To the Graduate Council:

I am submitting herewith a thesis written by Dale Sikkema entitled "Optimal Profit for Hotel Operation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Julius Smith

Julius Smith, Major Professor

We have read this thesis
and recommend its acceptance:

Carl G. Wagner

Accepted for the Council:

Associate Vice Chancellor and
Dean of the Graduate School

OPTIMAL PROFIT FOR HOTEL OPERATION

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Dale Sikkema
May 1996

DEDICATION

This thesis is dedicated to my wife and son

Mrs. Charlene Sikkema

and

Dale Sikkema Jr.

who have given me emotional support, without which I never would have finished.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all of the people who helped make my graduate work at the University of Tennessee a great success. These include the faculty with whom I have been associated, my fellow graduate students, and the staff of the Department of Mathematics. Special thanks go to my Thesis Committee, Dr. Balram S. Rajput, Dr. Carl G. Wagner, and the Director of my Thesis, Dr. Julius Smith. Their help in learning the material I needed to know and suggestions on how to proceed were invaluable.

Other people I would like to thank include Pam Armentrout for her help in everything from applying to Graduate school through filing the correct graduation paperwork, Dr. William R. Wade for his support and the Bible study he gave, and Rick Moran for patient help in showing me how to use the computer system. Also, I would like to thank the Department of Mathematics as a whole for supporting me financially as a Graduate Teaching Associate.

ABSTRACT

This study deals with finding an optimal rule for accepting customers at a hotel through the course of a day. Previous endeavors have developed heuristic methods for approximating the optimal rule. Found here are the optimal rules for the cases with no reservations allowed and either a single room type or multiple room types. A Dynamic Programming method is used to generate the optimal rules. There does not seem to be any reason why the method used here could not be extended to include the possibility of reservations for some of the customers. Also, this approach may be effective in solving the more general problem of considering multiple night stays. Finally, the approach used here can be applied to similar problems in other industries such as the Airline industry.

TABLE OF CONTENTS

INTRODUCTION

The problem considered here is that of specifying an optimal rule, based on the current time and remaining capacity of a hotel, that dictates whether to accept or decline a customer's request for a room. What is meant by an optimal rule is a rule that, when followed, results in the maximum expected profit. In order to accomplish this, it must first be specified what assumptions are made about the problem and what considerations are taken to find the profit.

Assumptions made about the problem are as follows. First, reservations for any customer are not considered. To begin with, only hotels with one type of room are considered, and later results will be extended to the case of multiple room types. Multiple types of customers are considered. Each customer type is assumed to arrive independently from the others, each by a non-homogeneous Poisson process. The amount of time required to process a customers request, i.e. the time it takes for the customer to check in, is not taken into consideration. Finally, only requests for single night stays are considered.

Customers are divided into different types based on the profit that the hotel will receive from the customer and the loss the hotel will incur for rejecting the customer's request. Any two customers that have the same potential profit and loss associated with them are of the same type. The profit the hotel will receive from a customer is the price the customer will pay for the room minus the operating costs, with some customers willing to pay more than others. The cost the hotel will incur for rejecting a customer must be measured in monetary units, which can be hard to do practically, and this is based both on monetary compensation given to the customer and loss of good will from the customer. Thus, the total profit the hotel receives from a particular customer is either the profit for accepting if the customer is accepted or minus the cost for rejecting if the customer is rejected.

Problems like this have been examined in several variations (Weatherford and Bodily, 1992, and Bitran and Mondschein, 1995). Bitran and Mondschein have the most advanced study to date. In their paper they look at the cases here as well as multiple night stays and reservations, however they resort to heuristic methods of solving the problem. They claim that for the simplest case, the recursive formula that they assert along with properties that characterize this formula form a tractable dynamic programming problem (1995). It will be shown here how the formula asserted by Bitran and Mondschein arises from the dynamic programming algorithm. It will then be shown how this formula can be used to actually solve the problem and find the optimal rule for accepting and rejecting customers in both the single and multiple product case.

The remainder of the paper will be divided into the following chapters. Chapter one will explore the problem in continuous time. Chapter two will discuss the dynamic programming algorithm. Chapter three will use the dynamic programming algorithm to derive a formula for maximum expected profit in the single room type case. Chapter four will summarize results that Bitran and Mondschein derived for the single room type case. Chapter five will use a right to left technique to describe an algorithm that will specify the various acceptance thresholds that define the optimal rule in the case of only one room type. Chapter six will generalize the problem to multiple room types and explore a similar solution to the single room type case. Chapter seven will conclude by pointing out areas that still require exploration.

CHAPTER ONE

CONTINUOUS TIME FORMULATION

The first goal of this paper is to describe the continuous time formulation of the problem. To do this, first define the real and random variables involved in the formulation as follows:

- $\mathcal{T}$:= [0,T] is the planing horizon.
- C is the total number of hotel rooms.
- $\mathcal{C}$:= {0, 1, 2, ..., C} is the set of possible capacities.
- s is the total number of customer types.
- $\mathcal{S}$:= {1, 2, ..., s} is the set of customer types.
- Any function BP : $\mathcal{T}\times\mathcal{C}\times\mathcal{S}\rightarrow$ {0,1} with BP(t,0,i) = 0 for all t,i in the domain is a booking policy.
- $\mathcal{BP}$:= {BP| BP is a booking policy}.
- π_i is the profit for accepting a type i customer.
- χ_i is the cost for rejecting a type i customer.
- $N_i(t)$ is a non-homogeneous Poisson process with parameter $\lambda_i(t)$ counting the arrivals of type i customers.
- $N(t) = N_1(t) + N_2(t) + \ldots + N_s(t)$.
- $Y_k(t)$ is the arrival time of the k^{th} customer, regardless of type.

- $Z_k(t)$ is the type of the k^{th} customer.

$BP(t,c,i) = 0$ means reject the customer and $BP(t,c,i) = 1$ means accept the customer, hence the term booking policy. $BP(t,0,i)$ must be 0 since the hotel cannot accept a customer if they have no vacancy. Thus, the profit made from a type i customer is $g(t,c,i) = [BP(t,c,i)][\pi_i + \chi_i] - \chi_i$ at time t and capacity c. Obviously, if $BP(t,c,i) = 0$, $g(t,c,i) = -\chi_i$, and otherwise, $g(t,c,i) = \pi_i$.

Since $N_i(t)$ is a non-homogeneous Poisson process with parameter $\lambda_i(t)$ counting the arrivals of type i customers for $i = 1, 2, \ldots, s$; $N(t)$ is a non-homogeneous Poisson process with parameter $\lambda(t) := \lambda_1(t) + \lambda_2(t) + \ldots + \lambda_s(t)$ counting the arrivals of all customers. $Y_k(t)$ is the sum of the wait times of the random variable $N(t)$. If $\lambda(t) = \lambda$ for all t, then the $Y_k(t)$'s have gamma distributions with parameters λ and k. Finally, $\Pr\{Z_k(t) = i\} = \lambda_i(t)/\lambda(t)$, for any k.

Define $F^* : \mathscr{T} \times \mathscr{C} \times \mathscr{BP} \to \mathscr{R}$ to be the profit from time t until time T with capacity c while applying booking policy BP. In order to give a formulation of F^*, first define

$$q(c,k) = c - \sum_{j=1}^{k-1} BP(Y_j(t), q(c,j), Z_j(t)),$$

where $q(c,1) = c$. With this,

$$F^*(t,c,BP) = \sum_{k=N(t)+1}^{N(T)} g\left(Y_k(t), q(c,k), Z_k(t)\right)$$

With this, finding the optimal rule would require calculating $\max_{BP} E[F^*(0,C,BP)]$. This expected value is with respect to three random variables. Instead of taking this approach, this paper will consider a discrete version of the problem where the time interval [0, T] is broken into equal pieces, each of which can have at most one customer arrival. This is a standard approach (Bitran and Mondschein, 1995).

CHAPTER TWO

THE DYNAMIC PROGRAMMING ALGORITHM

This paper will make use of the dynamic programming algorithm (DP) later. Here, the DP algorithm is explained. DP is an algorithm that has a discrete-time dynamic system and optimizes the expected value of an additive function. The system is of the form $x_{k+1} = f_k(x_k, u_k, w_k)$, $k = 0, 1, \ldots, N - 1$, where

- k indexes discrete time.
- x_k is the state of the system.
- u_k is the decision variable to be selected at time k.
- w_k is a random parameter.
- N is the number of times the control is applied.

The additive function is $\sum_{k=0}^{N} g_k(x_k, u_k, w_k)$, where g_k is the value incurred at each time k. Since g_k depends on a random variable w_k, optimization takes place on the expected value of the sum.

g_k is usually thought of as a cost or a profit function, and optimizing the expected value means to minimize or maximize the expected value, respectively. The rest of this chapter will refer only to maximizing the profit since this is what this paper will deal with. In order to explain how to maximize the profit, Bellman's principle of optimality must be introduced.

The principle of optimality says that if $\{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ is a rule that maximizes the expected profit from time 0 to time N, then the rule $\{\mu_i, \mu_{i+1}, \ldots, \mu_{N-1}\}$ maximizes the expected profit from time i to time N. Each component of this rule is a function of the state variable and the random variable that specifies what decision is to be made at that time, i.e. $\mu_k (x_k, w_k) = u_k^*$, where u_k^* is the optimal decision at time k. A common analogy used to explain this is that of finding the shortest route between two cities. If the shortest route between city A and city C passes through city B, then the shortest route between city B and city C is the B to C leg of the shortest route from A to C. This is intuitively clear in the example.

Now, it will be shown how the DP algorithm solves the maximization problem using the principle of optimality through a right to left technique. To begin with, DP first considers the problem only at the last time period, time N. At this point, $\mu_N(x_N, w_N) = u_N^*$ is calculated to maximize $E[g_N(x_N, u_N, w_N)]$, i.e.

$$E[g_N(x_N, u_N^*, w_N)] \geq E[g_N(x_N, u_N, w_N)]$$

for all possible values of u_N. Define $J_N (x_N) := E[g_N(x_N, u_N^*, w_N)]$. J_N is the maximal expected profit from time N to time N. Finding this maximal expected value is much simpler than finding the entire maximal rule at once. Next, DP considers one step back in time to time N-1. Going backwards in time from the end to the beginning is what is meant by a right to left technique. At time N-1, DP calculates the component $\mu_{N-1}(x_{N-1}, w_{N-1}) = u_{N-1}^*$ to find

$$\max \{E[g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})] + E[J_N(x_N)]\}. \quad (1)$$

It should be noted that the expectations in (1) are with respect to w_{N-1}. Since it is known J_N will be maximal, the principle of optimality tells us that (1) is the maximal expected profit from time N-1 to time N. Remembering that $x_{k+1} = f_k(x_k, u_k, w_k)$, a substitution gives that u_{N-1}^* solves the problem

$$\max \{E[g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + J_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}))]\}.$$

Now, define

$$J_{N-1}(x_{N-1}) := E[g_{N-1}(x_{N-1}, u_{N-1}^*, w_{N-1}) + J_N(f_{N-1}(x_{N-1}, u_{N-1}^*, w_{N-1}))].$$

DP continues this process back in time where component $\mu_k(x_k, w_k) = u_k^*$ solves

$$\max \{E[g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k))]\} \text{ and}$$

$$J_k(x_k) := E[g_k(x_k, u_k^*, w_k) + J_{k+1}(f_k(x_k, u_k^*, w_k))],$$

where the expectation is with respect to w_k. By the principle of optimality, each J_k is optimal from time k to time N since each J_{k+1} is optimal. Finally, when time is 1, $J_1(x_1)$ is the maximum expected profit for k = 1 to N, and the optimal rule $\{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ is calculated along the way. Each of the individual maximization problems for finding components of the optimal rule is much simpler than the single maximization problem for finding the entire rule. Thus, the DP algorithm can take an ugly maximization problem and turn it in to a sequence of much easier maximization problems.

Here are some final notes about DP in general. Often times, the additive formula to be optimized will contain some final non-random exit value that must be taken into account when finding the optimal rule (Bertsekas, 1987). This was not included in the above discussion since it does not play a role in the problem explored here. Also, the actual optimization at each step might be done either analytically or numerically. If the optimization can be done analytically, a closed form expression for the optimal expected profit and optimal rule can sometimes be found. This is the best case scenario. If numerical methods must be applied, problems of too great a magnitude are often times still not tractable since the computer time and memory needed to execute the DP algorithm can be too great to be of any practical use (Bertsekas, 1987).

CHAPTER THREE

FORMULA FOR MAXIMUM EXPECTED PROFIT

Bitran and Mondschein consider this problem in a discrete time setting with time intervals small enough so that the number of customers that arrive in one time interval is either zero or one (1995). In their paper, the following formula is given as the maximum expected profit form period t onwards, given that a type i customer arrives at time t:

$$F_t(c,i) = \max\begin{cases} \pi_i + E_j[F_{t+1}(c-1,j)] \\ -\chi_i + E_j[F_{t+1}(c,j)] \end{cases} \text{if } c \cdot i \neq 0 . \qquad (2)$$

Boundary conditions for this formula are

$$F_t(c, i) = -\chi_i + E_j[F_{t+1}(c, j)] \text{ if } c \cdot i = 0 . \qquad (2i)$$

$$F_T(c, i) = \pi_i \text{ if } c \cdot i \neq 0 \text{ and } F_T(c, i) = -\chi_i \text{ if } c \cdot i = 0. \qquad (2ii)$$

The following are variable definitions for the above formulas.

- t is the time since the beginning of the planning horizon.
- c is the hotel's current capacity.
- i is the type of customer requesting a room, where i = 0 means no customer.
- π_i is the profit associated with accepting a type i customer, with $\pi_0 = 0$.
- χ_i is the loss associated with rejecting a type i customer, with $\chi_0 = 0$.
- j is a random variable, where j = i iff the next customer is of type i.

- E_j is the expected value of a function with respect to the random variable j.

This chapter shows how the formula above is derived from the DP algorithm.

Now, define the variables and functions used by DP as they exist in this problem as follows.

- t indexes discrete time, ranging from 0 (beginning of the planing horizon) to T (end of the planing horizon).
- The state variable x_t represents the capacity of the hotel at time t.
- The random variable w_t represents the type of the customer that arrives at time t.
- The decision variable u_t represents the decision made at time t. If the customer is rejected, $u_t = 0$. If the customer is accepted, $u_t = 1$. Restraints put on u_t are that if $w_t = 0$ (no customer arrives at time t) or $x_t = 0$ (no vacancy) then $u_t = 0$.
- T is the number of times the control is applied.
- At time t, $x_{t+1} = f_t(x_t, u_t, w_t) = x_t - u_t$.
- At time t, $g_t(x_t, u_t, w_t) = u_t \cdot (\pi_{w_t} + \chi_{w_t}) - \chi_{w_t}$. If $u_t = 0$ then $g_t(x_t, u_t, w_t) = -\chi_{w_t}$, and if $u_t = 1$ then $g_t(x_t, u_t, w_t) = \pi_{w_t}$,. This is the profit incurred at time t.

The DP algorithm begins at time T, so

$$J_T(x_T) = \max\left\{E_{w_T}\left[u_T \cdot (\pi_{w_T} + \chi_{w_T}) - \chi_{w_T}\right]\right\}.$$

If $x_T = 0$ then $J_T(x_T) = E_{w_T}\left[-\chi_{w_T}\right]$, and if $x_T > 0$ then $J_T(x_T) = E_{w_T}\left[\pi_{w_T}\right]$. Clearly, this is the expected value of the boundary condition (2ii), so

$J_T(x_T) = E_{w_T}\left[F_T(x_T, w_T)\right]$. At time T-1,

$$J_{T-1}(x_{T-1}) = \max\left\{E_{w_{T-1}}\left[u_{T-1}\cdot\left(\pi_{w_{T-1}} + \chi_{w_{T-1}}\right) - \chi_{w_{T-1}} + J_T(x_{T-1} - u_{T-1})\right]\right\}.$$

If $x_{T-1} \neq 0$ then there are two choices for u_{T-1}, so

$$J_{T-1}(x_{T-1}) = \max\begin{cases} E_{w_{T-1}}\left[\pi_{w_{T-1}} + E_{w_T}\left[F_T(x_{T-1} - 1, w_T)\right]\right] \\ E_{w_{T-1}}\left[-\chi_{w_{T-1}} + E_{w_T}\left[F_T(x_{T-1}, w_T)\right]\right] \end{cases}$$

$$= E_{w_{T-1}}\left[\max\begin{cases} \pi_{w_{T-1}} + E_{w_T}\left[F_T(x_{T-1} - 1, w_T)\right] \\ -\chi_{w_{T-1}} + E_{w_T}\left[F_T(x_{T-1}, w_T)\right] \end{cases}\right] = E_{w_{T-1}}\left[F_{T-1}(x_{T-1}, w_{T-1})\right].$$

If $x_{T-1} = 0$, then $u_{T-1} = 0$, so

$$J_{T-1}(x_{T-1}) = E_{w_{T-1}}\left[-\chi_{w_{T-1}} + E_{w_T}\left[F_T(x_{T-1}, w_T)\right]\right] = E_{w_{T-1}}\left[F_{T-1}(x_{T-1}, w_{T-1})\right].$$

Continuing this right to left method, it is found that

$$\text{if } x_t \neq 0,\ J_t(x_t) = E_{w_t}\left[\max\begin{cases} \pi_{w_t} + E_{w_{t-1}}\left[F_{t+1}(x_t - 1, w_{t+1})\right] \\ -\chi_{w_t} + E_{w_{t-1}}\left[F_{t+1}(x_t, w_{t+1})\right] \end{cases}\right] = E_{w_t}\left[F_t(x_t, w_t)\right], \text{ and}$$

$$\text{if } x_t = 0,\ J_t(x_t) = E_{w_t}\left[-\chi_{w_t} + E_{w_{t-1}}\left[F_{t+1}(x_t, w_{t+1})\right]\right] = E_{w_t}\left[F_t(x_t, w_t)\right].$$

Now, to match the notation used by Bitran and Mondschein exactly, simply replace the state variable x_t with c, and replace the random variable w_t with the random variable j. Formula (2) is found to be the formula such that when the expected value of (2) is taken, the result of DP is found.

CHAPTER FOUR

SUMMARY OF RESULTS

This chapter summarizes results found by Bitran and Mondschein on this problem. From the formula above, a type i customer is accepted at time t and capacity c iff

$\pi_i + E_j[F_{t+1}(c - 1, j)] \geq -\chi_i + E_j[F_{t+1}(c, j)]$, which is equivalent to

$\pi_i + \chi_i \geq E_j[F_{t+1}(c, j)] - E_j[F_{t+1}(c - 1, j)]$.

Defining $\alpha_t(c) := E_j[F_{t+1}(c, j)] - E_j[F_{t+1}(c - 1, j)]$, it follows that a type i customer is accepted at time t and capacity c iff $\pi_i + \chi_i \geq \alpha_t(c)$ and a type i customer is rejected at time t and capacity c iff $\pi_i + \chi_i < \alpha_t(c)$. The following lemmas and propositions are given concerning the functions F and α. The proofs were given by Bitran and Mondschein (1995).

Lemma 1: *The function $F_t(c, i)$ is monotonically increasing as a function of the capacity:*

$F_t(c, i) \geq F_t(c-1, i)$, *for all c, t, i.*

Proof: Consider the optimal policy for the problem starting at time t and capacity c-1. This policy can be obeyed at time t when the capacity is c and it gives the same value (see appendix A) , call it X, of the objective function in either case.

However, this will not necessarily be the optimal policy for capacity c. Thus, the value of the objective function at time t with capacity c using the actual optimal policy for this case cannot be less than X, so

$$F_t(c, i) \geq F_t(c-1, i), \text{ for all } c, t, i \blacksquare$$

Lemma 2: *The objective function satisfies the inequality:*

$$F_t(c+2, i) - F_t(c+1, i) \leq F_t(c+1, i) - F_t(c,i) \textit{ for all } t, i, c. \qquad (3)$$

Proof: The proof is by induction on t. For t = T,

$$F_T(c+2, i) - F_T(c+1, i) = 0 \text{ for any values of c and i.}$$

If $i = 0$ or $c \neq 0$ then

$$F_T(c+1, i) - F_T(c,i) = 0,$$

but if $i \neq 0$ and $c = 0$ then

$$F_T(c+1, i) - F_T(c,i) = \pi_i + \chi_i > 0.$$

In any case, (3) holds for time T. Assume (3) holds for some $t+1 \leq T$, and consider the objective function at time t. Assume $i \neq 0$. For capacity $c \neq 0$ at time t, the objective function has the value of either

(a) $\pi_i + E_j[F_{t+1}(c-1, j)]$, or

(b) $-\chi_i + E_j[F_{t+1}(c, j)]$.

For capacity c+1 at time t, the objective function has the value of either

(c) $\pi_i + E_j[F_{t+1}(c, j)]$, or

(d) $-\chi_i + E_j[F_{t+1}(c+1, j)]$.

For capacity c+2 at time t, the objective function has the value of either

(e) $\pi_i + E_j[F_{t+1}(c+1, j)]$, or

(f) $-\chi_i + E_j[F_{t+1}(c+2, j)]$.

To prove that (3) holds, it must be shown that every possible combination of the choosing one value from (a) & (b), one value from (c) & (d), and one value from (e) & (f) satisfies (3).

(a), (c), (e): this combination substituted into the inequality gives

$$\pi_i + E_j[F_{t+1}(c+1,j)] - \pi_i - E_j[F_{t+1}(c,j)] \leq \pi_i + E_j[F_{t+1}(c,j)] - \pi_i - E_j[F_{t+1}(c-1,j)] \text{ iff}$$

$$E_j[F_{t+1}(c+1,j)] - E_j[F_{t+1}(c,j)] \leq E_j[F_{t+1}(c,j)] - E_j[F_{t+1}(c-1,j)] \text{ iff}$$

$$E_j[F_{t+1}(c+1,j) - F_{t+1}(c,j)] \leq E_j[F_{t+1}(c,j) - F_{t+1}(c-1,j)].$$

Since the probability of a type-i arrival in the next unit of time depends only on t, this latest inequality is implied by the induction hypothesis, and so this combination satisfies (3). The same proof holds for combinations (b), (c), (e); (b), (d), (e); and (b), (d), (f).

(a), (d), (e): In this case, a type i customer is accepted at capacity c and rejected at capacity c+1. This gives that

$$\pi_i + \chi_i \geq E_j[F_{t+1}(c,j)] - E_j[F_{t+1}(c-1,j)], \text{ and}$$

$$\pi_i + \chi_i < E_j[F_{t+1}(c+1,j)] - E_j[F_{t+1}(c,j)].$$

Combining these two inequalities gives

$$E_j[F_{t+1}(c+1,j)] - E_j[F_{t+1}(c,j)] > E_j[F_{t+1}(c,j)] - E_j[F_{t+1}(c-1,j)] \text{ iff}$$

$$E_j[F_{t+1}(c+1,j) - F_{t+1}(c,j)] > E_j[F_{t+1}(c,j) - F_{t+1}(c-1,j)].$$

This latest inequality contradicts a consequence of the induction hypothesis, and so this combination is infeasible, along with combinations (a), (d), (f); (a), (c), (f); and (b), (c), (f). The same type of proof works for c = 0 except there is only one possible value for the objective function when the capacity is 0. Finally, if i = 0, the values of the objective function substituted into the inequality gives

$$E_j[F_{t+1}(c+2,j)] - E_j[F_{t+1}(c+1,j)] \le E_j[F_{t+1}(c+1,j)] - E_j[F_{t+1}(c,j)],$$

which, as has been seen, is implied by the induction hypothesis ■

Proposition 1: The function $\alpha_t(c)$ is a nonincreasing function of c:

$$\alpha_t(c-1) \ge \alpha_t(c).$$

Proof: By lemma 2,

$$F_{t+1}(c-1, j) - F_{t+1}(c-2, j) \ge F_{t+1}(c, j) - F_{t+1}(c-1, j) \text{ for all } t, j, c.$$

Since the probability of a type-i arrival in the next unit of time depends only on t, taking the expected value of each side of the inequality still holds, so

$$E_j[F_{t+1}(c-1, j)] - E_j[F_{t+1}(c-2, j)] \ge E_j[F_{t+1}(c, j)] - E_j[F_{t+1}(c-1, j)],$$

which is precisely $\alpha_t(c-1) \ge \alpha_t(c)$ ■

Proposition 2: *The function $\alpha_t(c)$ is decreasing in t:*

$$\alpha_t(c) \ge \alpha_{t+1}(c).$$

Proof: First note that $\alpha_t(c)$ is only defined for c > 0.

For i ≠ 0, consider the following cases.

Case 1, c = 1: consider the difference

$$F_{t+1}(1,i) - F_{t+1}(0,i) = \max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(0,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(1,j)\right]\end{matrix}\right\} - \left(-\chi_i + E_j\left[F_{t+2}(0,j)\right]\right)$$

$\geq -\chi_i + E_j\left[F_{t+2}(1,j)\right] - \left(-\chi_i + E_j\left[F_{t+2}(0,j)\right]\right) = E_j\left[F_{t+2}(1,j)\right] - E_j\left[F_{t+2}(0,j)\right]$, and so

$F_{t+1}(1,i) - F_{t+1}(0,i) \geq E_j\left[F_{t+2}(1,j)\right] - E_j\left[F_{t+2}(0,j)\right] = \alpha_{t+1}(1)$.

Case 2, c > 1: consider the difference

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) = \max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-1,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c,j)\right]\end{matrix}\right\} - \max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-2,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c-1,j)\right]\end{matrix}\right\}$$

subcase a: assume $\max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-2,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c-1,j)\right]\end{matrix}\right\} = -\chi_i + E_j\left[F_{t+2}(c-1,j)\right]$.

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) = \max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-1,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c,j)\right]\end{matrix}\right\} - \left(-\chi_i + E_j\left[F_{t+2}(c-1,j)\right]\right)$$

$\geq -\chi_i + E_j\left[F_{t+2}(c,j)\right] - \left(-\chi_i + E_j\left[F_{t+2}(c-1,j)\right]\right) = E_j\left[F_{t+2}(c,j)\right] - E_j\left[F_{t+2}(c-1,j)\right]$, so

$F_{t+1}(c,i) - F_{t+1}(c-1,i) \geq E_j\left[F_{t+2}(c,j)\right] - E_j\left[F_{t+2}(c-1,j)\right] = \alpha_{t+1}(c)$.

subcase b: assume $\max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-2,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c-1,j)\right]\end{matrix}\right\} = \pi_i + E_j\left[F_{t+2}(c-2,j)\right]$

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) = \max\left\{\begin{matrix} \pi_i + E_j\left[F_{t+2}(c-1,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c,j)\right]\end{matrix}\right\} - \left(\pi_i + E_j\left[F_{t+2}(c-2,j)\right]\right)$$

$\geq \pi_i + E_j\left[F_{t+2}(c-1,j)\right] - \left(\pi_i + E_j\left[F_{t+2}(c-2,j)\right]\right) = E_j\left[F_{t+2}(c-1,j)\right] - E_j\left[F_{t+2}(c-2,j)\right]$,

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) \geq E_j\left[F_{t+2}(c-1,j)\right] - E_j\left[F_{t+2}(c-2,j)\right] = \alpha_{t+1}(c-1).$$

By proposition 1, $\alpha_{t+1}(c-1) \geq \alpha_{t+1}(c)$, and thus

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) \geq \alpha_{t+1}(c).$$

Hence, in all cases and subcases, the following inequality holds:

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) \geq \alpha_{t+1}(c) \text{ for } i \neq 0.$$

Also, $F_{t+1}(c,0) - F_{t+1}(c-1,0) = \alpha_{t+1}(c)$, so

$$F_{t+1}(c,i) - F_{t+1}(c-1,i) \geq \alpha_{t+1}(c) \text{ for all } i. \qquad (4)$$

As has already been seen,

$$E_j\left[F_{t+1}(c,j) - F_{t+1}(c-1,j)\right] = E_j\left[F_{t+1}(c,j)\right] - E_j\left[F_{t+1}(c-1,j)\right] = \alpha_t(c),$$

so noting that the right hand side of (4) is constant, taking the expected value of both sides of (4) gives the inequality

$$\alpha_t(c) \geq \alpha_{t+1}(c) \blacksquare$$

In proving these results, Bitran and Mondschein have in some cases proven more general results than appear here. The above proofs only refer to the specific case that is dealt with here, and so some of the proofs only use the same techniques and not the actual proofs given by Bitran and Mondschein.

With these results, the following characteristics of the optimal policy can be found. Proposition one gives that for each instant t, there is a capacity threshold C^*_{it}, where a type i customer is accepted iff the capacity is greater than or equal to C^*_{it}. This threshold is found to be $C^*_{it} = \min\{c | \pi_i + \chi_i \geq \alpha_t(c)\}$. If the customers are ordered such that $\pi_i + \chi_i \leq \pi_k + \chi_k, \forall i \leq k$ then the sequence $\left(C^*_{it}\right)_{0 \leq i \leq s}$ is decreasing in i. This means that if it is optimal to accept a type i customer, then it is optimal to accept a type k customer for all k > i.

Proposition two gives that for each capacity level c, there is a time threshold τ^*_{ic}, where a type i customer is accepted iff the time is at least τ^*_{ic}. The sequence $\left(\tau^*_{ic}\right)_{0 \leq i \leq s}$ is decreasing in i, so the hotel rents to higher valued customers before it rents to lower valued customers. These characteristics are to be expected in the optimal policy.

CHAPTER FIVE

DIFFERENCE EQUATION

This chapter will begin by analyzing the expected profit associated with the set of acceptance policies that obey the *optimal property*. An acceptance policy obeys the optimal property if for any customer type i that is accepted, a type k customer is accepted also for all k > i, where the customers are ordered such that $\pi_i + \chi_i \leq \pi_k + \chi_k, \forall i \leq k$. The previous chapter proved that the optimal policy does obey the optimal property. The following are the new definitions needed to analyze the expected profit using a general acceptance policy that obeys the optimal property.

- An acceptance policy is a function of time, capacity, and customer type that returns a 1 if the customer is accepted and a 0 if the customer is rejected, with the restriction that at capacity 0 or for customer type 0, the customer is always rejected.
- Z:= {acceptance policies that obey the optimal property}.
- $z^* \in A$ is the optimal policy.

- $f_T^z(c,i) := \begin{cases} \pi_i \text{ if customer is accepted} \\ -\chi_i \text{ if customer is rejected} \end{cases}$, where $z \in A$.

  $f_t^z(c,i) := \begin{cases} \pi_i + E_j\left[f_{t+1}^z(c-1,j)\right] \text{if customer is accepted} \\ -\chi_i + E_j\left[f_{t+1}^z(c,j)\right] \text{if customer is rejected} \end{cases}$, for $0 \le t \le T-1$.

- $p_j(t)$ is probability that a type j customer will arrive at time T-t.

- $o_j := \pi_j + \chi_j$.

- $E_{t,c}^z := E_j\left[f_{T-t}^z(c,j)\right]$.

- $K_t := \sum_{j=0}^{s} \left(p_j(t) \cdot \chi_j\right)$.

- $R_t := \sum_{j=0}^{s} \left(p_j(t) \cdot \pi_j\right)$.

- $m_{t,c}^z := \min\{i \mid \text{customer type } i \text{ is accepted by policy } z \text{ at time } T-t \text{ \& capacity } c\}$

  where $m_{t,c}^z = s+1$ if no customer type is accepted. Note that $m_{t,c}^z \ge 1$ since a type 0 customer is never accepted.

- $a_{t,c}^z := \sum_{j=0}^{m_{t,c}^z - 1} p_j(t)$.

- $b_{c,t}^z := \sum_{j=m_{t,c}^z}^{s} \left(p_j(t) \cdot o_j\right)$.

Clearly, $F_t(c,i) = f_t^{z^*}(c,i)$, and so any result found for a general $z \in Z$ will hold for the optimal policy. The advantage of this general approach is that the relationship can be explored between certain suboptimal heuristics that can be applied to this problem and the optimal solution since the heuristic approaches are developed to obey the optimal property.

The next objective here is to develop a recursive formula for the expected profit from time T-t until time T starting with a capacity of c and using policy $z \in Z$, i.e., a recursive formula for $E^z_{t,c}$. This will use a right to left technique in time and examine the behavior of $E^z_{t,c}$ for c = 0, 1, 2, ..., C. It is assumed here that at each time T-t and capacity c, $m^z_{t,c}$ is known. Thus, this technique works if one can calculate $m^z_{t,c}$ at time t with only knowledge of $E^z_{t_0,c}$ for $t_0 < t$, or if $m^z_{t,c}$ is deterministic.

For c = 0:

Keep in mind that any policy must reject all customers when c is 0. This gives:

$$f^z_T(0,i) = -\chi_i \text{, so}$$

$$E^z_{0,0} = \sum_{j=0}^{s}\left(p_j(0)\cdot(-\chi_j)\right) = -K_0 .$$

$$f^z_{T-1}(0,i) = -\chi_i + E^z_{0,0} = -\chi_i - K_0 \text{, so}$$

$$E^z_{1,0} = \sum_{j=0}^{s}\left(p_j(1)\cdot\left(f^z_{T-1}(0,j)\right)\right) = \sum_{j=0}^{s}\left(p_j(1)\cdot(-\chi_j - K_0)\right) = -K_1 - K_0 = -(K_0 + K_1)$$

$$f^z_{T-2}(0,i) = -\chi_i + E^z_{1,0} = -\chi_i - (K_0 + K_1) \text{, so}$$

$$E^z_{2,0} = \sum_{j=0}^{s}\left(p_j(2)\cdot\left(f^z_{T-2}(0,j)\right)\right) = \sum_{j=0}^{s}\left(p_j(2)\cdot\left(-\chi_j-(K_0+K_1)\right)\right) = -(K_0+K_1+K_2).$$

From here, it seems obvious that

$$E^z_{t,0} = -\sum_{d=0}^{t} K_d. \qquad (5)$$

This will be proved by induction on t. It has been shown above that (5) is true for $t = 0$. Assume that for some $t \geq 0$, (5) is true. By the induction hypothesis,

$$f^z_{T-t-1}(0,i) = -\chi_i + E^z_{t,0} = -\chi_i - \left(\sum_{d=0}^{t} K_d\right), \text{ so}$$

$$E^z_{t+1,0} = \sum_{j=0}^{s}\left(p_j(t+1)\cdot\left(f^z_{T-t-1}(0,j)\right)\right) = \sum_{j=0}^{s}\left(p_j(t+1)\cdot\left(-\chi_j - \left(\sum_{d=0}^{t} K_d\right)\right)\right)$$

$$= -K_{t+1} - \sum_{d=0}^{t} K_d = -\sum_{d=0}^{t+1} K_d, \text{ so}$$

$$E^z_{t+1,0} = -\sum_{d=0}^{t+1} K_d.$$

Thus, since $E^z_{0,0} = -K_0$ and $E^z_{t,0} = -\sum_{d=0}^{t} K_d \Rightarrow E^z_{t+1,0} = -\sum_{d=0}^{t+1} K_d$, by induction,

$$E^z_{t,0} = -\sum_{d=0}^{t} K_d \text{ for all t.}$$

For c = 1:

The policy here is not necessarily to reject everyone since there is one room left.

$$f^z_T(1,i) = \begin{cases} \pi_i \text{ if customer is accepted} \\ -\chi_i \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{0,1} = \sum_{j=0}^{s}\left(p_j(0)\cdot f^z_T(1,j)\right) = \sum_{j=0}^{m^z_{0,1}-1}\left(p_j(0)\cdot\left(-\chi_j\right)\right) + \sum_{j=m^z_{0,1}}^{s}\left(p_j(0)\cdot\pi_j\right)$$

$$= -\sum_{j=0}^{s}\left(p_j(0)\cdot\chi_j\right) + \sum_{j=m^z_{0,1}}^{s}\left(p_j(0)\cdot\chi_j\right) + \sum_{j=m^z_{0,1}}^{s}\left(p_j(0)\cdot\pi_j\right)$$

$$= -K_0 + \sum_{j=m^z_{0,1}}^{s}\left(p_j(t)\cdot o_j\right) = -K_0 + b^z_{0,1}$$

$$f^z_{T-1}(1,i) = \begin{cases} \pi_i + E^z_{0,0} \text{ if customer is accepted} \\ -\chi_i + E^z_{0,1} \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{1,1} = \sum_{j=0}^{m^z_{1,1}-1}\left(p_j(1)\cdot\left(-\chi_j + E^z_{0,1}\right)\right) + \sum_{j=m^z_{1,1}}^{s}\left(p_j(1)\cdot\left(\pi_j + E^z_{0,0}\right)\right)$$

$$= -K_1 + \sum_{j=m^z_{1,1}}^{s}\left(p_j(1)\cdot\chi_j\right) + a^z_{1,1}\cdot E^z_{0,1} + \sum_{j=m^z_{1,1}}^{s}\left(p_j(1)\cdot\pi_j\right) + \left(1-a^z_{1,1}\right)\cdot E^z_{0,0}$$

$$= -K_1 + a^z_{1,1}\cdot E^z_{0,1} + \left(1-a^z_{1,1}\right)\cdot E^z_{0,0} + \sum_{j=m^z_{1,1}}^{s}\left(p_j(1)\cdot o_j\right)$$

$$= -K_1 + a^z_{1,1}\cdot E^z_{0,1} + \left(1-a^z_{1,1}\right)\cdot E^z_{0,0} + b^z_{1,1}\,.$$

$$f^z_{T-2}(1,i) = \begin{cases} \pi_i + E^z_{1,0} \text{ if customer is accepted} \\ -\chi_i + E^z_{1,1} \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{2,1} = \sum_{j=0}^{m^z_{2,1}-1}\left(p_j(2)\cdot\left(-\chi_j + E^z_{1,1}\right)\right) + \sum_{j=m^z_{2,1}}^{s}\left(p_j(2)\cdot\left(\pi_j + E^z_{1,0}\right)\right)$$

$$= -K_2 + \sum_{j=m^z_{2,1}}^{s}\left(p_j(2)\cdot\chi_j\right) + a^z_{2,1}\cdot E^z_{1,1} + \sum_{j=m^z_{2,1}}^{s}\left(p_j(2)\cdot\pi_j\right) + \left(1-a^z_{2,1}\right)\cdot E^z_{1,0}$$

$$= -K_2 + a^z_{2,1} \cdot E^z_{1,1} + \left(1 - a^z_{2,1}\right) \cdot E^z_{1,0} + \sum_{j=m^z_{2,1}}^{s} \left(p_j(2) \cdot o_j\right)$$

$$= -K_2 + a^z_{2,1} \cdot E^z_{1,1} + \left(1 - a^z_{2,1}\right) \cdot E^z_{1,0} + b^z_{2,1}.$$

$$f^z_{T-3}(1,i) = \begin{cases} \pi_i + E^z_{2,0} \text{ if customer is accepted} \\ -\chi_i + E^z_{2,1} \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{3,1} = \sum_{j=0}^{m^z_{3,1}-1} \left(p_j(3) \cdot \left(-\chi_j + E^z_{2,1}\right)\right) + \sum_{j=m^z_{3,1}}^{s} \left(p_j(3) \cdot \left(\pi_j + E^z_{2,0}\right)\right)$$

$$= -K_3 + \sum_{j=m^z_{3,1}}^{s} \left(p_j(3) \cdot \chi_j\right) + a^z_{3,1} \cdot E^z_{2,1} + \sum_{j=m^z_{3,1}}^{s} \left(p_j(3) \cdot \pi_j\right) + \left(1 - a^z_{3,1}\right) \cdot E^z_{2,0}$$

$$= -K_3 + a^z_{3,1} \cdot E^z_{2,1} + \left(1 - a^z_{3,1}\right) \cdot E^z_{2,0} + \sum_{j=m^z_{3,1}}^{s} \left(p_j(3) \cdot o_j\right)$$

$$= -K_3 + a^z_{3,1} \cdot E^z_{2,1} + \left(1 - a^z_{3,1}\right) \cdot E^z_{2,0} + b^z_{3,1}.$$

From this, is seems that

$$E^z_{t,1} = -K_t + a^z_{t,1} \cdot E^z_{t-1,1} + \left(1 - a^z_{t,1}\right) \cdot E^z_{t-1,0} + b^z_{t,1} \text{ for } t \geq 1.$$

To prove this, consider any $t \geq 1$.

$$f^z_{T-t}(1,i) = \begin{cases} \pi_i + E^z_{t-1,0} \text{ if customer is accepted} \\ -\chi_i + E^z_{t-1,1} \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{t,1} = \sum_{j=0}^{m^z_{t,1}-1} \left(p_j(t) \cdot \left(-\chi_j + E^z_{t-1,1}\right)\right) + \sum_{j=m^z_{t,1}}^{s} \left(p_j(t) \cdot \left(\pi_j + E^z_{t-1,0}\right)\right)$$

$$= -K_t + \sum_{j=m^z_{t,1}}^{s} \left(p_j(t) \cdot \chi_j\right) + a^z_{t,1} \cdot E^z_{t-1,1} + \sum_{j=m^z_{t,1}}^{s} \left(p_j(t) \cdot \pi_j\right) + \left(1 - a^z_{t,1}\right) \cdot E^z_{t-1,0}$$

$$= -K_t + a^z_{t,1} \cdot E^z_{t-1,1} + \left(1 - a^z_{t,1}\right) \cdot E^z_{t-1,0} + \sum_{j=m^z_{t,1}}^{s} \left(p_j(t) \cdot o_j\right)$$

$$= -K_t + a^z_{t,1} \cdot E^z_{t-1,1} + \left(1 - a^z_{t,1}\right) \cdot E^z_{t-1,0} + b^z_{t,1}.$$

In the above proof, induction was not needed, and so it seems that it is not required to realize a pattern before proving the recursive formula for $E^z_{t,c}$. With this in mind, consider finding a recursive formula for $E^z_{t,c}$ when $c \geq 1$ and $t \geq 1$.

$$f^z_{T-t}(c,i) = \begin{cases} \pi_i + E^z_{t-1,c-1} \text{ if customer is accepted} \\ -\chi_i + E^z_{t-1,c} \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{t,c} = \sum_{j=0}^{m^z_{t,c}-1} \left(p_j(t) \cdot \left(-\chi_j + E^z_{t-1,c}\right)\right) + \sum_{j=m^z_{t,c}}^{s} \left(p_j(t) \cdot \left(\pi_j + E^z_{t-1,c-1}\right)\right)$$

$$= -K_t + \sum_{j=m^z_{t,c}}^{s} \left(p_j(t) \cdot \chi_j\right) + a^z_{t,c} \cdot E^z_{t-1,c} + \sum_{j=m^z_{t,c}}^{s} \left(p_j(t) \cdot \pi_j\right) + \left(1 - a^z_{t,c}\right) \cdot E^z_{t-1,c-1}$$

$$= -K_t + a^z_{t,c} \cdot E^z_{t-1,c} + \left(1 - a^z_{t,c}\right) \cdot E^z_{t-1,c-1} + \sum_{j=m^z_{t,c}}^{s} \left(p_j(t) \cdot o_j\right)$$

$$= -K_t + a^z_{t,c} \cdot E^z_{t-1,c} + \left(1 - a^z_{t,c}\right) \cdot E^z_{t-1,c-1} + b^z_{t,c}.$$

If $c \geq 1$ and $t = 0$,

$$f^z_T(c,i) = \begin{cases} \pi_i \text{ if customer is accepted} \\ -\chi_i \text{ if customer is rejected} \end{cases}, \text{ so}$$

$$E^z_{0,c} = \sum_{j=0}^{s} \left(p_j(0) \cdot f^z_T(c,j)\right) = \sum_{j=0}^{m^z_{0,c}-1} \left(p_j(0) \cdot \left(-\chi_j\right)\right) + \sum_{j=m^z_{0,c}}^{s} \left(p_j(0) \cdot \pi_j\right)$$

$$= -\sum_{j=0}^{s}\left(p_j(0)\cdot\chi_j\right) + \sum_{j=m_{0,c}^{z}}^{s}\left(p_j(0)\cdot\chi_j\right) + \sum_{j=m_{0,c}^{z}}^{s}\left(p_j(0)\cdot\pi_j\right)$$

$$= -K_0 + \sum_{j=m_{0,c}^{z}}^{s}\left(p_j(t)\cdot o_j\right) = -K_0 + b_{0,c}^{z}.$$

This gives the recursive formula

$$E_{t,c}^{z} = -K_t + a_{t,c}^{z}\cdot E_{t-1,c}^{z} + \left(1-a_{t,c}^{z}\right)\cdot E_{t-1,c-1}^{z} + b_{t,c}^{z} \text{ for } c \geq 1 \text{ and } t \geq 1 \qquad (6)$$

with boundary conditions

i) $E_{t,0}^{z} = -\sum_{d=0}^{t} K_d$ for all t, and (6i)

ii) $E_{0,c}^{z} = -K_0 + b_{0,c}^{z}$ for $c \geq 1$. (6ii)

Two suboptimal heuristic acceptance policies that obey the optimal property are a) accepting all customers as long as there is a room to accommodate the request and b) accepting a type i customer at time t iff the sum of the expected number of arrivals of type k customers for k > i is less than the current capacity. These options are both explored in the paper by Bitran and Mondschein, and they are standard approaches to finding "good" rules when finding the optimal rule is considered infeasible (1995). What follows is a description of these rules using (6).

Let z be the rule that accepts all customers as long as there is a room to accommodate the request. In this case, $m_{t,c}^{z} = 1$ when $c \geq 1$, and $m_{t,0}^{z} = s+1$.

This gives $a^z_{t,c} = p_0(t)$ when $c \geq 1$ and $a^z_{t,0} = 1$, and $b^z_{t,c} = R_t + K_t$ when $c \geq 1$ since $o_0 = 0$ and $b^z_{t,0} = 0$. Applying these values to (6) gives

$$E^z_{t,c} = p_0(t) \cdot E^z_{t-1,c} + \left(1 - p_0(t)\right) \cdot E^z_{t-1,c-1} + R_t \text{ for } c \geq 1 \text{ and } t \geq 1 \tag{7}$$

with boundary conditions

i) $E^z_{t,0} = -\sum_{d=0}^{t} K_d$ for all t, and (7i)

ii) $E^z_{0,c} = R_0$ for $c \geq 1$. (7ii)

If the remaining capacity is strictly greater than the time remaining, i.e. c > t, it seems to make since that no customer will be turned away, so $E^z_{t,c}$ should be equal to $\sum_{d=0}^{t} R_d$, the sum of the expected value of the π_i's. In fact, induction shows that indeed

$$E^z_{t,c} = \sum_{d=0}^{t} R_d \text{ for all } c > t. \tag{8}$$

The case for t = 0 is true by (7ii). For some $t \geq 0$, assume (8) holds. Consider any c > t+1. It must be the case that c > t and c-1 > t, so by the induction hypothesis,

$$E^z_{t+1,c} = p_0(t) \cdot E^z_{t,c} + \left(1 - p_0(t)\right) \cdot E^z_{t,c-1} + R_{t+1}$$

$$= p_0(t) \cdot \sum_{d=0}^{t} R_d + \left(1 - p_0(t)\right) \cdot \sum_{d=0}^{t} R_d + R_{t+1} = \sum_{d=0}^{t} R_d + R_{t+1} = \sum_{d=0}^{t+1} R_d .$$

Therefore, since $E^{z}_{0,c} = R_0$ and $E^{z}_{t,c} = \sum_{d=0}^{t} R_d \Rightarrow E^{z}_{t+1,c} = \sum_{d=0}^{t+1} R_d$ for the appropriate values of c, by induction $E^{z}_{t,c} = \sum_{d=0}^{t} R_d$ for all c > t.

Let y be the rule that accepts a type i customer at time t iff the sum of the expected number of arrivals of type k customers for k > i is less than the current capacity. This gives that

$$m^{y}_{t,c} = \min\left\{ i \,\middle|\, \sum_{j=i+1}^{s}\left(\int_{T-t}^{T} \lambda_j(t) \right) < c \right\} \text{ for } c > 0, \text{ and}$$

$$m^{y}_{t,0} = s + 1,$$

where $\lambda_j(t)$ is the arrival rate per time unit of type j customers. Recalling that $\lambda(t) = \sum_{j=1}^{s} \lambda_j(t)$ and that at most one customer arrives per time unit, it is seen that

$$\sum_{j=i+1}^{s}\left(\int_{T-t}^{T} \lambda_j(t)dt \right) = \int_{T-t}^{T}\left(\sum_{j=i+1}^{s} \lambda_j(t)dt \right) \le \int_{T-t}^{T} \lambda(t)dt \le \int_{T-t}^{T} dt = t.$$

Thus, for c > t, $m^{y}_{t,c} = 1$, $a^{y}_{t,c} = p_0(t)$, and $b^{y}_{t,c} = R_t + K_t$, so again (8) should hold. All that is needed to prove this is to show the case where t = 0 since the same induction proof used above will work here too. By (7ii),

$$E^{y}_{0,c} = -K_0 + b^{y}_{0,c} = -K_0 + R_0 + K_0 = R_0 \text{ for } c \ge 1,$$

so (8) holds here too.

In their paper, Bitran and Mondschein defined the quantity

$$OB = \frac{\sum_{i=1}^{s}\left(\int_{1}^{T}\lambda_i(t)dt\right)}{C},$$

the booking index, to relate the hotel's total capacity to the total expected number of arrivals. When they tested the heuristics y and z as defined above, they noted that it seems to be the case that the profit generated using rule z was less than the profit using rule y, however as the capacity increased and OB decreased, the difference between the profits decreased (1995). Mathematically, this can be stated as $\lim_{OB\downarrow 1}(y) = z$, i.e. as the booking index decreases to one, the heuristic y approaches the "accept everybody" rule. This is true since

$$OP \downarrow 1 \text{ iff for a fixed} \sum_{i=1}^{s}\left(\int_{1}^{T}\lambda_i(t)dt\right), C \uparrow \sum_{i=1}^{s}\left(\int_{1}^{T}\lambda_i(t)dt\right),$$

and so $m_{t,c}^{z} \downarrow 1$.

Finally, consider the optimal rule z^*. As has been seen already,

$$m_{t,c}^{z^*} = \min\left\{i \middle| o_i \geq \left(E_{t-1,c}^{z^*} - E_{t-1,c-1}^{z^*}\right)\right\} \text{ for } t \geq 1 \text{ and } c \geq 1,$$

$m_{t,0}^{z^*} = s+1$ for all t, and since $F_T(c,i) = \pi_i$ for all $c \cdot i \neq 0$, $m_{0,c}^{z^*} = 1$ for $c \geq 1$. Thus, for any $t > 0$, $m_{t,c}^{z^*}$ depends only on $E_{t-1,c}^{z^*}$ and $E_{t-1,c-1}^{z^*}$, so $E_{t,c}^{z^*}$ can be found using (6) and (7), and the optimal rule z^* can be found along the way.

Again, consider the case where $c > t$. Here, since the values of $m_{t,c}^{z^*}$ are not known at the beginning, it will be proved by induction that

$$E_{t,c}^{z^*} = \sum_{d=0}^{t} R_d \text{ for all } c > t, \text{ and} \qquad (9)$$

$$m_{t,c}^{z^*} = 1 \text{ for all } c > t. \qquad (10)$$

For the case $t = 0$, it has already been shown that $m_{0,c}^{z^*} = 1$, and therefore $E_{0,c}^{z^*} = R_0$. Assume that (9) and (10) hold for some $t \geq 0$. For any $c > t+1$, note that $c > t$ and $c-1 > t$. Thus, by the induction hypothesis,

$$m_{t+1,c}^{z^*} = \min\left\{i \geq 1 \middle| o_i \geq \left(E_{t,c}^{z^*} - E_{t,c-1}^{z^*}\right)\right\} = \min\left\{i \geq 1 \middle| o_i \geq \left(\sum_{d=0}^{t} R_d - \sum_{d=0}^{t} R_d\right)\right\}$$

$$= \min\left\{i \geq 1 \middle| o_i \geq 0\right\} = 1, \text{ and}$$

$$E_{t+1,c}^{z^*} = -K_{t+1} + a_{t+1,c}^{z^*} \cdot E_{t,c}^{z^*} + \left(1 - a_{t+1,c}^{z^*}\right) \cdot E_{t,c-1}^{z^*} + b_{t+1,c}^{z^*}$$

$$= -K_{t+1} + a_{t+1,c}^{z^*} \cdot \sum_{d=0}^{t} R_d + \left(1 - a_{t+1,c}^{z^*}\right) \cdot \sum_{d=0}^{t} R_d + \sum_{j=1}^{s} \left(p_j(t) \cdot o_j\right)$$

$$= -K_{t+1} + \sum_{d=0}^{t} R_d + R_{t+1} + K_{t+1} = \sum_{d=0}^{t+1} R_d \blacksquare$$

Thus, the rules z, y, and z^* all coincide for $c > t$.

To recap the results of the optimal rule, let $E_{t,c} := E_{t,c}^{z^*}$, $a_{t,c} := a_{t,c}^{z^*}$, $b_{t,c} := b_{t,c}^{z^*}$, and $m_{t,c} := m_{t,c}^{z^*}$. This gives that

$$E_{t,c} = -K_t + a_{t,c} \cdot E_{t-1,c} + \left(1 - a_{t,c}\right) \cdot E_{t-1,c-1} + b_{t,c} \text{ for all } t \geq c > 0,$$

with boundary conditions

i) $E_{t,0} = -\sum_{d=0}^{t} K_d$ for all t, and

ii) $E_{t,c} = \sum_{d=0}^{t} R_d$ for all c > t.

The optimal acceptance policy is found to be

$$m_{t,c} = \min\left\{i \middle| o_i \geq \left(E_{t-1,c} - E_{t-1,c-1}\right)\right\} \text{ for all } t \geq c > 0, \qquad (11)$$

$$m_{t,0} = s + 1 \text{ for all t,} \qquad (11i)$$

$$\text{and } m_{t,c} = 1 \text{ for } c > t. \qquad (11ii)$$

In order to implement the optimal rule, all one must do is calculate the array $m_{t,c}$ as described by (11) and from this array $m_{t,c}$, find the time threshold array τ^*_{ic} defined in chapter four. To find τ^*_{ic}, for a fixed capacity c consider each t for which $i = m_{t,c} < m_{t+1,c} = k$. For all $i \leq j < k$, let $\tau^*_{jc} := T - t$. This sets the earliest time for the capacity c at which type i through k-1 customers are accepted. The array τ^*_{ic}, which is of dimension $(s \times (C+1))$, is then stored and must be checked whenever a customer requests a room.

In order to make daily acceptance or rejection decisions, it is not necessary to calculate the expected profit. The expected profit need only be calculated once, and so the time required to execute an algorithm to calculate the expected profit is irrelevant (see appendix B). Thus, it seems that it would be advantageous to develop similar results to those found here for more complicated cases such as multiple products, reservations, and multiple night stays.

A final note here is that formulas (6), (6i) and (6ii) are actually satisfied by the expected value of any objective function found by using any acceptance policy with the following changes in definitions:

- $D^z_{t,c} = \{i | \text{a type i customer is accepted by policy z}\}$ where z is **any** acceptance policy.
- $a^z_{t,c} = \sum_{j \notin D^z_{t,c}} p_j(t)$.
- $b^z_{t,c} = \sum_{j \notin D^z_{t,c}} (p_j(t) \cdot o_j)$.

These definitions for $a^z_{t,c}$ and $b^z_{t,c}$ are simply generalizations of the definitions given before for $a^z_{t,c}$ and $b^z_{t,c}$.

CHAPTER SIX

MULTIPLE ROOM TYPES

This chapter will deal with an expansion of the solution to the single room type case to the multiple room type case. To do this, chapter six will be divided into three sections. Section 6.1 will use the dynamic programming algorithm to derive a formula for maximum expected profit in the multiple room type case. Section 6.2 will summarize results that Bitran and Mondschein derived for the multiple room type case. Section 6.3 will use a right to left technique to describe an algorithm that will specify the various acceptance thresholds that define the optimal rule in the multiple room type case.

6.1

Before giving the formula for finding the expected maximal profit, an important new feature involved in this case must be introduced, the downgrading of room types. First, each customer type requests only one type of room. With this, $A_r := \{k | \text{a type } k \text{ customer requests a type } r \text{ room}\}$. The room types are ordered so that room type 1 is the best and room type r is better than room type q iff $r < q$. The assumption is made that if a type i customer requests a type r room, then the customer will be willing to accept any room of type 1 through r if the price remains the same. The hotel has the option of downgrading a room of type p to type r if p < r, which means that if a type i customer requests a type r room then the hotel may give that customer a type p room but only charge π_i, the price that the customer is willing to pay.

The following formula was given by Bitran and Mondschein for the maximum expected profit from time t onwards (1995):

$$F_t(c,i) = \max\begin{cases} \max_{p \in B(c,i)} \pi_i + E_j\left[F_{t+1}(c - e_p, j)\right] \\ -\chi_i + E_j\left[F_{t+1}(c,j)\right] \end{cases} \text{if } i \cdot c(k) \neq 0 \text{ for some } k \leq r,$$

$$F_t(c,i) = -\chi_i + E_j\left[F_{t+1}(c,j)\right] \text{ if } i = 0 \text{ or } c(k) = 0 \text{ for all } k \leq r,$$

with boundary conditions

$$F_T(c,i) = \pi_i \text{ if } i \neq 0 \text{ and } c(k) \neq 0 \text{ for some } k \leq r, \text{ and}$$

$$F_T(c,i) = -\chi_i \text{ if } i = 0 \text{ or } c(k) = 0 \text{ for all } k \leq r.$$

The following definitions are necessary to understand this formula.

- The hotels capacity c is now a vector with as many components as there are room types.
- c(k) is the k-th component of c, and it represents the number of available type k rooms.
- e_p is a vector with the same dimension as c. The p-th component of e_p is a 1. All other components are 0.

- r_i is the r such that $i \in A_{r_i}$
- $B(c,i) := \{p | 1 \le p \le r_i \& c(p) > 0\}$.

For all other definitions, refer back to chapter three.

In order to use DP to derive this formula, the following changes are made to the DP variables and functions.

- x_t is a vector representing the capacity of each room type at time t.
- $x_{t+1} = f_t(x_t, u_t, w_t) = x_t - u_t \cdot e_p$, where p is the type of room given to the customer if a room is assigned, and p is arbitrary if no room is assigned.

The rest of the definitions given in the relating part of chapter three remain the same.

Now, beginning the DP algorithm, let t = T.

$J_T(x_T) = \max\left\{E_{w_T}\left[u_T \cdot \left(\pi_{w_T} + \chi_{w_T}\right) - \chi_{w_T}\right]\right\}$, so

$J_T(x_T) = E_{w_T}\left[-\chi_{w_T}\right] = E_{w_T}\left[F_T(x_T, w_T)\right]$ if $x_T(p) = 0 \; \forall 1 \le p \le r$, and

$J_T(x_T) = E_{w_T}\left[\pi_{w_T}\right] = E_{w_T}\left[F_T(x_T, w_T)\right]$ if $x_t(p) \ne 0$ for some $1 \le p \le r$,

where $w_T \in A_r$. For any t < T, finding the maximization requires deciding whether or not to rent a room and if a room is rented, deciding which room to rent. This gives

$$J_t(x_t) = \max_{u_t}\left\{E_{w_t}\left[g_t(x_t,u_t,w_t) + \max_{p\in B(x_t,w_t)} E_{w_{t-1}}\left[F_{t+1}(x_t - u_t \cdot e_p, w_{t+1})\right]\right]\right\}.$$

This gives that

$$J_t(x_t) = E_{w_t}\left[\max\begin{cases}\pi_{w_t} + \max_{p\in B(x_t,w_t)} E_{w_{t-1}}\left[F_{t+1}(x_t - u_t \cdot e_p, w_{t+1})\right] \\ -\chi_{w_t} + E_{w_{t-1}}\left[F_{t+1}(x_t, w_{t+1})\right]\end{cases}\right], \text{ or}$$

$$J_t(x_t) = E_{w_t}\left[F_t(x_t, w_t)\right].$$

If $x_t(k) = 0$ for all k then

$$J_t(x_t) = E_{w_t}\left[-\chi_{w_t} + E_{w_{t-1}}\left[F_{t+1}(x_t, w_{t+1})\right]\right] = E_{w_t}\left[F_t(x_t, w_t)\right].$$

This completes the derivation of the formula found in Bitran and Mondschein for the multiple room type case. It is important to note here that the type of room to be rented has not yet been determined. As will be seen after the proof of proposition four in section 6.2, the optimal room to rent, if renting a room is the decision reached, is the room $p = \max B(x_t, w_t)$. This means that the room to rent is lowest value room that the customer is willing to accept.

6.2

The formula derived in 6.1 gives that a type i customer is accepted at time t and capacity c iff

$$\pi_i + E_j\left[F_{t+1}(c - e_{p(i)}, j)\right] \geq -\chi_i + E_j\left[F_{t+1}(c, j)\right]$$

where $p(i) := \max B(c,i)$. Defining

$$\alpha_t(c, p(i)) := E_j[F_{t+1}(c, j)] - E_j\left[F_{t+1}\left(c - e_{p(j)}, j\right)\right],$$

it is clear that a type i customer is accepted at time t and capacity c iff

$$\pi_i + \chi_i \geq \alpha_t(c, p(i)).$$

The following lemmas and propositions are found concerning F and α. The proofs were given by Bitran and Mondschein (1995). Lemmas 1.1 and 2.1 are generalizations of Lemmas 1 and 2 given earlier.

Lemma 1.1: *The function* $F_t(c,i)$ *is monotonically increasing as a function of the capacity:*

$F_t(c,i) \geq F_t(c - e_p, i)$ for all c, p, t, i, and

$F_t(c - e_q, i) \geq F_t(c - e_p, i)$ for all $q \geq p$ and for all c, t, i.

Proof: Consider the optimal policy to be used starting at time t and capacity $c - e_p$ and customer type i. At each step of this optimal rule, a room to be rented to a particular customer type at a particular time will be available at that same time for that same customer type if the beginning capacity was c. Thus, the optimal rule for capacity $c - e_p$ can be used for capacity c, and since this rule rents to exactly the same customer types with the same room types for beginning capacities c and $c - e_p$, the value of their objective function will be the same (see appendix A), call it X. Now the optimal rule for beginning capacity c by definition must have an objective function value of at least X, so $F_t(c,i) \geq F_t(c-e_p,i)$ for all c, p, t, i. The same argument holds for beginning capacity $c-e_q$ in place of c for all $q \geq p$ since a type p room can be downgraded to a type q room ■

Lemma 2.1: *The objective function satisfies the inequality:*

$$F_t(c+e_l+e_p,i) - F_t(c+e_l,i) \leq F_t(c+e_p,i) - F_t(c,i)\text{, for all t, i, c, p, l.} \qquad (12)$$

Proof: The proof is by induction on t. For t = T, for any $i \neq 0$, c, p, and l, there are four cases:

Case i) There is a q such that $1 \leq q \leq r$ and $c(q) > 0$. This gives that

$$F_t(c+e_l+e_p,i) = F_t(c+e_l,i) = F_t(c+e_p,i) = F_t(c,i) = \pi_i\text{,}$$

so inequality (12) holds.

Case ii) For all q such that $1 \leq q \leq r$, $c(q) = 0$, and $r \geq l$. This gives that

$$F_t(c+e_l+e_p,i) = F_t(c+e_l,i) = \pi_i\text{,}$$

$$F_t(c+e_p,i) = \begin{cases} \pi_i \text{ if } p \leq l \\ -\chi_i \text{ if } p > l \end{cases} \geq -\chi_i\text{, and}$$

$$F_t(c,i) = -\chi_i.$$

This gives that

$$F_t(c+e_l+e_p,i) - F_t(c+e_l,i) = 0 \text{ and } F_t(c+e_p,i) - F_t(c+e_l,i) \geq -\chi_i + \chi_i = 0\text{,}$$

so inequality (12) holds.

Case iii) For all q such that $1 \le q \le r$, $c(q) = 0$, and $r \ge p$. This gives that

$$F_t(c+e_l+e_p,i) = F_t(c+e_p,i) = \pi_i,$$

$$F_t(c+e_l,i) = \begin{cases} \pi_i \text{ if } l \le p \\ -\chi_i \text{ if } l > p \end{cases} \ge -\chi_i, \text{ and}$$

$$F_t(c,i) = -\chi_i.$$

This gives that

$$F_t(c+e_l+e_p,i) - F_t(c+e_l,i) \le \pi_i + \chi_i \text{ and } F_t(c+e_p,i) - F_t(c+e_l,i) = \pi_i + \chi_i,$$

so inequality (12) holds.

Case iv) For all q such that $1 \le q \le r$, $c(q) = 0$, $r < p$, $r < l$. This gives that

$$F_t(c+e_l+e_p,i) = F_t(c+e_l,i) = F_t(c+e_p,i) = F_t(c,i) = -\chi_i,$$

so inequality (12) holds. Now, if $i = 0$,

$$F_t(c+e_l+e_p,i) = F_t(c+e_l,i) = F_t(c+e_p,i) = F_t(c,i) = -\chi_i,$$

and so inequality (12) holds. Thus, (12) is true for $t = T$. Assume (12) is true for some $t+1 \le T$, and consider the objective function at time t. Assume $i \ne 0$. Again, the four cases shown above should be shown here, however only case i) is shown since the other cases have fewer options and are similar yet easier. For capacity c the objective function takes on the maximum of either

(a) $\pi_i + E_j\left[F_{t+1}(c-e_{q1},j)\right]$, or

(b) $-\chi_i + E_j\left[F_{t+1}(c,j)\right]$.

For capacity $c+e_p$ the objective function takes on the maximum of either

(c) $\pi_i + E_j\left[F_{t+1}(c + e_p - e_{q2}, j)\right]$, or

(d) $-\chi_i + E_j\left[F_{t+1}(c + e_p, j)\right]$.

For capacity $c + e_l$ the objective function takes on the maximum of either

(e) $\pi_i + E_j\left[F_{t+1}(c + e_l - e_{q3}, j)\right]$, or

(f) $-\chi_i + E_j\left[F_{t+1}(c + e_l, j)\right]$.

For capacity $c + e_l + e_p$ the objective function takes on the maximum of either

(g) $\pi_i + E_j\left[F_{t+1}(c + e_l + e_p - e_{q4}, j)\right]$, or

(h) $-\chi_i + E_j\left[F_{t+1}(c + e_l + e_p, j)\right]$.

To prove that (12) holds, it must be shown that every feasible combination of choosing one value from (a) & (b), one value from (c) & (d), one value from (e) & (f), and one value from (g) & (h) satisfies (12). The combinations (a), (c), (e), (g); (b), (c), (e), (g); (b), (d), (e), (g); (b), (d), (f), (g); and (b), (d), (f), (h) all have similar proofs. Only one will be shown here.

(b), (c), (e), (g): This combination gives

$$F_t(c + e_l + e_p, i) - F_t(c + e_l, i) = E_j\left[F_{t+1}(c - e_{q4} + e_l + e_p, j)\right] - E_j\left[F_{t+1}(c - e_{q3} + e_l, j)\right]$$

$$= E_j\left[F_{t+1}(c - e_{q4} + e_l + e_p, j) - F_{t+1}(c - e_{q3} + e_l, j)\right]$$

and

$$F_t(c + e_p, i) - F_t(c, i) > E_j\left[F_{t+1}(c - e_{q2} + e_p, j)\right] - E_j\left[F_{t+1}(c - e_{q1}, j)\right]$$

$$= E_j\left[F_{t+1}(c - e_{q2} + e_p, j) - F_{t+1}(c - e_{q1}, j)\right].$$

Now, the relationships between q1, q2, q3, q4, p, and l must be established. If q1 $\geq$ p and q1 $\geq$ l, then q1 = q2 = q3 = q4 = q, and so by the induction hypothesis,

$$F_{t+1}(c - e_q + e_l + e_p, i) - F_{t+1}(c - e_q + e_l, i) \leq F_{t+1}(c - e_q + e_p, i) - F_{t+1}(c - e_q, i),$$

which by taking expected values of both sides implies

$$F_t(c + e_l + e_p, i) - F_t(c + e_l, i) \leq F_t(c + e_l, i) - F_t(c, i).$$

If q1 < p and q1 $\geq$ l, then q2 = q4 = p and q1 = q3 = q with q < p. This gives that

$$F_t(c + e_l + e_p, i) - F_t(c + e_l, i) = E_j\left[F_{t+1}(c + e_l, j) - F_{t+1}(c - e_q + e_l, j)\right], \text{ and}$$

$$F_t(c + e_p, i) - F_t(c, i) > E_j\left[F_{t+1}(c, j) - F_{t+1}(c - e_q, j)\right].$$

By the induction hypothesis,

$$F_{t+1}(c - e_q + e_l + e_q, i) - F_{t+1}(c - e_q + e_l, i) \leq F_{t+1}(c - e_q + e_q, i) - F_{t+1}(c - e_q, i),$$

which is equivalent to

$$F_{t+1}(c + e_l, i) - F_{t+1}(c - e_q + e_l, i) \leq F_{t+1}(c, i) - F_{t+1}(c - e_q, i)$$

which by taking expected values of both sides implies

$$F_t(c + e_l + e_p, i) - F_t(c + e_l, i) \leq F_t(c + e_l, i) - F_t(c, i).$$

If q1 < p and q1 < l, then without loss of generality, assume p $\geq$ l. This gives that q1 = q, q2 = q4 = p, and q3 = l where q < l, and so

$$F_t(c + e_l + e_p, i) - F_t(c + e_l, i) = E_j\left[F_{t+1}(c + e_l, j) - F_{t+1}(c, j)\right], \text{ and}$$

$$F_t(c + e_p, i) - F_t(c, i) > E_j\left[F_{t+1}(c, j) - F_{t+1}(c - e_q, j)\right].$$

By Lemma 1.1, $F_{t+1}(c+e_l,i) \le F_{t+1}(c+e_q)$, which implies that

$$F_{t+1}(c+e_l,i) - F_{t+1}(c,i) \le F_{t+1}(c+e_q) - F_{t+1}(c,i).$$

By the induction hypothesis,

$$F_{t+1}(c-e_q+e_q+e_q,i) - F_{t+1}(c-e_q+e_q,i) \le F_{t+1}(c-e_q+e_q,i) - F_{t+1}(c-e_q,i),$$

which is equivalent to

$$F_{t+1}(c+e_q,i) - F_{t+1}(c,i) \le F_{t+1}(c,i) - F_{t+1}(c-e_q,i), \text{ so}$$

$$F_{t+1}(c+e_l,i) - F_{t+1}(c,i) \le F_{t+1}(c,i) - F_{t+1}(c-e_q,i)$$

which by taking expected values of both sides implies

$$F_t(c+e_l+e_p,i) - F_t(c+e_l,i) \le F_t(c+e_l,i) - F_t(c,i).$$

Thus, the combination (b), (c), (e), (g) satisfies (12).

The remaining 11 possible combinations all lead to a contradiction of the induction hypothesis. Only one combination will be shown.

(a), (d), (e), (g): Here, the customer is accepted at capacity c, but rejected at capacity $c+e_p$. Thus,

$$\pi_i + \chi_i \ge E_j\left[F_{t+1}(c,j)\right] - E_j\left[F_{t+1}(c-e_{q1},j)\right] = E_j\left[F_{t+1}(c,j) - F_{t+1}(c-e_{q1},j)\right], \text{and}$$

$$\pi_i + \chi_i < E_j\left[F_{t+1}(c+e_p,j)\right] - E_j\left[F_{t+1}(c-e_{q2}+e_p,j)\right]$$

$$= E_j\left[F_{t+1}(c+e_p,j) - F_{t+1}(c-e_{q2}+e_p,j)\right].$$

Combining these two inequalities gives

$$E_j\left[F_{t+1}(c+e_p,j)-F_{t+1}(c-e_{q2}+e_p,j)\right] > E_j\left[F_{t+1}(c,j)-F_{t+1}(c-e_{q1},j)\right].$$

If $q1 \geq p$, then $q1 = q2 = q$, and so

$$E_j\left[F_{t+1}(c+e_p,j)-F_{t+1}(c-e_q+e_p,j)\right] > E_j\left[F_{t+1}(c,j)-F_{t+1}(c-e_q,j)\right].$$

However, by the induction hypothesis,

$$F_{t+1}(c-e_q+e_p+e_q,i)-F_{t+1}(c-e_q+e_p,i) \leq F_{t+1}(c-e_q+e_q,i)-F_{t+1}(c-e_q,i).$$

This is a contradiction. If $q1 < p$, $q1 = q$ and $q2 = p$, and so

$$E_j\left[F_{t+1}(c+e_p,j)-F_{t+1}(c,j)\right] > E_j\left[F_{t+1}(c,j)-F_{t+1}(c-e_q,j)\right].$$

By Lemma 1.1 and the induction hypothesis,

$$\begin{aligned} F_{t+1}(c+e_p,i)-F_{t+1}(c,i) &\leq F_{t+1}(c+e_q,i)-F_{t+1}(c,i) \\ &= F_{t+1}(c-e_q+e_q+e_q,i)-F_{t+1}(c-e_q+e_q,i) \\ &\leq F_{t+1}(c-e_q+e_q,i)-F_{t+1}(c-e_q,i), \text{ so} \end{aligned}$$

$$F_{t+1}(c+e_p,i)-F_{t+1}(c,i) \leq F_{t+1}(c,i)-F_{t+1}(c-e_q,i).$$

This is also a contradiction, so the combination (a), (d), (e), (g) is infeasible.

Finally, if $i = 0$,

$$F_t(c+e_l+e_p,i)-F_t(c+e_l,i) = E_j\left[F_{t+1}(c+e_l+e_p,j)-F_{t+1}(c+e_l,j)\right], \text{ and}$$

$$F_t(c+e_p,i)-F_t(c,i) = E_j\left[F_{t+1}(c+e_p,j)-F_{t+1}(c,j)\right],$$

and so the induction hypothesis implies that (12) is true ■

Proposition 3: *The function* $\alpha_t(c,p)$ *is nonincreasing as a function of c:*

$\alpha_t(c-e_l,p) \geq \alpha_t(c,p)$ for all c, p, l, t.

Proof: By lemma 2.1,

$$F_{t+1}(c-e_l,i) - F_{t+1}(c-e_p-e_l,i) \geq F_{t+1}(c,i) - F_{t+1}(c-e_p,i).$$

Taking the expected value of both sides gives

$$E_j[F_{t+1}(c-e_l,j)] - E_j[F_{t+1}(c-e_p-e_l,j)] \geq E_j[F_{t+1}(c,j)] - E_j[F_{t+1}(c-e_p,j)]$$

which is precisely $\alpha_t(c-e_l,p) \geq \alpha_t(c,p)$ ■

Proposition 4: *The function* $\alpha_t(c,p)$ *is nonincreasing as a function of t:*

$\alpha_t(c,p) \geq \alpha_{t+1}(c,p)$ for all c, p, l, t.

Proof: Note that $\alpha_t(c,p)$ is only defined if c(p) > 0. Let $i \in A_r$, and assume that there is some $1 \leq q \leq r$ such that c(q) > 0. Let $l = \max\{1 \leq q \leq r \mid c(q) > 0\}$. If $l \neq p$ or c(p) > 1,

$$F_{t+1}(c,i) = \max\begin{cases} \pi_i + E_j[F_{t+2}(c-e_l,j)] \\ -\chi_i + E_j[F_{t+2}(c,j)] \end{cases} \text{ and}$$

$$F_{t+1}(c-e_p,i) = \max\begin{cases} \pi_i + E_j[F_{t+2}(c-e_p-e_l,j)] \\ -\chi_i + E_j[F_{t+2}(c-e_p,j)] \end{cases}.$$

For $i \neq 0$, consider the following two cases.

Case 1: assume $\max\begin{Bmatrix}\pi_i + E_j\left[F_{t+2}(c-e_p-e_l,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c-e_p,j)\right]\end{Bmatrix} = -\chi_i + E_j\left[F_{t+2}(c-e_p,j)\right]$. This gives

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) = \max\begin{Bmatrix}\pi_i + E_j\left[F_{t+2}(c-e_l,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c,j)\right]\end{Bmatrix} - \left(-\chi_i + E_j\left[F_{t+2}(c-e_p,j)\right]\right)$$

$$\geq -\chi_i + E_j\left[F_{t+2}(c,j)\right] - \left(-\chi_i + E_j\left[F_{t+2}(c-e_p,j)\right]\right)$$

$$= E_j\left[F_{t+2}(c,j)\right] - E_j\left[F_{t+2}(c-e_p,j)\right] = \alpha_{t+1}(c,p).$$

Case 2: assume $\max\begin{Bmatrix}\pi_i + E_j\left[F_{t+2}(c-e_p-e_l,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c-e_p,j)\right]\end{Bmatrix} = \pi_i + E_j\left[F_{t+2}(c-e_p-e_l,j)\right]$. Here,

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) = \max\begin{Bmatrix}\pi_i + E_j\left[F_{t+2}(c-e_l,j)\right] \\ -\chi_i + E_j\left[F_{t+2}(c,j)\right]\end{Bmatrix} - \pi_i - E_j\left[F_{t+2}(c-e_p-e_l,j)\right]$$

$$\geq \pi_i + E_j\left[F_{t+2}(c-e_l,j)\right] - \pi_i - E_j\left[F_{t+2}(c-e_p-e_l,j)\right]$$

$$= E_j\left[F_{t+2}(c-e_l,j)\right] - E_j\left[F_{t+2}(c-e_p-e_l,j)\right] = \alpha_{t+1}(c-e_l,p).$$

By Proposition 3, $\alpha_{t+1}(c-e_l,p) \geq \alpha_{t+1}(c,p)$, so

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) \geq \alpha_{t+1}(c,p) \text{ for } i \neq 0.$$

For i = 0, $F_{t+1}(c,0) - F_{t+1}(c-e_p,0) = \alpha_{t+1}(c,p)$, so

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) \geq \alpha_{t+1}(c,p) \text{ for all } i,$$

so by taking the expected value of both sides and noting that the right hand side is a constant,

$\alpha_t(c,p) \geq \alpha_{t+1}(c,p)$.

If l = p and c(p) = 1, then either a) there is a $1 \leq q < p$ with c(q) >0, or b) for all $1 \leq q < p$, c(q) = 0.

subcase a) Let $k = \max\{1 \leq q < p \mid c(q) > 0\}$. If

$$F_{t+1}(c-e_p,i) = \max\begin{cases} \pi_i + E_j[F_{t+2}(c-e_p-e_k,j)] \\ -\chi_i + E_j[F_{t+2}(c,j)] \end{cases} = -\chi_i + E_j[F_{t+2}(c,j)],$$

it has already been seen that $\alpha_t(c,p) \geq \alpha_{t+1}(c,p)$. If

$$F_{t+1}(c-e_p,i) = \max\begin{cases} \pi_i + E_j[F_{t+2}(c-e_p-e_k,j)] \\ -\chi_i + E_j[F_{t+2}(c,j)] \end{cases} = \pi_i + E_j[F_{t+2}(c-e_p-e_k,j)], \text{ then}$$

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) = \max\begin{cases} \pi_i + E_j[F_{t+2}(c-e_p,j)] \\ -\chi_i + E_j[F_{t+2}(c,j)] \end{cases} - \pi_i - E_j[F_{t+2}(c-e_p-e_k,j)]$$

$$\geq E_j[F_{t+2}(c-e_p,j)] - E_j[F_{t+2}(c-e_p-e_k,j)]$$

$$= \alpha_{t+1}(c-e_p,k).$$

By Proposition 3, $\alpha_{t+1}(c,k) \geq \alpha_{t+1}(c-e_p,k)$, and by Lemma 1.1, $\alpha_{t+1}(c,k) \geq \alpha_{t+1}(c,p)$, so the inequality $\alpha_t(c,p) \geq \alpha_{t+1}(c,p)$ holds. Finally, if c(q) = 0 for all $1 \leq q \leq r$,

$$F_{t+1}(c,i) - F_{t+1}(c-e_p,i) = -\chi_i + E_j[F_{t+2}(c,j)] + \chi_i - E_j[F_{t+2}(c-e_p,j)] = \alpha_{t+1}(c,p),$$

so for all cases, $\alpha_t(c,p) \geq \alpha_{t+1}(c,p)$ is true ■

It was mentioned at the end of section 6.1 that the optimal room to rent is $p(i) = \max B(c,i)$. This is a consequence of Lemma 1.1 since for any q < p, $F_t(c-e_p,i) \geq F_t(c-e_q,i)$, so the expected profit for renting the better room is no more than the expected profit for renting the lesser room.

Proposition three gives that for each time t, if a type i customer is accepted at capacity c, then a type i customer is accepted at any capacity c + b, where b is a vector of the same dimension as c and each component of b is non-negative. It is more difficult to define the capacity threshold defined for the single product case since it would first be necessary to define what is meant by vector c less than vector c', and since this will play no role in the rest of the paper, no capacity threshold will be found here.

Proposition four gives that for each capacity c, there is a time threshold $\tau^*_{ic} := \min\{t \mid o_i \geq \alpha_t(c,p)\}$. For capacity c, a type i customer is accepted iff $t \geq \tau^*_{ic}$. The sequence $\left(\tau^*_{ic}\right)_{0\leq i\leq s}$ is not necessarily decreasing in i as was the case in chapter five. This is because a hotel may accept a low value customer because there are plenty of low value rooms, however, a high value customer could be rejected because there are no high value rooms left. On the other hand, if customer types j and k each request the same room type then $\left(\tau^*_{ic}\right)_{j\leq i\leq k}$ is decreasing in i, so the hotel rents to higher valued customers that request a particular room type before it rents to lower valued customers who request the same room type.

6.3

Previously, it was defined that for customer types, $i \leq k$ iff $o_i \leq o_k$. This definition might no longer suffice in the multiple product case since a customer may have a higher opportunity cost than another and yet request a lower value room. This case would only practically arise in the case with reservations which is not discussed here, and so the assumption will be made here that if a customer requests a certain type of room then that customer must have a higher opportunity cost than any customer who requests a lower value room. This means that the same order given to the customer types in the single product case will hold in the multiple product case without reservations.

Chapter five found a recursive formula for the expected profit the hotel would receive when using a general rule that obeyed the optimal policy. This formula was then specialized for three specific rules. This section could be approached in the same manner, however since the optimal rule is of most interest, only the formula for the optimal rule will be developed. To do this, the following definitions will be helpful.

- $E_{t,c} := E_j\left[F_{T-t}(c, j)\right]$.
- $m_{t,c,q} := \min\left\{ i \in A_q \middle| o_i \geq \alpha_t(c, p(i)) \right\}$, where $m_{t,c,q} = \max A_q + 1$ if no customer is accepted and $m_{t,c,q} \geq 1$.
- $x_c := \max\left\{ i \middle| c(r_i) > 0 \right\}$, the best type of customer for whom a room is available.
- $D_{t,c} := \left\{ i \middle| o_i \geq \alpha_t(c, p(i)) \right\}$, the set of customer types who are accepted. Note that $D_{t,c} = \left\{ i \middle| m_{t,c,n} \leq i \leq \max A_n \text{ or } m_{t,c,n-1} \leq i \leq \max A_{n-1} \text{ or } \cdots \text{ or } m_{t,c,1} \leq i \leq s \right\}$.
- $a_{t,c} := \sum_{j \in D_{t,c}} p_j(t)$.

For other definitions, refer back to chapter five.

The objective here is to develop a recursive formula for $E_{t,c}$. In order to do this, it must first be specified what order the vector c must be considered in. Here, let the dimension of the capacity vector be n. Consider the two n-dimensional vectors $c \neq d$. Let $1 \leq v \leq n$ be the smallest integer such that $c(v) \neq d(v)$. Such a v must exist or c = d. Define $c \prec d$ iff $c(v) < d(v)$. Consider any $c \prec d \prec h$. Let v be the smallest integer such that $c(v) \neq d(v)$, and let w be the smallest integer such that $d(w) \neq h(w)$. If v = w then v is the smallest integer such that $c(v) \neq h(v)$ and $c(v) < d(v) < h(v)$ so $c \prec h$. If v < w then v is the smallest integer such that $c(v) \neq h(v)$ and $c(v) < d(v) = h(v)$ so $c \prec h$. If v > w then w is the smallest integer such that c(w) $\neq$ h(w) and c(w) = d(w) < h(w) so $c \prec h$. Thus, the relation $\prec$ is transitive. Also, by definition, for any c $\neq$ d, either $c \prec d$ or $d \prec c$ but not both. This shows that $\prec$ gives the finite set of possible capacities a well ordering.

Consider c = **0**, where **0** is the n dimensional vector consisting entirely of zeros. Clearly, $\mathbf{0} \prec d$ for all d $\neq$ **0**. Consider

$F_T(\mathbf{0},i) = -\chi_i$, so

$$E_{0,0} = \sum_{j \notin D_{t,0}} \left(p_j(0)\cdot(-\chi_j)\right) = \sum_{j=0}^{s}\left(p_j(0)\cdot(-\chi_j)\right) = -K_0.$$

$F_{T-1}(\mathbf{0},i) = -\chi_i + E_{0,0} = -\chi_i - K_0$, so

$$E_{1,0} = \sum_{j \notin D_{t,0}} \left(p_j(1)\cdot(-\chi_j - K_0)\right) = \sum_{j=0}^{s}\left(p_j(1)\cdot(-\chi_j - K_0)\right) = -(K_0 + K_1).$$

As before, it seems that

$$E_{t,0} = -\sum_{d=0}^{t} K_d \qquad (13)$$

and this will be proved by induction. For t = 0, it has already been seen that (13) is true. Assume that (13) holds for some t $\geq$ 0. By the induction hypothesis,

$F_{T-(t+1)}(\mathbf{0},i) = -\chi_i + E_{t,0} = -\chi_i - \sum_{d=0}^{t} K_d$, so

$$E_{t+1,0} = \sum_{j \notin D_{t,0}} \left(p_j(t+1)\cdot\left(-\chi_j - \sum_{d=0}^{t} K_d\right)\right) = \sum_{j=0}^{s}\left(p_j(t+1)\cdot\left(-\chi_j - \sum_{d=0}^{t} K_d\right)\right) = -\sum_{d=0}^{t+1} K_d,$$

and so (13) holds for t+1.

Next, consider any $c \succ \mathbf{0}$. For t = 0,

$$F_T(c,i) = \begin{cases} \pi_i & \text{if } i \le x_c \\ -\chi_i & \text{if } i > x_c \end{cases}, \text{so}$$

$$E_{0,c} = \sum_{j \in D_{0,c}} \left(p_j(0) \cdot \pi_j\right) - \sum_{j \notin D_{0,c}} \left(p_j(0) \cdot \chi_j\right).$$

for any t > 0,

$$F_{T-t}(c,i) = \begin{cases} \max\begin{cases} \pi_i + E_{t-1,c-e_{p(i)}} \\ -\chi_i + E_{t-1,c} \end{cases} & \text{if } i \le x_c \\ -\chi_i + E_{t-1,c} & \text{if } i > x_c \end{cases}, \text{ so}$$

$$E_{t,c} = \sum_{j \notin D_{t,c}} \left(p_j(t) \cdot \left(-\chi_j + E_{t-1,c}\right)\right) + \sum_{j \in D_{t,c}} \left(p_j(t) \cdot \left(\pi_j + E_{t-1,c-e_{p(j)}}\right)\right)$$

$$= -K_t + \sum_{j \in D_{t,c}} \left(p_j(t) \cdot o_j\right) + E_{t-1,c} \cdot \sum_{j \notin D_{t,c}} \left(p_j(t)\right) + \sum_{j \in D_{t,c}} \left(p_j(t) \cdot E_{t-1,c-e_{p(j)}}\right)$$

$$= -K_t + a_{t,c} \cdot E_{t-1,c} + \sum_{j \in D_{t,c}} \left(p_j(t) \cdot \left(o_j + E_{t-1,c-e_{p(j)}}\right)\right).$$

If c(1) > t, i.e. there are more of the best rooms available than there is time remaining, it seems that the formula for the multiple room types and single room type case should coincide, that is, it should be the case that $E_{t,c} = \sum_{d=0}^{t} R_d$ if c(1) > t. Although this can be shown to be true, it would save relatively few calculations since the total number of time units in a day, T, is much larger than the total number of type 1 rooms, C(1). Thus, this approach will not be taken here.

With all this said, the recursive formula for the optimal expected profit in the multiple room type case is

$$E_{t,c} = -K_t + a_{t,c} \cdot E_{t-1,c} + \sum_{j \in D_{t,c}} \left(p_j(t) \cdot \left(o_j + E_{t-1,c-e_{p(j)}}\right)\right) \text{ for all } c \succ \mathbf{0},\ t > 0$$

with boundary conditions

$$E_{t,\mathbf{0}} = -\sum_{d=0}^{t} K_d \text{ for any t and}$$

$$E_{0,c} = \sum_{j \in D_{t,c}} \left(p_j(0) \cdot \pi_j\right) - \sum_{j \notin D_{t,c}} \left(p_j(0) \cdot \chi_j\right) \text{ for any } c \neq \mathbf{0}.$$

As in chapter five, in order to implement the optimal rule, one must calculate the $m_{t,c,q}$'s as the above recursive formula is calculated. From this, the threshold array τ^*_{ic} is found. For any fixed capacity c, consider each t for which $i = m_{t,c,q} < m_{t+1,c,q} = k$. For all $i \leq j < k$, let $\tau^*_{jc} := T - t$. With this done, all a hotel must do when a customer arrives is check the time threshold array for the current capacity and customer type. If the time of arrival is later than the threshold time, the customer is accepted. Otherwise, the customer is turned away.

The time required to find all $E_{t,c}$ in the multiple room type case will be much longer than the single room type case. Let C be the total capacity of a hotel. In the single room type case, there are $T \cdot (C+1)$ different expected values to find. In the multiple room type case, defining $C'(q) := C(q) + 1$, there are $T \cdot C'(1) \cdot C'(2) \cdot \ldots \cdot C'(n)$ different expected values to find. However, since these expected values need to be calculated only once, the time required is not very important to the applicability of the solution (see appendix B).

CHAPTER SEVEN

CONCLUSION

This paper has considered the problem of finding the optimal acceptance or rejection rule for a hotel to follow during the course of a day when customers request a room from the hotel. This optimal rule maximizes the expected profit the hotel will receive from the time that the customer arrives until the end of the day. Assumptions made for this analysis were that there are multiple customer types, there are multiple room types, room types can be downgraded, customer types arrive independently and in no particular order, time progresses in discrete units in each of which at most one customer can arrive, no time is required to process a customer's request, each customer type has an associated profit for the hotel if the customer is accepted, each customer type has an associated loss for the hotel if the customer is rejected, reservations are not allowed, and only requests for single night stays are considered.

The optimal rule is found by following a recursive formula to calculate the expected profit at all possible combinations of time and capacity, and finding the minimal customer types accepted for a time and capacity by taking the difference of two expected values and comparing this difference to the opportunity costs of the customer types. From these lists of minimal accepted customers, an array of threshold times is calculated which defines the actual optimal rule. For any capacity and customer type, the customer is accepted if and only if the arrival time is at least the threshold time associated with the customer type and capacity.

A natural extension of these results would be the case of multiple room types and reservations. Another avenue to explore would be the case of multiple night stays. To improve on these results, one might consider the possibility of customer type arrival being dependent as has been suggested in previous works (Brumelle et al., 1990, and Weatherford and Bodily, 1992). Also, one might investigate the consequences of allowing for time to process a customers request and considering the possibility of a line forming. Finally, a continuous time approach similar to the one looked at in chapter one might be informative.

Further extensions of this method might be available to a variety of similar problems. There has been much work done on the problem of maximizing the profits of airlines using similar structure and assumptions to the problem presented here (Alstrup et al., 1986, and Brumelle et al., 1990). Also, it has been suggested that similar problems arise in the entertainment industry and train transportation (Alstrup et al., 1986, Brumelle et al., 1990, and Weatherford and Bodily, 1992). It would be interesting to see if the method presented here could be adapted for use on these and other similar problems.

BIBLIOGRAPHY

BIBLIOGRAPHY

Alstrup, Jens, Søren Boas, Oli B. G. Madsen, and René Victor Valqui Vidal. 1986. Booking Policy for Flights With Two Types of Passengers. *European Journal of Operational Research* **27**, 274-288.

Bertsekas, Dimitri P. 1987. Dynamic Programming: Deterministic and Stochastic Models. Printice-Hall, Inc.: Englewood Cliffs.

Bitran, Gabriel R., and Susana V. Mondschein. 1995. An Application of Yield Management to the Hotel Industry Considering Multiple Day Stays. *Operations Research* **43**, 427-443.

Brumelle, S. L., J. I. McGill, T. H. Oum, K. Sawaki, and M. W. Tretheway. 1990. Allocation of Airline Seats between Stochastically Dependent Demands. *Transportation Science* **24**, 183-192.

Weatherford, Lawrence R., and Samuel E. Bodily. 1992. A Taxonomy and Research Overview of Perishable-Asset Revenue Management: Yield Management, Overbooking, and Pricing. *Operations Research* **40**, 831-843.

APPENDICES

APPENDIX A

APPENDIX A

In the proofs of Lemma 1 and 1.1, it was stated that if the rule which is the optimal rule for a given capacity and time is applied to the problem at a capacity of one more of any type room at the same time then the expected profit achieved by using that rule in either capacity is the same. This may not be clear upon first inspection, so the following is a proof of this fact.

Lemma A: *For any capacity c' and time t', let* $z_{t',c'}$ *be the optimal rule associated with t' and c'. Let* $f_t^{z_{t',c'}}(c,i)$ *be the expected profit from time t and capacity c given a type i customer has arrived at time t while using rule* $z_{t',c'}$*. Then for any room type q and customer type i,*

$$F_t(c',i) = f_t^{z_{t',c'}}(c'+e_q,i) \text{ for all } t \geq t'. \qquad (14)$$

Proof: This will be proved by induction. For t' = T, there are two cases.

Case i) If there is no room to satisfy the customers request then $F_T(c',i) = -\chi_i$, i.e., the rule $z_{t',c'}$ specifies to reject the customer. Thus, $f_T^{z_{t',c'}}(c'+e_q,i) = -\chi_i$, so $F_T(c',i) = f_T^{z_{t',c'}}(c'+e_q,i)$.

Case ii) If there is a room to satisfy the customers request then $F_T(c',i) = \pi_i$, i.e., the rule $z_{t',c'}$ specifies to accept the customer. Thus, $f_T^{z_{t',c'}}(c'+e_q,i) = \pi_i$, so $F_T(c',i) = f_T^{z_{t',c'}}(c'+e_q,i)$. Thus, for t' = T, (14) holds for any c' and i.

Assume (14) is true for some $t'+1 \le T$. Again there are two cases.

Case i) Assume $F_{t'}(c',i) = -\chi_i + E_j\left[F_{t'+1}(c',j)\right]$, i.e., the rule $z_{t',c'}$ specifies to reject the customer. Thus, $f_{t'}^{z_{t',c'}}(c'+e_q,i) = -\chi_i + E_j\left[f_{t'+1}^{z_{t',c'}}(c'+e_q,j)\right]$, but by the induction hypothesis, $F_{t'+1}(c',j) = f_{t'+1}^{z_{t',c'}}(c'+e_q,j)$ for all j, so $E_j\left[F_{t'+1}(c',j)\right] = E_j\left[f_{t'+1}^{z_{t',c'}}(c'+e_q,j)\right]$. This gives that (14) holds.

Case ii) Assume $F_{t'}(c',i) = \pi_i + E_j\left[F_{t'+1}(c',j)\right]$, i.e., the rule $z_{t',c'}$ specifies to accept the customer. Thus, $f_{t'}^{z_{t',c'}}(c'+e_q,i) = \pi_i + E_j\left[f_{t'+1}^{z_{t',c'}}(c'+e_q,j)\right]$. As was seen in case i), $E_j\left[F_{t'+1}(c',j)\right] = E_j\left[f_{t'+1}^{z_{t',c'}}(c'+e_q,j)\right]$. Again, this gives that (14) holds. Thus, for any c' and i, (14) holds and so by induction, (14) holds for all t' ■

APPENDIX B

APPENDIX B

It was mentioned that the time required to execute an algorithm to calculate the expected profit is irrelevant. In order to support this argument, a program was written to calculate both the expected profit and the arrays of minimal accepted customers in the two room type case. When this program was run for the case with six customer types, twenty-five type one rooms, and ninety-five type two rooms, the computer did a little under 300,000,000 floating point operations. This means that if run on a machine that can handle 20,000,000 floating point operations per second, this would take less than fifteen seconds to run. Thus, if one was willing to invest twelve hours to run this program, a case of 2880 times the magnitude of the case mentioned here could be handled.

VITA

Dale Sikkema was born in Tallahassee, Florida on March 21, 1971. He attended Elementary and Middle School in Tallahassee. In 1985, he moved to Knoxville, Tennessee and attended Farragut High School. He graduated from high school in 1989. In the Fall of 1989, he began studying at the University of Tennessee, Knoxville. He graduated cum laude with a Bachelor of Science in Mathematics and a minor in Computer Science in December, 1993. In August of 1994, he returned to the University of Tennessee, Knoxville to pursue a Master's degree in Mathematics. He graduated with a Master of Science in Mathematics in May, 1996.