)$ and accepts the request only if

$$\Psi(\xi) \geq \Delta V(Q(30), 5; 30, 100).$$

The quantity on the right is decided after taking into account the availability of spaces on the neighbouring days too, namely

$$\ldots, Q(27), Q(28), Q(29), Q(31), Q(32), Q(33), \ldots$$

If the demand that has already been realised on *each* neighbouring day is less than 100 then *all* neighbouring days should have some spaces still available for sale, implying that any length of stay is theoretically valid.

Alternatively, if the carpark size was only $C = 10$ then the cash threshold to beat would have been $\Delta V(Q(30), 5; 30, 10)$ which should in general be different than $\Delta V(Q(30), 5; 30, 100)$, as some neighbouring days might have already sold out. In this case the customers would be restricted in only requests of particular lengths of stay.

In particular, it might happen that if spaces are sold out on days 29 and 31 the marginal value of the remaining spaces on day 30 is reduced, whereas the opposite holds when there is abundant capacity on those days. This complex structure renders the problem intractable and requires some further simplification.

Below we show the two additional assumptions imposed to make the problem tractable.

1. We assume that customer demand is time-invariant.
   Suppose a booking is made at time period $T^m$ to arrive at the future period $T^k$, with $m \leq k$. Under a time-invariant framework the actual time period

for which the booking happens does not matter, but only the time difference in the periods between booking and arriving, $k-m$. In other words, if the optimal decision uses the expected marginal value $\Delta V(Q,m;k,C)$ this will now be the same as $\Delta V(Q,0;k-m,C)$ effectively reducing the system down by one dimension. The investigation into relaxing this assumption is left to further work.

2. We assume that the expected marginal value $\Delta V(Q(k),m;k,C)$ is calculated based on the fact that all preceding days $m, m+1,\dots,k$ have as many spaces remaining ($=Q(k)$) as for the day $T^k$. Then the problem may be regarded as one where the carpark size $C$ becomes $Q(k)$ for all days $T=1,\dots,k-1$. Thus, the expected marginal value reduces to $\Delta V(Q(k),m;k,Q(k))$.

Combining these two assumptions, the expected value of the carpark of total capacity $C$ with $Q$ spaces remaining for time $T^k$ as observed at time $T^m$, $m \le k$ simplifies to

$$V(Q,m;k,C) = V(Q,0;k-m,Q), \tag{5.1}$$

and the expected marginal values become

$$\Delta V(Q,m;k,C) = \Delta V(Q,0;k-m,Q). \tag{5.2}$$

This simplification reduces the number of dimensions to two hence rendering the problem tractable. Although it does no longer solve the exact network problem, this simplified methodology is a reasonable approximation. Note that the latter can now be interpreted as the expected marginal value of the space for a carpark of size $Q$ and $k-m$ days remaining (this will be revisited below).

The second assumption is crucial to the understanding of our method and thus it requires some further explanation. Recall that under assumption 2 we may calculate the expected value (or the expected marginal value) on day $T$ with $Q$ spaces remaining using a carpark with size $C=Q$ for all days in the optimising horizon.

To see where our assumption stems from, we begin by considering a carpark with (trivial) size $C=1$. To maximise the expected revenues on a day $T$ we need to apply optimal decisions according to the expected values of $Q=0$ and $Q=1$

spaces remaining for that day, $V(Q = 0, 0; T, C = 1)$ and $V(Q = 1, 0; T, C = 1)$. We know that the expected value of 0 spaces remaining is zero, thus we only need to find $V(Q = 1, 0; T, C = 1)$. Notice that Q and C are the same for this carpark thus

$$V(Q = 1, 0; T, C = 1) = V(Q = 1, 0; T, C = Q).$$

Thus the problem reduces down to one where the carpark size and the capacity remaining are both equal to 1; this situation is shown in figure 5.1.

When we move to a carpark of size 2, we maximise expected revenues by applying optimal decisions according to the expected values $V(Q = 0, 0; T, C = 2)$, $V(Q = 1, 0; T, C = 2)$ and $V(Q = 2, 0; T, C = 2)$. As before $V(Q = 0, 0; T, C = 2) = 0$, but what about the other two terms?

Let us look at $V(Q = 1, 0; T, C = 2)$ first. This is the expected value of one space remaining on day $T$ for a carpark of total capacity two. Figure 5.2 illustrates this situation. Indeed it is not clear how to obtain this value at all. Therefore, we propose to approximate it by

$$V(Q = 1, 0; k, C = 2) \approx V(Q = 1, 0; k, C = 1),$$

since the latter value has been already derived from the preceding case of $C = 1$. Recall that this approximation corresponds to a carpark with size one, thus it will be correct when the state of the carpark looks like that in figure 5.3.

Finally, based on the above we may calculate $V(Q = 2, 0; k, C = 2)$ as the expected revenue to be generated for a carpark of total size $C = 2$, namely

$$V(Q = 2, 0; k, C) = V(Q = 2, 0; k, C = Q).$$

Generalising, for the carpark of size $\mathcal{C}$ the expected values for $Q = 1, \ldots, \mathcal{C} - 1$ would have been approximated by the preceding cases of smaller sized carparks, i.e.

$$\begin{aligned} V(Q = 0, 0; T, C = \mathcal{C}) &\approx V(Q = 0, 0; T, C = 0) \\ V(Q = 1, 0; T, C = \mathcal{C}) &\approx V(Q = 1, 0; T, C = 1) \\ &\cdots \\ V(Q = \mathcal{C} - 1, 0; T, C = \mathcal{C}) &\approx V(Q = \mathcal{C} - 1, 0; T, C = \mathcal{C} - 1). \end{aligned} \tag{5.3}$$

Note that this method is expected to work well when at any given time the state of the carpark is reasonably uniform with no great peaks and contrasts on capacity availabilities between neighbouring days. In general, this assumption is expected to work better for small-sized carparks where the peaks cannot be too large.

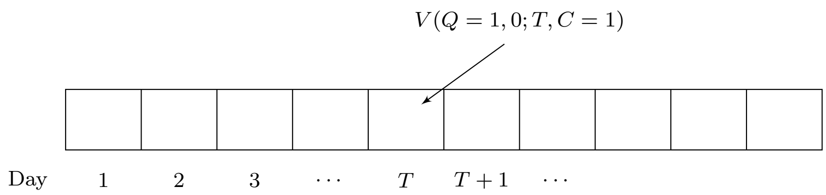



Figure 5.1: An empty carpark with size $C = 1$.

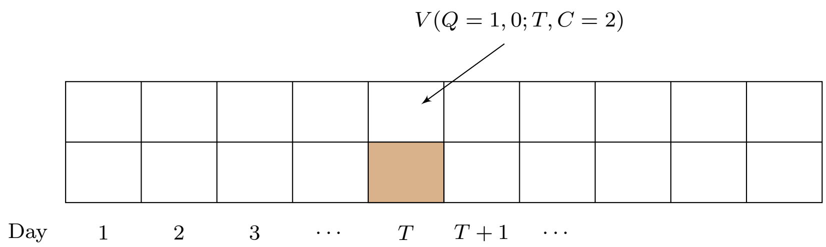



Figure 5.2: A partially filled carpark with size $C = 2$. Brown boxes correspond to spaces that have already been sold while white boxes are the remaining available ones.

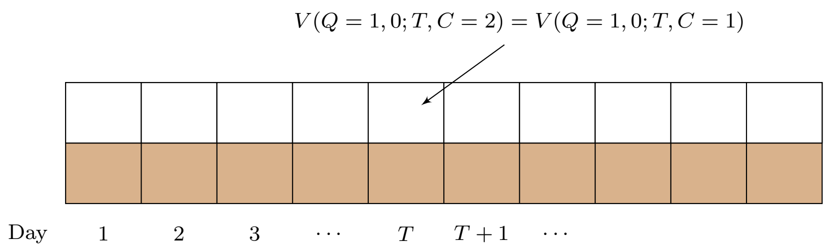



Figure 5.3: A semi-filled carpark with size $C = 2$ but only $Q = 1$ spaces available for all days. Brown boxes correspond to spaces that have already been sold while white boxes are the remaining available ones.

### 5.1.1 Numerical scheme

Define the value matrix $v$. This is a $2D$ array composed of the expected values at every time and state of the carpark. In particular, we have

$$V(Q_j, 0; k-m, Q_j) := v_j^{k-m} \tag{5.4}$$

Without loss of generality, under the time-invariant setting this may also be interpreted as the expected value of the carpark with $Q$ spaces remaining and $k-m$ periods remaining until all spaces are occupied. This observation here is crucial; it indicates the manner at which the expected revenues generated for a carpark of size $C$ on day $T^k$ will be translated into the expected values of a carpark with $Q = C$ spaces remaining and $\tau = T^k$ time left, in order to form the bid-price rejection policy.

From a simulated combined booking set (which may be referred to as a *path*), we can calculate the corresponding optimal value function. However, the optimal solution is tied to the path followed. Thus, in order to approximate the correct optimal solution we proceed using Monte-Carlo simulation, i.e. by simulating and averaging over thousands of such paths..

The procedure to determine the optimal solution $v$ and the optimal rejection policy depends upon iterations.

### 5.1.2 Rejection policy algorithm

1. Choose a booking horizon $[0, T]$ with $K$ periods sufficiently long to capture nearly all of bookings in each customer set and a maximum capacity for the carpark $C$.

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 estimate for $v$, we may denote it as $v^0$, where $v^r$ is the $r^{th}$ guess at the solution.

3. Choose a sufficiently large number $P$ and use Monte-Carlo to generate $P$ booking sets within the pre-specified time interval $[0, T]$.

4. Evaluate the expected value of the carpark at all time periods and all possible capacities $0 \leq Q_j \leq C$ (beginning with capacity 0) to generate the

matrix

$$v_j^{k,r+1} = \mathbb{E}\left[\sum_{i=i^*} f(B^i,k)\Psi(D^i)\right], \tag{5.5}$$

given that for the $i^{th}$ booking made in the period $T^m$ the added value is

$$EAMV = \sum_k f(B^i,k)\left[\Psi(D^i) - \left(v_j^{k-m,r} - v_{j-1}^{k-m,r}\right)\right]. \tag{5.6}$$

5. Go to step 3 and repeat until $||v^{r+1}-v^r|| < \epsilon$, where $\epsilon$ measures the tolerance level we are willing to accept.

Note:

(a) In step 4 the EAMV works by using the set of expected marginal values as follows; let us take for example, a customer who books on day 2 to arrive on day 6 and stay for 3 days. Let the spaces availability on day 6, 7 and 8 be 40, 25, 31 respectively. Then, in order for this customer to be accepted the total price of the 3-day stay should satisfy

$$3\Psi(3) \geq \underbrace{(v_{40}^{6-2} - v_{40-1}^{6-2})}_{\text{day 6}} + \underbrace{(v_{25}^{7-2} - v_{25-1}^{7-2})}_{\text{day 7}} + \underbrace{(v_{31}^{8-2} - v_{31-1}^{8-2})}_{\text{day 8}}.$$

Also to optimise a size $C$ carpark we would have used the solution for the expected marginal values for all smaller carparks $0, 1, \ldots, C-1$. In other words, we start off with a carpark of size $C = 1$ and maximise the expected revenues for all days in $[0, T]$. Then, the resulting revenues are *interpreted* as the expected values of a carpark with $Q = 1$ spaces remaining and $\tau = T^k$ time left, $v_1^k$ for every $k = 0, 1, \ldots, K$. Thus, the opportunity costs for the case of $Q = 1$ are generated as

$$\Delta v_1^k = v_1^k - v_0^k \quad \forall k.$$

When, moving onto the case of a size 2 carpark the optimisation would take into account the expected marginal values (opportunity costs) at $Q = 1$. From the resulting expected revenues we would obtain the expected marginal values for $Q = 2$. Thus, we will move to the case of a size 3 carpark where we will now need the expected marginal values for $Q = 1, 2$. In this manner, we gradually build up expected marginal values for the case of the carpark of maximum size $C$.

(b) We use the same paths at every iteration, to guarantee that at the $p^{th}$ path the marginal value is non-negative (i.e. $\Delta V^p(Q, t; T) \geq 0$) so that the estimate on the expected marginal value is non-negative as well,

$$\frac{1}{P}\sum_{p=1}^{P} \Delta V^p(Q, t; T) \geq 0.$$

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 stepping into 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, the relaxation parameter $\omega$ should now be a number between $0 \leq \omega < 1$. In this manner, the solution should steadily improve at every iteration. As an extra measure for reducing the variance from the paths, we propose a solution whereby the number of paths is increased by approximately $\sqrt{2}$ at every iteration. Therefore a stopping criterion on the maximum number of paths, $P_{\max}$, is imposed in order to prevent the algorithm from excessive use of time and RAM.

### 5.1.3 Successive under-relaxation rejection policy algorithm

1. Choose a booking horizon $[0, T]$ with $K$ periods sufficiently large to capture nearly all of bookings for each customer set and a maximum capacity for the carpark $C$.

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 to $v$ we denote it as $v^0$, where $v^r$ is the $r^{th}$ guess at the solution.

3. Choose the number of paths $P$ and the relaxation parameter $\omega$.

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

5. Evaluate the expected value of the carpark at time $T^k$ for all time periods $T^m$, $m = 0, 1, \ldots, k$ and all possible capacities $0 \leq Q_j \leq C$ (beginning from capacity 0) to generate the matrix

$$y_j^k = \mathbb{E}\left[\sum_{i=i^*} f(B^i, k)\Psi(D^i)\right], \tag{5.7}$$

given that for the $i^{th}$ booking made in the period $T^m$ the added value is

$$A = \sum_k f(B^i, k)\left[\Psi(D^i) - \left(v_j^{k-m,r} - v_{j-1}^{k-m,r}\right)\right]. \tag{5.8}$$

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

$$v_j^{k,r+1} = v_j^{k,r} + \omega\,(y_j^k - v_j^{r,k}). \tag{5.9}$$

7. Go to step 4 $P$ times.

8. Go to step 3, increase $P$ and decrease $\omega$, and repeat until $||v^{r+1} - v^r|| < \epsilon$ or until $P > P_{\max}$, whichever occurs first.

Note: A reasonable choice for $\omega$ 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 path are used directly in the evaluation for the next path. Under these assumptions we can show *experimentally* that the algorithm converges to the true solution in a finite number of iterations.

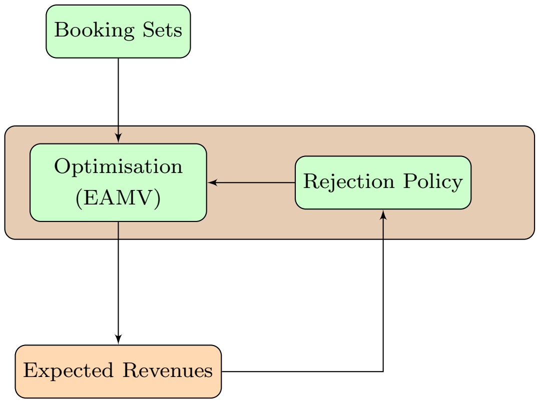



Figure 5.4: Schematic overview of the MC procedure

In figure 5.4 we illustrate a schematic overview of the program flow. At the initial phase of the program a booking set is generated through simulation. The rejection policy is initially set to zero. Next, we apply optimisation to the bookings using the $EAMV$ algorithm and we perform this on every simulated booking set. Hence the resulting expected revenues $v$ are calculated. The last phase of the program is to update the rejection policy by computing the expected marginal values $\Delta v$ and to re-run the optimisation algorithm. This iterative scheme terminates once the difference between the newly derived revenues and the previous ones is sufficiently small. Once the algorithm has terminated, the final set of expected marginal values is computed and defines the $MC$ bid-price rejection policy.

## 5.2 Probability Distributions

### 5.2.1 Non-stationary derivations of the distributions

Previously, we described the simulation process for a set of bookings that was characterised by the intensity parameters denoted $\lambda_b$, $\lambda_a$ and $\lambda_s$, using exponential inter-arrival times. Then working solely with the simulated bookings we derived an algorithm that generates the set of expected displacement costs for all times and capacities remaining.

In the next two sections, we will be looking at two other methods that are formulated in continuous time. We do realise that the vast literature assumes a discrete-time setting since most managers in reality would update their decisions at discrete points in time. However, we believe that a continuous-time formulation might now be more suitable as internet and e-commerce enable decisions to be made automatically and frequently.

Instead of taking simulations we do now properly define the underlying probability distributions for the arrival and stay plus any other relevant distributions that may be needed.

We note that although all of our models that we tested are time-invariant we prefer to present the probability distributions in their most general case, i.e. by taking into account their explicit dependence in time. Then, we can show how these are simplified once time-invariance is introduced.

Recall that each booking has three time parameters, the time of the booking, the time of the arrival and the time of the departure. The explicit reference to the departure time may be dropped once we include the required length of stay.

Assume that we concentrate on the bookings which are made on day $t$. For each of these bookings there is an associated pre-booking time $\eta$ and length of stay $\xi$ at the carpark. Thus, they can be represented in a $2D$ array showing the breakdown of bookings with respect to different durations of stay and various pre-bookings times. The sum of all the elements gives the total number of bookings made at day $t$. By scaling the array by the total number of bookings we can obtain statistical information for both $\eta$ and $\xi$. Let us define the *joint* probability density function $\phi(\eta, \xi; t)$ for bookings at $t$ with respect to $\eta$ and $\xi$ such that

$$\int_0^\infty \int_0^\infty \phi(\eta, \xi; t)\, d\eta\, d\xi = 1. \tag{5.10}$$

The density function $\phi(\cdot)$ gives the probability density of a customer arriving $\eta$ days after booking at $t$ and staying for $\xi$ days and it forms the basis for all probability distributions derived thereafter. In particular, we may define the probability density of arrival $\eta$ days after the booking made at $t$ as

$$\rho_a(\eta; t) = \int_0^\infty \phi(\eta, \xi, t)\, d\xi, \tag{5.11}$$

and, similarly, the probability density of staying $\xi$ days as

$$\rho_s(\xi; t) = \int_0^\infty \phi(\eta, \xi, t)\, d\eta. \tag{5.12}$$

Given these quantities, we may denote the cumulative probability density of arrival not more than $\eta$ days after booking as

$$P_a(\eta; t) = \int_0^\eta \rho_a(\eta'; t)\, d\eta' \tag{5.13}$$

and the cumulative probability density of staying not more than $\xi$ days as

$$P_s(\xi; t) = \int_0^\xi \rho_s(\xi'; t)\, d\xi'. \tag{5.14}$$

Let us now consider the probability that a customer departs from the carpark exactly $z$ days after making the booking at $t$. If we denote this by $\rho_d(z; t)$, we may write:

$$\rho_d(z; t) = \int_0^z \phi(\eta, z - \eta, t)\, d\eta. \tag{5.15}$$

In this expression we make sure that we sum over all instances where the length of stay plus the arrival time is equal to $z$. Consequently, the cumulative density of departure is given by

$$P_d(z, t) = \int_0^z \rho_d(z', t)\, dz'. \tag{5.16}$$

Our objective will be to maximise the revenue rate over a future time $T$. Thus,

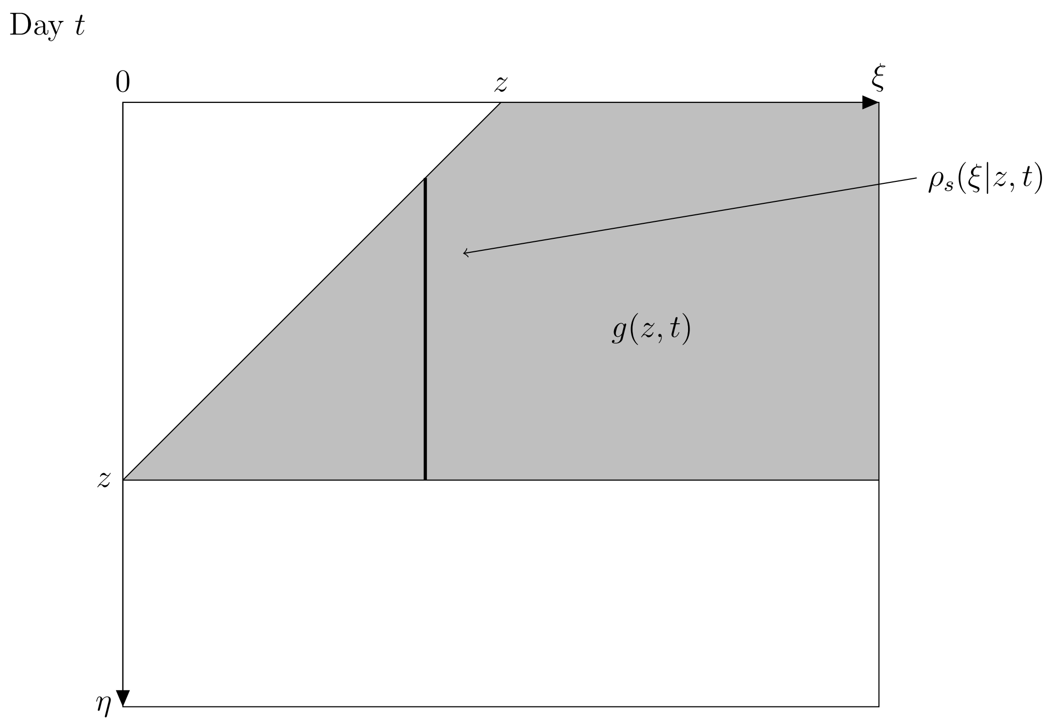



Figure 5.5: Breakdown of bookings for a particular day $t$, with respect to different durations of stay, $\xi$, and various pre-bookings times, $\eta$.

it makes sense to derive a quantity that will describe the number of people that are *present* in the carpark at $T$. In fact, these customers may have either arrived exactly at $T$ and are staying for any length of stay, or anytime $t$ before $T$ but they will stay for at least $z = T - t$ days.

Thus, we define the probability of *being present* in the carpark $z$ days after booking at $t$, namely $g(z, t)$, as

$$g(z,t) = \int_{0}^{z} \int_{z-\eta}^{\infty} \phi(\eta, \xi, t)\, d\xi\, d\eta. \tag{5.17}$$

Figure 5.5 shows the breakdown of bookings of a particular day $t$ with respect to the different durations of stay and pre-booking times. In this figure the probability $g(z, t)$ is found by integrating $\phi(\cdot)$ for the area under the shaded region. This probability takes into account customers that are going to be present $z$ days after their booking at $t$ irrespective of when they have arrived at the carpark or when they are about to depart. Thus, $g(z, t)$ will be referred to as the *occupancy* probability.

In particular, we can show that $g(z,t)$ satisfies

$$g(z,t) = P_a(z,t) - P_d(z,t). \tag{5.18}$$

Intuitively, the occupancy probability says that a customer is present $z$ days after the booking made at $t$ only if he arrives not more than $z$ days after the booking ($P_a(z,t)$) and does not depart before this time ($-P_d(z,t)$). In figure (5.5), the triangular region on the left and above the $g(z)$ correspond to customers that will depart by time $z$ and the rectangular region that lies below $g(z)$ corresponds to bookings that will not have arrived yet.

Furthermore, we are interested in finding the probability of staying for $\xi$ days *given* that the customer is present $z$ days after the booking made at $t$. This conditional probability will be denoted by $\rho_s(\xi|z,t)$ and it is given by,

$$\rho_s(\xi|z,t) = \frac{\int_{(z-\xi)^+}^{z} \phi(\eta,\xi,t)\, d\eta}{g(z,t)}. \tag{5.19}$$

This conditionality inside the probability is to differentiate it from the original probability density $\rho_s(\xi)$ which corresponds to all bookings made on day $t$. In fact, the conditional probability only accounts for the possible durations of customers that will definitely be present on that future time $T = t + z$.

Therefore, the cumulative probability density of a customer staying at most $\xi$ days given that he is present $z$ days after the booking made at $t$, is given by

$$P_s(\xi|z,t) = \int_0^{\xi} \rho_s(\xi'|z,t)\, d\xi'. \tag{5.20}$$

This cumulative probability density gives the *proportion* of customers present $z$ days after the booking is made at $t$, such that their duration of stay does not exceed $\xi$ days. This term will be vital in deriving the rejection policy for the PDE model and thus it will be revisited in the next section.

### 5.2.2 Adjusting for a time-invariant setting

The above densities are derived for the general non-stationary case in which the customers average booking intensities ($\lambda_b$) as well as the arrival and staying probability distributions depend on the current time $t$.

In this thesis we assume for simplicity that customers booking, arrival and staying intensities are time-invariant, i.e. they do not depend on the current time we calculate them.

Recall for each customer class $n$ we assume that bookings follow a Poisson process with *stationary* intensity $\lambda_{b_n}$. Moreover, the probability densities of $\eta$ and $\xi$ for the $n^{th}$ customer class are assumed to be *stationary* exponential distributions and they are independent of each other. Therefore, we have

$$\rho_{a_n}(\eta) = \lambda_{a_n} e^{-\lambda_{a_n}\eta} \quad \forall\, n \tag{5.21}$$

and

$$\rho_{s_n}(\xi) = \lambda_{s_n} e^{-\lambda_{s_n}\xi} \quad \forall\, n. \tag{5.22}$$

Using equation (4.2) and the stationary assumptions of the intensities we may drop the time-dependence of equations (5.11) and (5.12) and express them as

$$\rho_a(\eta) = \sum_n \alpha_n \rho_{a_n}(\eta) \tag{5.23}$$

and

$$\rho_s(\xi) = \sum_n \alpha_n \rho_{s_n}(\xi), \tag{5.24}$$

where the weight

$$\alpha_n = \frac{\lambda_{b_n}}{\sum_j \lambda_{b_j}} \quad \forall\, n$$

follows from the superposition property of the Poisson process and measures the probability of the next booking to be from booking class $n$.

Finally, the joint density $\phi(\cdot)$ itself is time-invariant and, consequently, we may use the relations (5.23) and (5.24), to express $\phi(\cdot)$ as

$$\phi(\eta, \xi) = \sum_n \alpha_n \rho_{a_n}(\eta)\rho_{s_n}(\xi). \tag{5.25}$$

We note that the assumption imposed on the distributions of arrivals and stays being independent enables us to write the joint density $\phi(\cdot)$ as the product of the individual densities for $\eta$ and $\xi$. In addition, the definition in (4.2) simplifies the structure of the model, as we can build the joint density by adding up the individual distributions of all customer classes (with appropriate weights). Such a typical density function may be seen in figure 5.6.

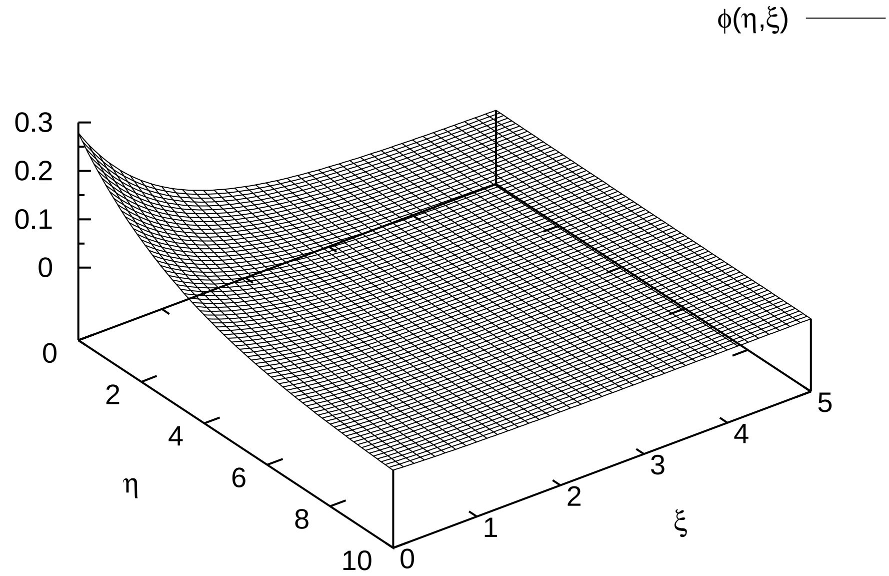



Figure 5.6: Joint density $\phi(\eta, \xi)$ as a function of the pre-booking time $\eta$ and the length-of-stay $\xi$. Two classes, 1 and 2, are considered with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$, $\lambda_{s_2} = 1$.

Moving on, the cumulative densities for arrival and stay, (5.13) and (5.14), become

$$P_a(\eta) = 1 - \sum_n \alpha_n e^{-\lambda_{a_n}\eta} \tag{5.26}$$

and

$$P_s(\xi) = 1 - \sum_n \alpha_n e^{-\lambda_{s_n}\xi}, \tag{5.27}$$

respectively.

Furthermore, the probability density of departure (5.15) is

$$\begin{aligned}
\rho_d(z) &= \sum_n \alpha_n \rho_{d_n}(z) \\
&= \sum_n \alpha_n \int_0^z \rho_{a_n}(t)\rho_{s_n}(z-t)\,dt \\
&= \sum_n \alpha_n \frac{\lambda_{a_n}\lambda_{s_n}}{\lambda_{s_n} - \lambda_{a_n}} \left(e^{-\lambda_{a_n}z} - e^{-\lambda_{s_n}z}\right).
\end{aligned} \tag{5.28}$$

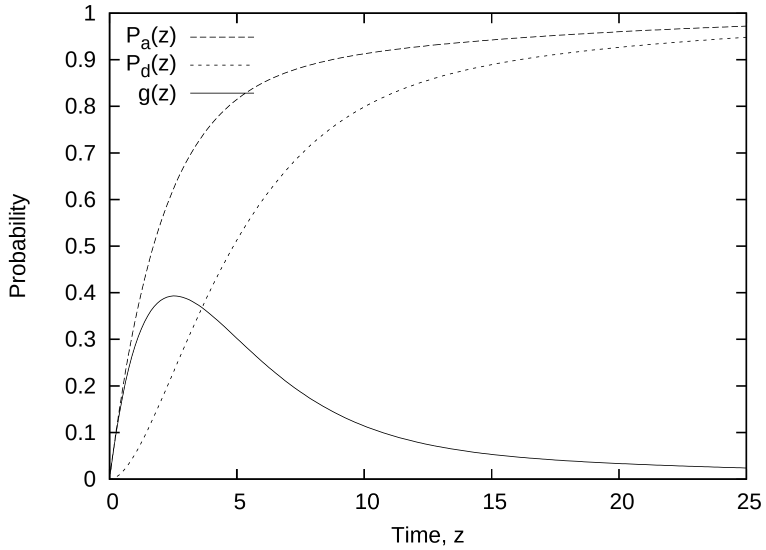



Figure 5.7: $P_a(z,t), P_d(z,t)$ and $g(z,t)$ for a given day $t$. Two classes, 1 and 2, are considered with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/2$ and $\lambda_{s_2} = 1/3$.

and, as such, the corresponding cumulative density (5.16) becomes

$$P_d(z) = 1 - \sum_n \alpha_n \left[ \frac{\lambda_{a_n} e^{-\lambda_{s_n} z} - \lambda_{s_n} e^{-\lambda_{a_n} z}}{\lambda_{a_n} - \lambda_{s_n}} \right]. \tag{5.29}$$

Consequently, the occupancy probability $g(z,t)$ reduces to

$$g(z) = P_a(z) - P_d(z). \tag{5.30}$$

Figure 5.7 presents the cumulative probabilities of arrival, $P_a(z,t)$, and departure, $P_d(z,t)$, and the occupancy probability $g(z,t)$ on a given day $t$. The cumulative probability $P_d(z,t)$ always lies below $P_a(z,t)$, since a departure occurs only if it is preceded by the corresponding arrival. Thus, $g(z)$ is non-negative with a distinct maximum occurring at the point where the distance between the distributions $P_a(z,t)$ and $P_d(z,t)$ is maximum.

Finally, the conditional density of stay in $\rho_s(\xi|z;t)$ simplifies to (appendix A)

$$\rho_s(\xi|z) = \sum_n \alpha_n \frac{\rho_{s_n}(\xi)\Big(P_{a_n}(z) - P_{a_n}((z-\xi)^+)\Big)}{g(z)} \tag{5.31}$$

and thus the resulting cumulative density becomes

$$P_s(\xi|z) = \int_0^{\xi} \rho_s(\xi'|z)\, d\xi'. \tag{5.32}$$

Figure 5.8 shows the conditional probability densities $\rho_s(\xi|z,t)$ and $P_s(\xi|z,t)$ on a given day $t$ as functions of $\xi$ for different lead times $z$. The solid line in the upper figure corresponds to the total probability density of stay $\rho_s(\xi,t)$ whereas the solid line in the lower figure to the cumulative probability density of stay $P_s(\xi,t)$.

In order to compute equation (5.32) we may use numerical integration to evaluate the integral, or compute it directly using its (rather cumbersome) analytical form which may be found in appendix B.

As mentioned before, the importance of this conditional distribution in the model will be made clear in the next section, where we derive the PDE.

## 5.3 A stochastic PDE model

Now that we have the probability distributions in place we can proceed to derive the partial differential equation for the revenue generated in the carpark. Previously, we have defined the $V(Q,m;k)$ as the expected revenue generated from bookings made after time $t^m$ for $T^k$. Now, with a slight change in notation we define $V(Q,t;T)$ as the instantaneous rate at which revenue is generated at time $t$ over the instant $T$.

Given that we are at time $t$ our aim is to optimally sell the carparking spaces of some future time $T$ so that to maximise the expected revenues for that instant $T$. We assume that all intermediate days do not affect the solution at $T$, i.e. that there is only a single-resource (the instant $T$) to optimise.

Let, $Q(t;T)$ be the number of spaces remaining for time $T$ as of $t$. Then our assumption implies that the expected value rate at $T$ depends only on the current time $t$ and on the available spaces for that time instant $Q(t;T)$ only, with zero dependence on the availabilities of the neighbouring times $Q(t;\mathcal{T})$ for any $\mathcal{T} \in [t,T]$.

This is in fact the only additional assumption that has to be imposed so that we can easily formulate the continuous-time model. In reality such an assumption can be reasonable when most bookings are for a single-day stay only; when

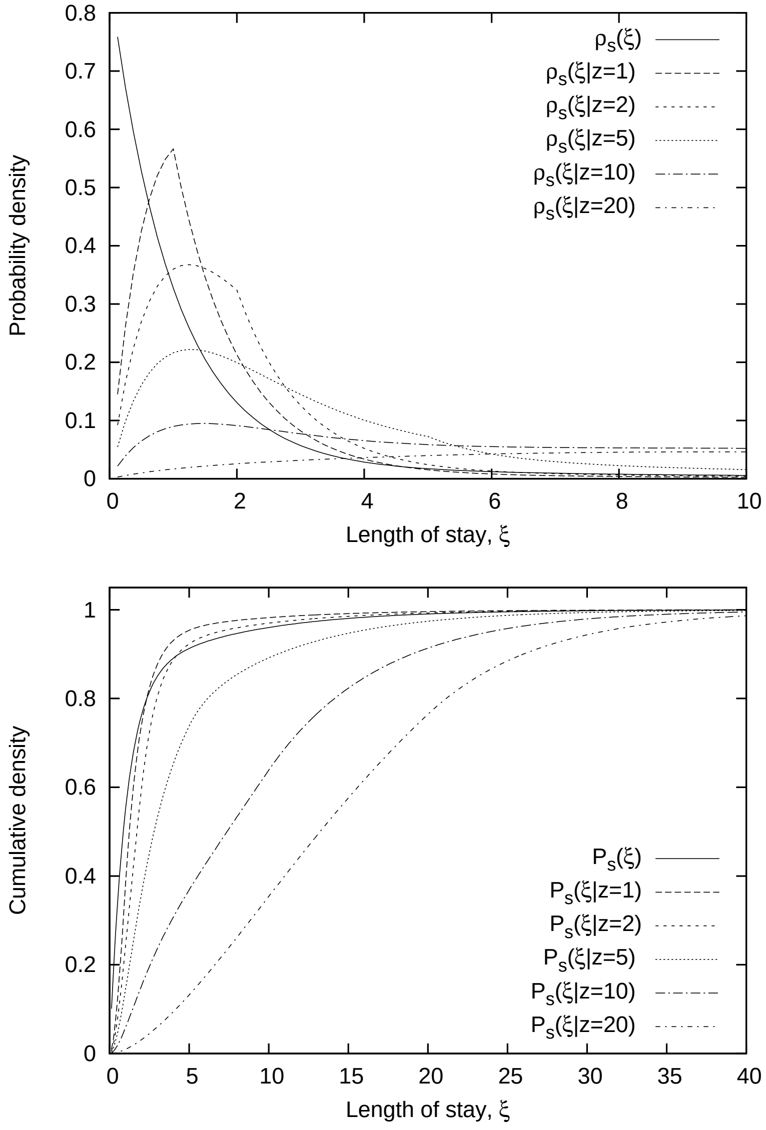



Figure 5.8: Conditional probability and cumulative density of stay, $\rho_s(\xi|z)$ and $P_s(\xi|z)$ on a given day $t$. Two classes, 1 and 2, are considered with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$, $\lambda_{s_2} = 1$.

intermediate effects from the nearby days might not be significant (see Ladany, 1976; Bitran and Gilbert, 1996).

Although we imposed this simplification to the problem, note that we still deal with multiple-day requests whereby customers can book for any duration of stay $\xi$ (which is a continuous quantity). This in some respect, distinguishes our work from the existing literature.

The average intensity at which bookings are made at time $t$ to be present over the instant $T$, $f(t;T)$, is given by

$$f(t;T) = \left( \sum_n \lambda_{b_n} \right) g(T-t,t). \tag{5.33}$$

Then $Q$ may be regarded as a jump process with jump size $-1$ (corresponding to one sale), with $Q(t=-\infty;T) = C$ be the initial carpark size. In particular, over the next time interval $dt$, we sell one space with probability $f(t;T)\,dt + o(dt)$, we do not sell a space with probability $1 - f(t;T)\,dt - o(dt)$ 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 $Q$ as

$$dQ = \begin{cases} 0 & \text{with probability } 1 - f(t;T)\,dt \\ -1 & \text{with probability } f(t;T)\,dt. \end{cases} \tag{5.34}$$

Therefore, according to equation (2.40) the differential of $V(\cdot)$ becomes

$$\mathrm{E}\left[dV(Q,t;T)\right] = \frac{\partial V(Q,t;T)}{\partial t}\,dt + f(t;T)\Big(V(Q-1,t;T) - V(Q,t;T)\Big)\,dt. \tag{5.35}$$

On the other hand, the expected change in the value rate is given by

$$\mathrm{E}\left[dV(Q,t;T)\right] = -f(t;T)\,dt \int_0^\infty \rho_s(\xi|T-t,t)\,\Psi(\xi)\,d\xi, \tag{5.36}$$

the average cashflow rate per customer at $t$ generated from bookings present at $T$ multiplied by the expected demand intensity. The minus sign indicates 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 realised as generated revenue from customers 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.

Equating (5.35) and (5.36) we obtain the differential equation for the rate at which cashflow is generated at $t$ for cars present at $T$

$$\frac{\partial V}{\partial t} + f(t;T)\Big(V(Q-1,t;T) - V(Q,t;T)\Big) = -f(t;T)\int_0^{\infty} \rho_s(\xi|T-t,t)\,\Psi(\xi)\,d\xi \tag{5.37}$$

Equation (5.37) has a corresponding *final* condition that at time $T$ the remaining value rate is zero. Also there is a boundary condition on the capacity stating that the value remaining for time $T$ is zero when there are no more free spaces in the carpark for that day. Mathematically, the two conditions are expressed as

$$V(Q,T;T) = 0 \quad \forall Q \tag{5.38}$$

$$V(0,t;T) = 0 \quad \forall t. \tag{5.39}$$

Therefore, the PDE in (5.37) is solved backwards in time and its solution gives the expected rate at which revenue is generated for the instant $T$ as observed at time $t$. This solution corresponds to a carpark where no optimisation policy is in effect and there is no restriction on the duration of stay for the customers.

### 5.3.1 A time-invariant model

In a time-invariant setting all days are the *same* in the sense that the probability distributions are kept identical, irrespective of the day we are at ($t$) or the day we target to optimise ($T$). Then there is no point in having explicit reference to either $t$ or $T$ as the only timescale that matters the most is the time lag between them $\tau = T - t$. If then $f(\tau)$ denotes the expected intensity of bookings to be present $\tau$ days later, the resulting model is

$$\frac{\partial V}{\partial \tau} + f(\tau)\Big(V(Q,\tau) - V(Q-1,\tau)\Big) = f(\tau)\int_0^{\infty} \rho_s(\xi|\tau)\,\Psi(\xi)\,d\xi, \tag{5.40}$$

where the $V(Q,\tau)$ is now defined as the rate at which value is generated for a future instant when there is $\tau$ time remaining. The corresponding *initial* and boundary conditions are given by

$$V(Q,0) = 0 \quad \forall Q \tag{5.41}$$

$$V(0,\tau) = 0 \quad \forall \tau. \tag{5.42}$$

### 5.3.2 Rejection policy

For the stationary model in (5.40) we are able to derive a rejection policy that is based on the length of stay of the customers. In fact, to maximise the expected value rate we accept a customer only if its corresponding price rate $\Psi$ is higher than or equal to some optimal minimum price rate $\Psi^*$ set by the carpark manager. Since, there is a one-to-one correspondence between the price $\Psi(\cdot)$ and the length of stay $\xi$, with the price rate monotonically decreasing in $\xi$, we can apply the rejection policy with regards to the latter. In other words, we accept a customer only if its duration of stay $\xi$ is *less than or equal to* some optimal maximum duration of stay $\xi^*$.

In fact, the proportion of customers that are present $\tau$ days later and do not stay for more than $\xi^*$ days is precisely given by $P_s(\xi^*|\tau)$ which has been derived in (5.32) previously.

Then the intensity of bookings that require to be present $\tau$ days later *and* will get accepted is given by

$$f(\tau)\, P_s(\xi^*|\tau). \tag{5.43}$$

The term in (5.43) may be interpreted as the *instantaneous booking acceptance rate* for customers present $\tau$ days later.

Similarly, the average cashflow rate *per customer* generated from the *accepted* customers that are present $\tau$ days later is given by

$$\frac{\int_0^{\xi^*} \rho_s(\xi|\tau)\, \Psi(\xi)\, d\xi}{P_s(\xi^*|\tau)}. \tag{5.44}$$

Multiplying the number of accepted customers (5.43) with the average price per accepted customer (5.44) we obtain the average instantaneous cashflow rate generated from accepted customers present $\tau$ days later,

$$f(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\, \Psi(\xi)\, d\xi. \tag{5.45}$$

Therefore, the PDE becomes

$$\frac{\partial V}{\partial \tau} + f(\tau) P_s(\xi^*|\tau)\Big(V(Q,\tau) - V(Q-1,\tau)\Big) = f(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\, \Psi(\xi)\, d\xi. \tag{5.46}$$

We can express (5.46) as a HJB equation (see section 2.3.3), where the control is the duration of stay $\xi$. In particular, we have

$$\frac{\partial V}{\partial \tau} = \max_{\xi} \left\{ f(\tau) P_s(\xi|\tau) \Big( V(Q-1,\tau) - V(Q,\tau) \Big) + f(\tau) \int_0^{\xi} \rho_s(\xi'|\tau)\, \Psi(\xi')\, d\xi' \right\}, \tag{5.47}$$

with the boundary conditions

$$V(Q,0) = 0 \quad \forall Q \tag{5.48}$$
$$V(0,\tau) = 0 \quad \forall \tau. \tag{5.49}$$

The solution to the optimisation problem in equation (5.47) is the optimal value rate $V(Q,\tau)$ and the values $\Psi(\xi^*)$, with $\xi^* = \xi^*(Q,\tau)$, that achieve the supremum form the optimal rejection policy.

Since the model in equation (5.47) is based upon the stochastic process $Q$ we may refer to it as the *stochastic PDE* or simply the *PDE approach*.

The PDE model seems to be closer to a dynamic pricing problem seen in the literature because the price varies continuously as a result of changing the maximum allowed stay $\xi$. 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.

Let us now differentiate (5.46) with respect to the control $\xi^*$ to obtain

$$V(Q,\tau) - V(Q-1,\tau) = \Psi(\xi^*). \tag{5.50}$$

This is equivalent to the condition (4.22) derived in the discrete-time case and it verifies that the optimal price charged should equal the corresponding opportunity cost. The only difference is that the PDE model regards each day individually and optimises with no reference to the neighbouring days.

Therefore, the optimal rejection policy for a given day $T$ can be alternatively expressed as a dynamic bid-price table[1] where the bid price is equal to the expected marginal value

$$\pi(Q,\tau) = \Delta V(Q,\tau), \tag{5.51}$$

[1]For a small but finite choice of $\Delta\tau$.

for every $Q = 0, 1, \ldots, C$ and $\tau \in [0, T]$.

### 5.3.3 Numerical scheme

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

Firstly, we construct the mesh. In this stationary case the mesh has only two dimensions, the advance-time $\tau$ and the capacity remaining $Q$. Suppose that the domain we will work on is rectangular with $\tau$ ranging from 0 to $T$ and $Q$ ranging from 0 to $C$. Divide [0, $T$] into $K$ equally spaced intervals at $\tau$ values indexed by $k = 0, 1, \ldots, K$. Similarly, we divide [0, $C$] into $C$ equally spaced intervals at $Q$ 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 $\Delta Q = 1$ in the state direction such that $\tau^k = k\Delta\tau \;\forall k$ and $Q_j = j \;\forall j$. We seek an approximation to the values of $V$ at the $(K+1)\times(C+1)$ grid points.

Therefore,

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

where $v$ is a $2D$ array.

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

Then, we may approximate the conditional probability distribution $P_s(\cdot)$ by

$$P_s(\xi^i|\tau^k) = P_s(i\Delta\xi|k\Delta\tau) \approx p_i^k.$$

Moreover, the average intensity in equation (5.33) may be written as

$$f(\tau^k) = f(k\Delta\tau) \approx f^k.$$

Consequently, the integral term on the RHS of the equation (5.46) may be written as

$$\int_0^{\xi^i} \rho_s(\xi'|\tau^k)\,\Psi(\xi')\,d\xi' = \int_0^{i\Delta\xi} \rho_s(\xi'|k\Delta\tau)\,\Psi(\xi')\,d\xi' \approx r_i^k$$

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

$$v_j^{k+1} = v_j^k + f^k \max_i \left\{ p_i^k \left( v_{j-1}^k - v_j^k \right) + r_i^k \right\} \Delta \tau,$$

with the boundary conditions

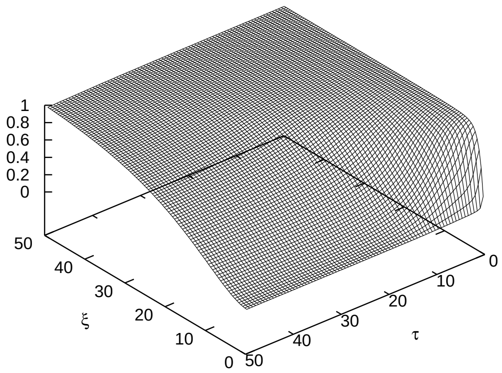
$$v_j^0 = 0 \quad \forall j \tag{5.52}$$

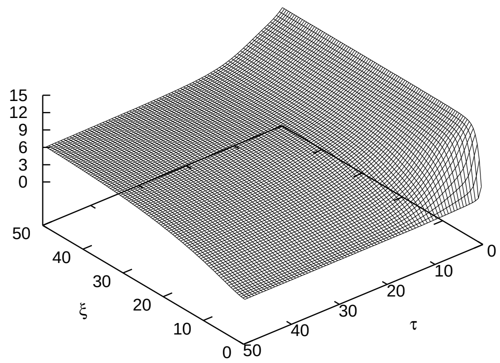
$$v_0^k = 0 \quad \forall k. \tag{5.53}$$

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.

### 5.3.4 Exhaustive PDE algorithm

Under this scheme, the calculation of the optimal solution depends on iterating for the optimal value $\xi^*$ at each time and state point. It performs an exhaustive linear search for the optimal $\xi$ value. In particular, for every point in the grid it searches through all possible length of stays (all $i$'s), one at a time, until it identifies the one that maximises the value function. Note that each value $p_i^k$ requires solving an integral (equation 5.32) or evaluating the complex analytical form derived in appendix B. Similarly, each value $r_i^k$ requires solving the integral term on the RHS of the equation (5.46). Thus, it will be very time inefficient to calculate the $p_i^k$ and $r_i^k$ values every time we need them. Therefore, we find it convenient to calculate all values $p_i^k$ and $r_i^k$ for all $i, k$ at the beginning and store them into $2D$ arrays denoted by $p$ and $r$, respectively. Appendix C illustrates a novel approach as to the manner at which these calculations can be performed. The resulting matrices $p$ and $r$ will be of the form shown in figure 5.9.




$(p)$ $(r)$

Figure 5.9: 2D matrices $p$ (left) and $r$ (right) as functions of the length-of-stay $\xi$ and time left $\tau$, with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$, $\lambda_{s_2} = 1$.

The algorithm is explained below:

1. Calculate the integrands and fill in the arrays $p$ and $r$.
2. Set the time $\tau^k$ starting from $k = 0$ (initial condition) to $k = K$
3. Set the capacity $j$ starting from $j = 0$ (boundary condition) to $j = C$
4. Calculate

$$v_j^k = v_j^{k-1} + f^{k-1} \max_i \left\{ p_i^{k-1} \, (v_{j-1}^{k-1} - v_j^{k-1}) + r_i^{k-1} \right\} \Delta\tau$$

5. Go to step 3 $C$ times.
6. Go to step 2 $K$ times.

The exhaustive PDE algorithm is guaranteed to find the optimal solution. However, searching through to find the optimal value in this manner may potentially be time consuming as it depends on the number of discretised points $I$ it has to go through.

### 5.3.5 Value-policy PDE algorithm

Next, we propose a more efficient method that can identify the optimal $\xi$ without the need of any iterations. This idea is actually simple and it is based on the optimal condition (5.50). In particular, through this relationship the optimal $\xi^*$ value at a grip point may be directly calculated by

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

since $\Psi(\cdot)$ is an invertible function. As such we end up with the optimal policy by completely avoiding any iterations. The proposed algorithm is as follows:

1. Calculate the integrand and fill in the matrix $r$.
2. Set the time $\tau^k$ starting from $k=0$ to $k=K$.
3. Set the capacity $j$ starting from $j=0$ (boundary condition) to $j=C$
4. Set $\xi$ using
$$i = \xi_j^k = \Psi^{-1}(v_j^{k-1} - v_{j-1}^{k-1})$$
5. Calculate
$$v_j^k = v_j^{k-1,r} + f^{k-1}\Big(P_s(\xi_j^k|\tau^{k-1})\,(v_{j-1}^{k-1,r} - v_j^{k-1,r}) + r_i^{k-1}\Big)\,\Delta\tau$$
6. Go to step 3 $C$ times
7. Go to step 2 $K$ times.

This method is expected to perform much faster than the exhaustive linear search. Note that in contrast to the exhaustive PDE algorithm, here we can use the analytical form of $P_s(\xi|\tau)$ because there will be no iterations to slow the process down. This achieves a further reduction in computation time as only the array $r$ has to be pre-computed.

In conclusion, both schemes are derived using explicit finite differences, thus the convergence is expected to be of order $O(\Delta\tau)$.

## 5.4 A deterministic model based on Pontryagins maximum principle

Recall the *stochastic* PDE in equation (5.40),

$$\frac{\partial V}{\partial \tau} + f(\tau)\Big(V(Q,\tau) - V(Q-1,\tau)\Big) = f(\tau)\int_0^\infty \rho_s(\xi|\tau)\,\Psi(\xi)\,d\xi, \qquad (5.54)$$

with the boundary conditions

$$V(Q,0) = 0 \quad \forall Q \tag{5.55}$$

$$V(0,\tau) = 0 \quad \forall \tau. \tag{5.56}$$

The stochastic nature of the PDE comes from the fact that $Q$ is a jump process with *integer* jump size 1 corresponding to one sale or no sale of a parking slot. This resulted in expected marginal values of the form $V(Q,\tau) - V(Q-1,\tau)$. If we could rather sell *fractions* of spaces, the change in remaining value created by selling the next fraction $\Delta Q$ of a slot is given by

$$V(Q,\tau) - V(Q-\Delta Q,\tau).$$

Then, the relevant marginal value is

$$\frac{V(Q,\tau) - V(Q-\Delta Q,\tau)}{\Delta Q}.$$

As we allow for finer and finer choices of $\Delta Q$ the capacity effectively becomes a *continuous* quantity and in the limit as $\Delta Q \to 0$ we obtain

$$\lim_{\Delta Q \to 0} \frac{V(Q,\tau) - V(Q-\Delta Q,\tau)}{\Delta Q} = \frac{\partial V}{\partial Q}.$$

In other words, when we treat $Q$ as a continuous quantity the expected marginal values can be computed by the partial derivative of $V$ with respect to $Q$ (assuming that such a derivative exists).

Therefore, (5.46) is transformed into

$$\frac{\partial V}{\partial \tau} + f(\tau)\frac{\partial V}{\partial Q} = f(\tau)\int_0^\infty \rho_s(\xi|\tau)\,\Psi(\xi)\,d\xi. \tag{5.57}$$

Finally, we may optimise this PDE by imposing a control on the length of stay $\xi$ as before, which gives

$$\frac{\partial V}{\partial \tau} = \max_\xi \left\{ f(\tau)\int_0^\xi \rho_s(\xi'|\tau)\,\Psi(\xi')\,d\xi' - f(\tau)P_s(\xi|\tau)\frac{\partial V}{\partial Q} \right\}. \tag{5.58}$$

### 5.4.1 Applying the maximum principle

An alternative way to express the problem in (5.58) is

$$\max_{\xi} \int_0^T r(\xi(\tau), \tau)\, d\tau \tag{5.59}$$

$$\begin{aligned} \text{subject to} \quad & dQ = f(\tau) P_s(\xi|\tau)\, d\tau, \\ & Q(\tau = T) = C, \end{aligned} \tag{5.60}$$

where the revenue rate is

$$r(\xi(\tau), \tau) = f(\tau) \int_0^{\xi} \rho_s(\xi'|\tau)\, \Psi(\xi')\, d\xi'. \tag{5.61}$$

In this representation, time is in reverse, meaning that $T$ should be regarded as the maximum time remaining until the day we seek to optimise. The idea here is that at the beginning of the booking horizon ($T$ time 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. $Q(\tau = T) = C$. 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 ($Q(\tau = 0) \geq 0$). The objective is to find the optimal selling rate of the spaces (optimal trajectory for $Q$) so that the generated revenue is maximised.

Also notice that from this representation it is made clear the state process $Q$ is *deterministic*, as it changes continuously and deterministically with time. Thus, to solve this model we apply the Pontryagins maximum principle, as discussed in section 2.3.2. As such we may refer to the PDE in (5.58) as the *Pontryagins method*, in order to distinguish it from the PDE model derived in the previous section 5.3.

The corresponding Hamiltonian $\mathcal{H}$ is thus given by

$$\mathcal{H}(\xi(\tau), \lambda(\tau), \tau) = f(\tau) \int_0^{\xi} \rho_s(\xi'|\tau)\, \Psi(\xi')\, d\xi' + \lambda(\tau) \Big( f(\tau) P_s(\xi|\tau) \Big). \tag{5.62}$$

The equations of motion are

$$\frac{d\lambda}{d\tau} = -\frac{\partial \mathcal{H}}{\partial Q} \tag{5.63}$$

$$\frac{dQ}{d\tau} = \frac{\partial \mathcal{H}}{\partial \lambda} = f(\tau) P_s(\xi|\tau). \tag{5.64}$$

Clearly, $\mathcal{H}$ is differentiable in $\xi$ which means that the optimal price $\xi^*(\tau)$ satisfies

$$\frac{\partial H}{\partial \xi^*} = 0, \tag{5.65}$$

or after simple calculation

$$\lambda(\tau) = -\Psi(\xi^*(\tau)). \tag{5.66}$$

Moreover, from the equation of motion for $\lambda$ (5.63) we have

$$\frac{\partial \lambda}{\partial \tau} = 0,$$

because the Hamiltonian does not depend on $Q$. Thus, the optimal path for $\lambda$ along the optimal $Q$-trajectory is in fact a constant value. Therefore, using this result in conjunction with (5.66) we deduce that the optimal price $\Psi(\cdot)$ has to be constant itself; this happens only if the optimal length of stay $\xi^*$ is constant along the optimal $Q$-trajectory.

Therefore, when the problem is deterministic we have shown that the optimal price policy is fixed-price policy; when the total demand to come is greater than the capacity this is the run-out price, $\Psi^0$, at which we sell exactly the entire capacity and when the total demand is less that capacity this becomes the revenue maximising price, $\Psi^*$. This result is also verified in Gallego and van Ryzin (1994) and has been illustrated in section 3.4.2.2.

### 5.4.2 Solution of the deterministic problem under the two cases

Under the deterministic setting the problem is greatly simplified. In this section, we examine our strategy in 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 $\Psi^* \to \psi_1$. In this case all customers are accepted irrespective of their duration of stay.

However, 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 $Q(\tau = 0) = 0$.

First, let us write down the initial value problem (IVP) that arises from the equation of motion for $Q$ (5.64)

$$\frac{dQ}{d\tau} = f(\tau)P_s(\xi|\tau), \quad Q(\tau = T) = C. \tag{5.67}$$

If we solve the IVP problem for a fixed value of $\xi$ we obtain

$$Q(\tau;\xi) = C - \int_{\tau}^{T} f(\tau')P_s(\xi|\tau')\, d\tau'.$$

Then, at expiry ($\tau = 0$) the remaining unsold spaces are given by

$$Q(0;\xi) = C - \int_{0}^{T} f(\tau')P_s(\xi|\tau')\, d\tau'.$$

We know that, if we have used the optimal length of stay $\xi^*$ we would sell the entire inventory exactly and therefore we would have

$$Q(0;\xi^*) = C - \int_{0}^{T} f(\tau')P_s(\xi^*|\tau')\, d\tau' = 0.$$

Let us define the function $\mathcal{F}$ as

$$\mathcal{F}(\xi) = C - \int_{0}^{T} f(\tau)P_s(\xi|\tau)\, d\tau. \tag{5.68}$$

Then, at the optimal $\xi^*$ we must have

$$\mathcal{F}(\xi^*) = 0. \tag{5.69}$$

This suggests the implementation of the Newton-Raphson method to find the root of $\mathcal{F}$, i.e. the optimal value $\xi^*$.

Finally, we can substitute the optimal $\xi^*$ in equation (5.59 ) to obtain the maximised total revenue rate generated within the time horizon $T$. More precisely, we evaluate the double integral

$$\int_0^T f(\tau) \int_0^{\xi^*} \rho_s(\xi|\tau)\Psi(\xi)\, d\xi\, d\tau. \tag{5.70}$$

Therefore, the algorithm to calculate the optimal value $\xi^*$ reads

**Pontryagins algorithm**

1. Set $\xi = 0$ at the beginning. This implies that no customers are accepted.

2. Solve the IVP problem in (5.67) using any numerical integration technique and calculate the resulting $\mathcal{F}(\xi)$.

3. If $||\mathcal{F}(\xi)|| < \epsilon$, where $\epsilon$ is the tolerance level, $\xi^*$ has been found and the iterations terminate.

4. Otherwise, update guess on $\xi$ using Newton-Raphson method (see Press et al., 2009),
$$\xi^{r+1} = \xi^r - \frac{\mathcal{F}(\xi^r)}{\mathcal{F}'(\xi^r)}$$

5. Go to step 2.

# Chapter 6

# Numerical Results

## 6.1 Monte Carlo Results

Unless otherwise stated, the intensity parameters regarding the two customer classes are set as follows:
The leisure booking class is given by

$$\mathcal{B}_1 \sim \begin{cases} \lambda_b = 5 \\ \lambda_a = 1/14 \\ \lambda_s = 1/7. \end{cases} \tag{6.1}$$

The business booking class is given by

$$\mathcal{B}_2 \sim \begin{cases} \lambda_b = 25 \\ \lambda_a = 1/3 \\ \lambda_s = 1. \end{cases} \tag{6.2}$$

Note: The leisure customers tend to book around two weeks in advance $(1/\lambda_a)$ and stay for around a week in the carpark $(1/\lambda_s)$, whereas the business customers book relatively close to the arrival day and stay on average for only a day. 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.

The total expected demand to be served per day can be calculated by

$$\mathrm{E}[TD] = \sum_n \frac{\lambda_{b_n}}{\lambda_{s_n}},$$

which in our scenario is 60 customers per day. In other words, we would *expect* that carparks of size 60 or more should be sufficiently large to meet the customer demand in the long run.

The parameters of the pricing function are set to

$$\Psi(\cdot) \sim \begin{cases} \psi_1 = 5 \\ \psi_2 = 10 \\ \mu = 1/5 \end{cases} \tag{6.3}$$

chosen specifically to replicate the behaviour of pricing functions that are commonly used in practise. Under this set of parameters, $\Psi(\xi)$ ranges from as 5 units per day (when $\xi \to \infty$) to 15 units per day (when $\xi = 0$).

Imagine we are at time $t = 0$ when the carpark starts its operations. We seek to evaluate the expected revenue per day to be generated in the carpark. We can present the expected revenue as of the current time $t$, $\forall t = 0, \ldots, 50$. Nonetheless, since intensities are stationary we find it more convenient to use a reversed time index and illustrate our findings as functions of the time remaining $\tau = T - t$.

### 6.1.1 A First-Come-First-Served (FCFS) Policy

Let us first begin with the simplest case, where the carpark operates with a FCFS policy. This policy is modelled mathematically by setting all entries of the bid-price table to zero, so that the only restriction of the bookings is the capacity availability. The carpark is initially assumed to accept bookings from only a single customer class. Then, the two customer classes, 1 and 2, are combined to obtain the general framework for our problem. Our results are based on Monte Carlo and are averaged over 10000 paths.

Figure 6.1 shows the expected revenue to be generated on day $T$, as a function of the time remaining. The left carpark corresponds to customers from the leisure class only, while the right deals with customers of the business class only. We observe that in both figures the expected revenues are concave and increasing in

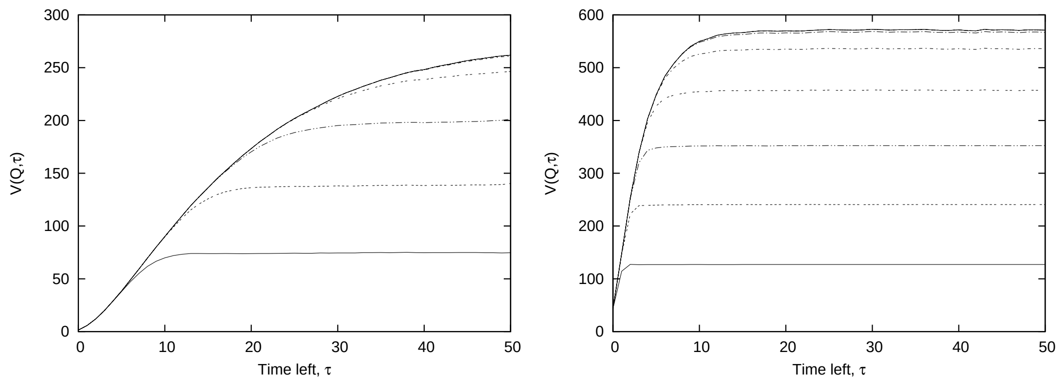



Figure 6.1: Expected revenue for a carpark on day $T$ with $\tau$ days remaining until $T$, using the FCFS policy. Single booking sets are used with the left figure representing the leisure customer class and the right figure the business customer class. The different lines correspond to different capacities remaining from 100 to 10 in steps of 10, top to bottom.

both the time remaining $\tau$ and capacity remaining $Q$. This is intuitively correct, as one would expect to make greater profits if there are more spaces to sell and/or if there is more time available until spaces are used. Looking at the business class, we also note that a booking horizon of around ten days seems to be sufficient in capturing the entire revenue from the business class, as by going further in the past does not produce significant increase on the expected revenues. In contrast the active booking horizon for the leisure class seems to span over fifty days in advance due to the longer average pre-booking times of these customers.

Let us now allow customers from both customer classes to arrive at the carpark. Figure 6.2 presents the resulting expected revenue as a function of the time remaining. We observe that when capacity is abundant ($Q > 60$) all booking requests from both customer classes are accepted. In this case the combined expected revenue is the sum of the individual expected revenues of the two customer classes and it is indeed increasing and concave in the time remaining. However, when the capacity is scarce (expected demand is greater than capacity) then this is no longer the case. In particular, for little time remaining ($\tau \leq 5$), business bookings dominate and thus the expected revenues are shown to be increasing. However, when there is much time remaining, the FCFS policy tends to accept too many leisure customers which results in displacing some lucrative business customers that would have arrived later in the horizon. As a result the expected values in this region drop and therefore the expected revenue is no

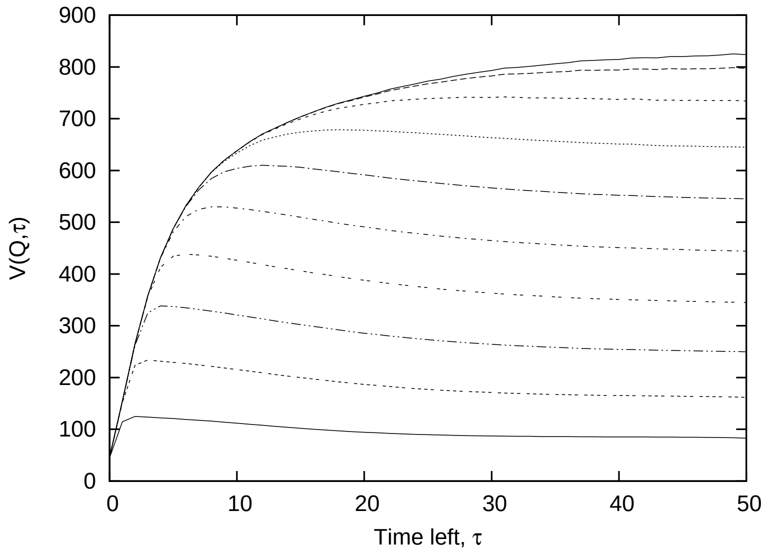



Figure 6.2: Expected revenue for a carpark on day $T$ when there are $\tau$ days remaining until $T$, using the FCFS policy. We use the combined booking set which results by merging the two customer classes. The different lines correspond to different capacities remaining from 100 to 10 in steps of 10, top to bottom.

longer an increasing function of the time remaining.

In order to understand the suboptimality of the FCFS policy we examine the structural properties of the expected marginal value of the space $\Delta V(Q,\tau) = V(Q,\tau) - V(Q-1,\tau)$. The two distributions in figure 6.3 illustrate $\Delta V(Q,t;T)$ as a function of capacity remaining and time-to-go, respectively. The two figures verify that expected value of the space is no longer a decreasing value of capacity remaining nor it increases with the more time remaining.

In a $3D$ plot (figure 6.4) we can show the resulting surface of expected marginal values for every $Q$ and $\tau$. The hump shown in the middle of the surface emphasises the fact that within this region optimality is not satisfied.

In conclusion, the graphs in figure 6.5 summarise our findings by comparing the three different carparks; a carpark that accepts customers only from the leisure booking set (dashed line), only from the business booking set (dotted line) and from the combined booking set (solid line). We notice that under the FCFS policy, there are instances when the carpark accepting customers from the combined booking set fails to generate higher revenues than the carpark with only business customers. This is again because there is nothing stopping the leisure customers filling up the spaces at the expense of the more profitable business customers.

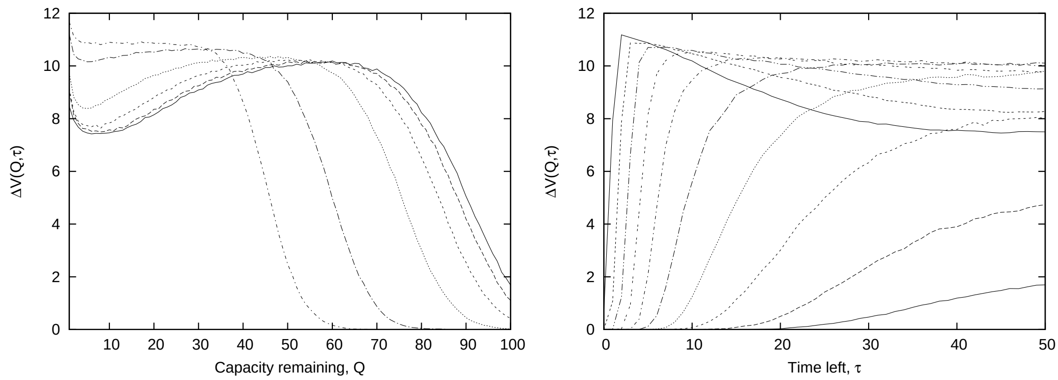



Figure 6.3: Expected marginal value of the space on day $T$ for the combined carpark, as a function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), using the FCFS policy. The different lines in the left figure correspond to different times remaining from 50 to 10 in steps of 10, right to left. The different lines in the right figure correspond to different capacities remaining from 10 to 100 in steps of 10, left to right.

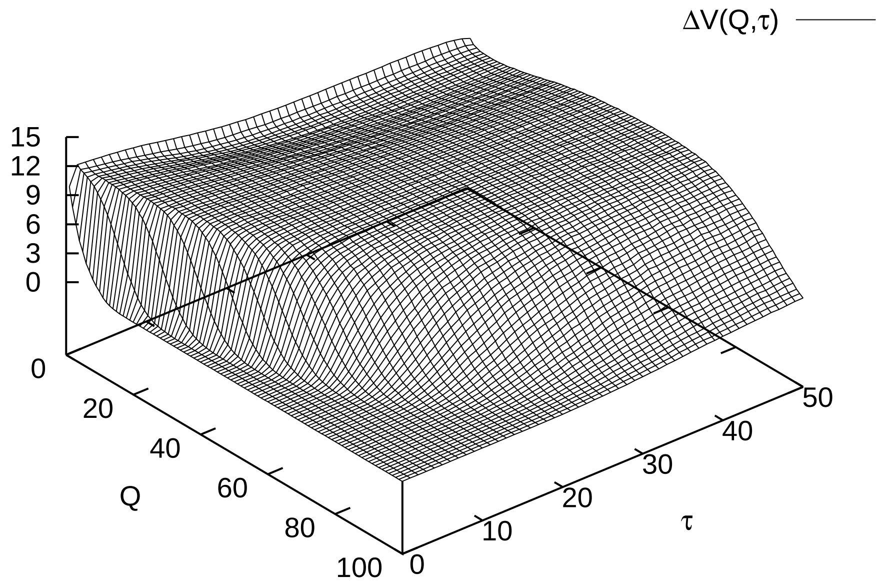



Figure 6.4: Expected marginal values for the FCFS policy, as a function of the capacity remaining $Q$ and the time left $\tau$.

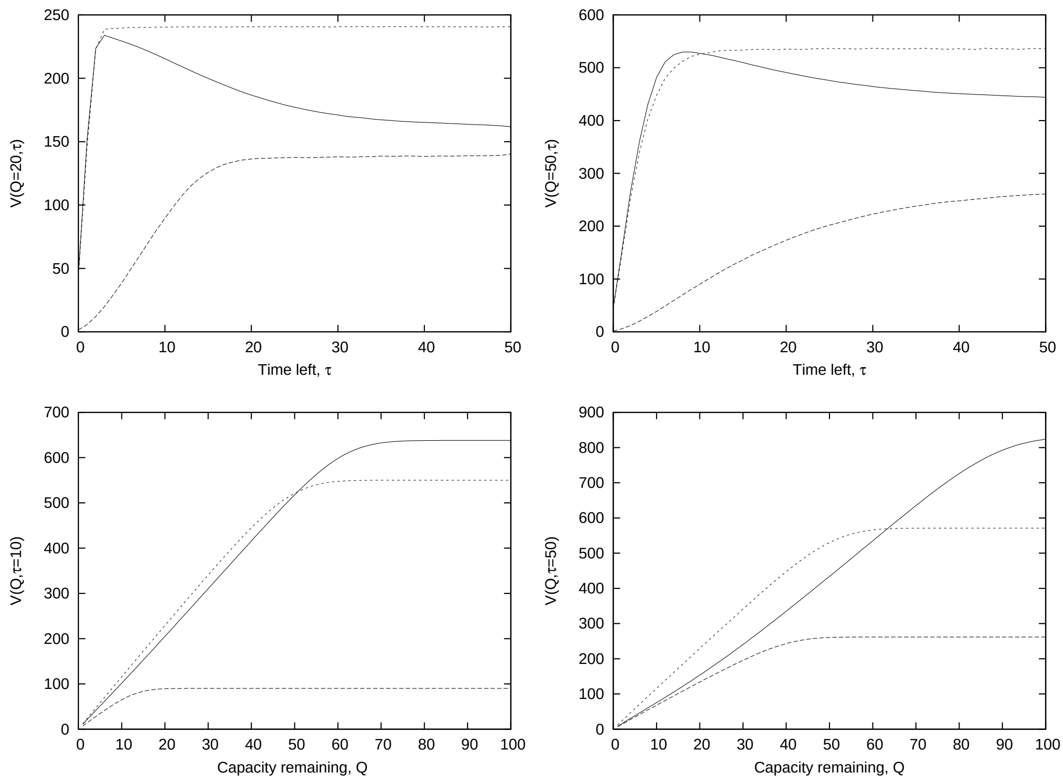



Figure 6.5: Expected revenue on day $T$ for carparks operating under a FCFS policy, as a function of the time remaining $\tau$ (upper figures) and capacity remaining $Q$ (lower figures). Each line corresponds to one carpark; one that operates with only leisure customers (dashed line), with only business customers (dotted line) and with a combined set of customers (solid line). The upper left figure is when $Q = 20$ and the upper right figure when $Q = 50$. The lower left and lower right figures correspond to having $\tau = 10$ and $\tau = 50$ respectively.

### 6.1.2 Implementing our Monte Carlo (MC) rejection policy

Instead of simply using a FCFS policy, we now set the bid prices equal to the expected marginal values which have been calculated by the algorithm in section 5.1.3. Convergence of our results has been achieved and the algorithm terminated after 15 iterations whereby the number of paths used reached 12672.

Figure 6.6 illustrates the optimal expected revenues in the carpark for the day $T$ as functions of the time remaining. It is clear that now the expected revenue retains its desirable features; it is an increasing function of the time remaining for all capacities.

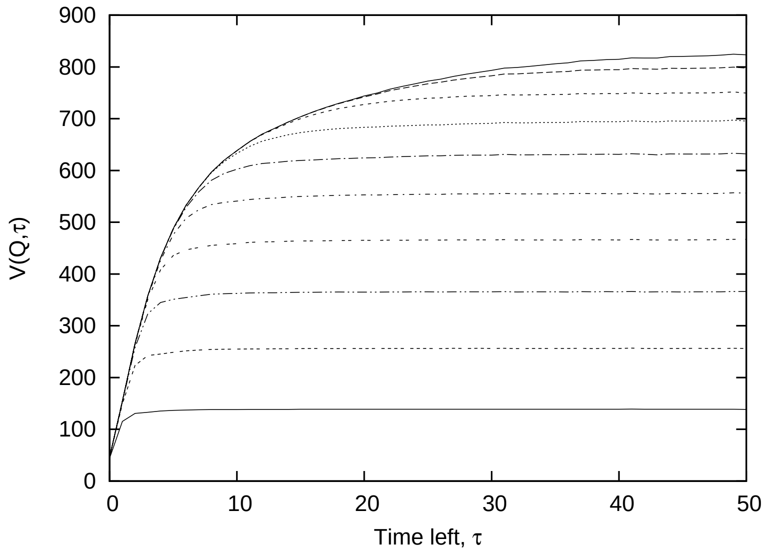



Figure 6.6: Expected revenue for the combined carpark on day $T$ with $\tau$ days remaining until $T$, using the MC policy. The different lines correspond to different capacities remaining from 100 to 10 in steps of 10, top to bottom.

Moreover, figures 6.7 present $\Delta V(Q, \tau)$ as functions of capacity and time remaining. We notice that $\Delta V(Q, \tau)$ maintains the desirable monotonicity properties; it decreases with capacity and increases with time left.

Once again, the optimised expected marginal values are presented in figure 6.8 as a $3D$ surface with respect to $Q$ and $\tau$. Compared to figure 6.4, the optimised marginal values are lower in regions where capacity is large and higher in regions where capacity is limited, to emphasise that the marginal values should adjust according to the carpark state in order to maximise the expected revenues. The bump that is seen along the first couple of capacities is created by the fact that customers are charged for the entire day even though they are only staying for fractions of a day.

Finally we provide a comparison between the FCFS policy and our improved MC policy. The plots in figure 6.9 show the expected revenues generated by the three carparks, as in figure 6.5, where we have added the expected revenue generated under the MC rejection policy. It is clear, that our rejection policy outperforms all other carparks for all times and capacities remaining. Looking at the lower figures we note the manner at which the optimisation algorithm worked to “fix” the region where the FCFS policy of a combined carpark was losing out against a single carpark of only business customers (carparks of size

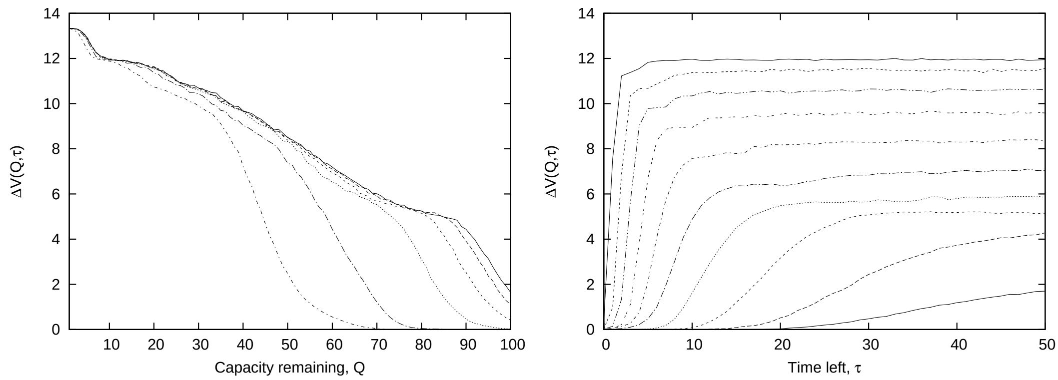



Figure 6.7: Expected marginal value of the space for the combined carpark on day $T$ as function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), using the MC policy. The different lines in the left figure correspond to different times remaining from 50 to 10 in steps of 10, right to left. The different lines in the right figure correspond to different capacities remaining from 10 to 100 in steps of 10, top to bottom.

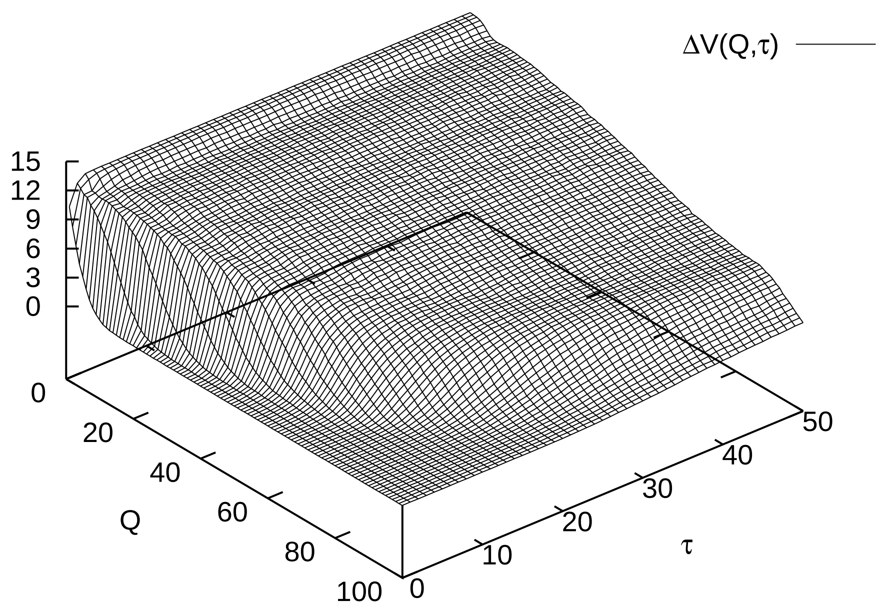



Figure 6.8: Bid-price table with expected marginal values for the MC policy, as a function of capacity remaining $Q$ and time left $\tau$.

less than 60). In fact, it is within this problematic region where the optimisation algorithm would have to perform well.

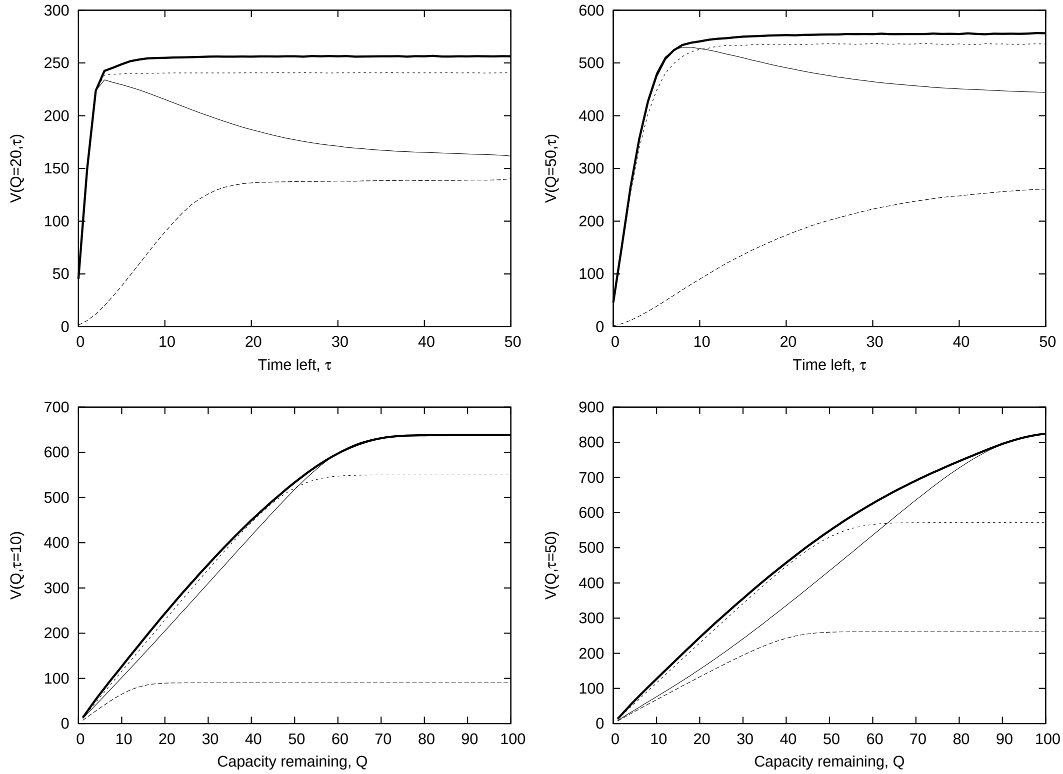



Figure 6.9: Expected revenue for different carparks on day $T$ as a function of the time remaining $\tau$ (upper figures) and capacity remaining $Q$ (lower figures). Each line corresponds to one carpark; one that operates under a FCFS basis with only leisure customers (dashed line), under a FCFS basis with only business customers (dotted line), under a FCFS basis with a combined set of customers (solid line) and under the implemented MC policy with a combined booking set of customers (thick solid line). The upper left figure is when $Q = 20$ and the upper right figure when $Q = 50$, both with $\tau = 50$ days left. The lower left and lower right figures correspond to having $\tau = 10$ and $\tau = 50$ respectively.

### 6.1.3 Convergence of the MC method

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 entire 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 $\Delta T$ in consideration. We would expect that the finer the $\Delta T$ the more accurately we charge customers according to their required duration of stay and thus these values should be decreasing and eventually converge to the value when each customer is charged the exact amount that corresponds to their exact duration of stay.

The pricing function can still calculate the price rate per day but $D$ now refers to the number of periods of length $\Delta T$ the customer is staying for. Therefore, the price rate per day for a stay of $D$ $\Delta T$-periods is given by

$$\Psi(D^i; \Delta T) = \psi_1 + \psi_2 \, e^{-\mu D \Delta T}, \tag{6.4}$$

which is again a step function where the size of each step equals $\Delta T$[1]. One can calculate the price rate *per period of size* $\Delta T$ by multiplying equation (6.4) by $\Delta T$.

In figure 6.10 we show the convergence of the Monte-Carlo method for the expected revenue of the carpark on time $T$ as a function of the time remaining. The different curves correspond to decreasing sizes $\Delta T$ ($\Delta T = 1$ is a day, $\Delta T = 0.0125$ is 18 minutes). It seems that the expected revenues are behaving in the right manner as the finer the $\Delta T$ is the less the resulting revenues. Again, this is because as we reduce $\Delta T$, customers are charged a price that better corresponds to their desired length of stay.

[1] In fact, (4.14) is a special case of (6.4), with $\Delta T = 1$.

Table 6.1: Convergence of the MC approach. Computation times (in seconds).

| $\Delta T$ | Comput. Time | Iterations | Paths |
|---|---|---|---|
| 1.0 | 126.7 | 15 | 12672 |
| 0.5 | 268.8 | 16 | 17920 |
| 0.25 | 621.1 | 17 | 25342 |
| 0.125 | 1334.6 | 18 | 35780 |
| 0.0125 | 10832.3 | 18 | 35780 |

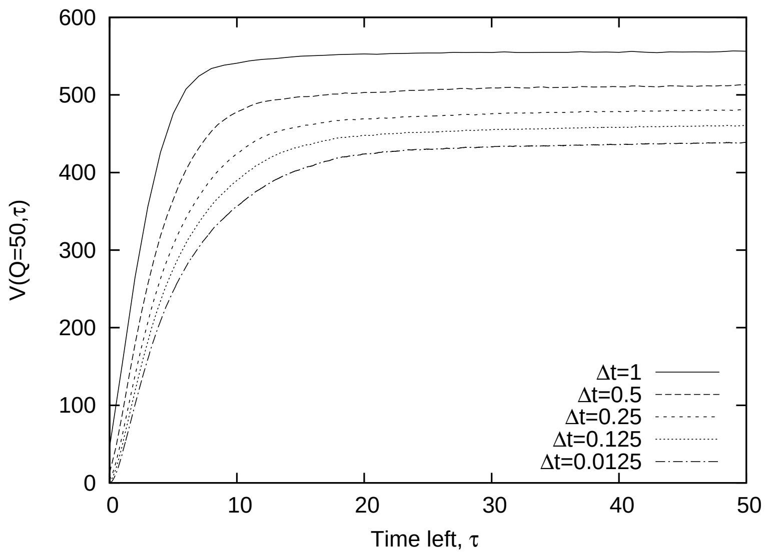



Figure 6.10: Convergence in $\Delta T$ of the expected revenue generated by the MC method. Figure shows the expected revenues as functions of the time left $\tau$, for a carpark with $Q = 50$ spaces remaining.

Table 6.1 shows the computation time it took to calculate the solution[2] for different sizes $\Delta T$. The computation time increases linearly with $\Delta T$. For the case of $\Delta T = 0.0125$ it took around three hours to compute the results. The disadvantage of this method is then apparent as the MC solution takes too long to compute the solution and thus it cannot be updated on a frequent hourly basis. However, the solution is a dynamic set of marginal values which by definition do no need to be frequently updated, perhaps this should be done just once every day. We note that the number of paths and iterations might increase as $\Delta T$ becomes finer.

As mentioned previously, the continuous-time methods can only be compared with the MC discrete-time method in the limit, as the size of the interval goes to

[2]The solution is a $2D$ array showing expected revenues for capacities from 1 to 100 and times from 0 to 50 in steps of $\Delta T$.

zero ($\Delta T \to 0$). Therefore, for this comparison we will be using the converged values of the MC method for $\Delta T = 0.0125$ as the corresponding limiting value.

The expected revenues are shown in figure 6.11 as functions of the time remaining. Comparing this to figure 6.6 we observe that the expected revenues have been reduced indicating that customers now tend to get charged a price that better reflects their required duration of stay. Also note that the curves are fewer indicating that we can satisfy the same demand with smaller carparks as we can sell the same space to more than one person per day, effectively squeezing in more customers.

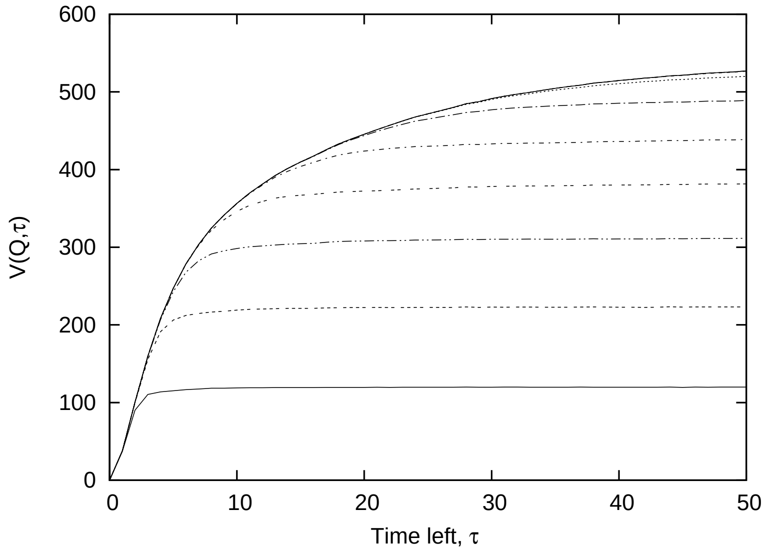



Figure 6.11: Expected revenues for the MC method as functions of the time remaining $\tau$, after setting $\Delta T \to 0.0125$. The different lines represent different carpark sizes from 100 to 10 in steps of 10, top to bottom.

The resulting surface of expected marginal values is shown in figure 6.12. We note that the surface now looks quite different from that in figure 6.8. In particular, relatively large capacities now are assigned very low or zero marginal values which indicates that now we can squeeze in more customers effectively.

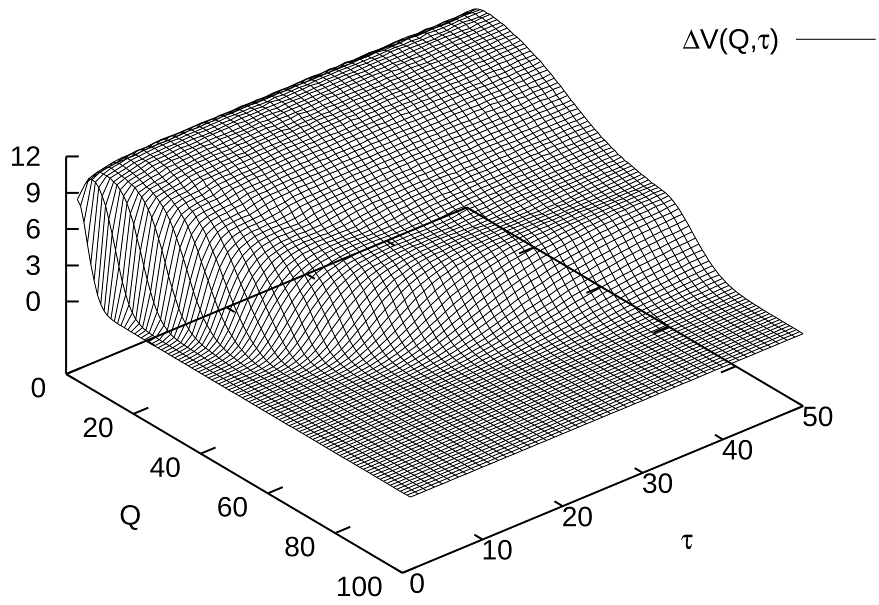



Figure 6.12: Resulting bid-price table with expected marginal values for the MC policy after setting $\Delta T \to 0.0125$.

This set of marginal values will be used as the bid prices to approximate the solution to the network problem. However, by looking closer at figure 6.12 and particularly in a cross section of time (say $\tau = 30$) we notice that for very small capacities the expected marginal values are sloping downwards and they no longer follow the increasing behaviour which arises from the inventory-monotonicity property. Recall that for $\Delta T = 1$ the properties are satisfied. Thus, the problem arises by choosing $\Delta T$ to be smaller than a certain level. Let us investigate this further.

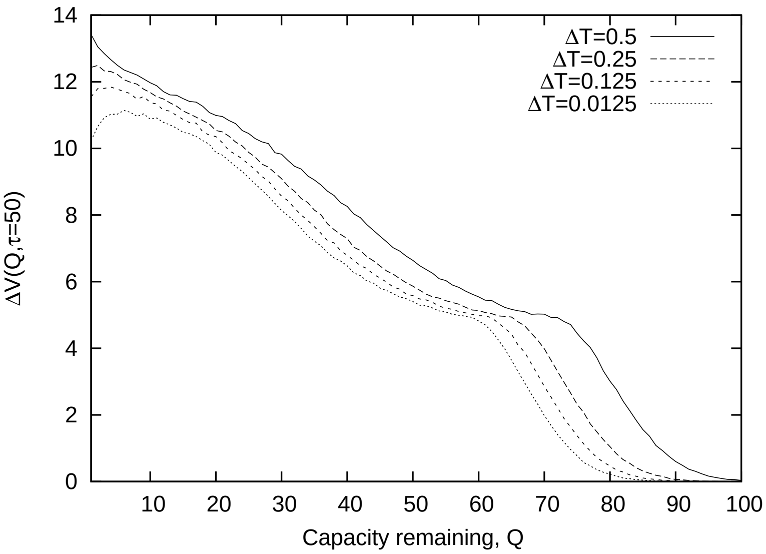



Figure 6.13: Expected marginal values with 50 days left as functions of the capacity remaining $Q$ for different sizes $\Delta T$.

Figure 6.13 shows the optimal expected marginal values for day $T = 50$ as function of the capacity remaining. The different distributions represent different sizes of $\Delta T$. For $\Delta T = 0.5$ the distribution shows monotonically decreasing expected revenues with capacity. However, for smaller choices of $\Delta T$ the curves seem to bend on the far left end and thus the inventory monotonicity property is no longer satisfied. So, what is it that forces the model to converge in these values, showing such a strange behaviour? This is because of the manner at which the pricing function works. When we set $\Delta T = 1$ to account for day intervals we seek to extract the maximum price rate from the customers who stay for only one day which is $\Psi(D = 1; 1) = 13.18$. For this to happen, we require the customer not only to want to stay for $\xi \leq 1$ but that they arrive and leave on the same day (blue request in the top figure in 6.14). If their duration of stay crosses over two days, then they will occupy two days and thus pay the price rate per day that corresponds to two days, $\Psi(D = 2; 1) = 11.70$ which will in fact be lower (purple request in the top figure of 6.14). Because the relatively big interval size such single-day-length single-occupancy customers exist for all days in the horizon and thus we allocate the last remaining spaces to them.

When $\Delta T = 0.5$ in order to extract the maximum amount per day $\Psi(D = 1; 0.5) = 14.04$ we require the customer to arrive and depart such that his overall stay covers only a single period, as shown in the middle figure in 6.14. In fact, we require *two* such customers per day, one appearing after the other so that we receive 2 times $\Psi(D = 1; 0.5)\,\Delta T$ which gives 14.04. Fortunately, the simulations generate such customers and therefore the expected marginal values have the correct shape. However, for $\Delta T = 0.25$ it becomes less possible to extract the maximum amount $\Psi(D = 1; 0.25) = 14.51$; it is harder to find *four* customers arriving one after the other who will stay for only a single $\Delta T$ period while not crossing over to the neighboring periods. The simulations show that even when such single- period customers are found, their request crosses over two periods (as in the bottom figure of 6.14) and as a result the maximum attainable price-rate is lowered to $\Psi(D = 2; 0.25) = 14.04$. This limitation creates that extra curvature on the distributions. In fact, as $\Delta T$ tends to zero the customer requests might cross more than two periods resulting in even lower maximum attainable price rates.

One more fact that could potentially cause this effect is the decaying coefficient $\mu$ of the pricing function. The greater this is, the greater the price differential from a single-period stay to a two-period stay. If our choice of $\mu$ is small then the algorithm might be indifferent in choosing between a single-period-stay customer (if such a customer exists) and a two-period-stay customer because the price differential is insignificant.

We conclude this section with some closing remarks on the Monte-Carlo method. First, the rejection algorithm optimises the day by treating the problem as a network problem whereby the decision on a booking request was taken after examining all the days involved. Second, calculating the correct optimal expected revenues using the MC approach is computationally intensive. This is because the optimal revenues rely on our estimates of the expected marginal values (rejection policy) which are based on the large number of paths and iterations taken. However, even after convergence is achieved the expected marginal value curves (figure 6.7) are still not sufficiently smooth.

The MC method calculated the dynamic set of expected displacement costs by taking into account the inter-dependence within the days. This set can be used to make decisions on a daily basis for a sequence of days without having to be frequently updated. If, however, the dynamic bid prices have to updated frequently, such a method cannot be used due to the large computational time of the method. Therefore, in the next section we examine the solution to the derived PDE model and later the solution to the Pontryagins method.

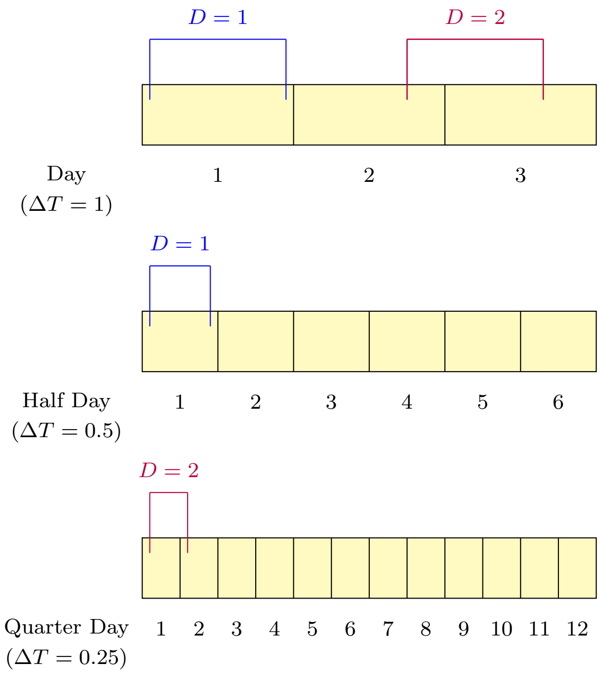



Figure 6.14: Three carparks that price according to day ($\Delta T = 1$), half-day ($\Delta T = 0.5$) and quarter-of-day ($\Delta T = 0.25$) intervals. The figures show different cases of bookings that require to stay for a length less than $\Delta T$. According to when they arrive, the blue bookings are allocated one period of stay whereas the purple are allocated two.

## 6.2 PDE Results

### 6.2.1 No optimisation

We begin by presenting the results for the case when no optimisation policy is in place (equivalent to the FCFS policy seen in the previous section).

Figure 6.15 presents the resulting expected revenues as functions of the time remaining for different remaining capacities. As in figure 6.2, the expected values fail to be increasing in the time remaining indicating a sub optimality to this (no optimisation) policy. Also, note that for capacities greater than 60, all curves appear to be the same, emphasising the fact that we are now solving in continuous time and thus larger capacities are not needed since cars may be allocated into parking spaces more easily.

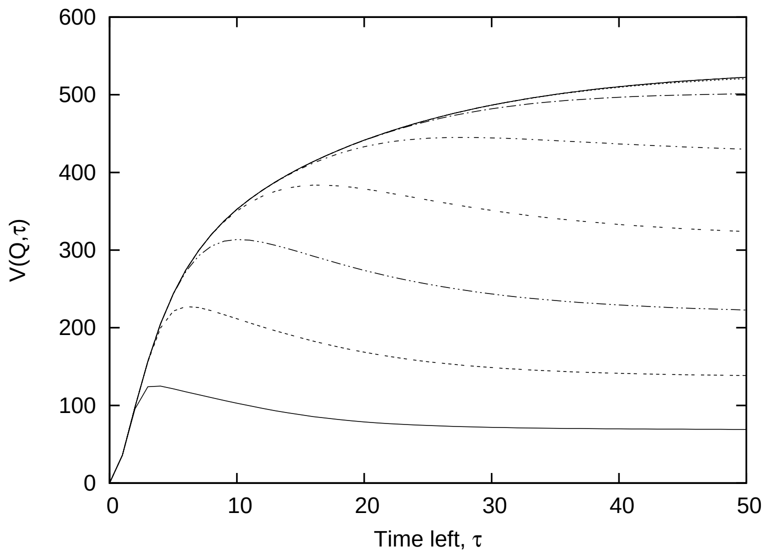



Figure 6.15: Expected revenue for a carpark on day $T$ when there are $\tau$ days remaining until $T$, using no optimisation. The different lines correspond to different capacities remaining from 100 to 10 in steps of 10, top to bottom.

Furthermore, figure 6.16 shows the expected marginal value of the spaces as functions of the capacity remaining (left) and time left (right). As before, the suboptimality of the model is noticeable as $\Delta V(Q, \tau)$ fails to be a decreasing function of the capacity remaining nor does it increase in the time left. The intuition here is that all customers are allowed entry with no preference with respect to how much revenue they contribute to the system. So when the time remaining is relatively large, the carpark accepts too many (leisure) low-paying

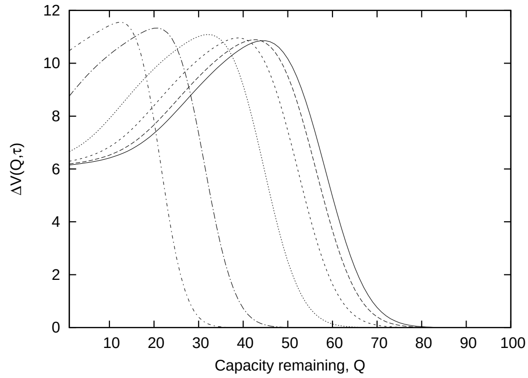



Figure 6.16: Expected marginal value of the space for a carpark on day $T$ as function of the capacity remaining $Q$ (left figure) and time remaining $\tau$ (right figure), with no optimisation in place. The different lines in the left figure correspond to different times remaining from 50 to 10 in steps of 10, right to left. The different lines in the right figure correspond to different capacities remaining from 10 to 100 in steps of 10, left to right.

customers and does not save enough spaces for the business customers that will arrive later, which as a result reduces the expected revenues.

In a $3D$ plot in figure 6.17 we can show the resulting set of expected marginal values for every $Q$ and $\tau$. The intuition here is similar to that in figure 6.4.

### 6.2.2 Optimal rejection policy implemented

Let us now solve the PDE with the rejection policy described by equation (5.50). In figure 6.18 we present the resulting optimal expected values, while figure 6.19 shows our findings on the optimal rejection policy.

The marginal values behave in the correct manner now, implying that the corresponding prices are closer to the optimal ones. In simple terms this means that under our rejection policy the more profitable business customers are prioritised for receiving service at the expense of the low paying leisure customers. The distributions here are much smoother than those in the MC method, which is obviously a desirable feature. Although the absolute values are slightly different, the shape and behaviour of the curves is quite similar.

In a $3D$ plot (figure 6.20) we can show the resulting set of expected marginal values for every $Q$ and $\tau$. This set of marginal values will be used as the bid prices to approximate the solution to the network problem.

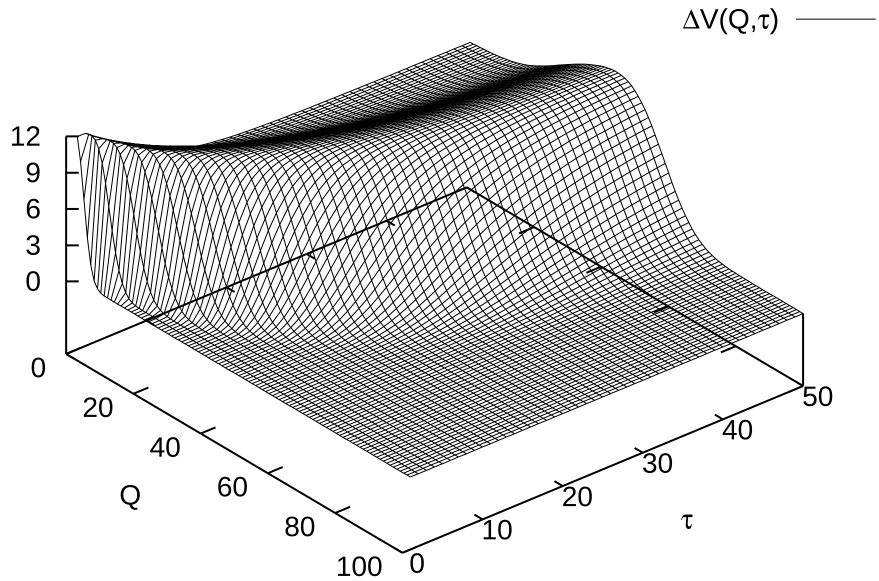



Figure 6.17: Bid-price table with expected marginal values without optimisation, as a function of capacity remaining $Q$ and time left $\tau$.

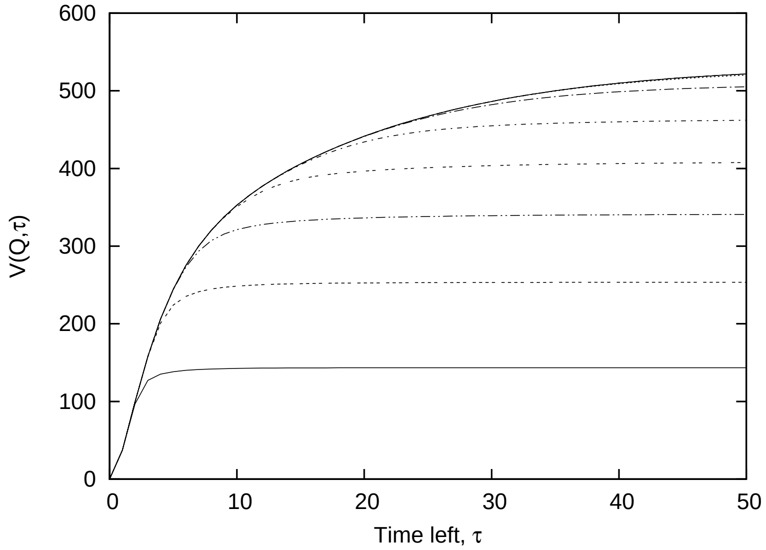



Figure 6.18: Expected revenues on day $T$ as functions of the time left $\tau$ after implementing the optimal PDE policy. The different lines correspond to different capacities remaining from 100 to 10, top to bottom.

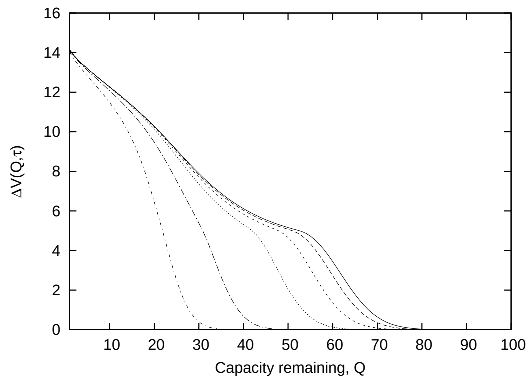



Figure 6.19: Expected marginal value of the space on day $T$ as a function of the capacity remaining $Q$ (left figure) and time left $\tau$ (right figure), using the optimisation policy. The different lines in the left figure correspond to different times remaining from 50 to 10 in steps of 10, right to left. The different lines in the right figure correspond to different capacities remaining from 10 to 100 is steps of 10, top to bottom.

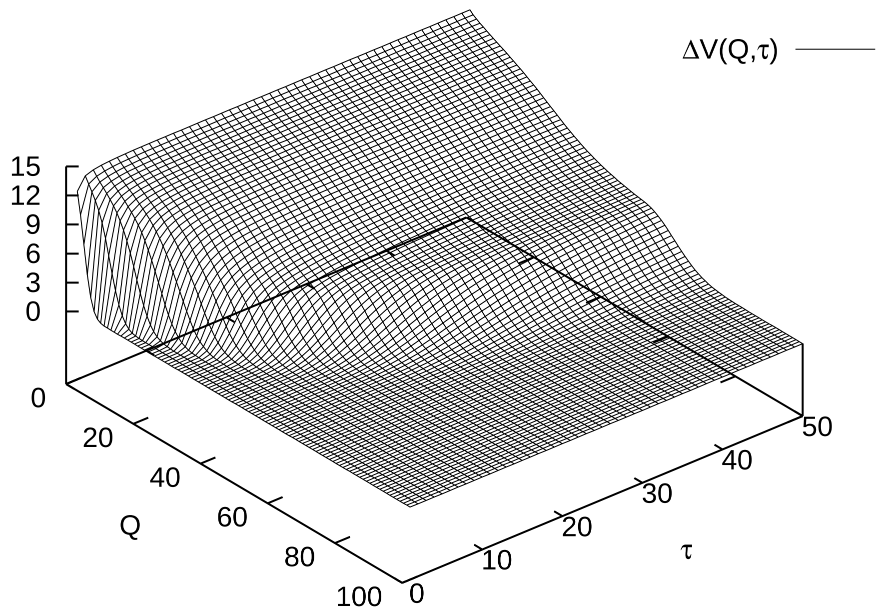



Figure 6.20: Bid-price table with expected marginal values from the optimal PDE policy, as a function of capacity remaining $Q$ and time left $\tau$.

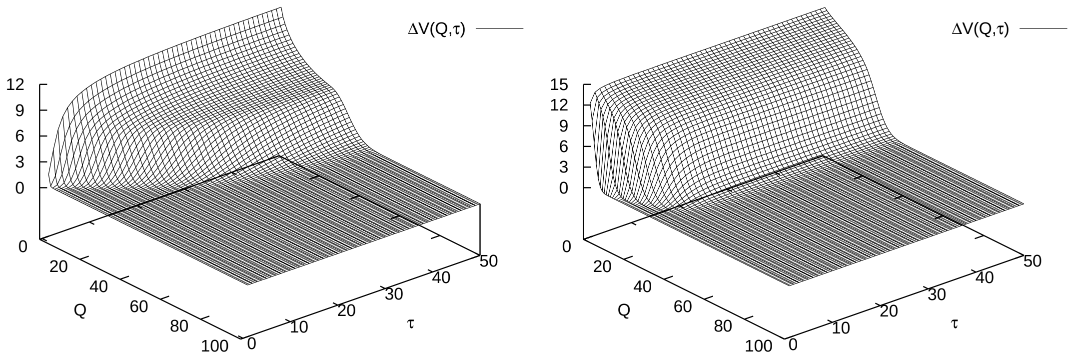
### 6.2.3 Further investigation on the rejection policy

In this section we seek to further investigate the PDE solution. In particular, we shed light to the shape of the optimal rejection policy and examine how this is affected by the various model parameters. Unless otherwise stated all parameter values in the following figures are set according to equations (6.1), (6.2) and (6.3).

Let us first begin by reviewing figure 6.20. What we observe is a surface that is increasing in both the time left and the capacity remaining, while it smooths out to zero after a certain region. In fact, this happens around capacity 60 which has been the expected total demand under the set of parameters chosen. Looking closer to this figure, and by focussing on the slice $\tau = 50$ in particular, we may notice a profound hump around the capacity 60 but also a smaller one (barely visible) at capacity 25.

As we will see below, this phenomenon is related to the expected demands of the single customer classes that constitute the combined set. So let us first start by investigating the solution of the PDE when only a single class is used at a time. In figure 6.21 we present the optimal rejection policies as these result from considering (a) only leisure customers and (b) only business customers in the carpark. In the case of the leisure class the expected marginal value of the space is nonzero for any carpark size less than around 35. This is because under our choice of parameters the expected total demand to come from the leisure class is $\lambda_{b_1}/\lambda_{s_1} = 35$, implying that any excess capacity has little or zero value. Note that after 35 the expected marginal values, although close to zero, may still be positive to compensate for the stochastic nature of the problem. Similarly, for the business carpark the expected marginal values drop to zero after 25 which is the total expected demand for this class ($\lambda_{b_2}/\lambda_{s_2} = 25$). The two distributions seem to differ in the general structure, a fact that emphasises the difference in customer behaviour between the these two customer classes (i.e. they have different average pre-booking times, average length of stays).

Let us examine in greater detail these differences. In figure 6.22 we plotted the expected marginal values at $\tau = 50$ for the leisure class and the business class when treated in isolation. For the leisure class the hump is clearly visible around capacity 35 and the expected marginal value of the spaces at this point is roughly 5 currency units which is indeed the long term price rate (according to our pricerate function parameters) when customers are allowed to stay for long periods. What actually happens in this case is that when there are only a small




(Leisure) (Business)

Figure 6.21: Optimal expected marginal values as functions of the capacity remaining $Q$ and time left $\tau$. The left figure corresponds to a carpark with customers from only the leisure class while the right figure to a carpark with customers from only the business class.

number of spaces we accept the leisure customers with shorter stays who bring higher rates into the carpark. For larger carparks, however, the leisure demand of only short stays in not enough to fill the carpark and thus we tend to accept leisure customers with longer stays, which eventually lowers the marginal values. By capacity 35 we would have accepted all potential customers, meaning that any length of stay has been allowed and due to the long-stay leisure behaviour the price rate approaches the long term rate set by the manager. However, in the business carpark the situation is slightly different. For the business class, due to the shorter stays the average price is higher on average. In particular, when there are few spaces remaining we accept short-stay business customers and thus the pricerate function calculates a higher rate. Even for larger carparks because of the characteristics of the business class there is still demand to come and stay for short periods, which justifies the slow decrease in the marginal value of the space. However, for even larger carparks, greater than around 25, the demand cannot fill out all spaces and thus the marginal values of the spaces drop rapidly to zero.

To support the above arguments, figure 6.23 is provided to illustrate the effect of varying the staying intensity $\lambda_s$ of the leisure class, which is achieved by varying the average length of stay $\bar{\xi} = 1/\lambda_s$.

If we now combine the two customer classes the resulting distribution (figure 6.24) shows some interesting behaviour. For capacities less than 25, priority goes

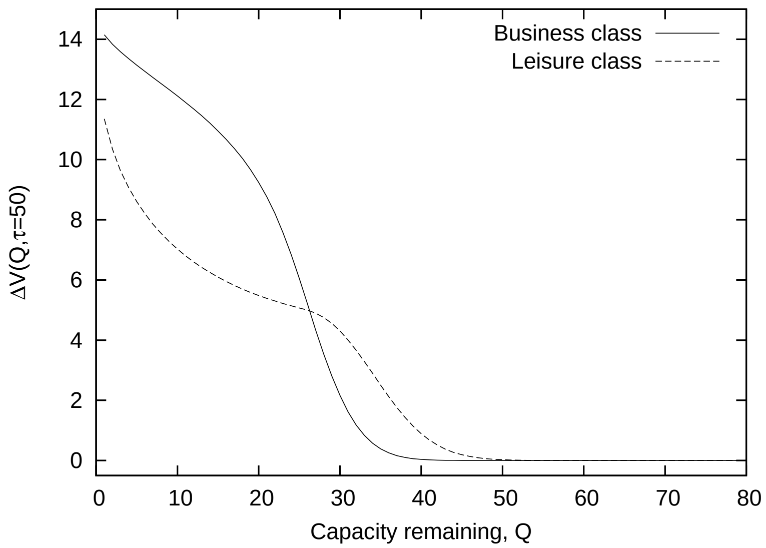



Figure 6.22: Optimal expected marginal values with 50 days left, as functions of the capacity remaining $Q$, for two carparks operating with only leisure customers (dashed line) and only business customers (solid line).

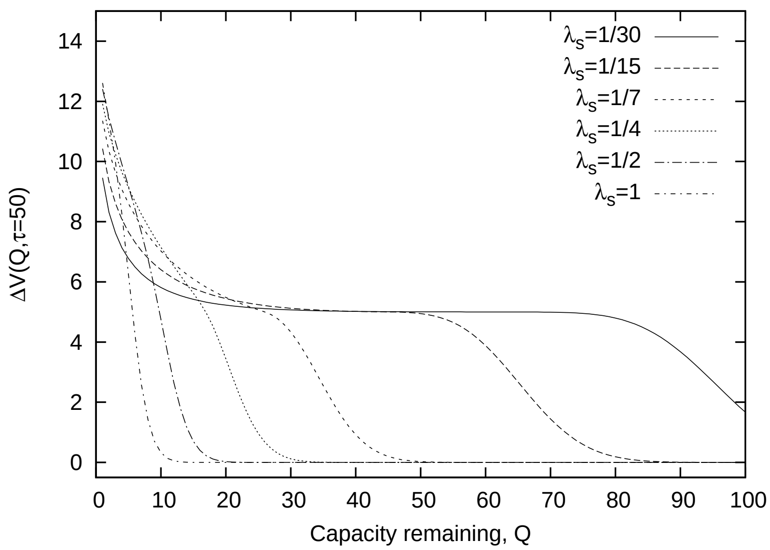



Figure 6.23: Effect on the expected marginal values of varying the staying intensity $\lambda_s$. We consider customers from a single class only, with $\lambda_b = 5$ and $\lambda_a = 1/14$.

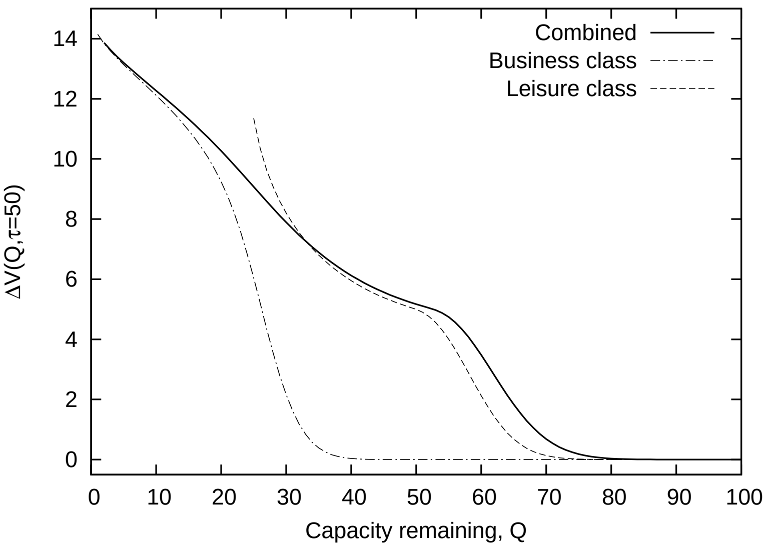



Figure 6.24: Optimal expected marginal values for a carpark with customers from both customer classes. We also plot the corresponding business class and the leisure class adjusted 25 units towards the right.

to the more high-paying business customers thus the spaces are kept highly priced. If however the carpark is large enough to meet all the business class, then there might be free spaces to serve some of the leisure demand too. Therefore, the expected marginal values adjust according to the leisure demand in this region. Roughly speaking, it is as if the leisure curve appears after the business curve with a lag of 25 units. (One may visualise this from figure 6.24 where we show the leisure curve placed 25 units further on the right.) Finally, when the carpark is so large that it can satisfy all the demand from both classes, the surplus spaces are worth very little and eventually worthless.

Now that we have explained the core of the solution we can return back to our main problem with the two customer classes combined and investigate the effect on the expected marginal values curves by changing some of the model's parameters. We believe that the shape of the rejection policy is affected in three manners;

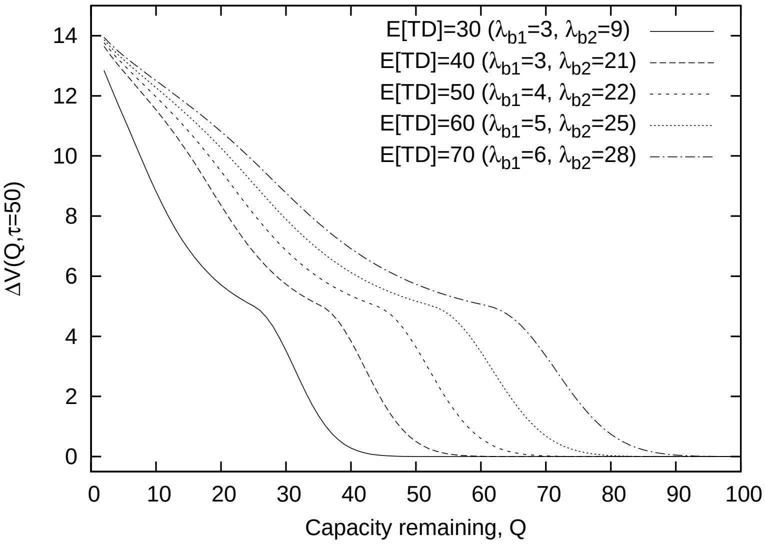



Figure 6.25: Effect on the expected marginal values of changing the expected total demand E$[TD]$. In this figure we change the expected total demand by changing the combination of booking intensities $\lambda_{b_n}$ of the two classes, while all other parameters are fixed.

1. Horizontal Shift:

   The rejection policy widens/shrinks: This is the result of varying the expected total demand of the combined customers. The greater this is, the wider the curve will become as more spaces will be needed to satisfy the demand and thus more spaces gain value. To justify this we provide the reader with figure 6.25 where we present the shape of the rejection policy at $\tau = 50$ as function of the capacity for varying expected total demand E$[TD]$.

2. Vertical Shift:

   The rejection policy moves up or down according to the choice of $\mu$, the decaying coefficient in the price-rate function. The bigger this is the greater the price differential between e.g. one-night and two-night stays. This implies that the expected-marginal-value curve slopes further downward to ensure that for greater capacities the restriction should be loosened further. The argument is supported by figure 6.26 where we present the expected marginal value for varying choices of $\mu$.

3. Balance between the customer classes:

   The rejection policy is affected by the ratio of the booking intensities of

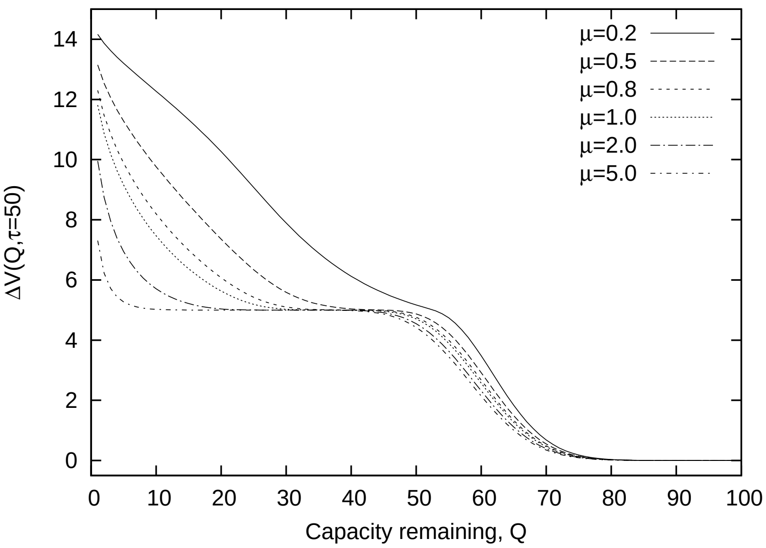



Figure 6.26: Effect on the expected marginal values of changing the decaying coefficient $\mu$ of the price-rate function.

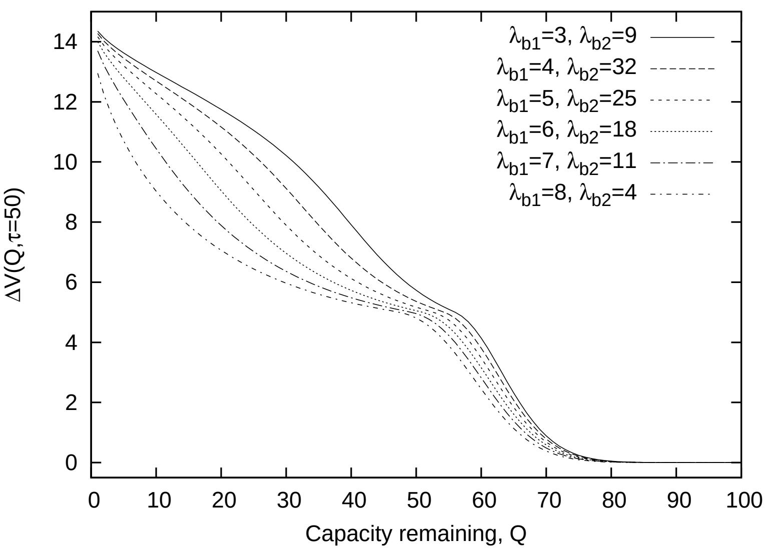



Figure 6.27: Effect on the expected marginal values of changing the ratio of the booking intensities of the two customer classes. In all tests we ensure that the expected total demand is kept fixed at $\mathrm{E}[TD] = 60$, while all other parameters are also fixed.

the two customer classes. Since each booking class represents different customer behaviour in terms of pre-booking times and length of stays, the way these classes appear in the combined set should certainly affect the resulting policy. When business customers dominate, the resulting values are a combination of the marginal values of the two single customer class with a greater weight on the business one. Alternatively, the situation reverses if leisure customers dominate over the business ones. In figure 6.27 we examine this effect for various combinations of the booking class while making sure that we kept the expected total demand the same in all cases.

### 6.2.4 Solution analysis: numerical integrity

Table 6.2: PDE optimal solution. First order convergence in time dimension for the two numerical schemes.

| | | Exhaustive search | | Value-policy | |
|---|---|---|---|---|---|
| $\Delta\xi$ | $\Delta\tau$ | $V(Q=30,\tau=50)$ | (%) Rel. Diff | $V(Q=30,\tau=50)$ | (%) Rel. Diff |
| 0.0125 | 0.05 | 334.723 | – | 334.723 | – |
| | 0.025 | 333.619 | 0.3308 | 333.619 | 0.3308 |
| | 0.0125 | 333.090 | 0.1589 | 333.090 | 0.1589 |
| | 0.00625 | 332.830 | 0.0780 | 332.830 | 0.0780 |
| | 0.003125 | 332.702 | 0.0387 | 332.702 | 0.0387 |

Table 6.2 shows the convergence for the two PDE algorithms. To obtain these results we set the control discrete step to $\Delta\xi = 0.0125$ and we vary the time step from 0.05 to 0.003125. The convergence is shown for the point $V(Q = 30, \tau = 50)$. As we notice both methods give identical results with the percentage relative difference to be decreasing linearly with halving the timestep, justifying that the convergence is of order $O(\Delta\tau)$. In particular, for $\Delta\tau = 0.003125$ the relative percentage difference drops to less than 0.05% indicating that the solution is correct in three significant figures.

Although, the two methods agree at the "final" solution, the computational times have not been the same, with the value-policy scheme needing only one third of the time to compute the full solution. Relevant results may be seen in table 6.3 where we present the computational times needed to calculate the entire solution for $C = 1, 2, \ldots, 100$ and $\tau = [0, 50]$ using $\Delta\xi = 0.025$. The computation time improvement of the value-policy scheme is explained by two factors: First,

Table 6.3: PDE optimal solution. Computation times (in seconds) of the two numerical schemes for different grids.

| $\Delta\xi$ | $\Delta\tau$ | Exhaustive search | Value-policy |
|---|---|---|---|
| | 0.05 | 5.541 | 1.955 |
| | 0.025 | 11.912 | 3.820 |
| 0.025 | 0.0125 | 24.579 | 7.491 |
| | 0.00625 | 51.433 | 14.707 |
| | 0.003125 | 111.111 | 29.321 |

Table 6.4: Pre-calculation of matrices $p$ and $r$. Computation times (in seconds) for different grids.

| | $\Delta\tau$ | | | | |
|---|---|---|---|---|---|
| $\Delta\xi$ | 0.05 | 0.025 | 0.0125 | 0.00625 | 0.003125 |
| 0.05 | 0.88 | 1.75 | 3.54 | 7.1 | 13.86 |
| 0.025 | 1.76 | 3.48 | 6.97 | 13.88 | 27.75 |
| 0.0125 | 3.49 | 6.93 | 13.99 | 27.8 | 55.46 |

obtaining the optimal $\xi's$ to use at each point in the mesh requires no iterations as it comes directly out of equation (5.50). Second, that we do not have to iterate in order to obtain the result means that we can use the analytical solution of the conditional density of stay instead (5.32), which is found in appendix B. This achieves a further reduction in time as only the $r$ matrix has to be pre-calculated.

To show the extent at which the pre-calculation of matrices $p$ and $r$ affects the total computation time we provide table 6.4. For example, when we take the pair $\Delta\xi = 0.025$ and $\Delta\tau = 0.0125$ we observe that the time needed to pre-calculate both $p$ and $r$ has been just below 7 seconds. Note that since the computational work is roughly the same for both matrices we expect around 3.5 seconds to have been consumed for each of the matrices in isolation. Looking at the corresponding time it took for the exhaustive search method to compute the entire solution (24.5s in table 6.3) we deduce that it needed around 17 additional seconds; the combined effect of iterations to find $\xi^*$ and evaluation of the finite difference grid. On the other hand, the value-policy method only took 7.5 seconds in total, which is the combined sum of around 3.5 seconds to pre-calculate $r$ (as $p$ does not need to be pre-calculated) and additional 4 seconds for the evaluation of the finite difference grid by using the optimal $\xi$'s directly.

Lastly, table 6.5 examines the effect on the optimal solution of changing the size $\Delta\xi$. For these results we have used $\Delta\tau = 0.003125$. We notice that changing the discrete step from 0.1 to 0.0125 we improve the solution by just 1 significant figure, from total five to six. This indicates that the discrete step $\Delta\xi$ does not play a major role in forming the final result.

Table 6.5: Effect on the solution of changing the size of the control step $\Delta\xi$.

| $\Delta\xi$ | $V(Q = 30, \tau = 50)$ | (%) Relative Diff |
|---|---|---|
| 0.1 | 332.7004082 | – |
| 0.05 | 332.7013901 | $2.9 \times 10^{-4}$ |
| 0.025 | 332.7016286 | $7.0 \times 10^{-5}$ |
| 0.0125 | 332.7016878 | $2.0 \times 10^{-5}$ |

## 6.3 Pontryagins deterministic results

In this section we examine results using the Pontryagins method. As before, we begin by showing some results for the case where no optimisation is implemented, and thus all customers are allowed to stay based on a FCFS order. Once again the objective is to maximise the revenue to be generated on a day $T$ given that there are $\tau = 50$ days left until $T$. Figure 6.28 shows the deterministic trajectories of the selling of spaces as a function of the time remaining $\tau$ for different carpark sizes $C$. On the far right we start with an empty carpark and 50 days to sell the spaces ($Q(\tau = 50) = C$). As time progresses we sell the spaces at the rate shown by a particular trajectory. What we observe is that for carparks of size 60 and more (which is the expected total demand under the set of parameters used) the total demand to come within a 50 day window is lower than the capacity and as a result there are unsold spaces. However, in the region of carpark sizes of 60 and less, we sell the spaces too early in the horizon leaving the carpark with no empty spaces, while there is still much time remaining until $T$. This behaviour is problematic as we do not fully exploit the more profitable business customers who tend to arrive later closer to $T$. Recall that it is in this region where the Pontryagins maximum principle is implemented in order to optimise the selling rate of the spaces.

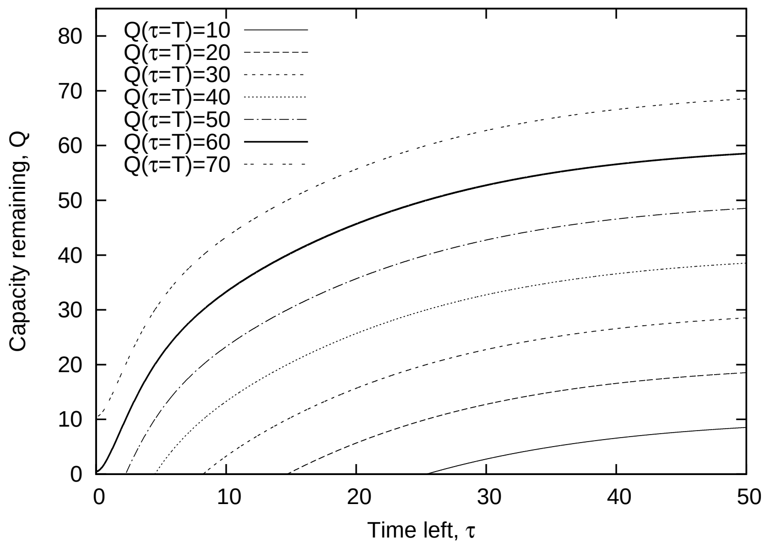



Figure 6.28: Non-optimised deterministic trajectories for the selling of the spaces.

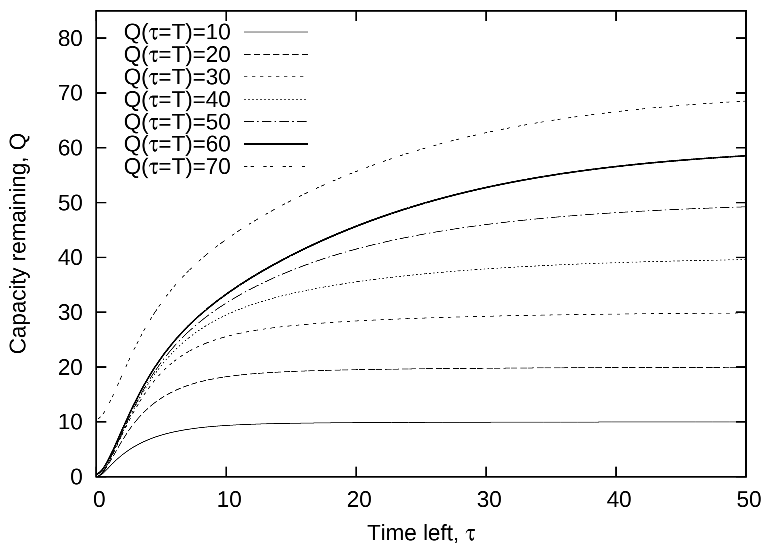



Figure 6.29: Optimal deterministic trajectories for the selling of the spaces.

In figure 6.29 we present the *optimal* deterministic trajectories of $Q$ as functions of the time left $\tau$, for varying carpark sizes $C$. Initially ($\tau = T = 50$) the carpark has the entire capacity available (i.e. $Q(\tau = 50) = C$). As time progresses the spaces are getting optimally sold effectively reducing the remaining spaces $Q$. At the target day ($\tau = 0$) we would have either sold the entire capacity ($Q(\tau = 0) = 0$) or are left with unsold capacity ($Q(\tau = 0) > 0$). For the trajectories that are left with zero capacity, we have that demand was greater than

Table 6.6: Optimal fixed lengths of stays $\xi^*$ for the Pontryagins method, under two numerical schemes.

| $C$ | Euler method | Runge-Kutta $4^{th}$ order method | Absolute % difference |
|---|---|---|---|
| 1 | 0.30794 | 0.30797 | 0.0078 |
| 5 | 0.80140 | 0.80143 | 0.0039 |
| 10 | 1.31566 | 1.31570 | 0.0030 |
| 15 | 1.87463 | 1.87468 | 0.0027 |
| 20 | 2.57408 | 2.57416 | 0.0031 |
| 25 | 3.57991 | 3.58003 | 0.0034 |
| 30 | 5.21663 | 5.21683 | 0.0038 |
| 35 | 7.63024 | 7.63046 | 0.0029 |
| 40 | 10.50440 | 10.50460 | 0.0019 |
| 45 | 13.91750 | 13.91770 | 0.0014 |
| 50 | 18.48930 | 18.48940 | 0.0005 |
| 55 | 26.58190 | 26.58190 | 0.0000 |
| 56 | 29.55280 | 29.55270 | 0.0003 |
| 57 | 34.07530 | 34.07480 | 0.0015 |
| 58 | 44.82300 | 44.81980 | 0.0071 |
| 60 | $\infty$ | $\infty$ | $\infty$ |
| 70 | $\infty$ | $\infty$ | $\infty$ |
| 80 | $\infty$ | $\infty$ | $\infty$ |

capacity and thus space optimisation was implemented to control the maximum length of stay $\xi$ allowed. For the remaining trajectories, capacity was abundant and therefore no restriction on the allowed length of stay $\xi$ has been imposed.

Recall that along each $Q$-trajectory the optimal value for $\xi^*$ is constant. Table 6.6 presents the optimal values of $\xi^*$ for a carpark with initial capacity $C$ and $\tau = 50$ days left. Two numerical methods for solving the IVP problem in (5.67) are shown; the Euler method and the $4^{th}$ order Runge-Kutta method (see Butcher, 2008).

As expected, the smaller the initial capacity, the more tight the restriction of the optimal maximum $\xi^*$. This is because, when capacity is scarce we should protect the spaces from low-paying customers so that we can sell them at higher price. On the other hand, for capacities of more than 60, no restriction on prices is needed and therefore the optimal $\xi^*$ is infinite (all customers accepted). The Euler method is the more desirable method (among the two presented) when speed is concerned; the computations took 5.61 seconds using the Euler method and 8.81 seconds using the Runge-Kutta method. If accuracy is more important however the Runge-Kutta $4^{th}$ order method might be more appropriate, although for a small timestep the relative percentage difference of the solutions has never been more than 0.008%, which may be regarded as insignificant.

Table 6.7: Calculate the optimal fixed length of stays $\xi^*$. Computation times (in seconds) for the two numerical schemes.

| $\Delta\tau$ | Euler method | Runge-Kutta method |
|---|---|---|
| 0.05 | 0.868 | 1.647 |
| 0.025 | 1.904 | 3.776 |
| 0.0125 | 5.284 | 8.383 |
| 0.00625 | 16.261 | 22.516 |
| 0.003125 | 49.052 | 60.462 |

In more detail, table 6.7 shows the computation times for the two methods to calculate the optimal values of $\xi$ for all carpark sizes $C = 1, \ldots, 100$.. The Euler method is quicker than the Runge-Kutta $4^{th}$ order method as it requires fewer calculations to approximate the derivative. However, the speed improvement works at the expense of solution accuracy.

### 6.3.1 Pontryagins vs PDE

Recall that with the Pontryagins method, once the optimal $\xi^*$ is found, the expected revenue generated in a carpark of size $C$ and $\tau$ time to go may be calculated by evaluating the double integral in equation (5.70).

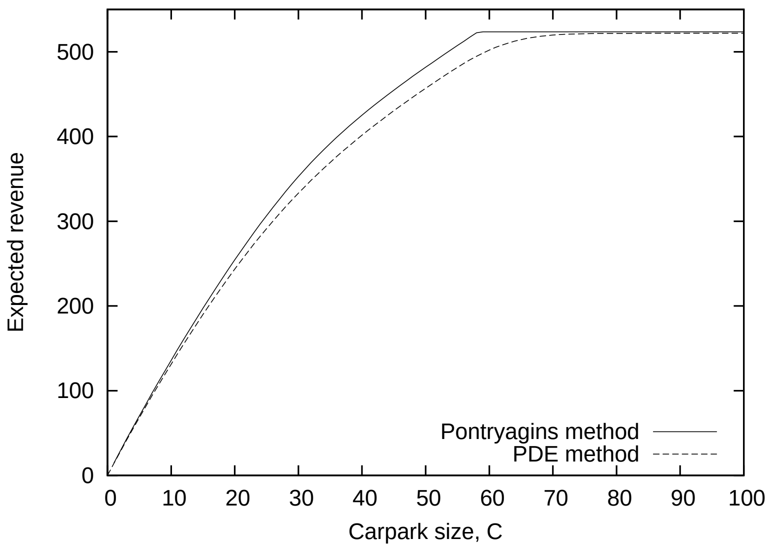



Figure 6.30: Comparison of the expected revenues of the PDE and the Pontryagins method. These are the expected revenues generated in a carpark with $\tau = 50$ days left, plotted as functions of the carpark capacity.

Figure 6.30 compares the expected revenues obtained by the PDE and the

Pontryagins method. These are the expected revenues generated in a carpark with $\tau = 50$ days left, plotted as functions of the carpark capacity. We notice that the Pontryagins deterministic method serves as an upper bound to our stochastic PDE model, formalising the idea that uncertainty in sales results in lower expected revenues.

### 6.3.2 Calculating the expected marginal values from Pontryagins solution

As mentioned already it is the expected marginal values that we wish to use and not the actual solution (revenues) directly. Unlike the case of the PDE method where the resulting expected values are used directly to obtain the expected marginal values for all $Q$ and $\tau$ and construct the bid-price control policy, with the Pontryagins this requires a rather more work. The Pontryagins method calculates the revenue to be generated from a carpark with $C$ available spaces and $\tau = T$ time remaining to sell them. In figure 6.29 $C$ was allowed to vary but the maximum advance time $T$ was kept fixed. 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 $\tau$ which is incredibly inefficient.

Fortunately, we do not have to do this because we can use the values of optimal $\xi^*$ to evaluate the marginal values through their relationship in equation (5.50). For example, from figure 6.29 if we want to calculate the marginal value when there are $Q$ spaces remaining and $\tau < 50$ time left, we can navigate to the required time and search through the different trajectories to find the one passing through $Q$. Once this is identified we can use the optimal fixed $\xi^*$ along this trajectory (or the interpolated $\xi^*$ between the two closest trajectories) to calculate $\Psi(\xi^*)$ which by (5.50) is the corresponding expected marginal value at that point.

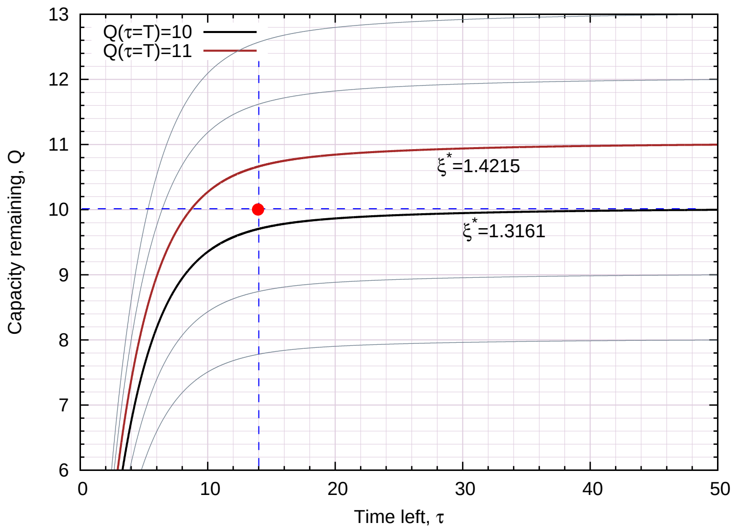



Figure 6.31: Calculation of marginal values through Pontryagins solution.

Figure 6.31 illustrates an example to calculate the marginal value when there are $Q = 10$ spaces remaining and $\tau = 14$ days to go. This is in fact the same as figure 6.29 but slightly magnified to emphasise the interpolating procedure we use. As we can see, there is no trajectory that falls exactly on the required combination (marked by the red dot). Thus, we take the two closest trajectories; these correspond to a carpark of initial size $C = 10$ and $C = 11$. In particular, from the trajectories $C = 10$ and $C = 11$ we use their corresponding optimal $\xi^*$ values which are $\xi^* = 1.3161$ and $\xi^* = 1.4215$, respectively. Linear interpolation suggests that

$$\xi^*(Q = 10, \tau = 14) = 1.3431$$

which implies that

$$\Delta V(Q = 10, \tau = 14) = \Psi(1.3431) = 12.64.$$

We can proceed in this way to construct the entire set of marginal values and thus create the bid-price control policy.

If we instead wanted to calculate the marginal value when there are $Q = 50$ spaces remaining and $\tau = 10$ days to go, we would obtain $\xi^*(Q = 50, \tau = 10) = \infty$ because the point lies above the last trajectory for which optimisation was implemented (see figure 6.29). Substituting $\xi^* = \infty$ in the pricing function leads to an optimal price $\Psi(\infty) = 5$ currency units. However, the opportunity cost does not equal to 5 since in this region $\Psi(\xi^*) \neq \Delta V(Q,\tau)$. This is because the opportunity cost only tells us the minimum price we should accept for the booking *not* how we should actually be *pricing* the product. Since expected demand to come when having only 10 days left is lower than 50, we should be willing to accept everyone with positive revenue contribution, i.e. the opportunity cost should be zero.

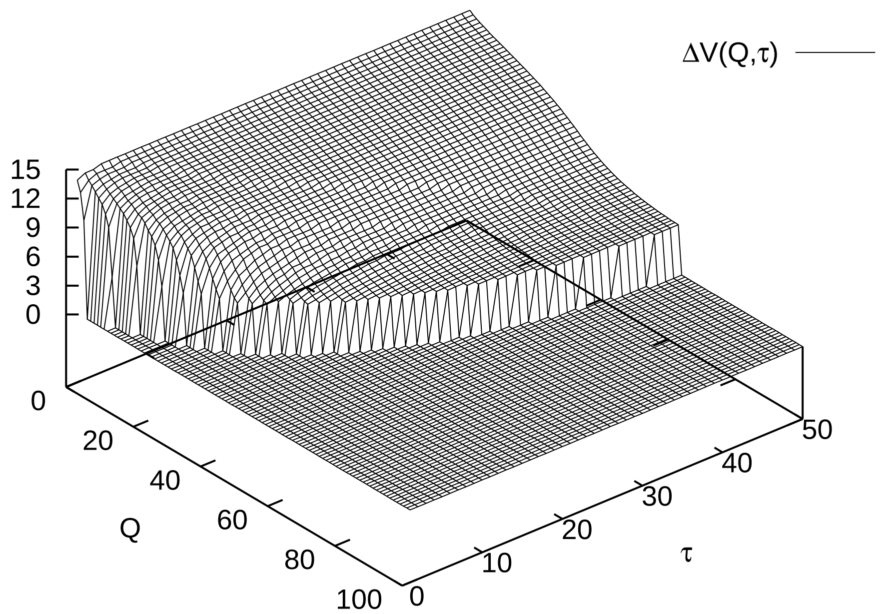



Figure 6.32: Bid-price table with expected marginal values derived from the Pontryagins solution.

The resulting surface of expected marginal values is illustrated in figure 6.32. The deterministic nature of the solution is apparent; the marginal values drop directly to zero once the level of capacity meets the expected demand (the $Q$-trajectory when $C = 60$). A comparison between this and figure 6.20 emphasises the role of uncertainty in the problem.

## 6.4 Performance of the three methods when approximating the network model

Finally we can examine the performance of the three derived methodologies, MC, PDE and Pontryagins, described in sections 5.1, 5.3 and 5.4 respectively, when these are used to approximate the network model. Recall that the DLP approach could serve as an upper bound to the solution and thus become an effective benchmark model for comparison among our derived methodologies; however, the resulting model has been to large to be solved in reasonable time and therefore it has not been used. Since there is no full network solution available to compare the results against, we need to define one of the methods as the benchmark model for the others. Note that any comparison between the MC and the other two methods can only be made in the limit $\Delta T \rightarrow 0$ because of the continuous-time nature of the PDE and Pontryagins models. Thus, in this section we consider that customers can stay for as little as $\Delta T = 0.0125$ and as a result we use the MC rejection policy in figure 6.12.

Our main objective has been to maximise the expected revenue that can be generated in the carpark on a future day $T$, given the size of the carpark and given the network structure of the problem. As pointed out in section 4.3.9, we examine this once the system is at a steady-state equilibrium position, i.e. when the initialisation effects have already been absorbed in the system.

Therefore, we have installed the rejection policies on day 0 when the carpark was entirely empty. Then we simulated a booking set and we let it run for a sufficient number of days to build up accordingly and eliminate all initialisation effects. Then, we have taken an interval of 20 consecutive days and calculated the expected total revenue to be made over this interval. This procedure is repeated under 1000 simulated paths. Finally, by dividing over the number of days in the interval and the number of booking sets we obtained an estimate of the expected value function, i.e the expected revenue per day for the given carpark as a perpetual quantity. Similarly, we can run this approach and calculate the expected perpetual revenue per day for any carpark-size $C$.

Figure 6.33 presents the expected perpetual revenues per day that are generated under the described procedure, as functions of the carpark size $C$. All the methods produce expected revenues that are increasing in the size of the carpark. Roughly speaking, the MC, PDE and Pontryagins seem to perform similarly,

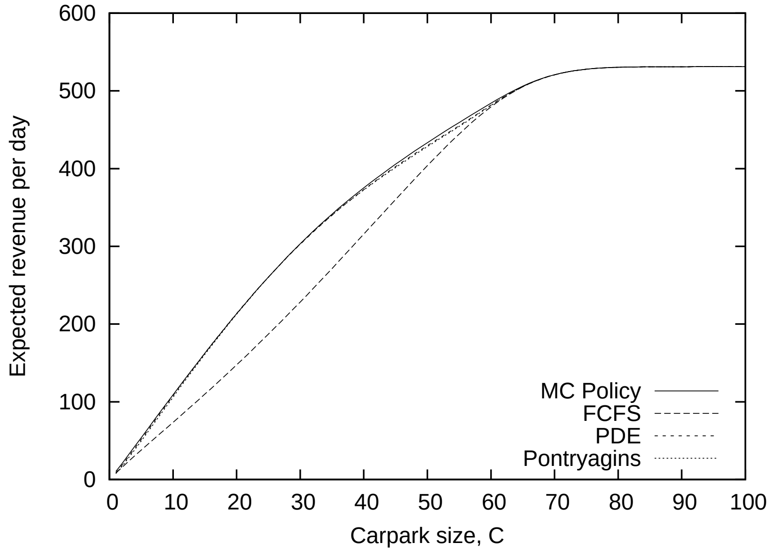



Figure 6.33: Expected perpetual revenue per day to be generated under the rejection policies of the four methods: MC, FCFS, PDE and Pontryagins.

while the FCFS fails to do so. For sufficiently large carparks (capacity greater than expected demand) we tend to accept *all* customers from both customer classes, irrespective of the rejection policy implemented. The difference in performance between the rejection policies, however, should be apparent especially when the expected demand is greater than supply, i.e. in the low-to-intermediate sized carparks. Thus, we seek to examine the results in more detail.

First, recall that the MC method in section 5.1 has been derived from the actual network model with only some simplifications in terms of the dimensionality of the problem. In contrast the PDE and Pontryagins methodologies differ mathematically and intuitively as well. Therefore, the MC is expected to better approximate the network solution for the given network model. As such, the solution (expected revenues) to the network model when using the MC rejection policy will form the benchmark model for which the other two approximation methods will be compared against. Let us denote by $V^{MC}$ the expected perpetual revenue for the network model after using the MC rejection policy with $\Delta T = 0.0125$. Also let us define by $V^{PDE}$ and $V^{Pontr}$ the expected perpetual revenues for the network model that result after implementing the PDE and the Pontryagins policy, respectively. Finally, $V^{FCFS}$ is the expected perpetual revenue under a FCFS policy. Therefore, by dividing over $V^{MC}$ gives us an indication of the methods' performance against the MC method.

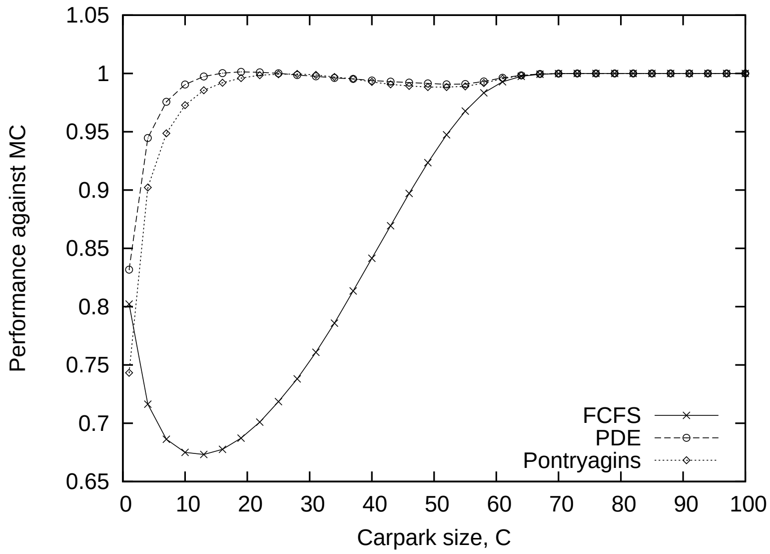



Figure 6.34: Relative comparison of the expected perpetual revenues for the three methods (against the MC).

Table 6.8 presents the expected perpetual revenues per day as these are calculated by implementing the four different rejection policies. We have evaluated this for any carpark size in $C = 1, 2, \ldots, 100$. In particular, when the capacity is highly congested, $1 \leq C \leq 10$, the MC method seems to work better producing greatest expected revenues. The improvement in performance ranges between $1 - 17\%$ against the PDE, and between $2.7 - 26\%$ against the Pontryagins, with the difference to be decreasing in carpark size. In the intermediate region $10 < C \leq 40$ the PDE and Pontryagins have achieved similar performance to the MC in the order of less than 1%. Finally, for capacities greater than 65 all methods work identically by allowing service to all the customers.

Figure 6.34 illustrates graphically the difference in performance of PDE, Pontryagins and FCFS methodologies as a relative measure against the MC solution.

According to our preliminary observations there are three regions of interest; low, intermediate and large capacities. Let us now attempt to give some insight into what is going on in these three regions, one by one. Clearly, the choice of a rejection policy to use in the network model should (at least partially) affect the result. Therefore, figure 6.35 is introduced and shows the expected marginal values for the three methods as functions of capacity.

At the region where the capacity is scarce $(1 \leq C \leq 10)$ the PDE and the Pontryagins marginal values are higher than the MC. This is because solving for

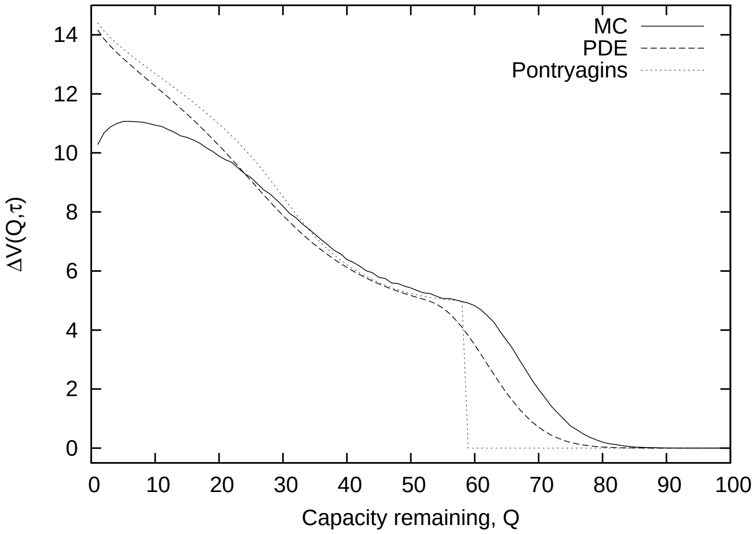



Figure 6.35: Expected marginal values for the three methods, as functions of capacity remaining $Q$, in the perpetual sense.

each day independently ignores the interdependence within the days and over-estimates the last spaces. As a result the EAMV algorithm falsely rejects too many low-paying customers and hence reduces the realised revenues. In addition, we recall that with these small capacities the assumption imposed on the MC method about how to obtain the marginal values should work reasonably well.

In the intermediate region the three methods (MC, PDE, Pontryagins) perform similarly. Interestingly, the PDE works even better than the MC in carparks of size $C = 17 - 23$. We believe that the explanation lies on the manner at which the MC marginal values work to optimise the bookings in intermediate sized carparks. To explain this, figure 6.36 illustrates a state of an intermediate sized carpark with $C = 11$ spaces, at some point in time. Imagine we want to decide whether to accept the next customer for a stay on day $T$. Since there is only one space remaining for $T$ we use the expected marginal value $\Delta V(Q = 1, 0; T, C = 11)$ which is approximated by $\Delta V(Q = 1, 0; T, C = 1)$ (see section 5.1). However, a customer requiring to stay on day $T$ *only* (and thus consume the last space on day $T$) would potentially block longer stay bookings from staying on and around $T$. Since capacity in the neighbouring days is abundant such bookings would have been accepted and thus contribute more in the total revenue. Therefore, for the single-stay booking to be accepted, the revenue generated should somehow compensate for the *loss* in expected revenue of blocking out these longer stay bookings. This implies that the *true* marginal value of the last space on day $T$ is higher than the marginal value computed by the MC approach. In contrast, the marginal value based on the PDE model is higher (as it is based on a single-resource model) and thus seems to be a better estimate on the true marginal value in this case (even though it was unintented).

| $C$ | $V^{MC}$ | $V^{PDE}/V^{MC}$ | $V^{Pontr}/V^{MC}$ | $V^{FCFS}/V^{MC}$ |
|---|---|---|---|---|
| 1 | 10.266 | 0.831 | 0.743 | 0.802 |
| 2 | 20.894 | 0.893 | 0.829 | 0.759 |
| 3 | 31.803 | 0.924 | 0.874 | 0.734 |
| 4 | 42.802 | 0.945 | 0.902 | 0.717 |
| 5 | 53.855 | 0.959 | 0.922 | 0.704 |
| 6 | 64.941 | 0.969 | 0.938 | 0.694 |
| 7 | 76.039 | 0.976 | 0.949 | 0.687 |
| 8 | 87.095 | 0.982 | 0.958 | 0.682 |
| 9 | 98.042 | 0.987 | 0.967 | 0.678 |
| 10 | 108.969 | 0.991 | 0.973 | 0.675 |
| 11 | 119.808 | 0.994 | 0.978 | 0.673 |
| 12 | 130.601 | 0.996 | 0.982 | 0.673 |
| 13 | 141.316 | 0.997 | 0.986 | 0.673 |
| 14 | 151.892 | 0.999 | 0.988 | 0.674 |
| 15 | 162.402 | 1.000 | 0.991 | 0.676 |
| 16 | 172.851 | 1.000 | 0.992 | 0.678 |
| 17 | 183.143 | 1.001 | 0.994 | 0.681 |
| 18 | 193.332 | 1.001 | 0.995 | 0.684 |
| 19 | 203.402 | 1.001 | 0.996 | 0.687 |
| 20 | 213.311 | 1.001 | 0.997 | 0.692 |
| 21 | 223.034 | 1.001 | 0.998 | 0.696 |
| 22 | 232.633 | 1.001 | 0.999 | 0.701 |
| 23 | 242.110 | 1.001 | 0.999 | 0.707 |
| 24 | 251.388 | 1.000 | 0.999 | 0.712 |
| 25 | 260.463 | 1.000 | 0.999 | 0.719 |
| 30 | 303.303 | 0.998 | 0.999 | 0.753 |
| 35 | 341.458 | 0.996 | 0.996 | 0.795 |
| 40 | 375.278 | 0.994 | 0.993 | 0.842 |
| 45 | 405.631 | 0.992 | 0.990 | 0.888 |
| 50 | 433.417 | 0.991 | 0.988 | 0.932 |
| 55 | 459.403 | 0.991 | 0.989 | 0.968 |
| 60 | 484.159 | 0.995 | 0.994 | 0.990 |
| 65 | 506.150 | 0.999 | 0.999 | 0.998 |
| 70 | 520.830 | 1.000 | 1.000 | 1.000 |
| 75 | 527.945 | 1.000 | 1.000 | 1.000 |
| 80 | 530.420 | 1.000 | 1.000 | 1.000 |
| 85 | 531.010 | 1.000 | 1.000 | 1.000 |
| 90 | 531.125 | 1.000 | 1.000 | 1.000 |
| 95 | 531.145 | 1.000 | 1.000 | 1.000 |
| 100 | 531.150 | 1.000 | 1.000 | 1.000 |

Table 6.8: Expected perpetual revenues for a carpark of size C.

It happens that for intermediate size carparks these peaks of having a scarce-capacity day surrounded by abundant-capacity days is a frequent observation and

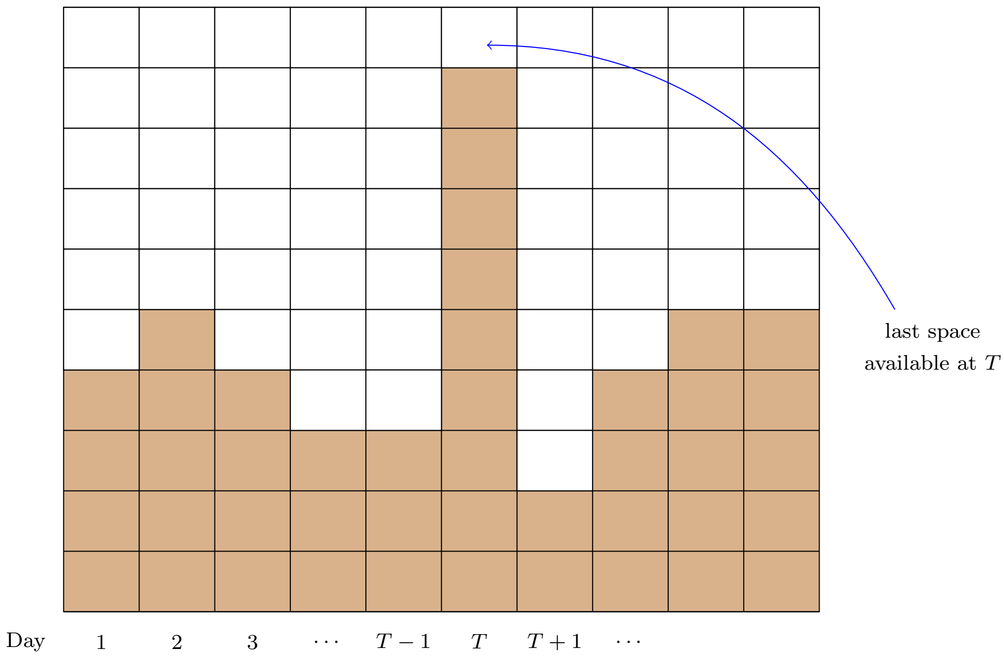



Figure 6.36: A state of a carpark of size $C = 11$, with a highly congested resource (peak) on day $T$. The brown boxes indicate spaces that have been already sold and the white boxes indicate the remaining available spaces.

as a result it creates this inefficiency of the MC method. Nonetheless the overall performance of the MC stays in reasonable levels.

Finally, for capacities greater than 65 all three methods seem to have performed identically. This is in fact the expected outcome because for relatively large capacities the role of a rejection policy is suppressed and turns into a FCFS policy, efficiently allowing everyone to stay.

For sake of discussion in figure 6.37 we present the average occupancy rates as these result from implementing the four rejection polices. As predicted, for low-to-intermediate capacities the FCFS policy naturally leads to higher utilisation of the spaces with up to 95% occupancy rates. However the high occupancy levels do not imply increased revenues as discussed previously. On the other hand, the MC method maintains a lower occupancy level (around 87%) in this region. For carparks greater than 60 the occupancy decreases proportionally to the size of the carpark as even accepting all the demand still cannot fill out the spaces. An important observation, however, is to notice the significant lower occupancy level of the method when capacity is too low, for example when $C = 1$. What happens here is that this single space is not occupied for every $\Delta T$-period within a day. In fact under the MC policy the slot stays empty for about 4.8 hours per day

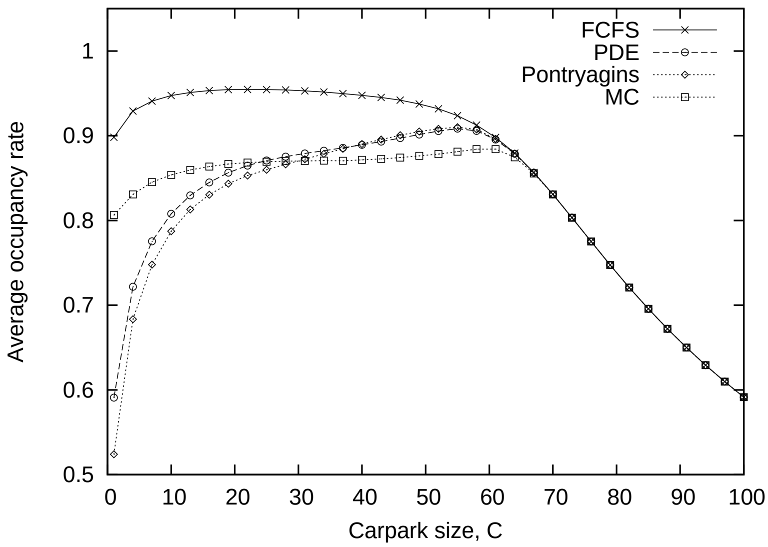



Figure 6.37: Average occupancy rate as function of the carpark size $C$, under the four rejection policies.

(20% of the day). Even worse, for the PDE and Pontryagins the overestimated price results in having the space empty for about 9 hours per day (40% of the day) and for about half a day, respectively.

### 6.4.1 Comparison under false estimates of the booking intensities

The PDE and the Pontryagins rejection policies performed reasonably well in maximising the perpetual expected revenue per day in the carpark, for all carpark sizes. However, the above results were based on the fact that the estimates of the average booking intensities used to calculate the rejection policies have been the correct values. In other words, we have defined from the very beginning that the average booking intensities for the two customer classes are set fixed and they are known; $\lambda_{b_1} = 5$ for the leisure class and $\lambda_{b_2} = 25$ for the business class. Then, when running the network model with an installed rejection policy, the bookings were simulated according to these exact same intensities.

In this section, we challenge the methods to see how well they would perform under false initial estimates. What we seek to do is to derive the rejection policies based on the probability distributions with slightly erroneous estimates but in the network model test them using bookings which are simulated according to

$\lambda_{b_1} = 5$ and $\lambda_{b_2} = 25$. These simulations will then represent the real-life scenario when bookings might behave slightly differently than what might have expected. Then, we will obtain an estimate of the expected revenue per day $\hat{V}$ which we compare against the correct estimate $V$. In particular, we have examined three such combinations which are summarised in table 6.9.

| Comb. | Leisure $\lambda_{b_1}$ | Business $\lambda_{b_2}$ | Expected demand |
|---|---|---|---|
| 1 | 6 | 18 | 60 (as before) |
| 2 | 6 | 22 | 64 (higher) |
| 3 | 4 | 21 | 49 (lower) |

Table 6.9: Combinations of average booking intensity estimates used to derive the PDE and Pontryagins rejection policies.

Table 6.10 presents the results from our tests regarding the PDE method. The three combinations show that for small carpark sizes the expected revenue per day will actually be greater than what it would have been under the correct estimates[3]. For combination 1 and 3 this happens because the marginal values are now lower due to the lower estimate on the high paying business class (from 25 to 18 and from 25 to 21, respectively). Thus, in the network model this results in more customers being accepted on average, rising the expected value from as little as 1% to as much as 5%. Combination 2 results in higher revenues purely because even if the total demand is higher, the estimate on the business demand is lower than the true estimate.

For medium-sized carparks the variations are in the order of at most 5% while for large carparks the error in the estimates seems not to have affected the resulting expected revenues.

Similar results are seen in the case of the Pontryagins policy. Table 6.11 presents the relevant numerical tests. Even though the Pontryagin's deterministic nature, the results of using erroneous estimates do not show great variations for the correct expected revenues.

[3]Recall that under the correct estimates the PDE policy overvalued the last remaining parking spaces and in the network model it accepted less sales, resulting in lower revenues than the MC policy.

| $C$ | $V^{PDE}$ | $\hat{V}_1/V^{PDE}$ | $\hat{V}_2/V^{PDE}$ | $\hat{V}_3/V^{PDE}$ |
|---|---|---|---|---|
| 1 | 8.483 | 1.048 | 1.019 | 1.029 |
| 2 | 18.674 | 1.035 | 1.015 | 1.020 |
| 3 | 29.341 | 1.031 | 1.013 | 1.018 |
| 4 | 40.384 | 1.029 | 1.012 | 1.017 |
| 5 | 51.712 | 1.024 | 1.010 | 1.015 |
| 6 | 63.021 | 1.018 | 1.007 | 1.011 |
| 7 | 74.152 | 1.017 | 1.006 | 1.010 |
| 8 | 85.413 | 1.015 | 1.007 | 1.010 |
| 9 | 96.736 | 1.012 | 1.006 | 1.009 |
| 10 | 107.850 | 1.010 | 1.005 | 1.008 |
| 15 | 162.614 | 0.999 | 1.002 | 1.002 |
| 20 | 214.050 | 0.990 | 0.999 | 0.996 |
| 25 | 261.068 | 0.982 | 0.996 | 0.988 |
| 30 | 303.348 | 0.975 | 0.994 | 0.980 |
| 35 | 340.648 | 0.972 | 0.993 | 0.973 |
| 40 | 373.506 | 0.971 | 0.993 | 0.967 |
| 45 | 402.794 | 0.970 | 0.994 | 0.962 |
| 50 | 429.535 | 0.972 | 0.995 | 0.964 |
| 55 | 454.968 | 0.982 | 0.996 | 0.980 |
| 60 | 481.501 | 0.994 | 0.998 | 0.994 |
| 65 | 506.512 | 0.999 | 1.000 | 0.999 |
| 70 | 522.744 | 1.000 | 1.000 | 1.000 |
| 75 | 531.090 | 1.000 | 1.000 | 1.000 |
| 80 | 534.126 | 1.000 | 1.000 | 1.000 |
| 85 | 534.899 | 1.000 | 1.000 | 1.000 |
| 90 | 535.019 | 1.000 | 1.000 | 1.000 |
| 95 | 535.019 | 1.000 | 1.000 | 1.000 |
| 100 | 535.019 | 1.000 | 1.000 | 1.000 |

Table 6.10: Expected perpetual revenues using the PDE rejection policy with wrong initial estimates for the average booking intensities.

| $C$ | $V^{PONTR}$ | $\hat{V}_1/V^{PONTR}$ | $\hat{V}_2/V^{PONTR}$ | $\hat{V}_3/V^{PONTR}$ |
|---|---|---|---|---|
| 1 | 7.629 | 1.062 | 1.024 | 1.038 |
| 2 | 17.352 | 1.049 | 1.021 | 1.030 |
| 3 | 27.839 | 1.037 | 1.014 | 1.022 |
| 4 | 38.616 | 1.034 | 1.014 | 1.021 |
| 5 | 49.636 | 1.033 | 1.014 | 1.019 |
| 6 | 60.939 | 1.029 | 1.010 | 1.017 |
| 7 | 72.255 | 1.024 | 1.010 | 1.014 |
| 8 | 83.537 | 1.020 | 1.010 | 1.013 |
| 9 | 94.946 | 1.018 | 1.007 | 1.010 |
| 10 | 106.204 | 1.017 | 1.007 | 1.010 |
| 15 | 161.312 | 1.009 | 1.005 | 1.006 |
| 20 | 213.219 | 1.001 | 1.003 | 1.002 |
| 25 | 260.812 | 0.990 | 1.000 | 0.996 |
| 30 | 303.709 | 0.977 | 0.996 | 0.983 |
| 35 | 340.601 | 0.971 | 0.993 | 0.974 |
| 40 | 372.926 | 0.968 | 0.993 | 0.965 |
| 45 | 401.801 | 0.968 | 0.993 | 0.960 |
| 50 | 428.203 | 0.974 | 0.994 | 0.962 |
| 55 | 453.673 | 0.984 | 0.998 | 0.981 |
| 60 | 480.908 | 0.995 | 0.999 | 0.995 |
| 65 | 506.398 | 0.999 | 1.000 | 0.999 |
| 70 | 522.740 | 1.000 | 1.000 | 1.000 |
| 75 | 531.090 | 1.000 | 1.000 | 1.000 |
| 80 | 534.126 | 1.000 | 1.000 | 1.000 |
| 85 | 534.899 | 1.000 | 1.000 | 1.000 |
| 90 | 535.019 | 1.000 | 1.000 | 1.000 |
| 95 | 535.019 | 1.000 | 1.000 | 1.000 |
| 100 | 535.019 | 1.000 | 1.000 | 1.000 |

Table 6.11: Expected perpetual revenues using the Pontryagins rejection policy with wrong initial estimates for the average booking intensities.

## 6.5 Discussion

Three approximation methods have been studied: MC, the PDE and the Pontryagins methods. All these methods attempt to estimate the expected marginal values of the spaces so that they can be used in a bid-price control to approximate the network model. We have demonstrated the methodology of each approach explicitly and presented some important numerical results. Starting from the MC method we have derived the optimal surfaces for the expected marginal values. These results are obtained by still solving a network but simplified problem. However, by using a large number of paths and simulations, the optimal surfaces were not sufficiently smooth (even after applying smoothing on them). What is more, obtaining the solution was computationally very expensive even for large choices of $\Delta T$.

Moving on to our next model, we have shown how to construct a continuous-time stochastic PDE model. Although this model no longer solves a network problem, the resulting surfaces have shown to be smooth and they were actually computed in less than a minute. Finally, building on the previous method we have solved its deterministic variant using the Pontryagins maximum principle and verified that the Pontryagins solution forms an upper bound to the stochastic PDE solution. The biggest advantage of the Pontryagins methods compared to the PDE is the computation time which has now been limited to just few seconds. However, further information is lost due to the assumption of the deterministic sales process.

In the previous section, we have used the resulting expected marginal values to construct a bid-price table and implement it to approximate the network model. Comparison between the performances of the three methods followed. Our findings show that overall the MC method has proved to perform better than the PDE or Pontryagins approaches. This is mainly because the MC still solves a network problem even though it is slightly simplified. However, this method is very computationally intensive and might not work well in real-life applications when policies might have to be updated frequently.

Therefore, what is more important was to find out how well the other two methods can perform in a network environment in an attempt to replace the MC method. Our results have shown that in small sized carparks although the MC performs best, the PDE generates higher revenues than the Pontryagins method. In fact, according to table 6.8 the absolute benefit of using the PDE method over the Pontryagins could potentially be from as little as 1.7% ($C = 10$) to as much as 9.5% ($C = 1$). Although for larger carparks the two methods perform similarly, in reality the important decision is to be made when there are only few spaces remaining; The negative effect of making a suboptimal decision and falsely accepting/rejecting a customer is more apparent when there is only one space available rather than when there are still a number of spaces available that may potentially balance out the losses later on.

Finally, a test was undertaken to check the performance of the two methods under false estimates of the booking, arrival and staying intensities. We would expect that in general the PDE would adapt better because it is derived based on stochastic demand compared to the Pontryagins approach which is based on deterministic demand. Nonetheless, both methods seem to be robust and perform always within 5% from the correct estimates. In general, both the PDE and the Pontryagins approaches may be viewed as stable policies in the longrun when are allowed to run into the system for sufficient time.

In conclusion, we believe that the PDE model is the favourable model to be used in practice because of its computation speed and better accuracy (compared to the Pontryagins) and thus the following chapters will be devoted in developing this approach even further.

# Chapter 7

# PDE model for a finite time interval

*Part of the work below has been published in ICORES 2013 International conference proceedings. "Continuous-Time Revenue Management in Carparks - Part Two: Refining the PDE" is available online in SciTePress Digital Library at http://www.scitepress.org/DigitalLibrary.*

The derived PDE model in (5.46) solves for the rate at which cash is generated through an infinitesimal time rather than over some discrete time as in the case of the MC method. Thus, a comparison on the optimal solution between the MC and the PDE methodologies could only be made in the limit i.e. as the time intervals in the MC tend to zero $\Delta T \to 0$.

However, most internet pre-booking systems work with pre-defined fixed time slots and customers are charged based on the number of time slots they stay rather than based on the length of stay (e.g. pre-booking systems used at Inventive IT). In fact, using fixed time slots is beneficial as customers are forced to pay more than their intended duration of stay.

On the one hand, the pricing structure of most carparks dictates that spaces are sold to customers in slots, typically an integer number of fixed periods of time, such as day or hour over which the space will be reserved (see Teodorović and Lučić, 2006; Guadix et al., 2011). On the other hand, Bitran and Caldentey (2003) argue that "the explosive growth of the Internet and e-commerce make the continuous-time model much more suitable in practice". Moreover, our results in chapter 6 have shown the superiority of the PDE method over the MC method with respect to both efficiency (smoothness of results) and computational speed, but the PDE method presented there could not capture discrete time intervals. These then provide the motivation for our next study to extend the PDE model so that it can solve for the rate at which value is generated within any time period of finite size $\Delta T$.

## 7.1 Model formulation

The derived PDE in (5.46) is based on the occupancy probability (5.30) and the booking acceptance rate (5.43) which are used to calculate the rate at which bookings turn up and stay as of time $t$ for the infinitesimal period at $T > t$, and the rate (per day) of cashflow running through that period.

We can still have the bookings arriving in a continuous time (by observing them at time $t$) but calculate the associated revenue rates generated within some discrete-time interval in the future, the size of which can be of any finite length $\Delta T$. What we want to evaluate then is $\tilde{V}(Q, t; T)$, the rate per day at which revenue is generated in the carpark with $Q$ spaces remaining for the future period $[T,\, T + \Delta T]$ as of time $t$.

In order to formulate this we still consider that bookings are made in continuous time but we now adjust the probability distributions in such a way that we capture the probability of a customer being present within the interval $[T,\, T+\Delta T]$, rather than the instant $T$, which takes place between $[z,\, z+\Delta T]$ days after the booking is made.

Given a stationary model the actual time in consideration $[T,\, T+\Delta T]$ does not matter, henceforth we concentrate on the expected revenue rate to be generated between $[z,\, z + \Delta T]$ days later and thus we need to evaluate the probability of a customer being present between $[z,\, z + \Delta T]$ days after booking.

Recall that $g(z)$ is the fraction of customers who are present in the carpark $z$ days after making their booking. Now let the *adjusted* occupancy probability $\tilde{g}(z; \Delta T)$ denote the fraction of customers who are present in the carpark between $z$ and $z + \Delta T$ days later. Thus, we get

$$\tilde{g}(z; \Delta T) = g(z) + \int_{z}^{z+\Delta T} \int_{0}^{\infty} \phi(\eta, \xi)\, d\xi\, d\eta. \tag{7.1}$$

The integral term in (7.1) calculates this extra region and gives the probability of a customer arriving between $[z,\, z + \Delta T]$ days later. Customers who arrive anytime within this period will contribute to the integral, irrespective of their duration of stay. In fact, it may be expressed analytically and consequently $g(z;\Delta T)$ simplifies to

$$\begin{aligned}\tilde{g}(z;\Delta T) &= g(z) + \Big(P_a(z+\Delta T) - P_a(z)\Big)\\ &= P_a(z+\Delta T) - P_d(z).\end{aligned} \tag{7.2}$$

Figure 7.1 shows the extra region (the light grey strip below $g(z,t)$) that needs to be accounted as well. This region is carefully selected so that the extra customers arriving are not counted more than once. In addition, figure 7.2 is provided which presents $\tilde{g}(\cdot)$ as function of $\Delta T$. Clearly, the greater the finite period the greater the probability of a customer to be present over that period and this is indeed what this figure shows.

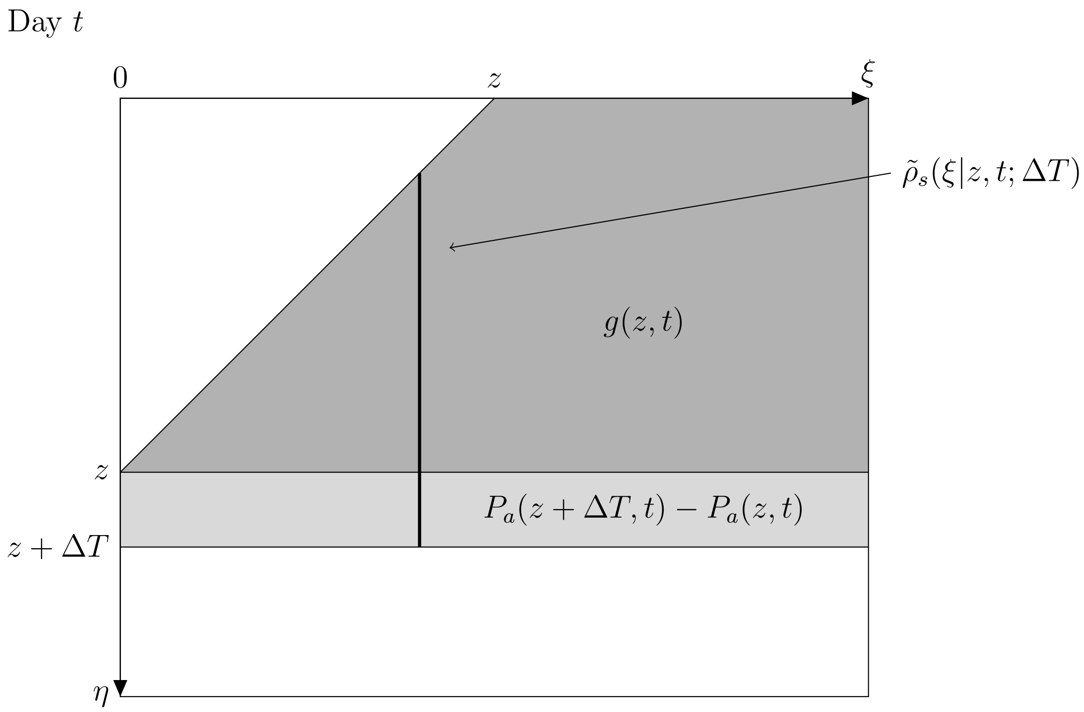



Figure 7.1: Extended region covered by $\tilde{g}(z,t;\Delta T)$ for bookings made at time $t$.

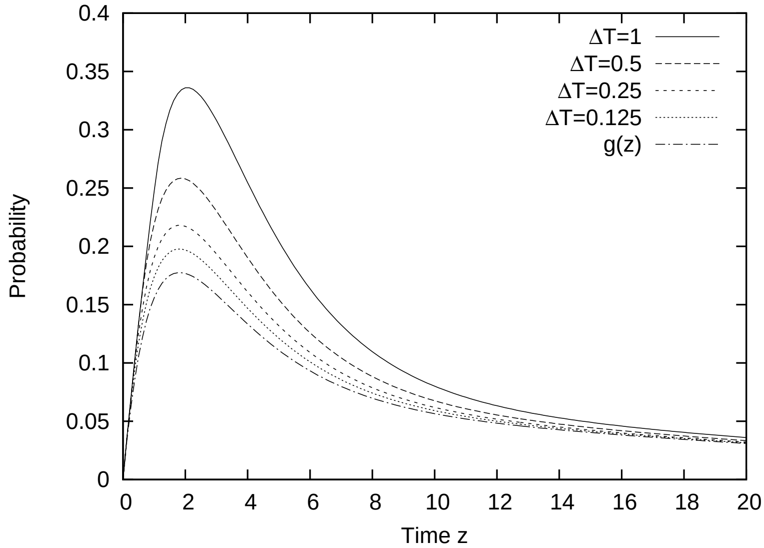



Figure 7.2: The adjusted occupancy probability $\tilde{g}(z;\Delta T)$, for different choices of $\Delta T$. $g(z)$ is also shown. Two classes, 1 and 2, are considered with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$ and $\lambda_{s_2} = 1$.

Similarly, we can show that the *adjusted* conditional probability density of a customer staying $\xi$ days given that the customer is present $[z, z+\Delta T]$ days later, $\tilde{\rho}_s(\xi|z;\Delta T)$, is given by

$$\begin{aligned}\tilde{\rho}_s(\xi|z;\Delta T) &= \frac{1}{g(z;\Delta T)}\int_{(z-\xi)^+}^{z+\Delta T} \phi(\eta,\xi)\, d\eta \\ &= \sum_n \alpha_n \frac{\rho_{s_n}(\xi)\Big(P_{a_n}(z+\Delta T) - P_{a_n}((z-\xi)^+)\Big)}{g(z;\Delta T)} \qquad (7.3)\end{aligned}$$

and it is shown in figure (7.1) as the thin vertical strip. Figure 7.3 is also added to present $\tilde{\rho}_s(\cdot)$ as a function of $\Delta T$, with $z$ fixed at $z = 2$.

In addition, the *adjusted* cumulative probability of staying not more than $\xi$ days given that the customer is present within $[z,\ z+\Delta T]$ days later, $\tilde{P}_s(\xi|z;\Delta T)$, is given by

$$\tilde{P}_s(\xi|z;\Delta T) = \int_0^{\xi} \tilde{\rho}_s(\xi'|z;\Delta T)\, d\xi'. \qquad (7.4)$$

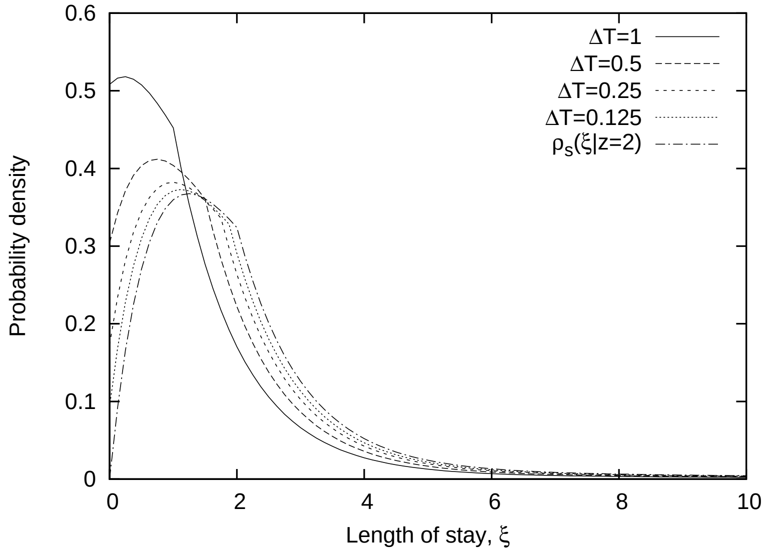



Figure 7.3: The adjusted conditional probability density $\tilde{\rho}_s(\xi|z = 2; \Delta T)$ for different choices of $\Delta T$. $\rho_s(\xi|z = 2)$ is also shown. Two classes, 1 and 2, are considered with $\lambda_{b_1} = 5$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$ and $\lambda_{s_2} = 1$.

Furthermore, let $\tilde{f}(z; \Delta T)$ to be the average intensity for bookings so that they are present between $[z,\ z + \Delta T]$ days later. This can be expressed as

$$\tilde{f}(z; \Delta T) = \left( \sum_n \lambda_{b_n} \right) \tilde{g}(z; \Delta T). \tag{7.5}$$

In section 5.3 we looked at all bookings that are present during the same infinitesimal and given that bookings could only be distinguished by their length of stay (the larger the length of stay, the less the price to be paid per day) we imposed a rule to reject those of length greater than $\xi^*$. In our new formation, the idea is similar with the only difference that we are now looking at all bookings that are present at any time within a finite time interval. Unfortunately, this increases the difficulty of the problem as bookings present in the same period, although being of the same length of stay, might have pay a different price rate according to how many time periods (of size $\Delta T$) each falls into in total; this depends on the time of arrival and departure given that they are staying during the period. Therefore, we suggest a simple procedure to estimate the expected number of time periods (E$[D]$) for which the customers are likely to occupy the slot, which will in turn enable us to determine a better estimate for the price rates that should be applied. We note that $D$ should strongly depend on the size of the time period, $\Delta T$.

In particular, we can find $k \in \mathrm{Z}$ such that the required length of stay $\xi$ is between $k$ and $k+1$ times larger than the length of the interval $\Delta T$, i.e. $k\Delta T \leq \xi \leq (k+1)\Delta T$. More precisely, $k$ satisfies

$$k = \left\lfloor \frac{\xi}{\Delta T} \right\rfloor, \tag{7.6}$$

where $\lfloor x \rfloor$ is the largest integer not greater than $x$.

To a first approximation, we assume that customers arrival times *given* a particular length of stay follow a uniform distribution, $u$.

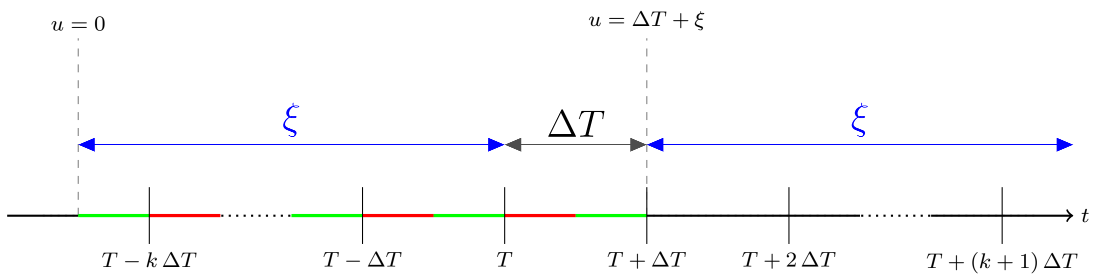



Figure 7.4: Calculate the expected number of stay periods, $\mathrm{E}[D]$, of the customers that are present between $[z,\, z + \Delta T]$ time after the booking and have length of stay $\xi$. We use a uniform distribution $u(0,\, \Delta T + \xi)$ for the customers conditional arrival times given the specified length of stay $\xi$.

Figure 7.4 illustrates the situation; in order for customers who stay for $\xi$ days to be accounted within the time interval $[T,\, T + \Delta T]$, they must have arrived no more that $\xi$ days before $T$ and no later than $T + \Delta T$. Thus the feasible region, within which customers contribute to the solution, is $T - \xi \leq t \leq T + \Delta T$. Therefore, the uniform distribution's endpoints become $0 \leq u \leq \Delta T + \xi$.

Regarding the number of periods $(D)$ the customers are likely to occupy a slot for, there are only two possible scenarios;

1. the customer stays $k+1$ periods when their required duration of stay covers $k$ periods plus a fraction of an additional period either before or after this interval (their arrival time lies on a red segment of figure 7.4),

2. or stays $k+2$ periods when their required duration of stay covers the same

$k$ periods plus a fraction before *and* after this interval (their arrival time lies on a green segment of figure 7.4).

In particular, we have

$$\begin{aligned} P\left(D=k+1\right) &= P(\xi - k\Delta T \leq u \leq \Delta T) + P(\xi - (k-1)\Delta T \leq u \leq 2\Delta T) \\ &\quad + \cdots + P(\xi \leq u \leq (k+1)\Delta T) \\ &= \frac{(k+1)\Delta T - \xi}{\xi + \Delta T} + \frac{(k+1)\Delta T - \xi}{\xi + \Delta T} + \cdots + \frac{(k+1)\Delta T - \xi}{\xi + \Delta T} \\ &= (k+1)\left(\frac{(k+1)\Delta T - \xi}{\xi + \Delta T}\right), \end{aligned} \tag{7.7}$$

while

$$\begin{aligned} P\left(D=k+2\right) &= 1 - P\left(n=k+1\right) \\ &= \frac{(k+2)\xi - \left(1-(k+1)^2\right)\Delta T}{\xi + \Delta T}. \end{aligned} \tag{7.8}$$

Therefore, $D$ is given by

$$D = \begin{cases} k+1 \quad \text{w.p} \quad P_1 = (k+1)\left(\frac{(k+1)\Delta T - \xi}{\xi + \Delta T}\right) \\ \\ k+2 \quad \text{w.p} \quad P_2 = \frac{(k+2)\xi - \left(1-(k+1)^2\right)\Delta T}{\xi + \Delta T} \end{cases}$$

Consequently, the expected number of periods becomes

$$\mathrm{E}[D] = (k+1)\,P_1 + (k+2)\,P_2. \tag{7.9}$$

The relationship between the required length of stay $\xi$ and the *expected length of stay* $\mathrm{E}[D]\,\Delta T$ (which is the expected number of periods multiplied by the size of the period) is better understood from figure 7.5 where we plot the second with regards to the first for different choices of $\Delta T$. In this figure the dotted line marks the equation $y = \xi$ and is added for better interpretation of the graphs. We notice, that under the discretisation of the time-slots customers are staying for more than their intended duration of stay (the curves lie above the line $y = \xi$) for all choices of $\Delta T$. Consequently, the demand for spaces rises and so is the

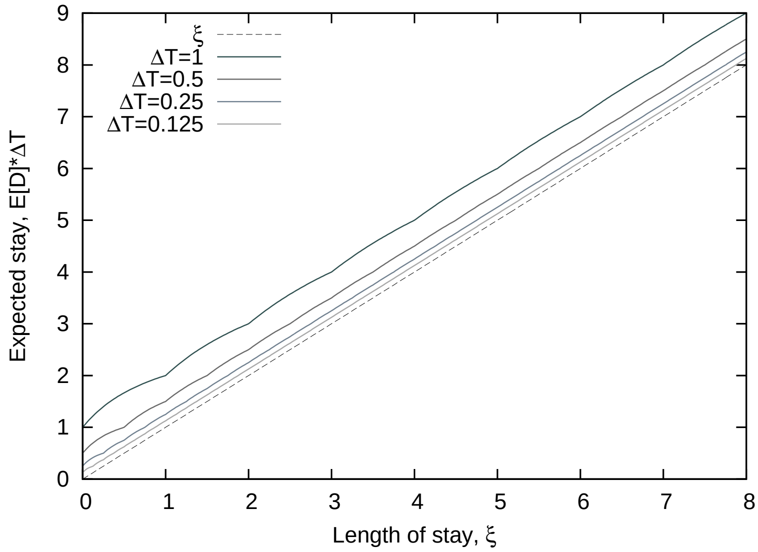



Figure 7.5: Relationship between required length of stay, $\xi$, and expected length of stay, $\mathrm{E}[D]\,\Delta T$, for different choices of $\Delta T$.

resulting revenue.

Therefore, for a given required duration of stay $\xi$, we use equations (7.6) and (7.9) to compute the expected number of time periods (of length $\Delta T$) the customer will be staying for ($E[D]$) and based on this number we can calculate the required price rate per day by the adjusted pricerate function as,

$$\tilde{\Psi}(\mathrm{E}[D]; \Delta T) = \psi_1 + \psi_2 e^{-\mu \mathrm{E}[D] \Delta T}. \tag{7.10}$$

Note, that this function will not be a discrete step function as, in contrast to the pricing function (6.4), the $\mathrm{E}[D]$ varies continuously with the choice of $\xi$ (as shown in figure 7.5).

Figure 7.6 compares the adjusted price rate function (7.10) with the corresponding discrete-jump price function in (6.4) that has been used for the MC convergence. The price rate is shown for different sizes of $\Delta T$. In each figure the continuous pricing function (4.3) is plotted as well. We notice that the adjusted price-rate function has the property that the expected price rate is always lower than that of the jump price function; this is because there is always the possibility that a booking spans over more time-slots that one might expect. This feature is more pronounce for larger time intervals $\Delta T$ because the price difference between a single-period and a two-period stay is more significant. Last, the two functions approach one another as $\Delta T$ decreases and, in fact, they become equal in the limit (as $\Delta T \to 0$). In the limiting case both pricing functions are equal to the continuous pricing function (4.3).

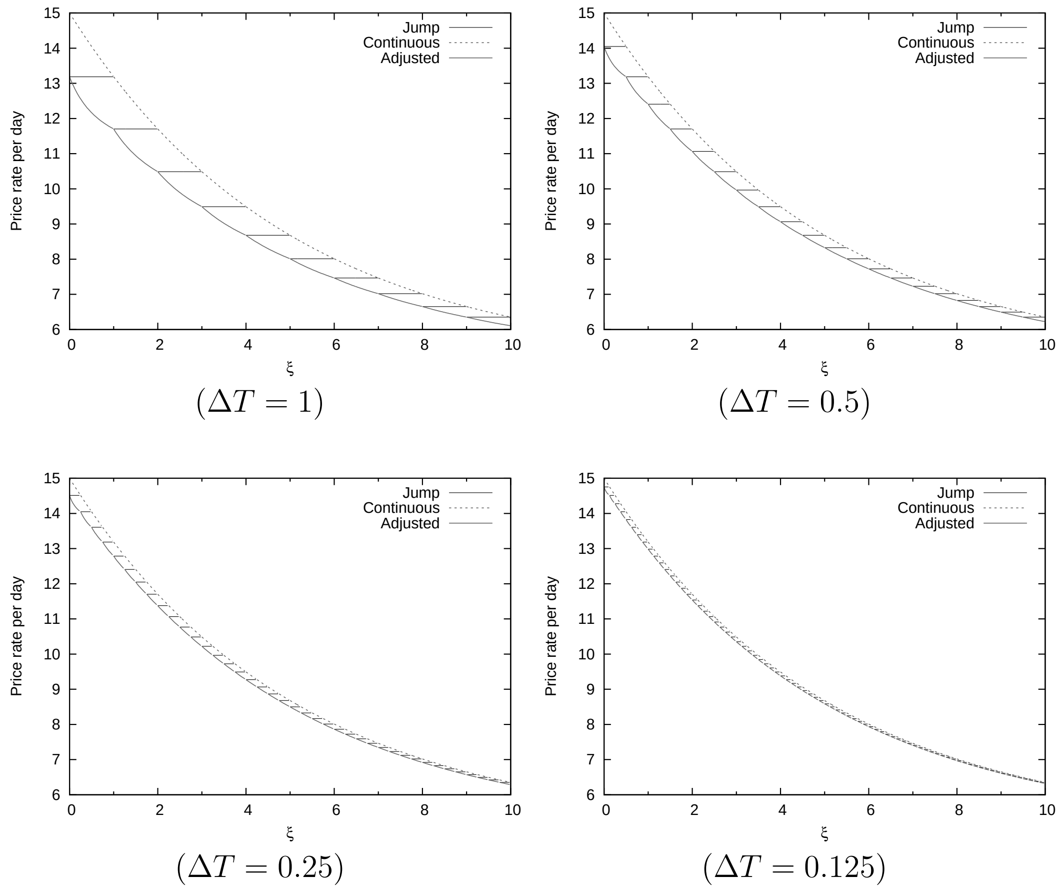



Figure 7.6: Plots of the adjusted price rate function used in the reformulated PDE and the discrete-jump price function used in the MC, for varying choices of $\Delta T$. The continuous pricing function (4.3) is shown as well.

Finally, we define $\tilde{V}(Q, \tau; \Delta T)$ as the *rate per day* at which revenue is generated from cars present over the period interval which is formed between $[\tau,\ \tau + \Delta T]$ days later.

Therefore, the *adjusted* PDE can be written as

$$\frac{\partial \tilde{V}}{\partial \tau} = \max_{\xi^*} \left\{ \tilde{P}_s(\xi^*|\tau) \tilde{f}(\tau) \Big( \tilde{V}(Q-1, \tau) - \tilde{V}(Q, \tau) \Big) + \tilde{f}(\tau) \int_0^{\xi^*} \tilde{\rho}_s(\xi|\tau) \tilde{\Psi}(\xi)\, d\xi \right\}, \tag{7.11}$$

with the same boundary conditions as before

$$\tilde{V}(Q,0) = 0 \quad \forall Q \tag{7.12}$$
$$\tilde{V}(0,\tau) = 0 \quad \forall \tau. \tag{7.13}$$

The solution to this system gives the optimal rate per day at which the value is generated in the future time period which takes place between $\tau$ and $\tau + \Delta T$ days later and the values $\xi^* = \xi^*(Q,\tau)$ that achieve the supremum construct the optimal rejection policy.

Note that the solution is always interpreted as *value rate per day* irrespective of the size of $\Delta T$. Also notice that if the revenue rate (second term to the right of equation 7.11) is multiplied by the size of the interval $\Delta T$ then the solution would calculate the *total revenue* generated between $\tau$ and $\tau + \Delta\tau$ days later. However, we prefer to factor out $\Delta T$ and keep the solution as a rate so that we can compare among the different $\Delta T$'s.

## 7.2 Numerical scheme

When it comes to solving equation (7.11) the procedure is similar to that for the original PDE in (5.46) and thus it will also be solved using an explicit finite difference scheme (see Smith, 1985, for details).

Firstly, we construct the mesh. In this stationary case the mesh has only two dimensions, the advance-time $\tau$ and the capacity remaining $Q$. Suppose that the domain we will work on is rectangular with $\tau$ ranging from 0 to $T$ and $Q$ ranging from 0 to $C$. Divide $[0,\ T]$ into $K$ equally spaced intervals at $\tau$ values indexed by $k = 0, 1, \ldots, K$. Similarly, we divide $[0,\ C]$ into $C$ equally spaced intervals at $Q$ 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 $\Delta Q = 1$ in the state direction such that $\tau^k = k\Delta\tau\ \forall k$ and $Q_j = j\ \forall j$. We seek an approximation to the values of $\tilde{V}$ at the $(K+1)\times(C+1)$ grid points.

Therefore,

$$\tilde{V}(Q_j,\tau^i) = \tilde{V}(j, k\Delta\tau) \approx \tilde{v}_j^k,$$

where $\tilde{v}$ is a $2D$ array.

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

Then, we may approximate the conditional probability distribution $P_s(\cdot)$ by

$$\tilde{P}_s(\xi^i|\tau^k) = \tilde{P}_s(i\Delta\xi|k\Delta\tau) \approx \tilde{p}_i^k.$$

Moreover, the average intensity in equation (5.33) may be written as

$$\tilde{f}(\tau^k) = \tilde{f}(k\Delta\tau) \approx \tilde{f}^k.$$

Consequently, the integral term on the RHS of the equation (5.46) may be written as

$$\int_0^{\xi^i} \tilde{\rho}_s(\xi'|\tau^k)\,\tilde{\Psi}(\xi')\,d\xi' = \int_0^{i\Delta\xi} \tilde{\rho}_s(\xi'|k\Delta\tau)\,\tilde{\Psi}(\xi')\,d\xi' \approx \tilde{r}_i^k.$$

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

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

Combining, the above we may write the numerical scheme as

$$\tilde{v}_j^{k+1} = \tilde{v}_j^k + f^k \max_i \left\{ \tilde{p}_i^k\,(\tilde{v}_{j-1}^k - \tilde{v}_j^k) + \tilde{r}_i^k \right\}\Delta\tau,$$

with the boundary conditions

$$\tilde{v}_j^0 = 0 \quad \forall j \tag{7.14}$$

$$\tilde{v}_0^k = 0 \quad \forall k. \tag{7.15}$$

For these results we have used the exhaustive search algorithm proposed in section 5.3.4.

In simple terms, we again solve with respect to $\tau$ starting from the initial condition at $\tau = 0$. However, at every point in the $\tau$ axis we rather compute the probabilities of being present between $\tau$ and $\tau + \Delta T$ days later (instead of the probabilities of being present *exactly* $\tau$ days later.).

Let us now investigate this scheme a little further. When solving this scheme we begin at $\tau^0$ with the initial condition $\tilde{V} = 0$. When we then move to evaluate

$\tau^1 = \Delta\tau$ the solution will give the value rate to be generated between $\tau^1$ and $\tau^1 + \Delta T$ time later, which is the time between $\Delta\tau$ and $\Delta\tau + \Delta T$ days later. As we proceed even further we will be evaluating the value rates between $[2\Delta\tau,\, 2\Delta\tau + \Delta T]$, $[3\Delta\tau,\, 3\Delta\tau + \Delta T], \ldots, [T,\, T + \Delta T]$ days later. An obvious limitation of this scheme is that in this manner there is no way to account for the value rate between 0 and $\Delta T$ days later.

Therefore, we propose to use a slightly modified scheme that will now calculate for the value rate between $\tau - \Delta T$ and $\tau$ days later[1]. In this way, we can actually evaluate the value rate between 0 and $\Delta T$ days later. In fact, this is achieved when we reach the time step $\tau^k = \Delta T$ for some $k$. As such, at $\tau^K = T$ the solution will refer to the value rate which is generated between $T - \Delta T$ and $T$ days later.

It is also important to notice that an issue arises for the time steps $\tau^k$ that are smaller than $\Delta T$ in absolute value. In these cases the left side of the relevant interval period (i.e. $\tau^k - \Delta T$) will be negative and the solution will fail. Thus, we fix this mathematically by setting it to $\max\{\tau^k - \Delta T,\, 0\}$.

Note that the above modifications are only made for computational convenience and they should not alter the interpretation of the approach described initially.

## 7.3 Numerical results

In this subsection we go through some numerical results for the cases $\Delta T = 1.0, 0.5, 0.25, 0.125$. Recall that $\Delta T = 1$ indicates that the parking slots are sold to customers for the minimum of a day, $\Delta T = 0.5$ for a minimum of half a day and so on. The parameters we use for the results are the same as those in chapter 6. For convenience we have re-stated them in table 7.1.

Table 7.1: Model parameters

| Class | $\lambda_b$ | $\lambda_a$ | $\lambda_s$ |
|---|---|---|---|
| $\mathcal{B}_1$ | 5 | 1/14 | 1/7 |
| $\mathcal{B}_2$ | 25 | 1/3 | 1 |

| $\Psi(\cdot)$ | $\psi_1$ | $\psi_2$ | $\mu$ |
|---|---|---|---|
| | 5 | 10 | 1/5 |

[1] This requires that the probability distributions described above are adjusted to account for the period $[z - \Delta T,\, z]$ instead of $[z,\, z + \Delta T]$

### 7.3.1 Results on the optimal solution of the adjusted PDE

For each choice of $\Delta T$ we solve the adjusted PDE and compute the resulting set of marginal values $\Delta\tilde{V}(Q,\tau)$ and store them in a $2D$ array as before. Once again these values will mark the rejection policy of our method and in section 7.3.2 they will be tested inside the network model. Table 7.2 shows the computation times to calculate these arrays, for each choice $\Delta T$. For comparison purposes we repeat ourselves by adding the computation times of the MC method that were initially presented in table 6.1. We observe that by reducing $\Delta T$ the computation time seems to increase but only for a tiny amount. In particular, computation times never go beyond one minute. This is because the scheme is independent of the choice of $\Delta T$ it only serves as a simple parameter in the model. Compared to the MC method where the computation time increases linearly in $\Delta T$ the PDE model improves computations times significantly as it only takes 40% of that time when $\Delta T = 1$ down to just to 4% when $\Delta T = 0.125$.

Table 7.2: Adjusted PDE solution. Computation times (in seconds) for different choices of $\Delta T$.

| $\Delta\xi$ | $\Delta\tau$ | $\Delta T$ | Adjusted PDE | MC method |
|---|---|---|---|---|
| 0.025 | 0.00625 | 1.0 | 51.2 | 126.7 |
| | | 0.5 | 53.7 | 268.8 |
| | | 0.25 | 55.1 | 621.6 |
| | | 0.125 | 56.7 | 1334.3 |

Let us now move on to figure 7.7 where we present the expected marginal values (optimal rejection policies) for the four choices of $\Delta T$ under investigation. A brief look at the $3D$ figures tells us that they all look quite similar in shape. However we cannot clearly see their differences, if any. Thus, in figure 7.8 we examine a slice of each figure at a fixed value of $\tau$. In particular we choose to work with $\tau = 50$ as the shape of this final curve foreshadows the pattern of the entire $3D$ figure. The differences between the four cases are now apparent. In particular, as $\Delta T$ increases we make two major observations:

- we tend to value spaces lower for small sized carparks.
  This is because when there is only one space ($Q = 1$) in the carpark on a given day, for instance, we seek to extract the maximum revenue for the day by targeting the space to the highest paying customers; these are a

single-day-stay customer when $\Delta T = 1$, two half-day-stay customers when $\Delta T = 0.5$, four six-hour-stay customers when $\Delta T = 0.25$ or eight three-hour-stay customers. These customers should not just arrive on the same day but also arrive in an order of one after the other so that the same parking slot is given to all. If there is indeed this demand every day then we will tend to receive $\tilde{\Psi}(D = 1; \Delta T = 1) = 13.24$, $\tilde{\Psi}(D = 1; \Delta T = 0.5) = 14.04$, $\tilde{\Psi}(D = 1; \Delta T = 0.25) = 14.51$ and $\tilde{\Psi}(D = 1; \Delta T = 0.125) = 14.75$, per day for each $\Delta T$ choice, which explains the curves heights on their left side.

- we tend to value spaces higher for bigger sized carparks.
  The expected total demand $\mathrm{E}[TD]$ for the spaces rises because the expected duration of stay $\bar{\xi} = 1/\lambda_s$ rises[2]. To understand this, recall figure 7.5 where we have seen that customers indeed stay for longer than intended. What is more, the greater the $\Delta T$ the more distant from the line $y = \xi$ the curve is, indicating that customers with a given intended stay will have to stay relatively longer when the spaces are sold per day rather than per hour.

  We can generalise this result to refer to the customer classes rather than the individual customers. In particular, under the purely continuous case and our model parameters in table 7.1, the expected total demand is $\mathrm{E}[TD] = 60$. Once we introduce the notion of the finite interval, the staying intensities ($\lambda_s$) of each customer class change. The exact amount by which they change can be evaluated using equation (7.9) with $k$ computed as

$$k = \left\lfloor \frac{\bar{\xi}_n}{\Delta T} \right\rfloor ,$$

  where $\bar{\xi}_n$ is the average stay time of class $n$, namely $\bar{\xi}_n = 1/\lambda_{s_n}$. The relevant results are summarised in table 7.3 for the different choices of $\Delta T$.

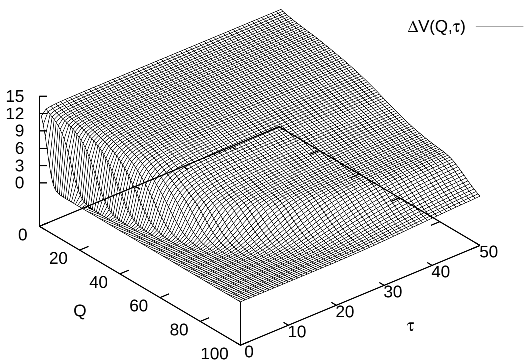
### 7.3.2 Performance of the adjusted PDE in the Network model

Now that we have explained the effect of the choice $\Delta T$ we may proceed to examine the performance of the adjusted PDE's optimal rejection policy in the network

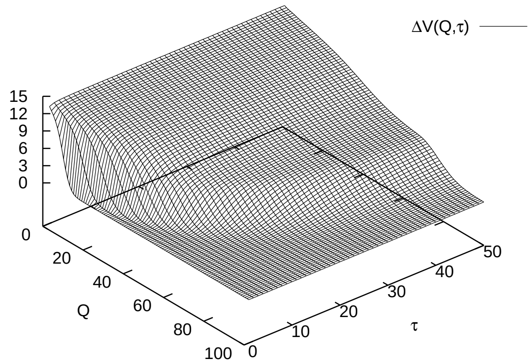
[2]Recall that the expected total demand is given by $\mathrm{E}[TD] = \sum_n \lambda_{b_n}/\lambda_{s_n}$ where $1/\lambda_{s_n}$ is expected duration of stay from class $n$.

Table 7.3: Effect of using discrete time-slots on the expected total demand.

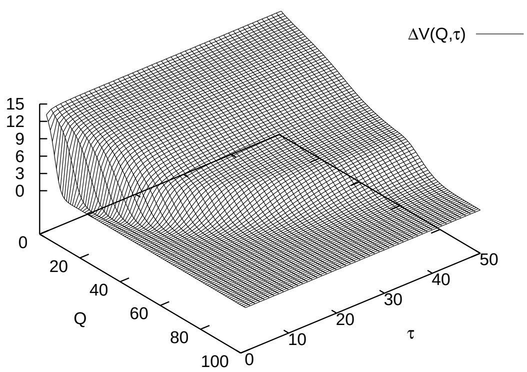
| | Continuous | | | With adjustment | | |
|---|---|---|---|---|---|---|
| $\Delta T$ | $\bar{\xi}_1$ | $\bar{\xi}_2$ | $\mathrm{E}[TD]$ | $\bar{\xi}_1$ | $\bar{\xi}_2$ | $\mathrm{E}[TD]$ |
| 1.0 | 7 | 1 | 60 | 8 | 2 | 90 |
| 0.5 | 7 | 1 | 60 | 7.5 | 1.5 | 75 |
| 0.25 | 7 | 1 | 60 | 7.25 | 1.25 | 67.5 |
| 0.125 | 7 | 1 | 60 | 7.125 | 1.125 | 63.75 |



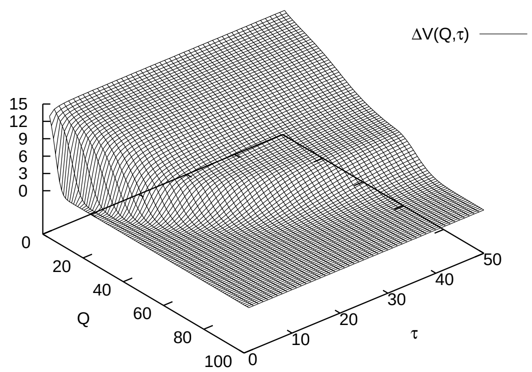

($\Delta T = 1$)




($\Delta T = 0.5$)




($\Delta T = 0.25$)




($\Delta T = 0.125$)

Figure 7.7: Bid-price tables with expected marginal values $\Delta\tilde{V}(Q,\tau)$ from the adjusted PDE, for varying choices of $\Delta T$, as functions of the time left $\tau$ and the capacity remaining Q.

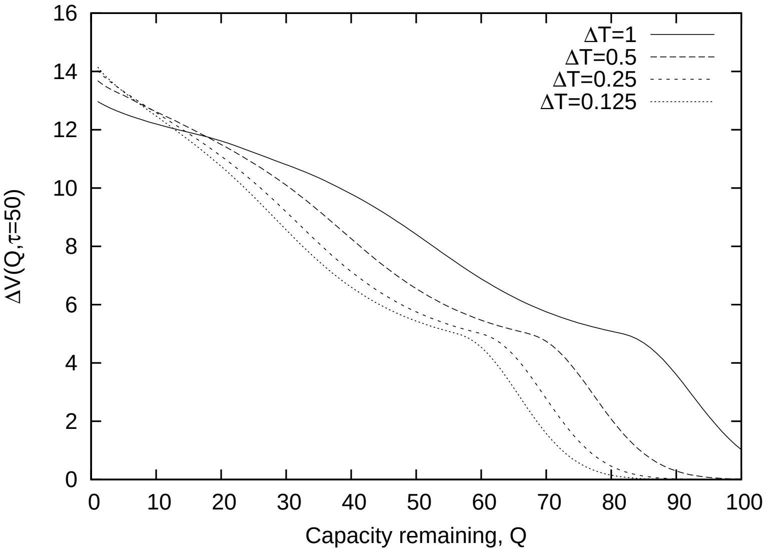



Figure 7.8: Expected marginal values from the adjusted PDE with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$.

model. In particular, we implement it in the network model and evaluate the expected revenue per day as a perpetual quantity. In other words we examine the results on a steady-state equilibrium which is achieved when the network model is let to run for sufficiently long so that any initialisation effects are vanished. Finally, the investigation is done for all choices of $\Delta T$ considered.

Our solution will be compared against the MC solution and the *adjusted* Pontryagins solution. The MC rejection policy corresponds to the MC model described in section 5.1 with $\Delta T$ to be set accordingly. As for the *adjusted* Pontryagins solution we prefer not to go through its derivation explicitly, as the Pontryagins model that we used is simply the solution to the adjusted PDE in (7.11) with $Q$ being a continuous quantity, i.e. we replace the difference $\tilde{V}(Q-1,\tau)-\tilde{V}(Q,\tau)$ by the first derivative $\partial\tilde{V}/\partial Q$ and solve it using the Pontryagins maximum principle similar to section 5.4. The relevant optimal rejection policies for the MC and the adjusted Pontryagins methods may be seen in figures 7.9 and 7.10, respectively. Recall that the marginal values of the MC method (figure 7.9) have also been discussed in section 6.1. In particular, we have shown that as $\Delta T$ decreases, the limitation of simulating single-period-stay customers in each period created the deep towards the left of the figure. As for the marginal values of the adjusted Pontryagins approach (figure 7.10) the characteristics are similar to those of the

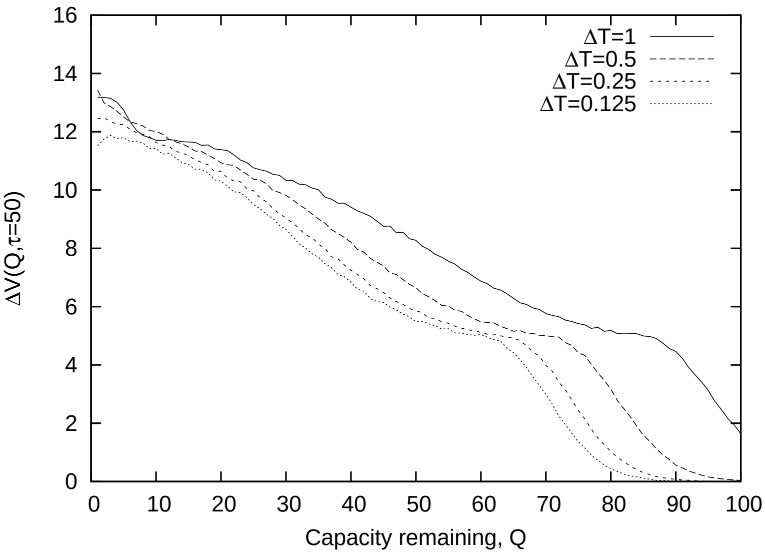



Figure 7.9: Expected marginal values from the MC with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$.

adjusted PDE (in figure 7.8). The only difference, however, is that the marginal values drops to zero once the expected demand is greater than the carpark size, emphasising on the deterministic nature of this method.

Let us denote by $V^{MC}$ the expected perpetual revenue for the network model after using the MC rejection policy with interval size $\Delta T$. Also let us define by $V^{Ad.PDE}$ and $V^{Ad.Pontr}$ the expected values for the network model that result after implementing the optimal policy of the *adjusted* PDE in (7.11) and the *adjusted* Pontryagins policy, respectively.

Furthermore, we want to compare the performance of the new adjusted PDE against the original PDE model found in equation (5.46) of section 5.3 which could only solve for infinitesimal periods. This, will show the extent at which the adjusted PDE improves the results in the case of finite time intervals. Thus, we define by $V^{PDE}$ the expected perpetual revenues for the network model that result after implementing the optimal policy of the original PDE.

Similar to section 6.4 we use the MC solution as the point of reference for the others and thus, by dividing over $V^{MC}$ gives us an indication of the methods performance against the MC method. All results have been obtained after running the network model and averaging over one thousand paths.

Figure 7.11 compares the performance of the methods for day intervals, i.e.

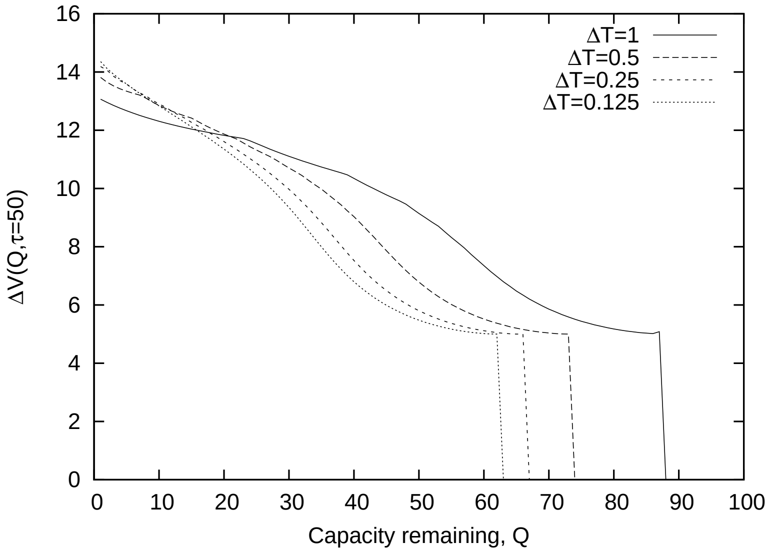



Figure 7.10: Expected marginal values from the adjusted Pontryagins with 50 days left. These are shown as functions of the capacity remaining $Q$, for varying choices of $\Delta T$.

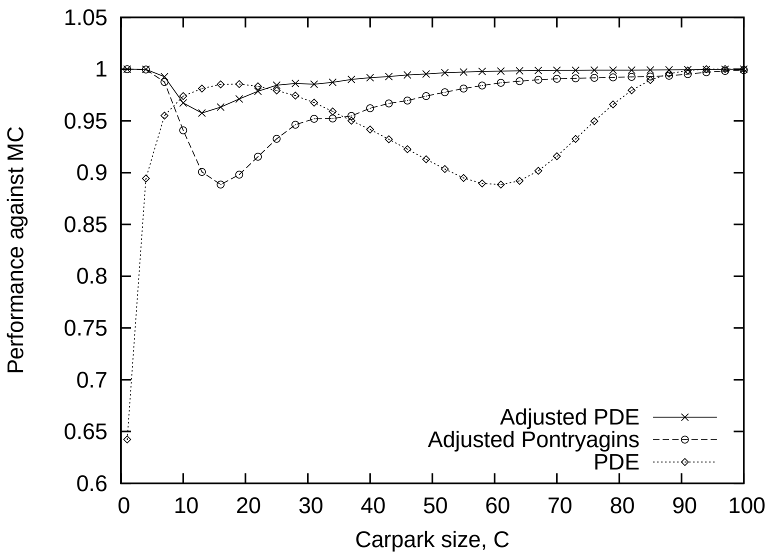



Figure 7.11: Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 1$ (against the MC).

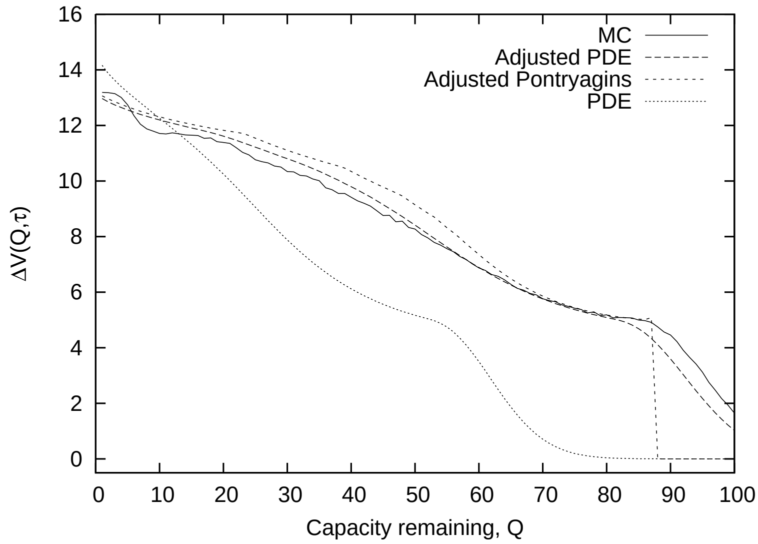



Figure 7.12: Expected marginal values for the three methods with $\Delta T = 1$, as functions of the capacity remaining $Q$ in the perpetual sense.

$\Delta T = 1$. The results are shown in terms of the expected revenue per day (perpetual sense) for varying carpark sizes $C$. The MC method has been shown to perform better overall than all the other methods (no curve crosses over 1). Again, this is mainly because of the nature of the MC solution, that it incorporates more information about the network when evaluating the rejection policy. In the discussion to follow we will be quite frequently referring to figure 7.12 in order to seek some sort of explanation to the observations. In particular, figure 7.12 plots the expected marginal values of the spaces under $\Delta T = 1$ for all methods as function of the capacity remaining. This will shed light into the manner at which the different policies in consideration value the parking spaces.

In the region with the lowest-size carparks ($C < 5$ ) the adjusted PDE and Pontryagins solutions remarkably match up with the MC. This is because the parking spaces are priced identically under all three policies - MC, adjusted PDE, adjusted Pontryagins - and this might actually be seen clearly in figure's 7.12 top left corner. The economic interpretation here is that under this valuation there is indeed such customer demand to come and fill out the spaces and thus pay the corresponding prices that maximise the revenues. On the other hand the original PDE policy does poorly here because the valuation of the spaces are set so high (7.12) that even the higher paying customers are not paying enough money to satisfy the acceptance price threshold and thus get denied service.

Still in figure 7.11 but for carparks of size $C \geq 35$ we observe that the adjusted PDE performs remarkably well (although a little worse than in the region $C < 5$) with the adjusted Pontryagins slightly lower while the original PDE is unsatisfactory once again. In particular, the performance steadily improves with the capacity. The maximum deviation from the MC value is 1.2% and 4.8% for the adjusted PDE and the adjusted Pontryagins, respectively, both incurred when $C = 35$. Once again notice that the performance difference between these two stems from the fact that we used a deterministic solution (adjusted Pontryagins) to approximate a stochastic problem.

From figure 7.12 we observe that the adjusted PDE and adjusted Pontryagins value the spaces very similar to the MC method, which is the reason why the resulting performance is very close. In contrast, the rejection policy of the original PDE constitutes of marginal values which are significantly lower than the others; this implies that spaces are priced low resulting in relatively too many low-paying customers being accepted at the expense of the most profitable business customers that would have arrived later in the horizon.

For large carparks all methods perform more or less identically because the actual role of any rejection policy in this region is suppressed by the excess in capacity over the expected total demand.

Intentionally, I have skipped a particular region to discuss it at the end. This is the region $10 < C < 25$ where the results show some rather interesting pattern. Within this region, the original PDE outperforms both the adjusted PDE and the adjusted Pontryagins. In fact, within this region its performance against the MC is never less than 2.6% (incurred at $C = 10$), as opposed to 4.2% (incurred at $C = 12$) and 10.0% (incurred at $C = 13$) for the adjusted PDE and the adjusted Pontryagins. Such a result may seem a bit surprising at a first glance as the original PDE seems to be spotting something the other "improved" schemes might have missed out on. So let us try to give some explanation to what might actually be happening in this region. We first recall that the manner at which we compute the expected revenues per day in the network model is based on the additive bid-price methodology - the booking is accepted if it generates greater revenue than the sum of the bid prices on the days it will occupy the space for - which is still a heuristic approximation to the true network solution. As a result, one may find an alternative approach that might work even better in some cases. This is what partially might happen here. The original PDE, although it completely ignores the network inter-dependence of the days, calculates a more representative set of marginal values (bid prices) when the carpark size lies in this region and therefore the additive bid-price approach picks out the most suitable booking requests for parking. In fact, this is the exact same region we have observed the PDE to perform better than the MC in the previous chapter (table 6.8).

To sum up, the adjusted PDE performed significantly better than the adjusted Pontryagins and the original PDE for most carpark sizes. In particular, the PDE achieves the same performance with the MC within only 0.8% deviation on average[3] whereas the Pontryagins achieved 3.1% deviation on average. As for the original PDE, its performance was around 5.2% lower than the MC solution on average. For the exact numerical results of all four methods the reader is referred to table 7.4.

Finally, figure 7.13 shows the average daily occupancy rates under the four methods. In general, the occupancy levels are now higher than those achieved in the infinitesimal case, seen in figure 6.37. For $C = 1$ when there is only one space to manage, under the MC method as well as under the two adjusted schemes we always have the space sold out for the day. In contrast under the original PDE policy the space is rented only about two thirds of the time which naturally lead to poor revenues. For greater capacities, however, it happens the opposite, as the original PDE lets too many people to park (occupancy rate rises up to 98%) but still misses out on extracting the maximum revenue out of the customers.

Next, we proceed to the case $\Delta T = 0.5$ and observe that the MC is again the dominant method. Relevant results may be seen in figure 7.14 with a detailed analysis in Table 7.5. The three methods perform similarly but significantly lower than the MC in the region $C < 5$. In fact, this is verified by figure 7.15 where the three methods generate similar (if not exactly the same) marginal values in this region but significantly higher than those of the MC. As a result too many customers are denied service leading to lower expected revenues. It is remarkable to notice how the adjusted methodology adjusts accordingly in $\Delta T$ to generate marginal values that lie close to the MC ones.

Overall, the PDE again performed better than the Pontryagins. In particular, the adjusted PDE produced similar results to the MC only with a maximum

[3]The average deviation of one method from the MC is computed by summing the revenue performances over all carparks sizes $C = 1, 2, \ldots, 100$ and then subtracting this number from 100.

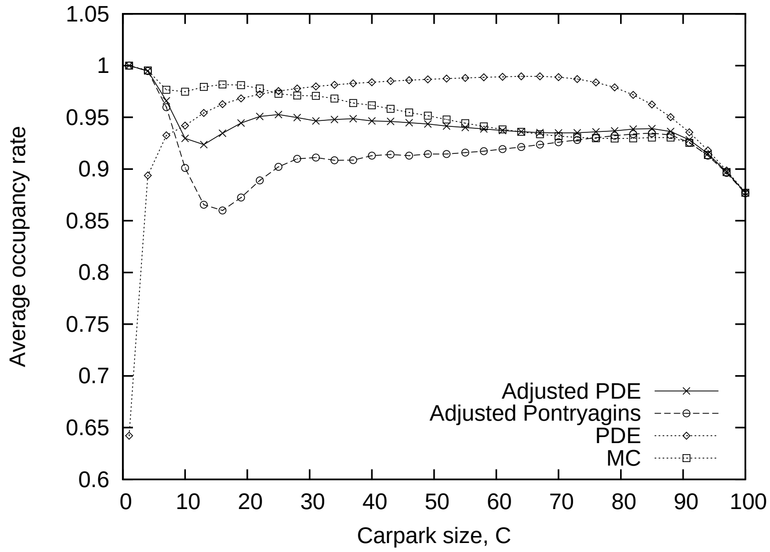



Figure 7.13: Average occupancy rate as function of the carpark size $C$, under the four rejection policies for $\Delta T = 1$.

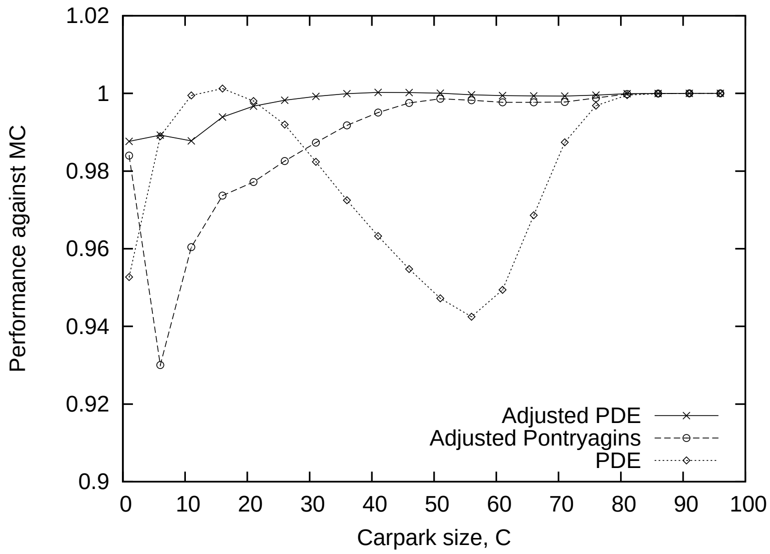



Figure 7.14: Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.5$ (against the MC).

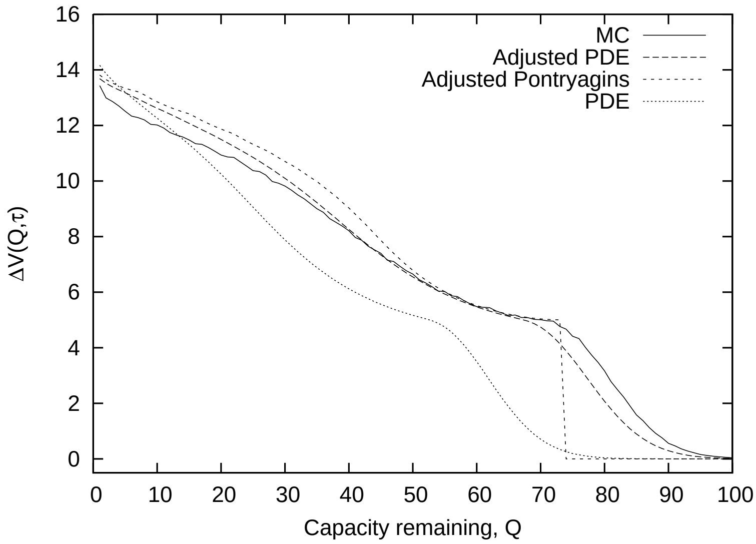



Figure 7.15: Expected marginal values for the three methods with $\Delta T = 0.5$, as functions of capacity remaining $Q$ in the perpetual sense.

deviation 4.4% (at $C = 3$), as opposed to a maximum deviation of 7.0% (at $C = 7$) for the Pontryagins. We notice that the original PDE although underperforming in most cases, it actually generates higher revenues than even the MC for carparks $C = 12-18$; similar situation was also discussed in the preceding case of $\Delta T = 1$.

Similarly, in figure 7.16 and table 7.6 we have the results for $\Delta T = 0.25$, while the results for $\Delta T = 0.125$ are presented in figure 7.17 and table 7.7. For carpark sizes $C = 20 - 25$ the adjusted PDE also achieves a higher performance than the MC when $\Delta T = 0.125$.

Another key observation is that the original PDE policy seems to improve as $\Delta T$ increases. This is the expected behaviour since the original PDE is designed to work for infinitesimal time intervals and consequently should perform better the closer we get to this infinitesimal setting. In fact, this is justified by figures 7.18 and 7.19 where the marginal values of the original PDE are now located closer to those of the adjusted PDE and the MC.

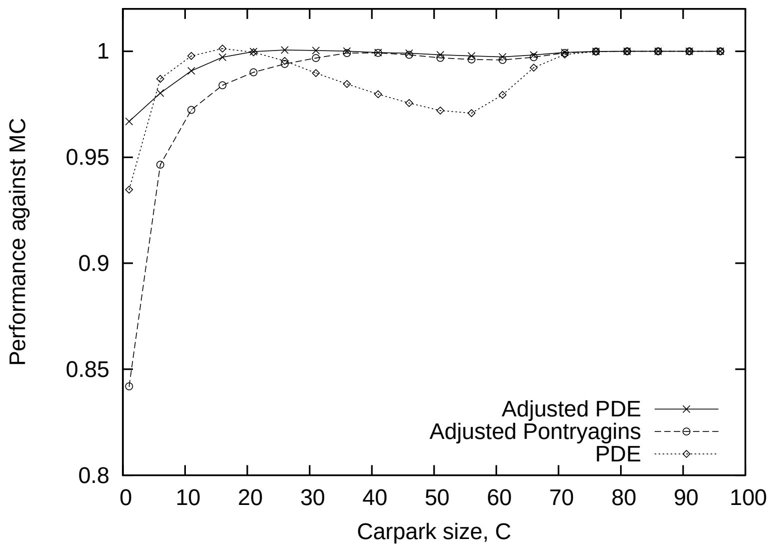



Figure 7.16: Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.25$ (against the MC).

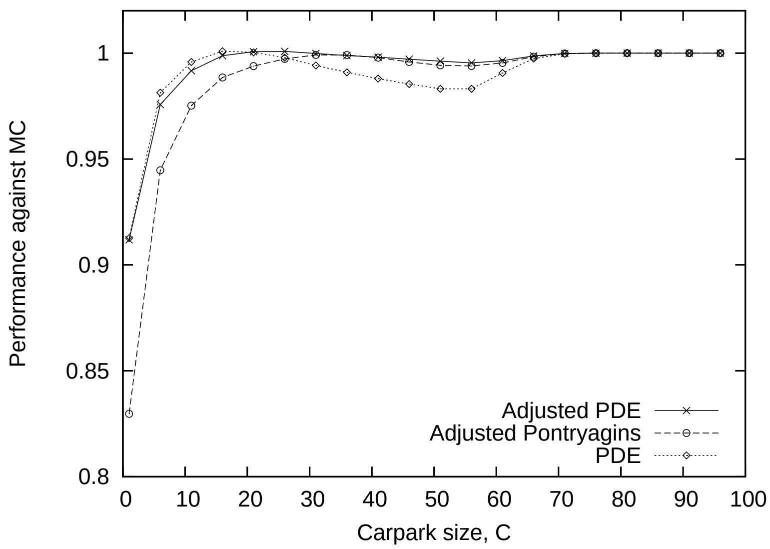



Figure 7.17: Relative comparison of the expected perpetual revenues for the methods with $\Delta T = 0.125$ (against the MC).

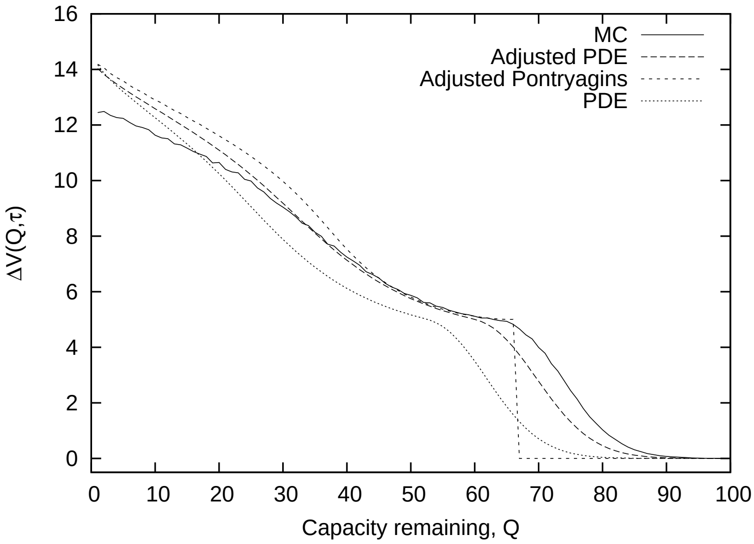



Figure 7.18: Expected marginal values for the three methods with $\Delta T = 0.25$, as functions of capacity remaining $Q$ in the perpetual sense.

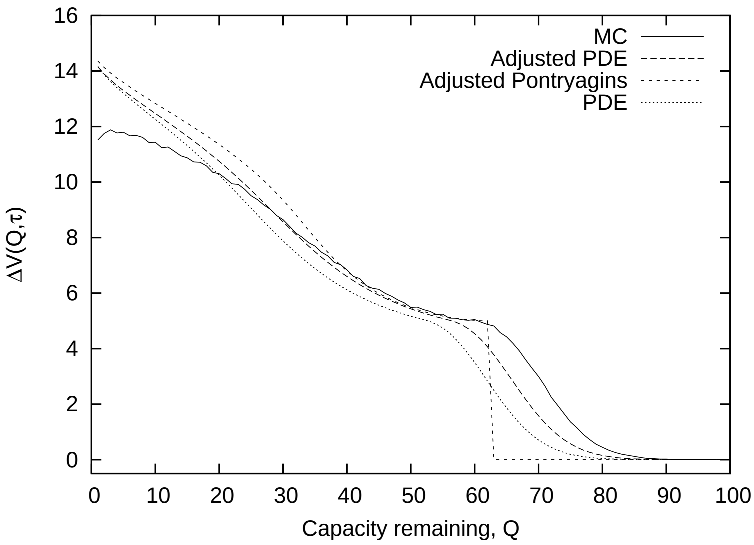



Figure 7.19: Expected marginal values for the three methods with $\Delta T = 0.125$, as functions of capacity remaining $Q$ in the perpetual sense.

## 7.4 Closing remarks

In real carparks it is up to the manager to decide which pricing strategy to follow. For instance, they can choose to sell the slots for a minimum length of a day ($\Delta T = 1$), half a day ($\Delta T = 0.5$) or an hour ($\Delta T = 1/24$). Whichever their choice is, the problem becomes one that we should maximise the revenues to be made within a time period of size $\Delta T$. Since the original PDE model could not capture this situation we have reformulated the PDE in a way that it could now solve for the expected value rate that is generated within a finite $\Delta T$-period in the future.

Numerical results have been conducted and showed that this adjusted PDE even though not incorporating any information about the inter-dependence within the days it performed undoubtedly well when tested under the network environment for any given choice of $\Delta T$. In particular, by implementing the adjusted PDE rejection policies we managed to generate expected revenues within a 4.5% to the MC solution for all carparks and all $\Delta T$ sizes.

What is more, we achieved in keeping the computation times in less than a minute. Even though the policy is defined as a dynamic set of expected marginal values which would naturally mean that frequent re-optimisation is not necessary, that we kept the computation time in relatively low levels implies that such an algorithm is flexible enough to be used in a real environment for which frequent optimisation may be conducted.

| $C$ | $V^{MC}$ | $V^{Ad.PDE}/V^{MC}$ | $V^{Ad.Pontr}/V^{MC}$ | $V^{PDE}/V^{MC}$ |
|---|---|---|---|---|
| 1 | 13.241 | 1.000 | 1.000 | 0.640 |
| 2 | 26.418 | 1.000 | 1.000 | 0.796 |
| 3 | 39.549 | 1.000 | 1.000 | 0.856 |
| 4 | 52.549 | 1.000 | 1.000 | 0.893 |
| 5 | 65.285 | 0.999 | 0.999 | 0.920 |
| 6 | 77.640 | 0.998 | 0.995 | 0.941 |
| 7 | 89.704 | 0.993 | 0.988 | 0.955 |
| 8 | 101.535 | 0.985 | 0.975 | 0.965 |
| 9 | 113.348 | 0.976 | 0.958 | 0.971 |
| 10 | 125.103 | 0.967 | 0.941 | 0.974 |
| 11 | 136.867 | 0.962 | 0.924 | 0.977 |
| 12 | 148.591 | 0.958 | 0.911 | 0.979 |
| 13 | 160.328 | 0.957 | 0.900 | 0.981 |
| 14 | 172.037 | 0.958 | 0.893 | 0.982 |
| 15 | 183.675 | 0.960 | 0.889 | 0.984 |
| 16 | 195.274 | 0.962 | 0.888 | 0.985 |
| 17 | 206.829 | 0.965 | 0.889 | 0.986 |
| 18 | 218.334 | 0.968 | 0.893 | 0.987 |
| 19 | 229.756 | 0.971 | 0.897 | 0.986 |
| 20 | 241.117 | 0.974 | 0.903 | 0.985 |
| 21 | 252.411 | 0.976 | 0.909 | 0.984 |
| 22 | 263.491 | 0.979 | 0.915 | 0.984 |
| 23 | 274.485 | 0.981 | 0.921 | 0.983 |
| 24 | 285.377 | 0.983 | 0.927 | 0.982 |
| 25 | 296.131 | 0.984 | 0.933 | 0.981 |
| 30 | 348.896 | 0.985 | 0.950 | 0.972 |
| 35 | 399.666 | 0.988 | 0.952 | 0.959 |
| 40 | 447.741 | 0.991 | 0.962 | 0.945 |
| 45 | 493.304 | 0.994 | 0.968 | 0.930 |
| 50 | 535.865 | 0.996 | 0.975 | 0.914 |
| 55 | 575.289 | 0.997 | 0.981 | 0.900 |
| 60 | 611.532 | 0.998 | 0.986 | 0.894 |
| 65 | 644.808 | 0.999 | 0.989 | 0.900 |
| 70 | 675.215 | 0.999 | 0.991 | 0.920 |
| 75 | 703.544 | 0.999 | 0.992 | 0.947 |
| 80 | 730.268 | 1.000 | 0.993 | 0.973 |
| 85 | 756.270 | 1.000 | 0.993 | 0.991 |
| 90 | 780.957 | 1.000 | 0.995 | 0.998 |
| 95 | 800.119 | 1.000 | 0.998 | 1.000 |
| 100 | 812.141 | 1.000 | 0.999 | 1.000 |

Table 7.4: Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 1$.

| $C$ | $V^{MC}$ | $V^{Ad.PDE}/V^{MC}$ | $V^{Ad.Pontr}/V^{MC}$ | $V^{PDE}/V^{MC}$ |
|---|---|---|---|---|
| 1 | 13.432 | 0.988 | 0.984 | 0.953 |
| 2 | 26.477 | 0.957 | 0.931 | 0.955 |
| 3 | 39.284 | 0.956 | 0.896 | 0.954 |
| 4 | 51.944 | 0.969 | 0.894 | 0.967 |
| 5 | 64.453 | 0.982 | 0.911 | 0.980 |
| 6 | 76.804 | 0.989 | 0.930 | 0.989 |
| 7 | 89.076 | 0.990 | 0.947 | 0.993 |
| 8 | 101.279 | 0.988 | 0.958 | 0.995 |
| 9 | 113.331 | 0.987 | 0.963 | 0.997 |
| 10 | 125.345 | 0.986 | 0.962 | 0.998 |
| 11 | 137.223 | 0.988 | 0.960 | 0.999 |
| 12 | 148.956 | 0.991 | 0.960 | 1.000 |
| 13 | 160.625 | 0.993 | 0.962 | 1.001 |
| 14 | 172.173 | 0.994 | 0.966 | 1.002 |
| 15 | 183.705 | 0.994 | 0.970 | 1.001 |
| 16 | 195.098 | 0.994 | 0.974 | 1.001 |
| 17 | 206.437 | 0.994 | 0.975 | 1.001 |
| 18 | 217.597 | 0.995 | 0.975 | 1.000 |
| 19 | 228.737 | 0.996 | 0.975 | 0.999 |
| 20 | 239.699 | 0.996 | 0.976 | 0.999 |
| 21 | 250.564 | 0.997 | 0.977 | 0.998 |
| 22 | 261.401 | 0.997 | 0.979 | 0.997 |
| 23 | 272.033 | 0.997 | 0.981 | 0.996 |
| 24 | 282.526 | 0.998 | 0.982 | 0.995 |
| 25 | 292.974 | 0.998 | 0.982 | 0.993 |
| 30 | 343.243 | 0.999 | 0.986 | 0.984 |
| 35 | 389.992 | 1.000 | 0.991 | 0.974 |
| 40 | 432.777 | 1.000 | 0.994 | 0.965 |
| 45 | 471.402 | 1.000 | 0.997 | 0.956 |
| 50 | 506.252 | 1.000 | 0.998 | 0.949 |
| 55 | 537.819 | 1.000 | 0.998 | 0.943 |
| 60 | 566.725 | 1.000 | 0.998 | 0.947 |
| 65 | 593.689 | 0.999 | 0.998 | 0.964 |
| 70 | 619.384 | 0.999 | 0.998 | 0.984 |
| 75 | 643.919 | 0.999 | 0.999 | 0.996 |
| 80 | 663.620 | 1.000 | 1.000 | 0.999 |
| 85 | 675.727 | 1.000 | 1.000 | 1.000 |
| 90 | 681.249 | 1.000 | 1.000 | 1.000 |
| 95 | 683.238 | 1.000 | 1.000 | 1.000 |
| 100 | 683.771 | 1.000 | 1.000 | 1.000 |

Table 7.5: Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.5$.

| $C$ | $V^{MC}$ | $V^{Ad.PDE}/V^{MC}$ | $V^{Ad.Pontr}/V^{MC}$ | $V^{PDE}/V^{MC}$ |
|---|---|---|---|---|
| 1 | 12.448 | 0.967 | 0.842 | 0.935 |
| 2 | 24.919 | 0.960 | 0.914 | 0.963 |
| 3 | 37.292 | 0.968 | 0.921 | 0.972 |
| 4 | 49.614 | 0.973 | 0.927 | 0.977 |
| 5 | 61.826 | 0.974 | 0.942 | 0.983 |
| 6 | 73.911 | 0.980 | 0.946 | 0.987 |
| 7 | 85.874 | 0.982 | 0.952 | 0.990 |
| 8 | 97.770 | 0.985 | 0.960 | 0.993 |
| 9 | 109.561 | 0.987 | 0.964 | 0.994 |
| 10 | 121.200 | 0.989 | 0.968 | 0.996 |
| 11 | 132.758 | 0.991 | 0.972 | 0.998 |
| 12 | 144.201 | 0.993 | 0.975 | 0.999 |
| 13 | 155.542 | 0.994 | 0.978 | 1.000 |
| 14 | 166.799 | 0.995 | 0.981 | 1.001 |
| 15 | 177.963 | 0.996 | 0.982 | 1.001 |
| 16 | 188.989 | 0.997 | 0.984 | 1.001 |
| 17 | 199.926 | 0.998 | 0.986 | 1.001 |
| 18 | 210.730 | 0.999 | 0.987 | 1.001 |
| 19 | 221.413 | 0.999 | 0.988 | 1.001 |
| 20 | 232.045 | 0.999 | 0.989 | 1.000 |
| 21 | 242.503 | 1.000 | 0.990 | 1.000 |
| 22 | 252.849 | 1.000 | 0.991 | 0.999 |
| 23 | 263.019 | 1.000 | 0.992 | 0.998 |
| 24 | 273.049 | 1.000 | 0.993 | 0.997 |
| 25 | 282.933 | 1.000 | 0.993 | 0.996 |
| 30 | 329.916 | 1.000 | 0.996 | 0.991 |
| 35 | 372.454 | 1.000 | 0.999 | 0.985 |
| 40 | 410.655 | 1.000 | 0.999 | 0.981 |
| 45 | 444.680 | 0.999 | 0.999 | 0.976 |
| 50 | 475.444 | 0.998 | 0.997 | 0.972 |
| 55 | 503.594 | 0.998 | 0.996 | 0.970 |
| 60 | 529.892 | 0.997 | 0.996 | 0.977 |
| 65 | 555.138 | 0.998 | 0.997 | 0.990 |
| 70 | 578.084 | 0.999 | 0.999 | 0.998 |
| 75 | 594.698 | 1.000 | 1.000 | 1.000 |
| 80 | 603.426 | 1.000 | 1.000 | 1.000 |
| 85 | 606.878 | 1.000 | 1.000 | 1.000 |
| 90 | 607.865 | 1.000 | 1.000 | 1.000 |
| 95 | 608.065 | 1.000 | 1.000 | 1.000 |
| 100 | 608.107 | 1.000 | 1.000 | 1.000 |

Table 7.6: Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.25$.

| $C$ | $V^{MC}$ | $V^{Ad.PDE}/V^{MC}$ | $V^{Ad.Pontr}/V^{MC}$ | $V^{PDE}/V^{MC}$ |
|---|---|---|---|---|
| 1 | 11.541 | 0.912 | 0.830 | 0.913 |
| 2 | 23.290 | 0.941 | 0.881 | 0.945 |
| 3 | 35.142 | 0.954 | 0.905 | 0.960 |
| 4 | 46.949 | 0.963 | 0.923 | 0.969 |
| 5 | 58.743 | 0.970 | 0.935 | 0.976 |
| 6 | 70.467 | 0.976 | 0.945 | 0.981 |
| 7 | 82.128 | 0.979 | 0.953 | 0.985 |
| 8 | 93.682 | 0.983 | 0.960 | 0.988 |
| 9 | 105.142 | 0.987 | 0.966 | 0.991 |
| 10 | 116.536 | 0.989 | 0.970 | 0.994 |
| 11 | 127.784 | 0.992 | 0.975 | 0.996 |
| 12 | 138.933 | 0.994 | 0.979 | 0.998 |
| 13 | 149.990 | 0.996 | 0.982 | 0.999 |
| 14 | 160.982 | 0.997 | 0.985 | 1.000 |
| 15 | 171.859 | 0.998 | 0.987 | 1.001 |
| 16 | 182.613 | 0.999 | 0.988 | 1.001 |
| 17 | 193.276 | 0.999 | 0.990 | 1.001 |
| 18 | 203.791 | 1.000 | 0.991 | 1.001 |
| 19 | 214.173 | 1.000 | 0.993 | 1.001 |
| 20 | 224.489 | 1.000 | 0.993 | 1.001 |
| 21 | 234.626 | 1.001 | 0.994 | 1.000 |
| 22 | 244.624 | 1.001 | 0.995 | 1.000 |
| 23 | 254.459 | 1.001 | 0.996 | 0.999 |
| 24 | 264.081 | 1.001 | 0.996 | 0.999 |
| 25 | 273.600 | 1.001 | 0.997 | 0.998 |
| 30 | 318.485 | 1.000 | 0.999 | 0.995 |
| 35 | 358.679 | 0.999 | 0.999 | 0.992 |
| 40 | 394.420 | 0.998 | 0.998 | 0.989 |
| 45 | 426.350 | 0.997 | 0.996 | 0.986 |
| 50 | 455.479 | 0.996 | 0.994 | 0.983 |
| 55 | 482.307 | 0.996 | 0.994 | 0.983 |
| 60 | 507.720 | 0.996 | 0.995 | 0.989 |
| 65 | 531.736 | 0.998 | 0.998 | 0.996 |
| 70 | 550.744 | 1.000 | 1.000 | 0.999 |
| 75 | 561.832 | 1.000 | 1.000 | 1.000 |
| 80 | 566.449 | 1.000 | 1.000 | 1.000 |
| 85 | 567.867 | 1.000 | 1.000 | 1.000 |
| 90 | 568.158 | 1.000 | 1.000 | 1.000 |
| 95 | 568.217 | 1.000 | 1.000 | 1.000 |
| 100 | 568.229 | 1.000 | 1.000 | 1.000 |

Table 7.7: Expected perpetual revenues for a carpark of size $C$, with $\Delta T = 0.125$.

# Chapter 8

# Extensions of the PDE model

In this chapter, we present two possible extensions of the original PDE model. Each of these new models is still part of an ongoing study. Nonetheless, we do present the underlying methodology and some results to justify the need for future study in these directions.

The first model aims to apply the solution of the original PDE in a certain way as to incorporate more information about the network structure of the problem and the inter-dependence of the consecutive days in particular. This approach is expected to generate significant improvement in revenue over the original PDE solution, especially when the percentage of multi-day bookings is high.

The second model approximates the jump process that describes the remaining spaces by a Brownian motion with drift and leads to a second order PDE. As it will be shown, this model might be appropriate when the carpark size and customer demand is large, in which case a sale of a space happens quite frequently and it is small relative to the size of the carpark.

The parameters we use for the results are the same as in all previous chapters. For convenience we have re-stated them in table 8.1.

Table 8.1: Model parameters

| Class | $\lambda_b$ | $\lambda_a$ | $\lambda_s$ |
|---|---|---|---|
| $\mathcal{B}_1$ | 5 | 1/14 | 1/7 |
| $\mathcal{B}_2$ | 25 | 1/3 | 1 |

| $\Psi(\cdot)$ | $\psi_1$ | $\psi_2$ | $\mu$ |
|---|---|---|---|
| | 5 | 10 | 1/5 |

## 8.1 Hybrid PDE model that adjusts for the network solution.

Recall that the derived PDE in section 5.3 was based on the assumption that we can solve for the value rate for each day $T$ independently from the others. However, consecutive days are in general not independent due to multi-day bookings and thus the decision also depends upon the availability of spaces on these neighboring days. For example, if the carpark is sold out for the days surrounding $T$ then only single-stay customers should be accepted, while a looser restriction would apply if the carpark is empty on those days. In the original PDE (5.46) such a condition has been completely ignored from the model.

Therefore, in this section we attempt to incorporate into the $EAMV$ algorithm more information about the network structure and the inter-dependence within the neighbouring days.

In particular, we aim to examine to what extent the value function on a day $T$ is affected by the occupancy on its neighbouring days $T-k, \ldots, T-1, T+1, \ldots, T+k$ for every $k$. In other words, we aim to derive a model that describes how the solution of the carpark on a nearby day $T'$ affects the stays on the target day $T$. This should give more insight into how customers are distributed around a given a day and could potentially improve the decision process even further.

Let us recall the occupancy probability function $g(z, T-z)$. This function computes the probability of being present at day[1] $T$, $z$ days after the booking ($t = T - z$). However, it does not tell us anything about how likely will the customer also be present on the neighbouring days $T-k, \ldots, T-1, T+1, \ldots, T+k$ for every $k$. In fact, if a customer is present at $T$ it will be much likely that he will also be present just before or just after $T$.

### 8.1.1 Model formulation

So, let us suppose that we want to optimise the revenue on the day $T$, which lies $\tau$ days away from the current time, i.e. $T = t + \tau$. Let us then denote by $T_z$ a nearby day that lies $z$ days away from the current time, i.e. $T_z = t + z$ for any $z \in [0, \infty)$ (figure 8.1).

[1] To be precise, in the continuous-time formulation $T$ represents an infinitesimal instant of time. However, in order to provide an intuitive explanation of the limitation of the model we refer to $T$ as being a day.

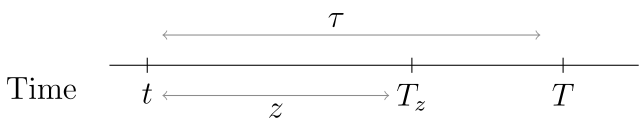



Figure 8.1: Illustration of the days

The starting point will be to find out the probability of a customer who booked at $t$, to be present on a nearby day $T_z$ *given* that he will definitely be present on day $T$. More precisely, we seek to find the probability of a customer being present $z$ days after booking at $t$ *given* that he is present $\tau$ days after booking at $t$. This is denoted by $\hat{g}(z|\tau, t)$. Obviously, there will be two cases; when $0 \leq z \leq \tau$ and when $\tau \leq z \leq \infty$. Figure 8.2 shows how the corresponding probabilities are formed under each case. The light grey shape corresponds to $g(\tau, t)$ which is kept fixed, while the medium gray shape to $g(z, t)$ with $z$ being a free variable. The darker gray shape is the intersection of these two probabilities and gives out the desired probability $\hat{g}(z|\tau, t)$.

Simple calculations can show that $\hat{g}(z|\tau, t)$ may be written as

$$\hat{g}(z|\tau, t) = \frac{1}{g(\tau, t)} \int_0^{\min(z,\tau)} \int_{\max(z,\tau)-\eta}^{\infty} \phi(\eta, \xi, t)\, d\xi\, d\eta \quad \forall\, t \leq z \leq \infty. \tag{8.1}$$

For a given fixed choice of $t$ and $\tau = T - t$ the distribution of $\hat{g}(z|\tau, t)$ as a function of $z$ is shown in figure 8.3. Note that the shape of the distribution might not necessarily be as the one used for the illustration here; however, it must have a maximum of one at the point where $z = \tau$.

Consequently, $\hat{g}(z|\tau, t)$ can potentially be used to create a weighting kernel that encapsulates some information from the nearby days. We can derive such a weighting kernel $W$ by dividing the distribution $\hat{g}(z|\tau, t)$ over its total area under the curve denoted by $Z_0(\tau, t)$. In particular, the total area under the curve reads

$$Z_0(\tau, t) = \int_0^{\infty} \hat{g}(z|\tau, t)\, dz. \tag{8.2}$$

Therefore, the weighting function $W(z|\tau, t)$ for fixed $\tau$ and booking time $t$ can then be expressed by

$$W(z|\tau, t) = \frac{\hat{g}(z|\tau, t)}{Z_0(\tau, t)}, \quad 0 \leq z < \infty \tag{8.3}$$

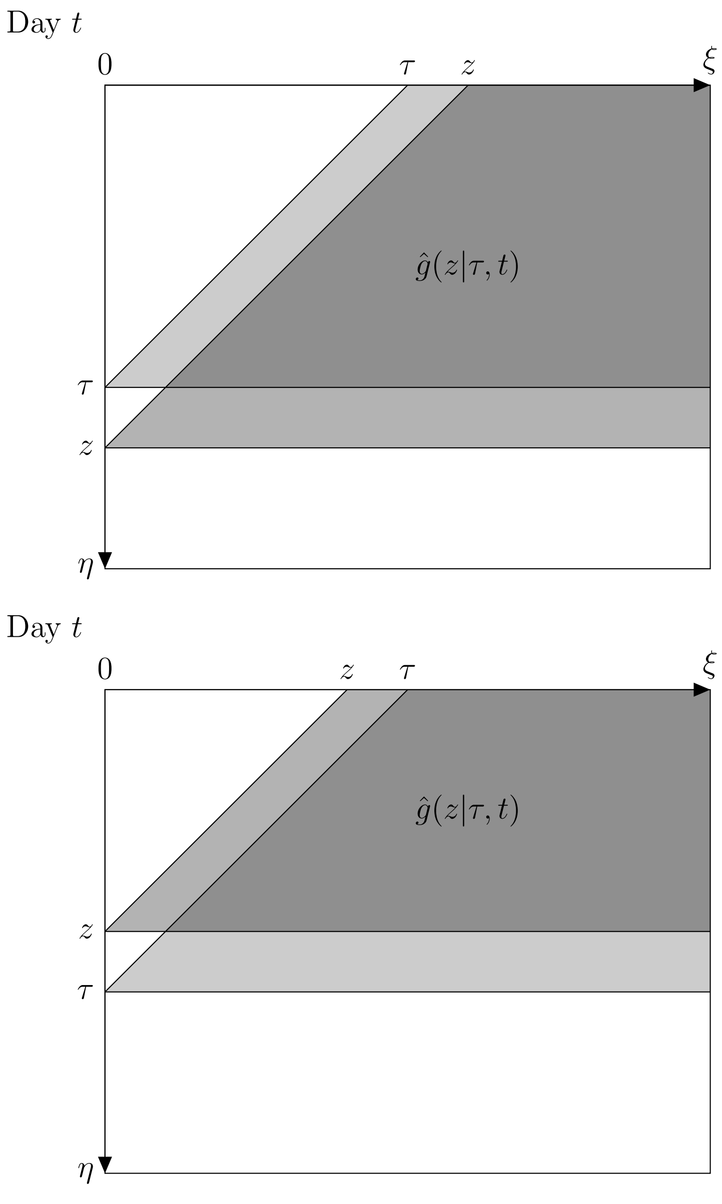



Figure 8.2: Probability $\hat{g}(z|\tau, t)$ with fixed $\tau$ and $t$, for the two cases $0 \leq z < \tau$ (bottom figure) and $\tau \leq z < \infty$ (top figure).

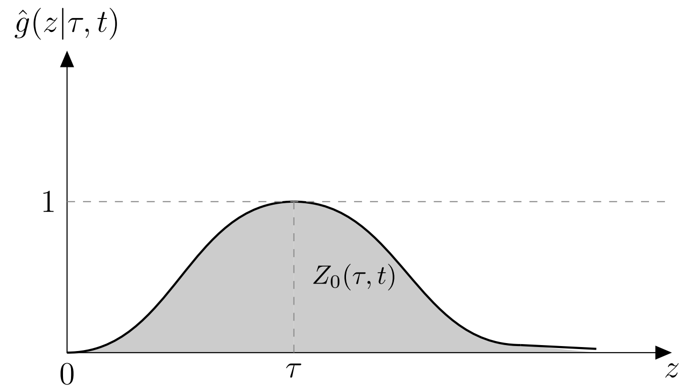



Figure 8.3: Possible shape of $\hat{g}(z|\tau,t)$ as function of $z$, for fixed $\tau$ and $t$.

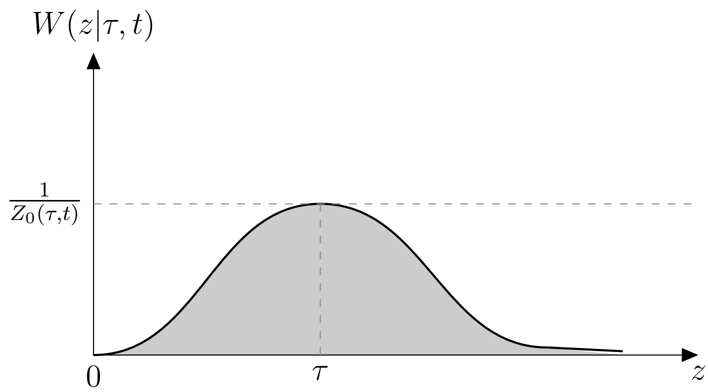



Figure 8.4: Possible shape of the weighting function $W(z|\tau,t)$ as function of $z$, for fixed $\tau$ and $t$.

In fact, $W(z|\tau,t)$ has the same shape with $\hat{g}(z|\tau,t)$ only scaled by $Z_0(\tau,t)$ (figure 8.4). This weighting kernel takes into account the multiple influence of different days on day $T$. The question is how can we actually use $W(\cdot)$ in order to improve our model?

We suggest that once the solution $V(Q,t;T)$ for all $Q,t,T$ to the *original* PDE is found, all values are refined according to a weighted average over the (infinite set of) neighbouring values, i.e. for a given combination $t,Q,T$ we update $V(t,Q;T)$ by

$$\bar{V}(t,Q;T) = \int_0^\infty W(z|\tau,t)\, V(t,Q(t;T_z);T_z)\, dz, \tag{8.4}$$

where $Q(t;T_z)$ is defined as the number of spaces remaining on a nearby day $T_z$ (which may be different than $Q$).

Such a "correction" attempts to transform the value function from a pure single-resource solution to one that incorporates more information about the entire network. Note that the value functions that will be integrated on the RHS of equation (8.4) will have already been computed when solving the original PDE model for different $T$'s.

Recall that inside the network model our decision is based on the additive bid-prices approach i.e. accept the booking only if the total price we receive for a stay is greater than or equal to the sum of the bid prices across the days the customer is staying for. This heuristic approach should improve with the quality of the bid prices used, which are defined by the expected marginal values $\Delta V(Q,t;T) = V(Q,t;T) - V(Q-1,t;T)$. Therefore, it might be more reasonable to apply the weighting on the marginal values themselves[2], namely

$$\Delta\bar{V}(t,Q;T) = \int_0^\infty W(z|\tau,t)\ \Delta V(t,Q(t;T_z);T_z)\,dz. \tag{8.5}$$

In this section we work under a time-stationary setting, and thus equation (8.5) simplifies to

$$\Delta\bar{V}(Q,\tau) = \int_0^\infty W(z|\tau)\ \Delta V(Q(z),\tau)\,dz, \tag{8.6}$$

where $Q(z)$ is the number of spaces remaining with $z$ time left.

Let us now attempt to average the marginal values appropriately, and in particular compute the expected marginal value $\Delta\bar{V}(\mathcal{Q},\tau)$ for a fixed parameter choice of $\mathcal{Q}$ and time left $\tau$. To do so we need the marginal values of all remaining times $z$ around the given time $\tau$ (already computed by solving the original PDE) *and* also the state of the carpark $Q(z)$ to apply on each of these neighbouring times.

However, the carpark availability $Q(z)$ with $z$ time left should *not* in general be equal to the availability $\mathcal{Q}$ with $\tau$ time left. In fact, there are infinite number of combinations of space availabilities to choose from and also an infinite number of nearby times-remaining to model. Thus, the challenging part is to determine the most representative state of the carpark at each remaining time $z$ to apply.

Under our model, in the steady-state equilibrium, the state of the carpark is expected to behave smoothly around a set of days. In other words, we do not expect to have large capacity-availability differences between neighbouring days;

[2]Otherwise we would have to apply the weighting on $V$'s according to equation (8.4) and then take the difference to compute the $\Delta V$'s.

a situation whereby we are sold out on Monday and Wednesday but empty on Tuesday is therefore not preferable.

Thus, we suggest to model the availability of the carpark on the nearby days by following the expected optimal[3] sales trajectory of the spaces. In particular, the expected optimal sales trajectory uses the optimal length of stay $\xi^*$ and is simply the solution to the IVP problem

$$\frac{\partial Q}{\partial \tau} = P_s(\xi^*|\tau)\, f(\tau), \quad Q(\tau = \infty) = C, \tag{8.7}$$

which may be solved by any numerical integration technique.

Then, if we seek to find the state of the carpark around $\tau$ days later, the idea is to choose the optimal sales trajectory that starts from some carpark size $C$ and passes through (or closer to) point $(\mathcal{Q}, \tau)$ and use that to specify the space availability before or after $\tau$.

In order to visualise this we may consider a discrete-time setting as illustrated in figure 8.5. The dark grey bar corresponds to the choice of time left $\tau$ and the number of spaces remaining that we seek to implement the weighting to, i.e. $Q(\tau) = \mathcal{Q}$. As mentioned previously, the state of the carpark in the nearby days will correspond to the optimal sales trajectory we choose. If we pick out a trajectory that passes relatively distant from the point of interest then the resulting carpark state we obtain will not correspond to the "correct" neighbourhood of $\mathcal{Q}$. Consequently, the weighting might degrade instead of improving the marginal value. This is in fact the case for the top and central figures. However, if we choose the trajectory that passes through the point of interest (bottom figure) then this is the correct expected sales path and therefore represents the states of the carpark on the nearby days that are most likely to behave accordingly.

Therefore, the optimal sales trajectory to choose is strongly related to the point of interest $(\mathcal{Q}, \tau)$. Moreover, each optimal sales trajectory is uniquely defined by the initial carpark size $C$ - the initial number of parking spaces before any sales have been made. As such, we may express mathematically that the optimal trajectory that passes through $(\mathcal{Q}, \tau)$ is the one that has initial carpark size $C_{\mathcal{Q},\tau}$.

Then, the refinement of the expected marginal value at the point $(\mathcal{Q}, \tau)$ is

[3]By optimal we mean the trajectory that maximises the reward functional in equation (5.59). The optimal trajectory is evaluated using the Pontryagins maximum principle (see section 2.3.2) and it specifies the optimal length of stay $\xi^*$ to be implemented.

performed as follows

$$\Delta\bar{V}(\mathcal{Q},\tau) = \int_0^\infty W(z|\tau)\ \Delta V(Q(z;C_{\mathcal{Q},\tau}),\tau)\,dz \tag{8.8}$$

and this will be referred to as the *weighted* PDE scheme.

### 8.1.2 Results

For the general set of parameters chosen, we plot $\hat{g}(z|\tau)$ as function of the time $z$ for fixed choices of $\tau$; this may be seen in figure 8.6. The first major observation we make is that the shapes of the distributions are actually spiky curves which are not differentiable at $z = \tau$. This indicates that the conditional probability of a customer being present on a neighbouring day drops down rapidly once we move away from the target day. In order to understand the shape of these curves, one may evaluate the derivative of $\hat{g}(z|\tau)$ with respect to $z$ to realise that the slope changes dramatically according to whether $z$ is smaller (steep positive gradient on the left side of $\tau$) or larger (steep negative gradient on the right of $\tau$) than $\tau$. We also observe a horizontal shift of the curves with a slight change in shape. First, the shift happens because the probability attains its maximum of one at the point where $z = \tau$. The change in the shape follows naturally as $\hat{g}(z|\tau)$ always runs between $z = 0$ (being present zero days later) and $z = \infty$ (being present infinitely later). In fact all the shapes have fixed endpoints at $z = 0$ and as $z \to \infty$.

Furthermore, in figure 8.7 we show the weighting function $W(z|\tau)$ that has been computed after dividing each curve by the total area. This is again shown as a function of $z$ for fixed choices of $\tau$. We observe that the main shape of the curve is retained but it is now scaled accordingly. The peaks seam to decay for larger values of $\tau$. This happens because the area covered by $\hat{g}(\cdot)$ increases with $\tau$, which is the result of accounting for the longer-staying leisure customers.

Now, let us shed some light on the rejection policy that results after applying the weighting on the original PDE's marginal values according to equation (8.8). As described, when evaluating the *weighted* marginal value at the point $(\mathcal{Q},\tau)$ we use the optimal sales trajectory that passes through this point. This sales trajectory will then specify the state of the carpark $Q$ to use at each and every time along the path, which is a continuous quantity. However, the list of marginal values we have to use at each time in the path is only defined for $Q$ being an

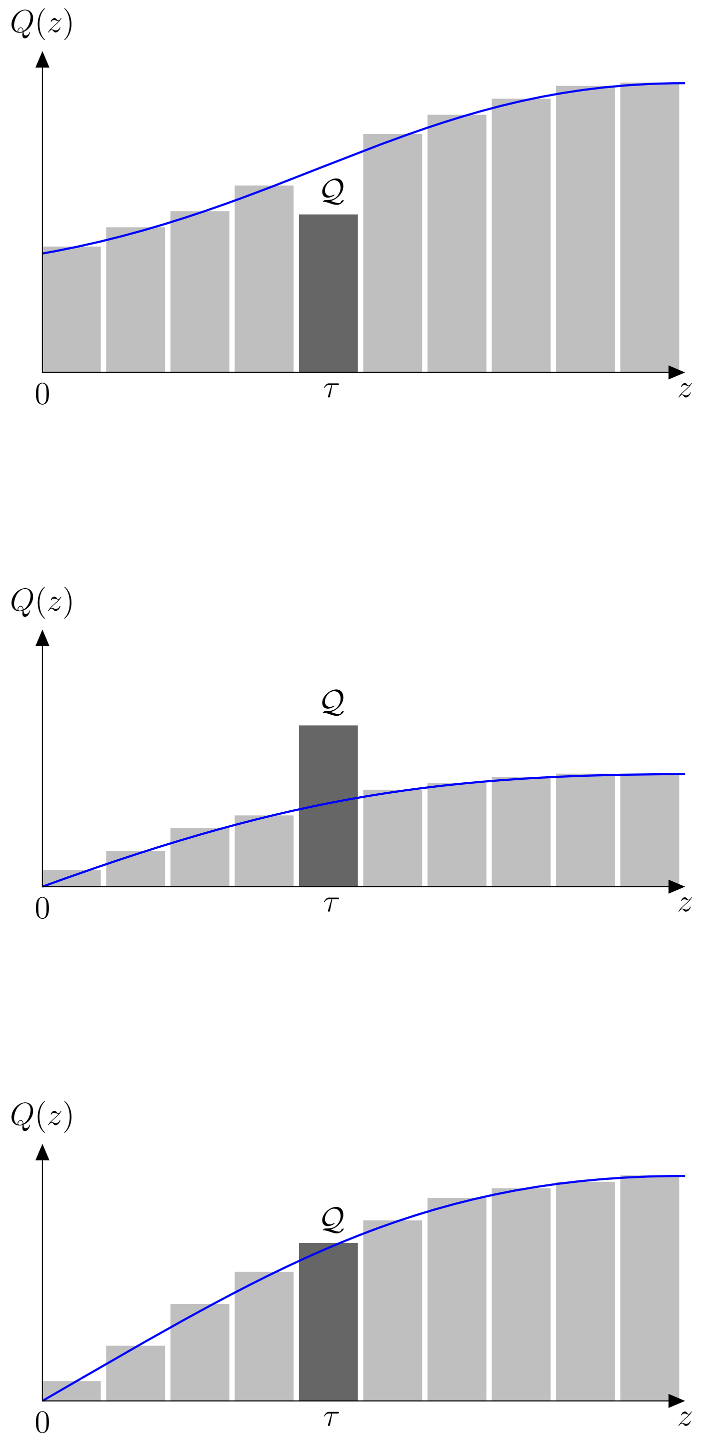



Figure 8.5: Effect of choosing an arbitrary optimal sales trajectory, on the resulting state of the carpark around time $\tau$. The top and centre figures show a sub-optimally chosen trajectory that results in relatively higher (top figure) and lower (centre figure) neighboring states. The bottom shape shows the correct optimal sales trajectory that has to implemented.

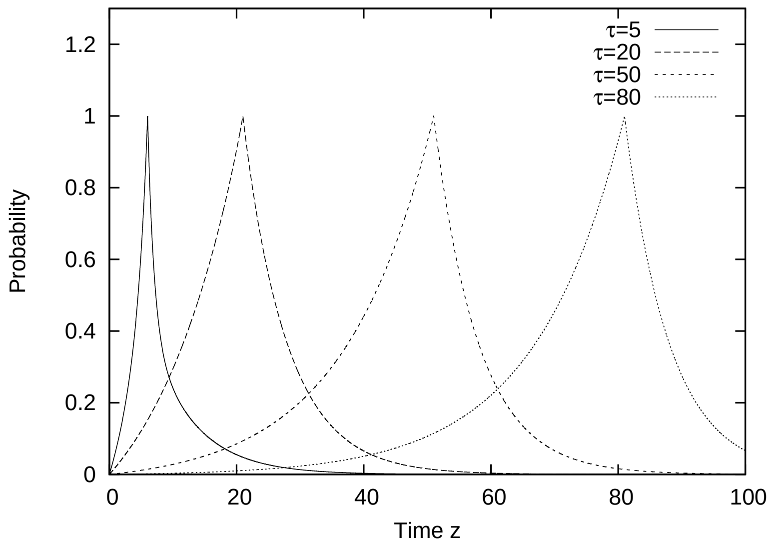



Figure 8.6: Conditional distribution $\hat{g}(z|\tau)$ as function of $z$, for different choices of $\tau$.

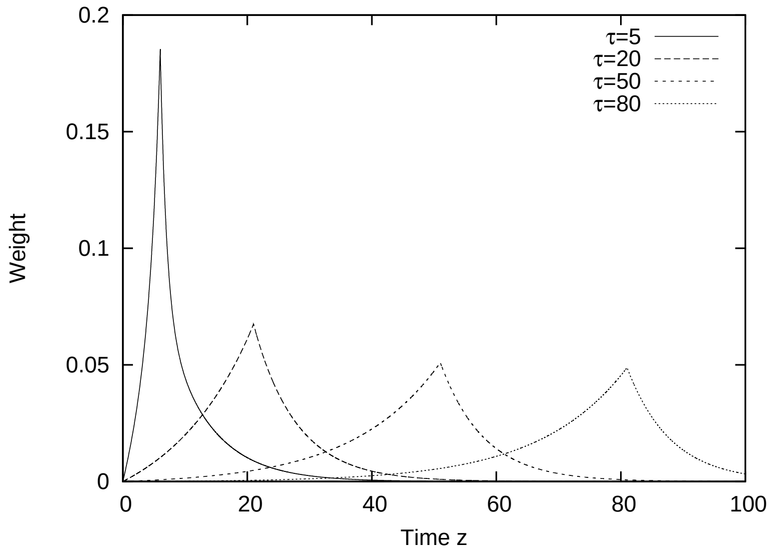


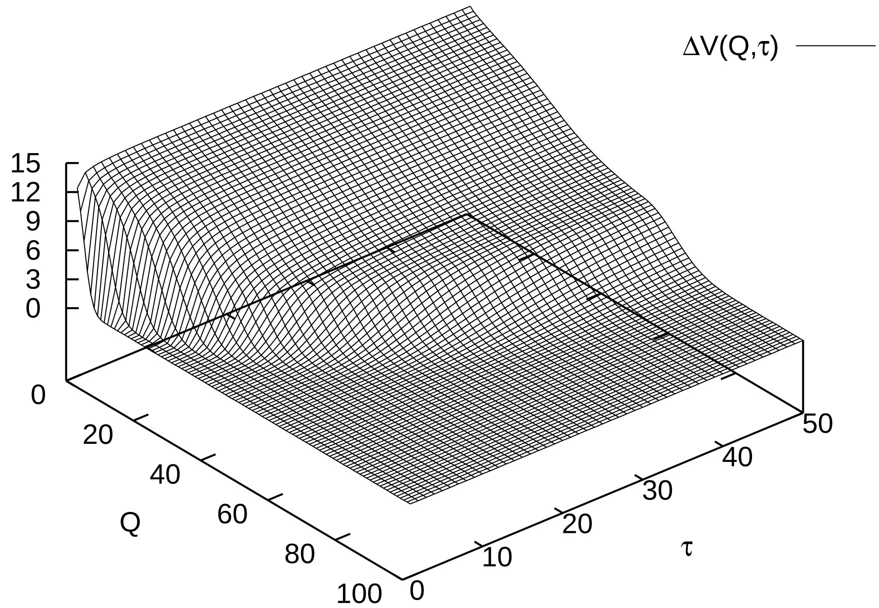

Figure 8.7: Weighting function $W(z|\tau)$ as function of $z$, for different choices of $\tau$.

integer. We believe that an interpolation scheme to work out the marginal value at a nearby non-integer state $Q$ does not really make sense under this setting, as we can only have integer spaces remaining (spaces are sold as integers not as fractions). Thus, the most reasonable approximation in this case is to use the marginal value at the nearest integer of $Q$.

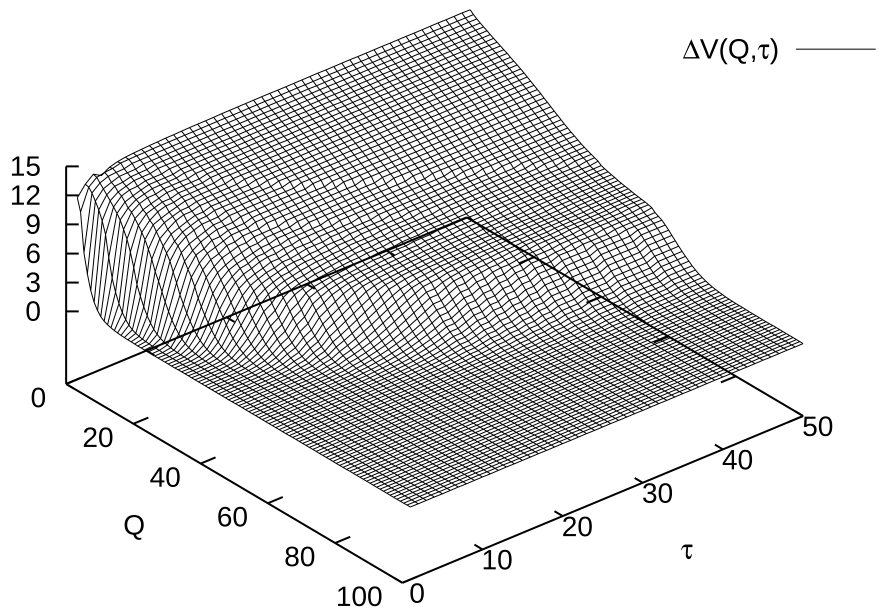
Figure 8.8 presents the weighted PDE policy (bottom figure) as a function of the capacity remaining and time left. The optimal rejection policy of the original PDE is also added (top figure). The algorithm to compute the weighted surface took around one and a half hours; however there is always room for improvement in the code efficiency. As a first look we do not spot any huge differences among the two policies; this is to be expected as the weighting function gives more emphasis (more weight) on to the point of interest, while the weights drops rapidly for the nearby times.

However there are some notable differences for small capacities as we show in figure 8.9, where we plot the expected marginal values for small capacities as functions of the time remaining. In particular, we observe that for the weighted PDE the marginal values are not continuous in time. Also for $\tau$ close to zero the weighting scheme calculates higher marginal values than the original PDE, whereas the converse happens for greater $\tau$. The former observation is the result of the numerical limitation that is created in the manner we pick out the optimal sales trajectory to follow. To explain this, consider two points of interest $(\mathcal{Q}, \tau)$ and $(\mathcal{Q}, \tau + \Delta\tau)$ that we seek to apply the weighting to. Then, the optimal trajectory to use is $C_{\mathcal{Q},\tau}$ and $C_{\mathcal{Q},\tau+\Delta\tau}$, respectively. In general, we would expect that the two optimal trajectories are the same or at least close to each other because the second point is only a "tiny" $\Delta\tau$ distance away from the first. However, if $\tau$ is relatively small ($\tau < 5$) then the resulting trajectories for $\tau$ and $\tau + \Delta\tau$ vary by a great amount (see figure 6.31 where a lot of trajectories pass through the region and thus a slight shift to the left or to right takes us to a trajectory that diverges away). Consequently, the marginal values that will be used inside the weighting scheme will be significantly different, resulting in the discontinuity of weighted values observed.

Let us now examine the performance of the weighted PDE solution inside the network model. As before we seek to evaluate the expected-revenue-per-day metric for varying carpark sizes by using the rejection policy (expected marginal values) as defined by the weighted PDE scheme; this is denoted by $V^{WPDE}$.




(Original PDE)




(Weighted PDE)

Figure 8.8: Bid-price table of the original PDE (upper figure) and of the weighted PDE scheme (lower figure), as functions of the capacity remaining $Q$ and time left $\tau$.

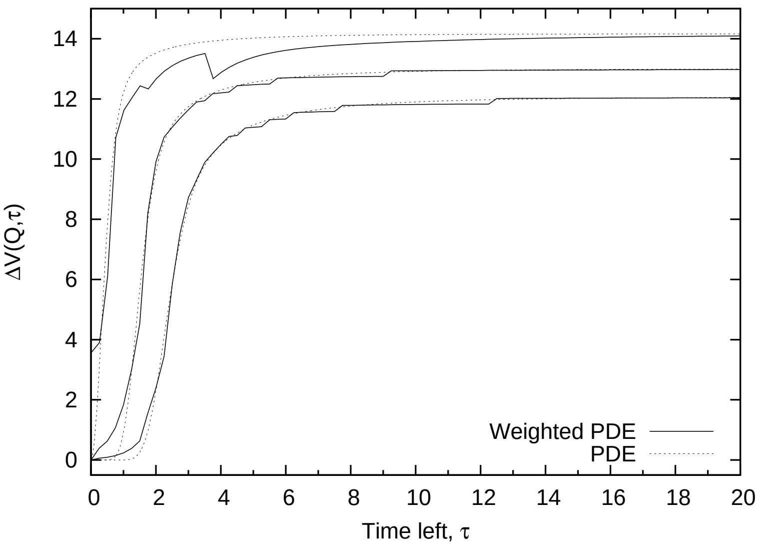



Figure 8.9: Expected marginal values of the weighted PDE for fixed $Q$, as functions of the time left. The different curves represent different choices of $Q$, 1, 5, 10, top to bottom. The marginal values of the original PDE are plotted as well.

As in the previous chapters the performance of the method is shown as the ratio over the MC expected revenue per day $V^{WPDE}/V^{MC}$. Figure 8.10 presents the relative performance of the original PDE and the weighted PDE against the MC as a function of the carpark size. Starting from the right, we can see that the weighting of the marginal values does not bring any additional improvement on the expected revenue per day to that of the original PDE. In fact the rejection policy implemented in this region is identical for the two methods. However, for small-sized carparks we observe significant improvement of the weighted PDE policy. Detailed results on the performance of the method are found in table 8.2. What is worth noting is that in the region $C = 17 - 23$ when the original PDE was shown to perform better than the MC (also recall table 6.8), the weighted PDE scheme generates identical expected revenues. This result “justifies” in some sense using the (single-resource) marginal values of the original PDE which will overprice the spaces for small carparks as it performs better for carparks $C = 17 - 23$.

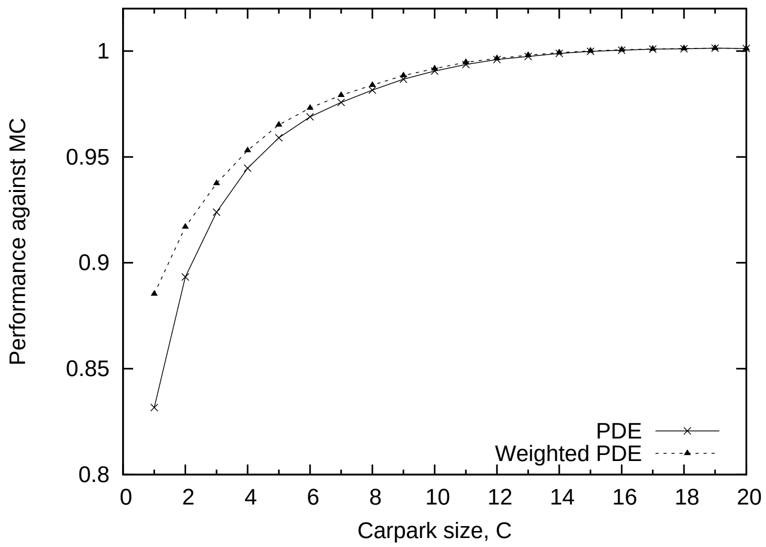



Figure 8.10: Relative comparison of the expected perpetual revenues for the original and weighted PDE against the MC, as functions of carpark size $C$.

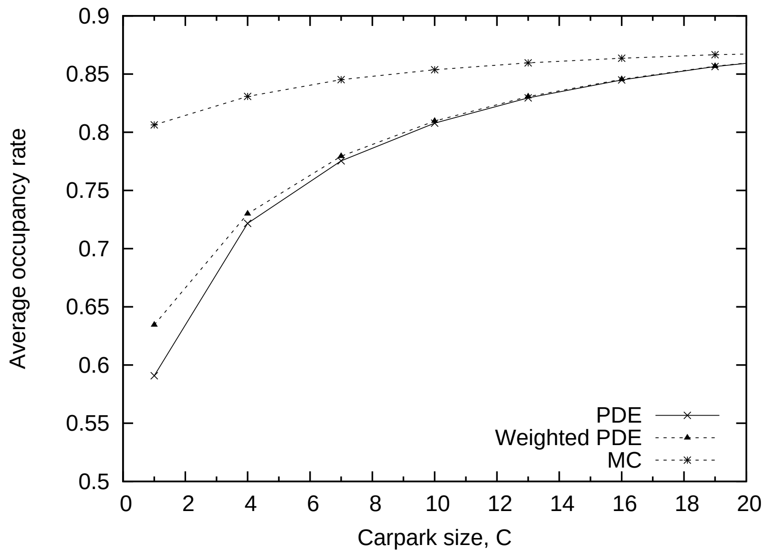



Figure 8.11: Average occupancy rate under the original PDE, the weighted PDE and the MC

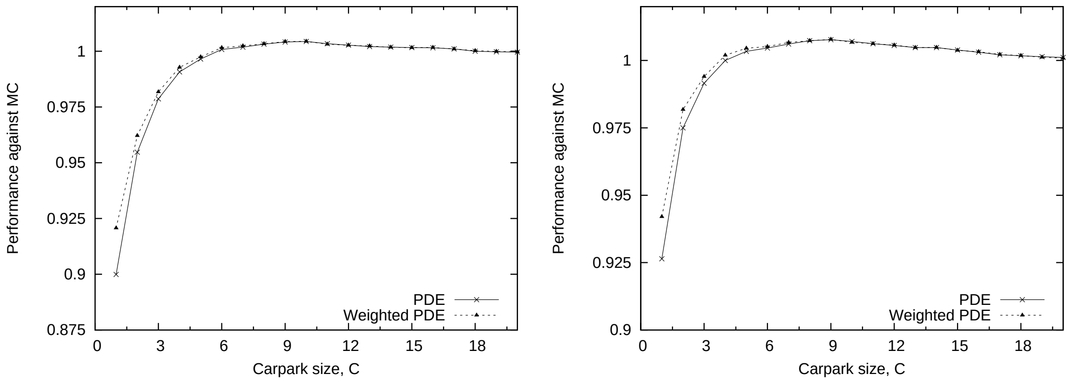
| $C$ | $V^{MC}$ | $V^{PDE}/V^{MC}$ | $V^{WPDE}/V^{MC}$ |
|---|---|---|---|
| 1 | 10.266 | 0.831 | 0.885 |
| 2 | 20.895 | 0.893 | 0.917 |
| 3 | 31.803 | 0.924 | 0.937 |
| 4 | 42.802 | 0.945 | 0.953 |
| 5 | 53.856 | 0.959 | 0.965 |
| 6 | 64.941 | 0.969 | 0.973 |
| 7 | 76.040 | 0.976 | 0.979 |
| 8 | 87.095 | 0.982 | 0.984 |
| 9 | 98.042 | 0.987 | 0.988 |
| 10 | 108.969 | 0.991 | 0.992 |
| 11 | 119.809 | 0.994 | 0.995 |
| 12 | 130.602 | 0.996 | 0.996 |
| 13 | 141.316 | 0.997 | 0.998 |
| 14 | 151.893 | 0.999 | 0.999 |
| 15 | 162.402 | 1.000 | 1.000 |
| 16 | 172.851 | 1.000 | 1.001 |
| 17 | 183.144 | 1.001 | 1.001 |
| 18 | 193.332 | 1.001 | 1.001 |
| 19 | 203.402 | 1.001 | 1.001 |
| 20 | 213.311 | 1.001 | 1.001 |
| 21 | 223.034 | 1.001 | 1.001 |
| 22 | 232.634 | 1.001 | 1.001 |
| 23 | 242.111 | 1.001 | 1.001 |
| 24 | 251.388 | 1.000 | 1.000 |
| 25 | 260.463 | 1.000 | 1.000 |
| 30 | 303.303 | 0.998 | 0.998 |
| 35 | 341.459 | 0.996 | 0.996 |
| 40 | 375.278 | 0.994 | 0.994 |
| 45 | 405.631 | 0.992 | 0.993 |
| 50 | 433.417 | 0.991 | 0.992 |
| 55 | 459.404 | 0.991 | 0.991 |
| 60 | 484.159 | 0.995 | 0.995 |
| 65 | 506.150 | 0.999 | 0.999 |
| 70 | 520.830 | 1.000 | 1.000 |
| 75 | 527.945 | 1.000 | 1.000 |
| 80 | 530.420 | 1.000 | 1.000 |
| 85 | 531.010 | 1.000 | 1.000 |
| 90 | 531.125 | 1.000 | 1.000 |
| 95 | 531.145 | 1.000 | 1.000 |
| 100 | 531.150 | 1.000 | 1.000 |

Table 8.2: Expected perpetual revenues for a carpark of size C.

Finally, in figure 8.11 we show the improvement in the average occupancy rate achieved for small capacity carparks under the weighted PDE scheme.

Although the improvement of the weighted PDE scheme is not significant if one takes into account the absolute difference in additional revenue (a 5% increase on a single-sized carpark of total worth 10.26 currency units is just 0.51), this is a promising result as it supports the reasoning of using the occupancy of the nearby days (through the weighting kernel) to compute a perhaps better estimate on the marginal value on a day. It is worth noting that these results are based on a set of parameters that assume a higher-percentage of business customers (the booking intensity $\lambda_b$ of business customers is five times more than that of the leisure customers, see table 8.1) and thus shorter-stay bookings. This indicates that the significance of the method could potentially be more pronounced when there is a higher percentage of longer-stay bookings, when the inter-dependence within the days is more apparent. Bearing in mind the rather long time needed to make this adjustment, it is worthwhile examining the solution in greater detail to identify whether such an improvement compensates for the additional computational effort.

To examine this we lower the booking intensity of the business class so that a higher proportion of leisure (or long-stay) customers is realised. In particular we run two experiments where the leisure booking intensity stays fixed at $\lambda_{b_1} = 5$, but the business booking intensity is set to (a) $\lambda_{b_2} = 5$ for an even proportion between the two classes and (b) $\lambda_{b_2} = 2$ so that long-stay customers dominate. The results on the performance are shown in figure 8.12. We notice that the improvement of the weighted PDE scheme is still not significant, even though the number of longer-stay bookings has increased considerably. We believe that the reason this happens is due to the manner at which the weighted marginal values are applied at the problem. Recall that the weighted marginal value at an arbitrary point $(\mathcal{Q}, \tau)$ have been computed *in advance* by using the optimal sales trajectory around that passes through that point (figure 8.5) and was then used inside the network model for examination. However, in reality and actually in our simulations, the states of the carpark around that point will be much different than the expected trajectory, as these are filled arbitrarily from accepting other customers. Thus, when tested inside the network model, once a booking arrives the weighting should rather be applied *in real time* according to the current state of the carpark at that particular point. In this way, the state of the carpark should be more representative of the real life scenario and the improvement might be significant.




$(\lambda_{b_2} = 5)$ $(\lambda_{b_2} = 2)$

Figure 8.12: Relative comparison of the expected perpetual revenues for the original and weighted PDE against the MC, as functions of carpark size $C$. The booking intensity for the leisure class is fixed to $\lambda_{b_1} = 5$. The booking intensity for the business class is lower to $\lambda_{b_2} = 5$ and $\lambda_{b_2} = 2$ in the left and right figures, respectively.

## 8.2 Second-order PDE

Let us recall the original PDE model with no optimisation

$$\frac{\partial V}{\partial t} + f(t;T)\Big(V(Q-1,t;T) - V(Q,t;T)\Big) = -f(t;T)\int_0^\infty \rho_s(\xi|T-t,t)\,\Psi(\xi)\,d\xi \tag{8.9}$$

with

$$V(Q,T;T) = 0 \quad \forall Q \tag{8.10}$$
$$V(0,t;T) = 0 \quad \forall t. \tag{8.11}$$

In this model we have described the available parking spaces $Q$ (state variable) by a jump process. In particular, we assumed that if a sale occurs over the next time interval $dt$, this happens with probability $f(t;T)dt$ and results in a jump of $\Delta Q = -1$ (a sale of a parking space). Thus, the *number of sales* over an interval of size $dt$ follows a Poisson distribution with mean and variance $f(t;T)\,dt$.

However, if we assume that we can sell a *fraction* of parking lot each time, i.e.

effectively reducing the jump size of the process ($\Delta Q \to 0$), while we increase the intensity of the sales ($f \to f/\Delta Q$) then the number of sales over an interval of size $dt$ might be approximated by a normal distribution with mean and variance $f(t;T)\,dt$. In particular, since $Q$ is regarded as a continuous quantity we may approximate it by the Brownian motion with drift,

$$dQ = -f(t;T)dt + \sqrt{f(t;T)}dW, \tag{8.12}$$

where $dW$ is a Wiener process and the minus sign on the drift term captures the fact that as time progresses the number of available parking spaces reduces.

This type of model might be appropriate when the carpark is sufficiently large ($C >> 1$) and sales intensity is high enough so that the expected demand is close to capacity; that is when the sale of one parking space does not create a significant change in the remaining capacity and sales are made frequently.

### 8.2.1 Brownian motion with drift as an approximation to the jump process

Support to the above model may be found through the relationship between the *Compound Poisson* process and the Brownian motion (with drift). Below we go through the main idea. First, notice that the remaining capacity of the carpark (state process) on day $T$ as of time $t$, $Q(t;T)$ may be written as

$$Q(t;T) = C - \sum_{i=1}^{N(t;T)} 1, \tag{8.13}$$

where $C = Q(0;T)$ is the initial carpark size minus the number of sales at $T$ that have occured by time $t$, $N(t;T)$. More precisely, $N(t;T)$ is a Poisson process with intensity $f(t;T)$.

If however we let the magnitude of the jump to be a random variable $H$, we result in a Compound Poisson process and thus $Q(t;T)$ may be written as

$$Q(t;T) = C - \sum_{i=1}^{N(t;T)} H. \tag{8.14}$$

In other words, every time a sale occurs the number of spaces change by the random amount $H$ (as opposed to a unit change).

These types of processes are common in Risk theory, especially when modelling the ruin probability of insurance companies, found in Asmussen and Rosiński (2001) and Mircea et al. (2010). In particular, when the intensity $f$ is stationary they often approximate it by a Brownian motion with drift. Given that $\mathrm{E}[H] = \mu$ and $Var[H] = \sigma^2$, we consider a sequence of the state process

$$Q_k(t;T) = C_k - \sum_{i=1}^{N(kt;T)} H^{(k)}$$

and assume that $C_k/\sqrt{k} \to C$, $\sqrt{k}\, H^{(k)} \to H$, $\sqrt{k}\, \mu^{(k)} \to \mu$ and $\sigma^{(k)} \to \sigma$. Then, as $n \to \infty$ one can show that

$$\frac{1}{\sqrt{k}} Q_k(t;T) \xrightarrow{\mathrm{D}} C - \mu f\, t + \sigma \sqrt{f}\, W(t),$$

with the approximation to improve as the magnitude of the jumps becomes small and by the same speed the expected number of jumps becomes large (i.e. take $H \to 0$ and set the intensity of the Compound Poisson process to $f/H$.)

Many textbooks have studied the relationship between the random walk and the Brownian motion (e.g. Neftçi, 2000). In particular, they show that one can approximate a Brownian motion by a *scaled symmetric random walk*, that is speeding up the time and taking down the jump size of a symmetric random walk. Likewise, one may think of a Compound Poisson process with high intensity and small jumps to behave like a scaled random walk because the variance of the jump times will be very small when the intensity is high.

For further details on approximating the jump process by a Brownian motion the reader is referred to Asmussen and Albrecher (2010) and Rydberg (1997). In the context of RM, the use of a Brownian motion to model the sale process has also been used in Raman and Chatterjee (1995), who further allowed the volatility to depend on the cumulative sales.

Therefore, according to (2.42), we can approximate (8.9) by

$$\frac{\partial V}{\partial t} - f(t;T)\frac{\partial V}{\partial Q} + \frac{1}{2} f(t;T)\frac{\partial^2 V}{\partial Q^2} = -f(t;T)\int_0^\infty \rho_s(\xi|T-t,t)\,\Psi(\xi)\,d\xi. \quad (8.15)$$

with

$$V(Q,T;T) = 0 \quad \forall Q \tag{8.16}$$

$$V(0,t;T) = 0 \quad \forall t. \tag{8.17}$$

Notice the diffusion term that is added in has effectively transformed the equation into a second order PDE. Consequently, to obtain a unique solution we require an additional boundary condition. This condition can be derived directly from the following logical argument. When the number of remaining spaces $Q$ is sufficiently[4] large then a small change in this quantity is not going to affect the value of the carpark. Mathematically, this reads

$$\frac{\partial V(\infty,t;T)}{\partial Q} = 0 \quad \forall t. \tag{8.18}$$

### 8.2.2 Optimal rejection condition

In the time-stationary setting we could potentially approximate the capacity remaining by

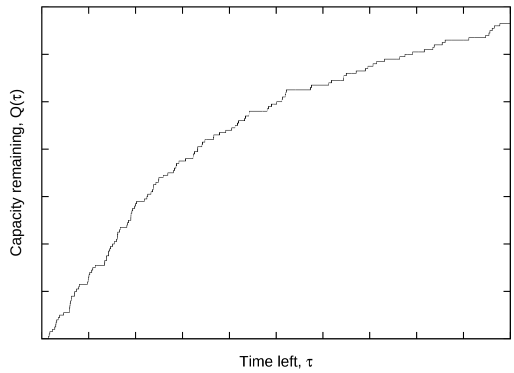
$$dQ(\tau) = f(\tau)\,d\tau + \sqrt{f(\tau)}\,dW. \tag{8.19}$$

where $\tau = T - t$. Figure 8.13 illustrates a realisation of the sales process as a function of the time left ($\tau = T - t$) when modelled by a Poisson process with intensity $f$ (left figure) and by a Brownian motion with drift $f$ and variance $\sqrt{f}$ (right figure). We observe that under these conditions the two processes look quite similar.

As in section 5.3, we may write the corresponding stationary model for $V(Q,\tau)$ and then implement the optimal rejection condition to obtain the HJB-type equation

$$\frac{\partial V}{\partial \tau} + f(\tau)P_s(\xi^*|\tau)\frac{\partial V}{\partial Q} - \frac{1}{2}f(\tau)P_s(\xi^*|\tau)\frac{\partial^2 V}{\partial Q^2} = f(\tau)\int_0^{\xi^*} \rho_s(\xi|\tau)\,\Psi(\xi)\,d\xi, \tag{8.20}$$

[4]The number of spaces should be relatively large compared to the expected customer demand.



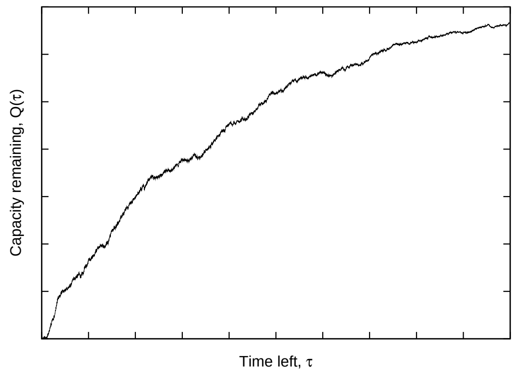



Figure 8.13: A Poisson process path with intensity $f(\tau)$ (left figure) and a Brownian motion path with drift term $f(\tau)$ and volatility $\sqrt{f(\tau)}$ (right figure). The intensity $f$ is computed based on two classes, 1 and 2, with $\lambda_{b_1} = 15$, $\lambda_{a_1} = 1/14$, $\lambda_{s_1} = 1/7$, $\lambda_{b_2} = 25$, $\lambda_{a_2} = 1/3$ and $\lambda_{s_2} = 1$.

with

$$V(Q, 0) = 0 \quad \forall Q \tag{8.21}$$

$$V(0, \tau) = 0 \quad \forall \tau \tag{8.22}$$

$$\frac{\partial V(\infty, \tau)}{\partial Q} = 0 \quad \forall \tau, \tag{8.23}$$

where $\xi^*$ is the optimal maximum duration of stay allowable. For notation purposes we will refer to the PDE model in (8.20) as the *second order PDE scheme.*

Differentiating (8.20) with respect to $\xi^*$ we obtain that the optimal price should satisfy

$$\Psi(\xi^*) = \frac{\partial V}{\partial Q} - \frac{1}{2}\frac{\partial^2 V}{\partial Q^2}. \tag{8.24}$$

Notice the extra term (diffusion term) that arises in the optimal condition. However, we have already shown (section 5.3) that for $Q$ being a Poisson process, the optimal price satisfies $\Psi(\xi^*) = V(Q, \tau) - V(Q - 1, \tau)$. How does then equation (8.24) relate to the marginal value? If we consider that $Q$ is continuous then applying a Taylor expansion to the function $V(Q+\Delta Q, \tau)$ around the point $Q+\Delta Q$ we obtain

$$V(Q + \Delta Q, \tau) = V(Q, \tau) + \Delta Q\frac{\partial V}{\partial Q} + \frac{1}{2}\Delta Q^2\frac{\partial^2 V}{\partial Q^2} + O(\Delta Q^3).$$

In the expansion $\Delta Q$ can be any number, so if we set it to -1 to match the case of the Poisson process, we have

$$V(Q-1,\tau) = V(Q,\tau) - \frac{\partial V}{\partial Q} + \frac{1}{2}\frac{\partial^2 V}{\partial Q^2} - O(\Delta Q^3),$$

which after rearranging and ignoring higher order terms gives

$$V(Q,\tau) - V(Q-1,\tau) \approx \frac{\partial V}{\partial Q} - \frac{1}{2}\frac{\partial^2 V}{\partial Q^2}. \tag{8.25}$$

Equation (8.25) indicates that (8.24) is a first order approximation to the marginal value of the space and thus is expected to behave similarly.

### 8.2.3 Numerical scheme

The second order PDE in (8.20) can be solved using an *implicit* finite difference scheme which is still first order accurate in time but a second order accurate in the state variable (see Smith, 1985, for details). Note that, a more accurate *Crank-Nicolson* scheme should theoretically increase (double) the convergence rate in time; however, the non-linearity of the PDE (the presence of the optimal control) as well as the presence of the convention term makes it difficult to obtain a stable scheme. We believe that using a change of variables to eliminate the convection term could potentially improve the stability of the Crank-Nicolson scheme (see Sachs and Strauss, 2008, for details), however for the current study we choose not to address this issue here.

Firstly, we must construct the mesh. In this stationary case the mesh has only two dimensions, the advance-time $\tau$ and the capacity remaining $Q$. Suppose that the domain we will work on is rectangular with $\tau$ ranging from 0 to $T$ and $Q$ ranging from 0 to $C$. Divide [0, $T$] into $K$ equally spaced intervals at $\tau$ values indexed by $k = 0, 1, \ldots, K$ and [0, $C$] into $J$ equally spaced intervals at $Q$ values indexed by $j = 0, 1, \ldots, J$. The length of these intervals is $\Delta\tau$ in the time direction and $\Delta Q$ in the state direction such that $\tau^k = k\Delta\tau\ \forall k$ and $Q_j = j\Delta Q\ \forall j$. We seek an approximation to the values of $V$ at the $(K+1)\times(J+1)$ grid points.

Therefore,

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

where $v$ is a $2D$ array.

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

As in section 5.3, we pre-calculate the conditional probability distribution $P_s(\cdot)$,

$$P_s(\xi^i|\tau^k) = P_s(i\Delta\xi|k\Delta\tau) \approx p_i^k$$

and the integral term on the RHS of the equation (8.20)

$$\int_0^{\xi^i} \rho_s(\xi'|\tau^k)\,\Psi(\xi')\,d\xi' = \int_0^{i\Delta\xi} \rho_s(\xi'|k\Delta\tau)\,\Psi(\xi')\,d\xi' \approx r_i^k.$$

Finally we recall that the average intensity $f$ in equation (5.33) may be written as

$$f(\tau^k) = f(k\Delta\tau) \approx f^k.$$

In the explicit scheme in section 5.3.3 we calculated the values at $\tau^k$ by using the already computed (known) values at the previous timestep $\tau^{k-1}$, resulting in an explicit relationship between *one* current value to a *set* of previous values. However, the implicit scheme entails that *one* previous value at $\tau^{k-1}$ is related to a *set* of new values at $\tau^k$. Under this representation, the new values still depend on the previous information but this happens in an *implicit* manner. In particular, we cannot solve for each new value *independently* but rather these values have to be *simultaneously* determined as the solution to a system of linear equations for every $j$.

Before explaining more on this, let us first approximate the derivatives of $v$. The implicit scheme uses a forward difference in time and and a central difference in space, i.e

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

$$\frac{\partial v}{\partial Q} = \frac{v_{j+1}^k - v_{j-1}^k}{2\Delta Q} + O(\Delta x^2)$$

$$\frac{\partial^2 v}{\partial Q^2} = \frac{v_{j+1}^k - 2v_j^k + v_{j-1}^k}{\Delta Q^2} + O(\Delta x^2),$$

where the spaces derivative are approximated at the future time step $\tau^k$. Note that in this discretisation, the only known quantity is $v_j^{k-1}$. We can use these terms to approximate the derivatives inside the PDE (8.20) and then separate the terms into known and unknown to obtain $J-1$ equations of the form

$$v_{j+1}^k\left[-f^k\,p_i^k\frac{\Delta\tau}{2\Delta Q}\left(1+\frac{1}{\Delta Q}\right)\right]+v_j^k\left[1+f^k\,p_i^k\frac{\Delta\tau}{\Delta Q^2}\right]$$

$$+\,v_{j-1}^k\left[f^k\,p_i^k\frac{\Delta\tau}{2\Delta Q}\left(1-\frac{1}{\Delta Q}\right)\right]=v_j^{k-1}+f^k\,r^k\Delta\tau,\quad \forall\, j=1,\ldots,J-1$$

This structure of the scheme enables us to rewrite this valuation problem as a system of $J+1$ equations, namely:

$$\begin{pmatrix} b_0 & c & 0 & 0 & . & . & . & 0 \\ a & b_1 & c & 0 & . & . & . & . \\ 0 & a & b_2 & c & 0 & . & . & . \\ 0 & 0 & a & b_3 & c & 0 & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & a & b_j & c & . \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & a & b_J \end{pmatrix}\begin{pmatrix} v_0^k \\ v_1^k \\ v_2^k \\ v_3^k \\ . \\ . \\ v_{J-1}^k \\ v_J^k \end{pmatrix}=\begin{pmatrix} d_0^k \\ d_1^k \\ d_2^k \\ d_3^k \\ . \\ . \\ d_{J-1}^k \\ d_J^k \end{pmatrix}$$

for every $k=1,\ldots,K$, where

$$a=-f^k\,p_i^k\,\frac{\Delta\tau}{2\Delta Q}\left(1+\frac{1}{\Delta Q}\right)$$

$$b_j=1+f^k\,p_i^k\,\frac{\Delta\tau}{\Delta Q^2}$$

$$c=f^k\,p_i^k\,\frac{\Delta\tau}{2\Delta Q}\left(1-\frac{1}{\Delta Q}\right)$$

and

$$d_j=v_j^{k-1}+f^k\,r^k\Delta\tau.$$

The above quantities are calculated for all capacities $j$ between 1 and $J-1$. For capacities $j=0$ and $j=C$ we need to impose the boundary conditions. The first row of the matrix refers to $j=0$ which corresponds to the capacity $Q=0$. From the boundary condition we know that at this state the value is zero for any choice of $\tau$. To preserve this condition we need to set $b_0 = c_0 = d_0 = 0$.

Similarly, the last row of the matrix corresponds to the capacity $Q = C$ the maximum capacity considered. At this state we have a boundary condition that sets the derivative of the value to zero. To include this condition in the scheme we need to discretise it first. In particular, using one-sided difference we may approximate it by

$$\frac{\partial V(C, \tau^k)}{\partial Q} \approx \frac{V(C, \tau^k) - V(C - \Delta Q, \tau^k)}{\Delta Q} \approx \frac{v_J^k - v_{J-1}^k}{\Delta Q}. \tag{8.26}$$

Setting the above equal to zero yields

$$v_J^k = v_{J-1}^k \quad \forall k, \tag{8.27}$$

which translates to $a_J = -1$, $b_J = 1$ and $d_J = 0$.

The optimal choice of $\xi^*$ (the index $i$ in the scheme) is calculated through the relationship in equation (8.24). In particular,

$$\xi^* = \Psi^{-1}\left(\frac{\partial V}{\partial Q} - \frac{1}{2}\frac{\partial^2 V}{\partial Q^2}\right), \tag{8.28}$$

which is expressed numerically as

$$i = \xi_j^k = \Psi^{-1}\left(\frac{v_{j+1}^{k-1} - v_{j-1}^{k-1}}{\Delta Q} - \frac{1}{2}\frac{v_{j+1}^{k-1} - 2v_j^{k-1} + v_{j-1}^{k-1}}{\Delta Q^2}\right), \tag{8.29}$$

where the terms on the right-hand side would have already been computed at the previous step.

Finally, *Thomas algorithm* (details in Smith, 1985) may be employed to solve the tridiagonal system of equations.

### 8.2.4 Results

#### 8.2.4.1 Solution analysis: numerical integrity

In the results below we use $T = 50$ and $C = 100$. Once again, we refer to the optimal solution as being the full set of values $v_j^k$ for all $j = [0, J]$ and $k = [0, K]$.

Table 8.3 presents the convergence in time for the implicit scheme as well as the relevant computation times in each case. As expected the convergence is linear in time. In particular, when using a timestep of $\Delta\tau = 0.003125$ we achieved solution accuracy to 3 significant figures. As for the computation times these range from as little as 5 seconds to as much as 74 seconds. As seen in the main PDE results of chapter 6 these times depend strongly on the time it takes for the pre-calculation of the matrices $p$ and $r$.

Similarly, table 8.4 presents the convergence in space with the relevant computation times in each case. Our results justify a second order convergence in space as by taking a step of half the size of the previous one, we improve the solution by (approximately) four times. In particular, the scheme is extremely accurate in the space dimension as even with considerably large space-steps we achieved a solution accuracy to 5 significant figures.

Table 8.3: Second-order PDE solution. First order convergence in time for the implicit scheme and computation times (in seconds) for different grids.

| $\Delta\xi$ | $\Delta\tau$ | $\Delta Q$ | $V(Q = 30, \tau = 50)$ | (%) Rel. Diff | Comp. Time (secs) |
|---|---|---|---|---|---|
| | 0.05 | | 330.6860622 | – | 4.94 |
| | 0.025 | | 331.6357739 | 0.2864 | 9.42 |
| 0.025 | 0.0125 | 0.025 | 332.1219689 | 0.1464 | 18.63 |
| | 0.00625 | | 332.3673601 | 0.0738 | 37.68 |
| | 0.003125 | | 332.4908955 | 0.0372 | 73.89 |

Table 8.4: Second-order PDE solution. Second order convergence in space for the implicit scheme and computation times (in seconds) for different grids.

| $\Delta\xi$ | $\Delta\tau$ | $\Delta Q$ | $V(Q = 30, \tau = 50)$ | (%) Rel. Diff | Comp. Time (secs) |
|---|---|---|---|---|---|
| | | 0.05 | 332.1217692 | – | 16.26 |
| | | 0.025 | 332.1219689 | $6.01 \times 10^{-5}$ | 18.47 |
| 0.025 | 0.0125 | 0.0125 | 332.1220239 | $1.65 \times 10^{-5}$ | 23.08 |
| | | 0.00625 | 332.1220402 | $4.91 \times 10^{-6}$ | 32.19 |
| | | 0.003125 | 332.1220439 | $1.11 \times 10^{-6}$ | 50.54 |

#### 8.2.4.2 Numerical comparison against the original PDE

Next, we compare the original PDE scheme to the new second order PDE scheme. Figure 8.14 presents the expected values with $\tau = 50$ days left as a function of the capacity remaining $Q$. The two schemes seem to generate very similar if not identical results, a fact that justifies our Brownian motion approximation to the Poisson process.

But how does the added diffusion term affect the solution? Let us recall the diffusion term (second derivative in equation (8.20))

$$\frac{1}{2} f(\tau)\, P_s(\xi^*(Q,\tau)|\tau)\, \frac{\partial^2 V}{\partial Q^2},$$

where $\xi^*(Q,\tau)$ is the optimal maximum duration of stay to implement when there are $Q$ spaces remaining and $\tau$ time left. By looking at the shape of the resulting value function in figure 8.14 we deduce that the second derivative should be close to zero in the region $Q < 50$ and $Q > 65$ because the value function $V$ shows little curvature in this region and thus behaves like a linear function.

If then the diffusion term is close to zero almost everywhere, one might think of removing it completely from the equation. However, by doing so we will end up with a first order PDE scheme, as the one in chapter 5 section 5.4, which will no longer be stochastic. So the diffusion term is not redundant everywhere.

Finally, it is interesting to examine the behaviour of the optimal condition $\Psi(\xi^*)$. Recall that the original PDE equates the optimal price to the marginal value of the space, namely

$$\Psi(\xi^*) = V(Q,\tau) - V(Q-1,\tau),$$

while the second order PDE uses equation (8.24) which reads

$$\Psi(\xi^*) = \frac{\partial V}{\partial Q} - \frac{1}{2}\frac{\partial^2 V}{\partial Q^2}. \tag{8.30}$$

Thus, we may calculate the expected marginal values out of the solution of the second order PDE and compare them against the above condition. Relevant results are shown in figure 8.15 which presents the optimal conditions as functions of the capacity remaining $Q$ with 50 days left. Even though the expected marginal value is slightly lower, we notice that the two curves are sufficiently close to one another, a result that justifies equation (8.25) about the approximation of the expected marginal value by the term $\partial V/\partial Q - 1/2\, \partial^2 V/\partial Q^2$.

An ongoing (and future) study on this model should provide further explanation in regards to the exact relationship between the original PDE and the second order PDE models and the role of the diffusion term in forming the final solution. Thus, at this stage of the work we do not extend into further analysis.

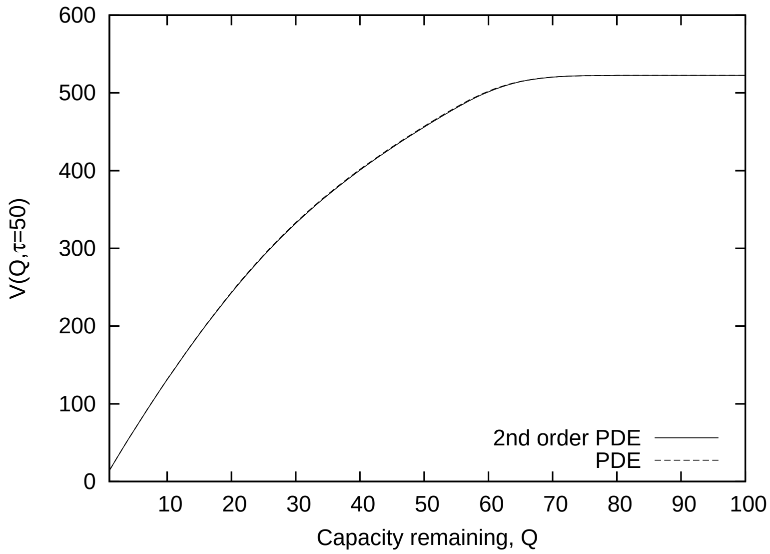



Figure 8.14: Expected revenues for the original and the second order PDE scheme with 50 days left, as a function of the remaining capacity $Q$.

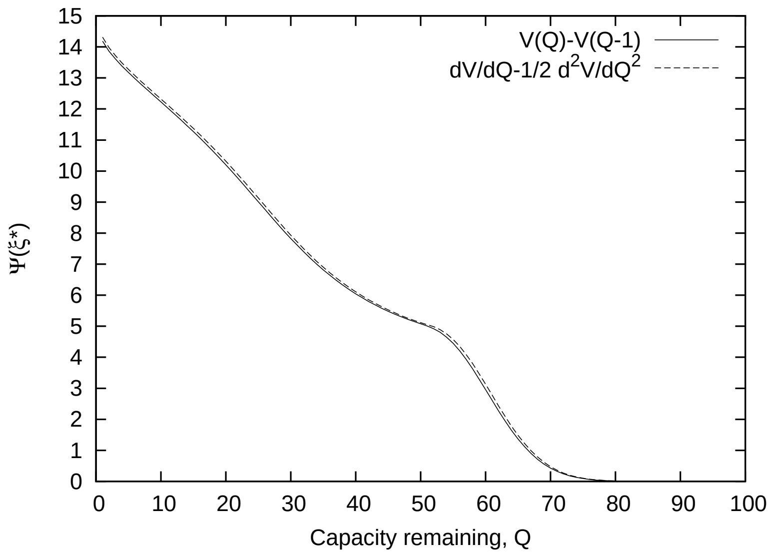



Figure 8.15: Comparison of the optimal condition $\Psi(\xi^*)$ as a function of the remaining capacity $Q$, with 50 days left.

# Chapter 9

# Conclusions

*"There are a lot of parallels between what we're doing and an expensive watch. It's very complex, has a lot of parts and it only has value when it's predictable and reliable."*

---

*Gordon Bethune, Chairman and CEO Continental Airlines 1997.*

In this thesis we have investigated the network RM problem encountered in an airport carpark. The objective has been to maximise the expected revenues on a given future day. Customers were assumed to arrive in random order and request a stay for any length of time. However, unlike in other industries, in our setting the price per day charged decreases monotonically in the length of stay of the customer. Consequently, operators must apply optimal accept/deny decisions to the booking requests in the order these arrive in the system taking into account both the length-of-stay and the total revenue to be generated.

Therefore, the inter-dependence of consecutive days on the one hand and the uncertainty in the demand process and/or length-of-stay on the other, resulted in a challenging optimisation problem.

The huge size of the problem meant that we had to develop several mathematical models to approximate the optimal solution. Even though each method can evaluate the expected revenue on a given day, $V(Q, \tau)$, for every time remaining $\tau$ and capacity remaining $Q$, the most useful output of these approximation methods was the set of expected marginal values , $\Delta V(Q, \tau) = V(Q, \tau) - V(Q-1, \tau)$, used to construct a bid-price control mechanism that enabled accept/deny decisions to be made in real time.

Three main methods have been developed in chapter 5, denoted as MC, PDE and Pontryagins. We note that usually managers prefer to sell the spaces for finite time intervals of size $\Delta T$, as in this way they extract more revenue from the customers. It is common the intervals to correspond to entire days ($\Delta T = 1$), but carpark operators are flexible enough to reduce it down to hours ($\Delta T = 1/24$) or even minutes.

However, the continuous-time nature of the PDE and Pontryagins models implied that a comparison among the three methods should be made in the limiting case, as $\Delta T \to 0$. Numerical tests have been conducted to examine the methods' performance in maximising the perpetual expected revenue per day. Taking into account both computation times and generated expected revenues, we have shown that the PDE model is the preferable method. This motivated us in developing the PDE model even further.

In particular, we have shown the conditions under which the PDE model can be transformed accordingly in order to calculate the expected revenue rate generated within a finite interval of size $\Delta T$, as opposed to an infinitesimal time. Then, the performance of the adjusted PDE model was examined for different choices of $\Delta T$ and the results revealed that the method performed always within 10% of the MC solution.

Note that the optimised PDE model is based on a time-stationary setting (equation (5.46)). In other words we assumed that booking, arrival and staying intensities are fixed. This means that we were able to compute a dynamic bid-price table with only two dimensions (the pre-booking time $\tau$ and the capacity remaining $Q$) as the actual target day $T$ did not matter and thus all days are treated in the same manner.

In general, customer demand is time-varying as different times of the year, days of the week or even hours in a day face a different stream of customers. What is more the distribution of length-of-stay of the customers might also change over time. Fortunately, our PDE model is flexible enough to model this situation; it can use non-stationary booking intensities $\lambda_b(t)$ and $\phi(\eta, \xi, t)$ can be any probability distribution.

A simple illustration of the time-varying setting is to assume that the weekly booking intensity of class $n$ is of the form

$$\lambda_{b_n}(t) = \lambda_{b_n} \sin\left(\frac{\pi t}{7}\right),$$

a half sinusoid peaking towards the middle of the week. In such a case, the non-stationary PDE formulation in equation (5.37) could easily be extended to incorporate a rejection policy based on length of stay and thus to obtain a set of *distinct* optimal expected marginal values for *each* target day $T$, namely

$$\Delta V(Q,\tau;T) = V(Q,\tau;T) - V(Q-1,\tau;T) \quad \forall T.$$

This will then lead to a set of bid-price tables, one for each day $T$ in the horizon.

It is worth noting that the PDE still remains simple to solve (since its dimensions do not increase), with the only difference that we will have to solve one PDE for each target day $T$ we choose.

However, we note that the non-stationary PDE is still based on a single-resource model. Also remember the observation of high peaks and asymmetries in capacities availabilities over neighbouring days, an issue that has been addressed in section 6.4. This indicated that in the network case the bid price of a given day $T$ should not just be the expected marginal value of that day (which was based on a single-resource solution) but should somehow relate to the capacity availabilities of the neighboring days, thus to their marginal values.

Let us take a simple example to illustrate the situation. Consider a booking request on day $t$ for a stay over $\xi$ days $T_1, \dots, T_\xi$, this booking if accepted it will generate a revenue equal to $\xi\Psi(\xi)$. The *EAMV* heuristic would accept the booking if

$$\sum_{T=T_1}^{T_\xi} \Big(\Psi(\xi) - \pi(Q,\tau = T-t;T)\Big) \geq 0,$$

where the bid price $\pi(Q,\tau;T)$ was given by

$$\pi(Q,\tau;T) = \Delta V(Q,\tau;T) \tag{9.1}$$

and the expected marginal value $\Delta V(Q,\tau;T)$ was computed based on the single-resource solution. The challenge is then to come up with a more representative set of bid prices that would encapsulate some information about the surrounding days too. Bearing in mind that under a continuous-time setting the number of products are infinite (bookings can stay for any length of stay which is a continuous quantity), we introduced a weighted PDE scheme (chapter 8) in an attempt to eliminate the peaks and make sure that the carpark is filling out uniformly. In particular, the proposed scheme applies a correction to the single-resource marginal values by applying a weighted average over the *infinite set* of marginal values of the neighbouring days, where the number of spaces remaining for each day is computed according to the optimal expected sales trajectory (see figure 8.5). The use of the weighting kernel $W(\cdot)$ is to assign a probability to the event that a customer is present on a neighbouring day *given* that he is present on a specified day $T$. This is in fact an estimate of the demand on the surrounding days of $T$. Finally, the bid prices are obtained by equating them to these "corrected" marginal values, as in equation (9.1).

However, using the optimal expected sales trajectory to determine the capacity availabilities on these surrounding days might not be the best way in practice. As the state of the carpark unfolds it might become asymmetric in capacity availabilities over consecutive days. If in the extreme case, the carpark on day $T$ is highly congested while in the surrounding days capacity is abundant, the bid price of day $T$ should rise to prevent short stay customers of filling out day $T$ reducing demand for the surrounding days. In contrast, if on day $T$ the carpark is almost empty while in the neighbouring days the spaces are sold out the bid price on day $T$ should reduce to make sure that the spaces are sold. These scenarios may be seen in figure 9.1.

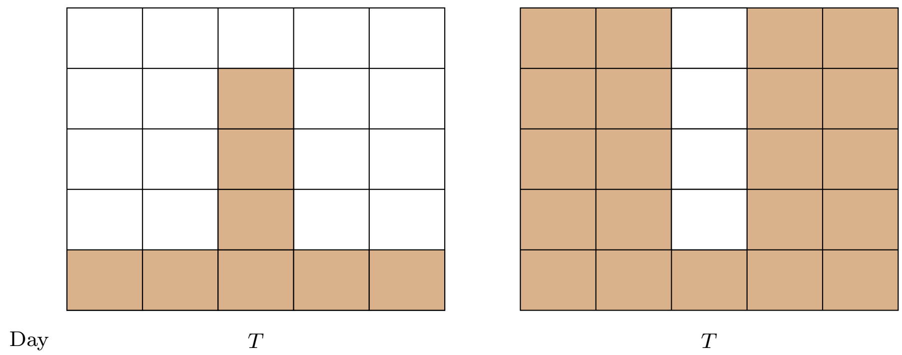



Figure 9.1: States of a carpark with size $C = 5$ under extreme scenarios. The left figure shows a carpark being highly congested on day $T$ but with abundant capacity on its surrounding days, whereas the right figure shows a carpark almost empty on day $T$ but sold out on the neighbouring days.

We believe that the proposed weighting kernel might be the key in understanding how to deal with this problem. Intuitively, the weighting kernel should "add in" value to the bid price of day $T$ in the former case, while it should "take out" value from the bid price in the latter. The exact manner at which the weighting kernel should adjust according to the state of the carpark is not obvious at this stage of the research.

In conclusion, this problem naturally goes to further work and its outcome could potentially bring new insights as to how network problems could be tackled in the future.

# Case study: RM in car supermarkets

*Aside to my Ph.D programme I had an excellent opportunity to work as a pricing analyst with one of the UK's leading used-car supermarkets. This has been a great experience for myself (and them too!) and my future plans as I have been given the chance to apply novel RM techniques I acquired in a real environment. The study was undertaken for a period of three months and this section is dedicated on the main models, results and analysis performed on investigating car sales data provided.*

The main question we tried to answer is how should the prices of each car be set so that the firm's revenues are maximised. Thus, our main objective is to model the relationship between car prices and customer demand. We have delivered a statistical model which, for a given period of time and a given stock distribution, can accurately replicate the expected number of sales (demand) and the expected resulting revenue from the sales. Using this model, we are able to determine the price elasticity of demand (customer sensitivity to price) for a given set of car sales data. Once the price elasticity has been determined, we are able to carry out two pieces of analysis. Firstly, we are able to forecast the number of sales and revenue generated, both if prices stay the same or if the price of the stock is changed (within certain limits of significance). Secondly, it provides a good way to group particular cars according to their sensitivity to price. This targets groupings on customer behaviour, which is what drives how likely a customer is to pay a certain price for a car, and hence the revenues.

# Model formulation

## Defining the price

In order to build a pricing model we first need to examine customer behaviour and in particular how the volume of sales is affected by changes in advertised prices. Since most sales are driven through online channels, we focused on modelling the impact of the web-advertised prices in the customer demand. For notational purposes we refer to the advertised web price of car $j$ by $WP_j$.

The value of any car depends upon many factors such as make, model, mileage, engine, transmission. As such, the advertised web prices will vary between even cars of the same model. Therefore, to measure the response when pricing a car at a particular level we will use the *CAP* price at sale of that particular car as our theoretical estimate for the *fair* price. The *CAP* price efficiently encapsulates all previously mentioned factors and it assigns an estimated value to every single car. This is the closest estimate to the fair value as it is the only source of measure acknowledged from all distributors and courts.

In particular, we define the *relative* web price ($p$) for car $j$ to be:

$$p_j = \frac{WP_j}{CAP_j}, \tag{1}$$

Then, $WP_i$ can alternatively be expressed as

$$WP_i = p_i \, CAP_i. \tag{2}$$

For instance, if the $WP_i$ is 6% above the corresponding $CAP$ price, then $p_i = 1.06$.

Usually, the *final* prices ($FP$) at which cars are sold are slightly different from the advertised web prices at the time of sale. Thus, we define the *relative* final price ($f$) for car $i$ to be:

$$f_i = \frac{FP_i}{CAP_i}, \tag{3}$$

Then, $FP_i$ now reads

$$FP_i = f_i \, CAP_i. \tag{4}$$

## Defining the time unit

The historical stock data hold snapshots of the stock for each day throughout the period of interest. This implies that cars might have multiple records within the same week. Since daily prices at which a particular car is advertised, rarely (and if so, slightly) change within the same week, we can convert the data into a weekly stock data that will consist of unique weekly car records (but still multiple records, in general). Therefore, a one-week time span becomes the standard unit of time used in the analysis.

## Formatting the data

The historical sales data and the historical weekly stock data are merged together so that for each car we have a complete path showing the weekly price movements on that car until the time of sale. The variables $p$ and $f$ will form two new columns in the dataset, with $f$ taking zero values at the observations that come from the stock data.

A sample data example for a particular car model can be seen in table CS1. The column $i$ assigns a unique number to each of the observations in the data, and column $j$ refers to a particular car. The *State* column shows whether the corresponding observation $i$ of car $j$ was accounted as being a Stock or as being a Sale - the day at which car $j$ was sold. In this table, there are four unique cars. Cars 1 and 3 first appeared in stock on 14/12/09, car 2 on 28/12/09 and car 4 on 21/12/2009.

## Modelling the demand

The methodology presented below, can be applied to any given time period (1 month, 2-months, 1 year) for which data is available. We assume that weekly sales evolve according to a Poisson process with stationary intensity $\lambda$. Thus, the probability of $n$ sales in a week is Poisson distributed, namely

$$P(n;\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}. \tag{5}$$

In this setting, $\lambda$ denotes the expected number of sales per week (demand). This is expressed as rate (# sales/week) that depends only[1] on the prices the firm

[1]Competitors prices are not considered in this model setting for the time being.

Table CS1: Sample data example

| i | j | Week beginning | State | CAP | $p_i$ | $f_i$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 14/12/09 | stock | 8000 | 1.20 | 0 |
| 2 | 1 | 21/12/09 | stock | 8000 | 1.18 | 0 |
| 3 | 1 | 28/12/09 | sale | 7900 | 1.18 | 1.23 |
| 4 | 2 | 21/12/09 | stock | 11500 | 1.17 | 0 |
| 5 | 2 | 28/12/09 | sale | 11300 | 1.13 | 1.14 |
| 6 | 3 | 14/12/09 | stock | 10000 | 1.20 | 0 |
| 7 | 3 | 21/12/09 | stock | 10000 | 1.18 | 0 |
| 8 | 3 | 28/12/09 | stock | 9500 | 1.12 | 0 |
| 9 | 3 | 04/01/10 | stock | 9500 | 1.10 | 0 |
| 10 | 3 | 11/01/10 | sale | 9500 | 1.10 | 1.10 |
| 11 | 4 | 21/12/09 | stock | 12300 | 1.19 | 0 |
| 12 | 4 | 28/12/09 | stock | 12300 | 1.19 | 0 |
| 13 | 4 | 04/01/10 | stock | 12300 | 1.19 | 0 |
| 14 | 4 | 11/01/10 | stock | 10200 | 1.14 | 0 |
| 15 | 4 | 18/01/10 | stock | 10200 | 1.14 | 0 |
| 16 | 4 | 25/01/10 | stock | 10200 | 1.13 | 0 |
| 17 | 4 | 02/02/10 | sale | 9300 | 1.11 | 1.12 |

advertises their cars for, through a function $\lambda = \lambda(p)$.

Although, this simple model assumes that the data is stationary, i.e. that customers intensities have been constant along the year, it is still useful to study this model before proceeding into more complex models.

The available historical data suggests that the price-demand function is of an exponential form i.e.

$$\lambda(p) = \beta_1 e^{-\beta_2 p}, \tag{6}$$

where $\beta_1$ and $\beta_2$ are fitting parameters that are explicitly determined from the data. Usually, the decaying coefficient $\beta_2$ is referred to as the *price elasticity* parameter.

The range of prices $p$ is a continuous quantity and lies in the interval $p \in [a, b]$. This interval is then divided into $K$ bins of size $\Delta p$. Thus, we can express the $k^{th}$ price-bin as

$$p^k = \big(a + (k-1)\Delta p\,,\, a + k\Delta p\big] \qquad \forall\, k = 1, 2, \ldots, K. \tag{7}$$

For each unique car $j$ in the data we can find:

1. The time elapsed from when it was first advertised on the web until the time it eventually was sold.

2. The time spent at each price interval prior to being sold.
3. The price interval it was advertised at the time of sale.
4. The actual price it was sold for.

By only considering the prices at the time of sale, we ignore the valuable information that can be extracted from the time for which cars were sitting in particular price-bins *without being sold*. Thus, intensities are determined using the following procedure:

Let us define $ST_{jk}$ to be the number of weeks that the $j$th car was advertised at the price $p^k$ before it was sold. In other words, this is all instances for which car $j$ has been counted as a Stock. Then, the *total* number of observations for which all cars in Stock were priced at $p^k$ is given by

$$\sum_j ST_{jk}\,, \tag{8}$$

so that

$$\sum_k \sum_j ST_{jk} \tag{9}$$

is the total number of observations in the data.

Now, let us denote the number of sales made at $p^k$ by $SA_k$. Then, the average time needed for a car being priced at price-bin $p^k$ to sell is given by

$$\bar{t_k} = \frac{\sum_j ST_{jk}}{SA_k}. \tag{10}$$

Therefore, the observed intensity ($\lambda$) at which sales are made when cars are priced in $p^k$ is

$$\lambda(p^k) = \frac{1}{\bar{t_k}}. \tag{11}$$

We show the observed intensities against the price-bins in figure CS1. Since each of the points in this plot have different statistical significance[2], we assign to them appropriate weights and then perform a weighted exponential regression to obtain the fitted intensities $\hat{\lambda}(p^k) \approx \lambda(p^k) \quad \forall k$. The blue line in figure CS1 shows

[2]For the derivation of each of these points we use equations (8) and (11). The accuracy of each point depends upon the information provided by its corresponding equation (8); the greater this is, the more accurate the derived intensity is.

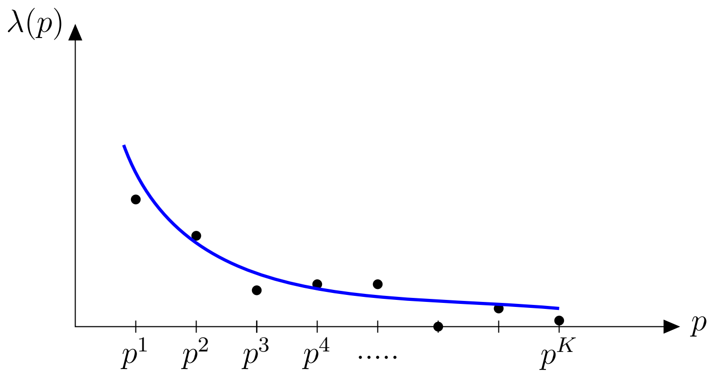



Figure CS1: Estimating the price-demand function $\lambda(p)$

the curve of best fit and it outputs the values for parameters $\beta_1$ and $\beta_2$ that are used to convert this discrete price-demand function into a continuous function.

## Actual sales and revenue

As described before, the *actual* number of sales occurred (over the pre-specified time range) by pricing at $p^k$ was given by $SA_k$. In fact, using equations (8) and (11) we may write it as

$$\sum_j ST_{jk}\,\lambda(p^k), \tag{12}$$

and therefore, the *total* number of sales over all price-bins $k$ can be expressed as

$$\sum_k \left[\sum_j ST_{jk}\,\lambda(p^k)\right]. \tag{13}$$

The *actual* revenue[3] realised by pricing at $p^k$ is given by

$$\sum_{\substack{i \\ p_i \in k}} f_i\,CAP_i. \tag{14}$$

This formula simply sums up all final prices for cars that were advertised at price bin $p^k$. Consequently, the *actual total* revenue realised over all price-bins can be expressed as

$$\sum_k \sum_{\substack{i \\ p_i \in k}} f_i\,CAP_i. \tag{15}$$

[3]By the term *revenue* we mean the amount of money collected from sales without subtracting the purchase prices or any other costs that the firm might incur.

## Estimated sales and revenue

The *estimated* number of sales occurred by pricing at $p^k$ is given by

$$\mathrm{E}\left\{\sum_j ST_{jk}\,\hat{\lambda}(p^k)\right\} \tag{16}$$

and therefore, the *estimated total* number of sales over all price-bins can be expressed as

$$\mathrm{E}\left\{\sum_k \left[\sum_j ST_{jk}\,\hat{\lambda}(p^k)\right]\right\}. \tag{17}$$

Equation (17) calculates the expected number of sales by using the fitted intensities $\hat{\lambda}(k)$ as opposed to the observed intensities $\lambda(k)$ which have been used to derive the actual number of sales in (13). The *estimated* revenue realised by pricing at $p^k$ is given by

$$\mathrm{E}\left\{\sum_{\substack{i\\ p_i\in k}} \hat{\lambda}(p_i)\,p_i\,CAP_i\,\alpha\right\} \tag{18}$$

and therefore, the *estimated total* revenue realised over all price-bins can be expressed as

$$\mathrm{E}\left\{\sum_k \sum_{\substack{i\\ p_i\in k}} \hat{\lambda}(p_i)\,p_i\,CAP_i\,\alpha\right\}, \tag{19}$$

where the parameter $\alpha$ will be explained shortly.

Let us now try to explain these equations. Each observation $i$ in our data file is assigned a probability $\hat{\lambda}(p_i)$ according to the (exact) price $p_i$ it is advertised at. This is the intensity at which observation $i$ is expected to sell at by being priced at $p_i$. In this way we calculate the expected revenue that is generated from each observation $i$ to be the intensity of sale $\hat{\lambda}(p_i)$ times the price $p_i$ times $CAP_i$. Summing all expected revenues that result from observations of the same price-bin we obtain (18), and after summing over all price-bins we obtain the total expected revenue as shown in (19).

The extra parameter $\alpha$ captures the fact that final prices often differ from the web prices at the time of sale, and it can be adjusted accordingly so that the error in revenue is minimised. Starting values for the estimation of $\alpha$ are

$$\alpha = \frac{\sum_i^I f_i}{\sum_i^I p_i}, \tag{20}$$

where $I$ is the subset of observations $i$ that are noted as a sale. In practice, $\alpha$ determines the percentage average price above $p$ that the firm would aim to receive at the time of sale.

## Numerical results on car model $XXX$

In this section we present the main results of our analysis on a particular car model $XXX$. Similar analysis has been performed to all major (based on popularity) car models.

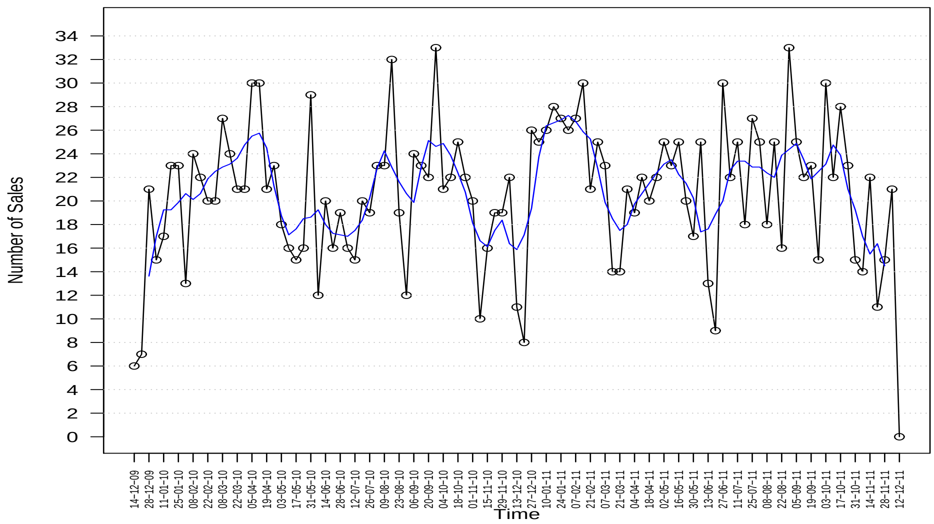



Figure CS2: Car model $XXX$ sales over a two-year period.

Figure CS2 presents a time series data on the sales of model $XXX$ over a two year period from December 2009 to December 2011. Although, the number of sales varies significantly between different time regions, the assumption of using a stationary intensity in our model is reasonable, as the mean of the time series is approximately constant.

In figure CS3 we show the average sales intensities (number of sales per week) for each price-bin $p^k$ along with the fitted price-demand intensity $\hat{\lambda}(p)$. The size of price-bins used here (and throughout this report) is 0.01 (1%) for better resolution. The blue curve is the fitted intensity after applying a weighted exponential regression[4]. The exponential model is

$$\lambda(p) = 0.242e^{-0.971p}. \tag{21}$$

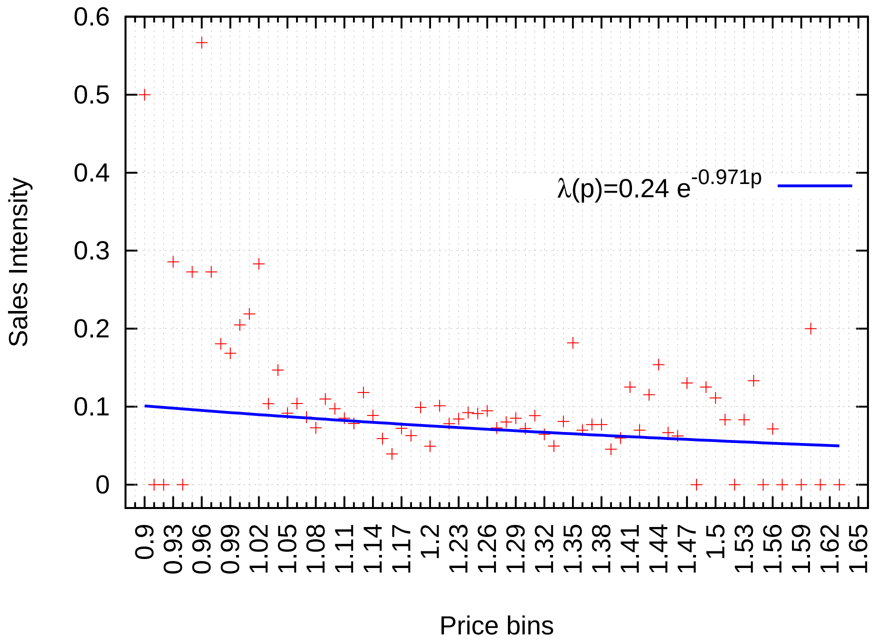



Figure CS3: Average sales intensities (# sales/week) per price-bin $p^k$ for the period 14/12/2009 to 12/12/2011.

We can then use equation (17) to estimate the total number of sales made within this 2-year period. Table CS2 summarises our results on the estimated number of sales. Notably, our proposed model estimates 2161 car sales, a highly accurate result leaving an error of less than a tenth of one percent.

Similarly, the estimated revenue is shown in table CS3. In this case, our proposed model estimates the total generated revenue with accuracy of just above three percent. This result is achieved with $\alpha$ set to 1.039, which means that company $XYZ$ can afford, on average, to charge up to an extra 3.9% above the cars' advertised web prices at the time of sale.

[4]As previously explained (figure CS1) the fitted curve will be drugged up or down according to the weights of the data points. Thus, although the curve does seem closer to being linear, this is actually the most representative fit of the data. Once more data become available these curves could easily be updated. Of course, such data to be generated it requires a bit of price experimentation.

Table CS2: Results on estimating the total number of sales

| Actual total # sales | Estimated total # sales | Relative difference in # sales |
|---|---|---|
| 2163 | 2161 | $-0.1\%$ |

Table CS3: Results on estimating the total revenue

| Relative difference in revenue estimation | $\alpha$ |
|---|---|
| 3.2% | 1.039 |

## Investigating sensitivity of prices

We now investigate the degree of change in total revenue and number of sales by shifting the stock distribution by a small amount $\Delta p = 0.01$. In other words, given the prices all cars of this model have been advertised at (mean advertised price is $\bar{p} = 1.20$), for the entire two-year period, we examine the revenue difference that would have occurred if the prices had *all* been shifted either up or down by the same small amount $\Delta p$.

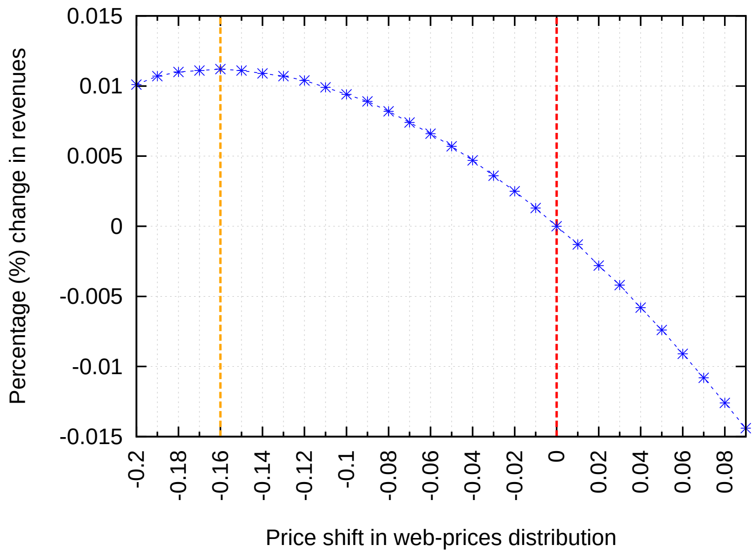



Figure CS4: Expected % change in total revenue by shifting the mean of the stock distribution.

Figure CS4 shows the expected percentage change in revenue as we shift (increase and decrease) the mean of the stock distribution (the point of no price change is indicated by the 0 on the x-axis, as indicated by the red vertical line).

Table CS4: Revenue and sales sensitivity on prices.

| Price shift from mean $\bar{p}$ by | Expected # sales | % Change in expected # sales | % Change in expected revenue |
|---|---|---|---|
| -0.03 | 2226 | 0.03 | 0.0036 |
| -0.02 | 2205 | 0.02 | 0.0025 |
| -0.01 | 2182 | 0.01 | 0.0013 |
| 0 | 2161 | 0 | 0.0000 |
| 0.01 | 2142 | -0.01 | -0.0013 |
| 0.02 | 2120 | -0.02 | -0.0028 |
| 0.03 | 2102 | -0.03 | -0.0042 |

This line indicates the current state where no shift on prices has been performed. As we increase prices the total expected revenue decreases; although the marginal revenue from each car increases, the resulting revenue is outperformed by a drop in the total number of sales. However, as we lower the prices, the expected number of sales as well as the total expected revenue increases up to a point where the marginal revenue from each car is so low that the high number of sales is not adequate to stimulate the total revenue even further, that is when total revenue begins to decrease again. The optimal price shift percentage is the point when the maximum revenue has been achieved. From figure CS4 the optimal point is found when pricing on average 0.16 below the current mean level ($\bar{p} = 1.20$). This point is indicated by the vertical orange line.

The result does not imply that the optimal price for each car is $p^* = 1.20 - 0.16 = 1.04$. It simply tells us that for the given stock distribution and the given time period, revenue would have been maximised if *all* web prices were lowered by an amount of 0.16.

Some key notes to consider are that there is a possibility that once other related costs (such as transaction costs, car delivery, car service) are taken into consideration, the excess revenue of 1.1% could be wiped out. In addition, if the optimal point that maximises revenues requires a significant price drop, the company $XYZ$ will enter a competition zone and, henceforth, potential interactions with the other competitors could be an issue.

Therefore, $XYZ$ might not be interested to determine this optimal point (which requires a massive and perhaps ineffective price movement) but just to investigate the sensitivity of sales and revenues for small price change. For this, table CS4 is provided.

### Out-of-sample forecasts

Given a time period the derived model produced encouraging in-sample forecasts. However, the question is how well the model performs for periods that lie outside the period from which the model is estimated (out-of-sample forecasts).

To illustrate this idea we provide an example which forecasts the number of sales and the total revenue for the month in October of 2011. Then, in order to derive the appropriate price-demand function $\lambda(p)$, we can use the available data from 14/12/2009 to 30/09/2011. Results about the performance of our model can be viewed in tables CS5.

Table CS5: Forecast sales and revenue for October 2011.

| Actual total # sales | Estimated total # sales | Relative difference in # sales | Relative difference in revenue |
|---|---|---|---|
| 103 | 93 | $-9.7\%$ | -3.5% |

Our out-of-sample forecast results in a relative sales error and a relative revenue error of less than 10% and less than 4%, respectively. We note that forecast accuracy varies between different time periods and that the longer the period of interest the lower the accuracy of the out-of-sample forecast.

## Concluding remarks

Based on the data available, we have developed a model that derives the relationship between price and demand using an exponential function and we showed the optimal pricing point that would have maximised the total revenue. However, the expected increase in revenue was relatively low compared to the price movement needed. This is caused by two factors;

1. The elasticity parameter $\beta_2 = -0.971$ indicates that customers are not very price sensitive and, thus, a small change in the prices does not affect the total outcome significantly.

2. Often, the sensitivity of customers, and the demand in general, depends upon the time period (Christmas, Summer) and therefore shifts in prices might result in different (and sometimes opposite) outcomes. Thus, extra revenues and opportunities generated by the model in a period-interval

might have off set by losses made in other time-intervals. This suggests a further study whereby seasonal factors may be taken into account.

As previously mentioned this study has been undertaken for every major car model in stock, generating a list of price elasticities ($\beta_2$) of the customers according to the make and model of the car. Working mostly with these parameters we could group car models into different clusters with respect to their price elasticities. These clusters have revealed some interesting insights into the customer behaviour, as car models that target *different* customer markets (low, middle, higher classes) may still end up in the *same* cluster and should, therefore, be treated similarly. Such a result also establishes other advantages in terms of the applicability of the model, as a manager will now have fewer parameters to control.

We believe that the statistical model in this report enables a simple way for company $XYZ$ to analyse the data. So long as the data can be combined and formatted correctly, the calculations are simple enough to be built into a spread sheet, and can report back with estimates of expected demand and revenue. It can be run over a selected period of time allowing it to be constantly updated and re-calibrated as new data is revealed. There is also potential for competitor data to be included in the model as long as the corresponding $CAP$'s can be identified for each car in their inventory. $XYZ$ will be able to use this information along with their own analysis (factoring in purchase costs, cost of processing etc.) to determine which groups of stock should be targeted for price changes. In particular, they can use the model to predict which price changes will drive the most profit.

# Bibliography

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

Asmussen, S. and Albrecher, H. (2010). *Ruin probabilities*, volume 14. World Scientific.

Asmussen, S. and Rosiński, J. (2001). Approximations of small jumps of lévy processes with a view towards simulation. *Journal of Applied Probability*, 38(2):482–493.

Baker, T. K. and Collier, D. A. (1999). A comparative revenue analysis of hotel yield management heuristics. *Decision Sciences*, 30(1):239–263.

Baker, T. K. and Collier, D. A. (2003). The benefits of optimizing prices to manage demand in hotel revenue management systems. *Production and Operations Management*, 12(4):502–518.

Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, 53(1):51–67.

Belobaba, P. P. (1987). Airline yield management. an overview of seat inventory control. *Transportation Science*, 21(2):63–73.

Belobaba, P. P. (1989). Application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37(2):183–197.

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

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

Bertsimas, D. and Popescu, I. (2003). Revenue management in a dynamic network environment. *Transportation Science*, 37(3):257–277.

Bitran, G. and Caldentey, R. (2003). An overview of pricing models for revenue management. *Manufactoring and Service Operations Management*, 5:203–230.

Bitran, G., Caldentey, R., and Mondschein, S. (1998). Coordinating clearance markdown sales of seasonal products in retail chains. *Operations Research*, 46(5):609–624.

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.

Bitran, G. S. and Mondschein, S. (1997). Periodic pricing of seasonal product in retailing. *Management Science*, 43:427–443.

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

Butcher, J. C. (2008). *Numerical methods for ordinary differential equations*. John Wiley & Sons.

Çinlar, E. (1975). Introduction to stochastic processes. 1975. *Printice-Hall, Englewood Cliffs, NJ*.

Chen, V. C., Günther, D., and Johnson, E. L. (1998). A markov decision problem based approach to the airline ym problem. In *YM Problem, Working Paper, The Logistics Institute, Georgia Institute of Technology*. Citeseer.

Chiang, A. C. (1992). *Elements of Dynamic Optimization*. New York: McGraw-Hill.

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. L. (2002). Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50(4):720–727.

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

Cross, R. G. (1997). *Revenue management: hard-core tactics for market domination*. New York: Broaway Books.

Curry, R. E. (1990). Optimal airline seat allocation with fare classes nested be origins and destinations. *Transportation Science*, 24(3):193–204.

D'Acierno, L., Gallo, M., and Montella, B. (2006). Optimisation models for the urban parking pricing problem. *Transport Policy*, 13(1):34–48.

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.

Feng, Y. and Gallego, G. (1995). Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management Science*, 41(8):1371–1391.

Feng, Y. and Gallego, G. (2000). Asset revenue management with markovian time dependent demand intensities. *Management Science*, 46(7):941–956.

Feng, Y. and Xiao, B. (2000a). A continuous-time yield management model with multiple prices and reversible price changes. *Management Science*, 46(5):644–657.

Feng, Y. and Xiao, B. (2000b). Optimal policies of yield management with multiple predetermined prices. *Operations Research*, 48(2):332–343.

Feng, Y. and Xiao, B. (2001). A dynamic airline seat inventory control model and its optimal policy. *Operations Research*, 49(6):938–949.

Gallager, R. G. (1996). *Discrete stochastic processes*, volume 101. Kluwer Academic Publishers Boston.

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

Gallego, G. and van Ryzin, G. J. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41.

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.

Guadix, J., Cortés, P., Muñuzuri, J., and Onieva, L. (2008). Parking revenue management. *Journal of Revenue & Pricing Management*, 8(4):343–356.

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

Günther, D. P., Chen, V. C. P., and Johnson, E. L. (1999). Airline yield management - optimal bid prices for single-hub problems without cancellations.

Hochbaum, D. S., editor (1996). *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, MA, USA.

Ivanov, S. and Zhechev, V. (2012). Hotel revenue management–a critical literature review. *Turizam: znanstveno-stručni časopis*, 60(2):175–197.

Kimes, S. E. (2000). *A Strategic Approach to Yield Management*. Yield management: Strategies for the service industries. In Ingold A., McMahon-Beattie, U., and I.Yeoman (Eds), London: Continuum.

Kincaid, W. M. and Darling, D. A. (1963). An inventory pricing problem. *Journal of Mathematical Analysis and Applications*, 7:183–208.

Koide, T. and Ishii, H. (2005). The hotel yield management with two types of room prices, overbooking and cancellations. *International Journal of Production Economics*, 93-94:417–428.

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

Lai, K.-K. and Ng, W.-L. (2005). A stochastic approach to hotel revenue optimization. *Computers and Operations Research*, 32(5):1059–1072.

Lautenbacher, C. J. and Stidham, S. J. (1999). The underlying markov decision process in the single-leg airline yield-management problem. *Transportation Science*, 33(2):136–146.

Lee, T. C. and Hersh, M. (1993). A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, 27(3):252–265.

Liang, Y. (1999). Solution to the continuous time dynamic yield management model. *Transportation Science*, 33(1):117–123.

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

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.

Luenberger, D. G. (2003). *Linear and nonlinear programming.* Springer.

Maglaras, C. and Meissner, J. (2006). Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing and Service Operations Management*, 8(2):136–148.

McGill, J. I. and van Ryzin, G. J. (1999). Revenue management: Research overview and prospects. *Transportation Science*, 33:233–256.

Mircea, I., Covrig, M., and Todose, D. (2010). Brownian motion for estimating the ruin probability of a risk process. In *International Conference On Applied Economics–ICOAE*, page 529.

Neftçi, S. (2000). *An introduction to the mathematics of financial derivatives.* Access Online via Elsevier.

Øksendal, B. (1995). *Stochastic differential equations: an introduction with applications*, volume 6. Springer New York.

Øksendal, B. K. and Sulem, A. (2007). *Applied stochastic control of jump diffusions*. Springer.

Olver, P. J. and Shakiban, C. (2006). *Applied linear algebra*. Prentice Hall.

Pak, K. and Piersma, N. (2002). overview of or techniques for airline revenue management. *Statistica Neerlandica*, 56(4):479–495.

Phillips, R. L. (2005). *Pricing and Revenue Optimization*. Stanford Business Books.

Pölt, S. (1998). Forecasting is difficult–especially if it refers to the future. In *Reservation and Yield Management Study Group Annual Meeting Proceedings*, Melbourne, Australia. AGIFORS.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2009). *Numerical recipes in C++: the art of scientific computing*, volume 994. Cambridge University Press Cambridge.

Qian, Z. S. and Rajagopal, R. (2013). Optimal parking pricing in general networks with provision of occupancy information. *Procedia - Social and Behavioral Sciences*, 80(0):779–805. ¡ce:title¿20th International Symposium on Transportation and Traffic Theory (ISTTT 2013)¡/ce:title¿.

Queenan, C. C., Ferguson, M., Higbie, J., and Kapoor, R. (2007). A comparison of unconstraining methods to improve revenue management systems. *Production and Operations Management*, 16(6):729–746.

Raman, K. and Chatterjee, R. (1995). Optimal monopolist pricing under demand uncertainty in dynamic markets. *Management Science*, 41(1):144–162.

Robinson, L. W. (1995). Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research*, 43(2):252–263.

Rojas, D. (2006). Revenue management techniques applied to the parking industry. Master's thesis, School of Industrial and Systems Engineering, University of Florida.

Rothstein, M. (1971). An airline overbooking model. *Transportation science*, 5:180–192.

Rydberg, T. H. (1997). The normal inverse gaussian lévy process: simulation and approximation. *Communications in statistics. Stochastic models*, 13(4):887–910.

Sachs, E. and Strauss, A. (2008). Efficient solution of a partial integro-differential equation in finance. *Applied Numerical Mathematics*, 58(11):1687–1703.

Schrijver, A. (1998). *Theory of linear and integer programming*. Wiley.

Shen, Z.-J. M. and Su, X. (2007). Customer behavior modeling in revenue management and auctions: A review and new research opportunities. *Production and operations management*, 16(6):713–728.

Simpson, R. (1989). *Using Network Flow Techniques to Find Shadow Prices for Market Demands and Seat Inventory Control*. FTL memorandum. MIT, Department of Aeronautics and Astronautics, Flight Transportation Laboratory.

Smith, B. and Penn, C. (1988). Analysis of alternate origin-destination control strategies. In *AGIFORS Symposium Proceedings*, pages 123–144. AGIFORS.

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.

Smith, S. A. and Achabal, D. D. (1998). Clearance pricing and inventory policies for retail chains. *Management Science*, 44(3):285–300.

Subramanian, J., Stidham Jr., S., and Lautenbacher, C. J. (1999). Airline yield management with overbooking, cancellations, and no-shows. *Transportation Science*, 33(2):147–167.

Talluri, K. and Van Ryzin, G. (1999). A randomized linear programming method for computing network bid prices. *Transportation Science*, 33(2):207–216.

Talluri, K. and van Ryzin, G. J. (1998). An analysis of bid-price controls for network revenue management. *Management Science*, 44(11):1577–1593.

Talluri, K. T. and van Ryzin, G. J. (2005). *The Theory and Practice of Revenue Management*. Springer.

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.

Todd, L. (2006). Parking management: Strategies, evaluation and planning. Technical report, Victoria Transport Policy Institute.

Todd, L. (2010). Parking pricing implementation guidelines: How more efficient pricing can help solve parking problems, increase revenue, and achieve other planning objectives. Technical report, Victoria Transport Policy Institute.

Todd, L. (2012). Parking management: Comprehensive implementation guide. Technical report, Victoria Transport Policy Institute.

Van Den Eijnden, F. O. (2009). Revenue management at park'n fly.

van Ryzin, G. and Vulcano, G. (2008). Simulation-based optimization of virtual nesting controls for network revenue management. *Operations research*, 56(4):865–880.

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.

Vinod, B. (2004). Unlocking the value of revenue management in the hotel industry. *Journal of Revenue and Pricing Management*, 3(2):178–190.

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.

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.

Weatherford, L. R. and Bodily, S. E. (1992). A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking, and pricing. *Operations Research*, 40(5):831–844.

Weatherford, L. R. and Kimes, S. E. (2003). A comparison of forecasting methods for hotel revenue management. *International Journal of Forecasting*, 19(3):401–415.

Williamson, E. L. (1992). *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts*. PhD thesis, Flight Trasportation and Operations research, Massachusetts Institute of Technology.

Wollmer, R. D. (1986). A hub-spoke seat management model. *Unpublished Internal Report, Mc Donnell Douglas Corporation, Long Beach, CA*.

Wollmer, R. D. (1992). An airline seat management model for a single leg route when lower fare classes book first. *Operations Research*, 40(1):26–37.

Zhang, D. and Weatherford, L. (2012). Dynamic pricing for network revenue management: A new approach and application in the hotel industry.

Zhao, W. (1999). *Dynamic and static yield management models*. PhD thesis, The Wharton School, Operations and Information Management Department, University of Pensylvania, Philadelphia.

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

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

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.

Zukerman, M. (2012). *Introduction to queueing theory and stochastic teletraffic models*. Online, Melbourne, Australia.

# Appendix A

# Derivation of analytical form for $\rho_S(\xi|z)$ when time is stationary.

From the definition of conditional probabilities, we know that

$$\rho_s(\xi|z) = \frac{\rho_s(\xi \cap z)}{g(z)}. \tag{A.1}$$

If $\xi \geq z$, from figure (5.5) we have,

$$\rho_s(\xi \cap z) = \int_0^z \phi(\eta, \xi)\, d\eta. \tag{A.2}$$

If $\xi < z$, from figure (5.5) we have,

$$\rho_s(\xi \cap z) = \int_{z-\xi}^z \phi(\eta, \xi)\, d\eta. \tag{A.3}$$

Let us define $(x)^+$ as,

$$(x)^+ = \begin{cases} x & \text{if} \quad x > 0 \\ 0 & \text{if otherwise} \end{cases}$$

$$= \max\{x, 0\}. \tag{A.4}$$

Thus, for every $\xi \in [0, \infty)$ we have,

$$\begin{aligned}
\rho_s(\xi \cap z) &= \int_{(z-\xi)^+}^{z} \phi(\eta, \xi)\, d\eta \\
&= \int_{(z-\xi)^+}^{z} \sum_n \alpha_n \rho_{a_n}(\eta) \rho_{s_n}(\xi)\, d\eta \\
&= \int_0^z \sum_n \alpha_n \rho_{a_n}(\eta) \rho_{s_n}(\xi)\, d\eta - \int_0^{(z-\xi)^+} \sum_n \alpha_n \rho_{a_n}(\eta) \rho_{s_n}(\xi)\, d\eta \\
&= \sum_n \alpha_n \rho_{s_n}(\xi) \left( \int_0^z \rho_{a_n}(\eta)\, d\eta - \int_0^{(z-\xi)^+} \rho_{a_n}(\eta)\, d\eta \right) \\
&= \sum_n \alpha_n \rho_{s_n}(\xi) \Big( P_{a_n}(z) - P_{a_n}\big((z-\xi)^+\big) \Big). \qquad \text{(A.5)}
\end{aligned}$$

Therefore, the conditional probability density of stay can be expressed analytically as

$$\rho_s(\xi|z) = \sum_n \alpha_n \frac{\rho_{s_n}(\xi) \Big( P_{a_n}(z) - P_{a_n}\left((z-\xi)^+\right) \Big)}{g(z)}. \qquad \text{(A.6)}$$

# Appendix B

# Derivation of analytical form for $P_s(\xi|z)$ when time is stationary.

In a time stationary setting the conditional cdf of stays is given by,

$$P_s(\xi|z) = \int_0^{\xi} \rho_s(\xi'|z)\, d\xi', \tag{B.1}$$

where

$$\rho_s(\xi|z) = \sum_n \alpha_n \frac{\rho_{s_n}(\xi)\Big(P_{a_n}(z) - P_{a_n}((z-\xi)^+)\Big)}{g(z)}. \tag{B.2}$$

is the conditional probability density of staying for $\xi$ days given that the customer is present $z$ days after the booking. In fact, there are two regions of interest according to which $\rho_s(\xi|z)$ changes discontinuously; these are $z \geq \xi$ and $z < \xi$. Let us examine each case separately.

- Case 1: $z \geq \xi$

In this case the conditional probability density becomes

$$\rho_s(\xi|z) = \sum_n \alpha_n \frac{\rho_{s_n}(\xi)\Big(P_{a_n}(z) - P_{a_n}(z-\xi)\Big)}{g(z)}$$

and, therefore, if we integrate this with respect to $\xi$ we obtain

$$
\begin{aligned}
P_s^1(\xi|z) &= \int_0^{\xi} \rho_s(\xi' s|z)\, d\xi', \\
&= \frac{1}{g(z)} \sum_n \alpha_n \left( \frac{\lambda_{s_n}}{\lambda_{a_n} - \lambda_{s_n}} e^{-\lambda_{a_n} z} \left( e^{(\lambda_{a_n} - \lambda_{s_n}) z} - 1 \right) + e^{-\lambda_{a_n} z} \left( e^{-\lambda_{s_n} \xi} - 1 \right) \right)
\end{aligned}
\tag{B.3}
$$

- Case 2: $z < \xi$

This case is a bit more interesting as within the process of integrating from 0 to $\xi$ there will be instances when $z < \xi$ but also other instances when $z \geq \xi$. Since (B.2) behaves differently in either case, before evaluating the integral in (B.2) we should split it in two regions as follows:

$$
\begin{aligned}
P_s^2(\xi|z) &= \int_0^{\xi} \rho_s(\xi'|z)\, d\xi' \\
&= \int_0^{z} \rho_s(s|z)\, ds + \int_z^{\xi} \rho_s(\xi'|z)\, d\xi' \\
&= P_s^1(z|z) + \int_z^{\xi} \rho_s(\xi'|z)\, d\xi'
\end{aligned}
\tag{B.4}
$$

In $P_s^1(z|z)$ the integration variable will run for values being always less than $z$ whereas in i in the second term the integration variable will always run above $z$. In fact, it is the same expression to $P_s^1(\xi|z)$ with $\xi$ being replaced by $z$.

For the second term $\rho_s(\xi|z)$ is given by,

$$
\rho_s(\xi|z) = \sum_n \alpha_n \frac{\rho_{s_n}(\xi) P_{a_n}(z)}{g(z)}
$$

and after integrating from $z$ to $\xi$ we obtain

$$
\int_z^{\xi} \rho_s(\xi'|z)\, d\xi' = \frac{1}{g(z)} \sum_n \alpha_n (e^{-\lambda_{a_n} z} - 1)(e^{-\lambda_{s_n} \xi} - e^{-\lambda_{s_n} z}) \tag{B.5}
$$

In conclusion, the analytical form for $P_s(\xi|z)$ is given by

$$
P_s(\xi|z) = \begin{cases} P_s^1(\xi|z) & \text{if } z < \xi \\[2ex] P_s^1(z|z) + \frac{1}{g(z)} \sum_n \alpha_n (e^{-\lambda_{a_n} z} - 1)(e^{-\lambda_{s_n} \xi} - e^{-\lambda_{s_n} z}) & \text{if } z \geq \xi \end{cases}
$$

# Appendix C

# Calculation of the value matrices $p$ and $r$

Recall that we seek to compute the integrals

$$p_i^k = P_s(i\Delta\xi|k\Delta\tau) = \int_0^{i\Delta\xi} \rho_s(\xi|\tau^k)\,d\xi \tag{C.1}$$

and

$$r_i^k = \int_0^{i\Delta\xi} \rho_s(\xi|\tau^k)\,\Psi(\xi)\,d\xi. \tag{C.2}$$

Notice that equation (C.1) may be written as

$$\begin{aligned} p_i^k &= \int_0^{(i-1)\Delta\xi} \rho_s(\xi|\tau^k)\,d\xi + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\xi|\tau^k)\,d\xi \\ &= p_{i-1}^k + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\xi|\tau^k)\,d\xi. \end{aligned} \tag{C.3}$$

Thus, the next value of $p_i$ can be expressed in terms of the previous value $p_{i-1}$ plus an integral term. Similarly, for equation (C.2) we have

$$\begin{aligned} r_i^k &= \int_0^{(i-1)\Delta\xi} \rho_s(\xi|\tau^k)\,\Psi(\xi)\,d\xi + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\xi|\tau^k)\,\Psi(\xi)\,d\xi \\ &= r_{i-1}^k + \int_{(i-1)\Delta\xi}^{i\Delta\xi} \rho_s(\xi|\tau^k)\,\Psi(\xi)\,d\xi. \end{aligned} \tag{C.4}$$

Again, the next value of $r_i$ may be computed as the previous value $r_{i-1}$ plus an integral term.

Now, if the difference of the two limits $\Delta\xi = \xi^i - \xi^{i-1}$ is sufficiently small then the integral term can be approximated by the *one-step Simpsons Rule*.

Therefore the above formualtions for $p$ and $r$ are powerfull as they enable us to recursively calculate the values $p$ and $r$ based on the previous ones. The recursion begins from the trival case $p_0^k = 0$ and $r_0^k = 0$ for all $k$.

Below is a sample code in C++ that incrementally builds up the matrices using the one-step Simpsons rule.

```
//define the step in time
double T=50;
int K=5000;
double dt=T/K;

//define the step in xi
double Xi_max=50;
int I=5000;
double dxi=Xi_max/I;


for (int t=0;t<Kmax;t++)
{
 for (int s=0;s<Imax;s++)
 {
  if (s==0)
  {
   p[s][t]=0;
   r[s][t]=0;
  }
  else
  {
   //The upper limit of xi is s*dxi
     double upper=s*dxi;

   //The lower limit of xi is (s-1)*dxi
     double lower=(s-1)*dxi;

   //Calculate the half step to use in the Simpsons rule
     double h=(upper-lower)/2.;

     //Find next p value
     //Add previous value plus a single segment simpsons rule
```

```
     p[s][t]= p[s-1][t]+(h/3.)*(conditional_rho_s(lower,t*dt)
                                +4.*conditional_rho_s(lower+h,t*dt)
                                +conditional_rho_s(upper,t*dt)
                                );

    //Find next r value
    //Add previous value plus a single segment simpsons rule
     r[s][t]=
      r[s-1][t]+(h/3.)*(conditional_rho_s(lower,t*dt)*priceRate(lower)
                        +4.*conditional_rho_s(lower+h,t*dt)*priceRate(lower+h)
                        +conditional_rho_s(upper,t*dt)*priceRate(upper)
                       );

  }//end else

 }//end s loop

}//end t loop
```

This code is faster than the conventional method of applying a multistep Simpsons rule for every value, as previous calculated values may be used to evaluate the new ones.