



Dynamic Pricing in Hotels

Mr. Pradeep Mittal¹

Department Of Computer Science & Applications,
Kurukshetra University, Kurukshetra, India.

Dimmy²

Department Of Computer Science & Applications,
Kurukshetra University, Kurukshetra, India.

Abstract: In this paper we explain a dynamic pricing approach for the Hotel Revenue management. It is based on having “price multipliers” that provide a varying discount/premium over some seasonal reference price. The price multipliers are a function of certain influencing variables (for example hotel occupancy, time till arrival, etc). Then an optimization algorithm is applied for determining the parameters of these multipliers with the goal to maximize the revenue, taking into account current demand, and the demand-price sensitivity of the hotel’s guest. It includes C programming which can be efficiently used by hotels to maximize total revenue.

Keywords: Dynamic pricing, Hotel room forecasting, Monte Carlo simulation, Price Elasticity, Revenue Management System

I. INTRODUCTION

Revenue management (RM) is the science of managing a limited amount of supply to maximize revenue, by dynamically controlling the price/quantity offered. Here we explain a dynamic pricing approach which is based on having a seasonal reference price, and control variables in the form of multipliers. Each multiplier will adjust the price up or down around the reference price based on a certain influencing variables (for example hotel occupancy, time till arrival, etc). The parameters of these multipliers are optimized. The goal is to maximize the revenue, taking into account current demand, and the demand-price sensitivity of the hotel’s guest. Monte Carlo simulator is used that simulates all the hotel’s processes, such as reservations arrivals, cancellations, duration of stay, no shows, group reservations, seasonality, trend, etc, as faithfully as possible.

Section 2 discusses revenue management framework. Section 3 contains C coding and Section 4 provides conclusion.

II. RM FRAMEWORK

The proposed idea is based on having a seasonal reference price that is possibly a piecewise constant function that varies according to the major seasons’ classification. This reference price is set by the hotel managers. Moreover, we have four multipliers that represent the “control variables”. They will be multiplied by the reference price, to obtain the final price. They vary around the value of 1, where a value that is lower than one corresponds to a discount with respect to the reference price. Conversely, a value that is higher than 1 represents a premium over the reference price. The advantage of this formulation is that it will give the hotel manager or the revenue manager a suggested price that has some relation with the price that he has determined during his experience. So, he can relate to the new price by observing its discount/premium in relation to his reference price.

Each of the four multipliers corresponds to a variable that is known to have an influencing effect on pricing decisions. Specifically, the four variables that we selected are:

- Time from reservation till arrival date.
- The hotel’s remaining capacity at the time of the reservation
- The length of stay, abbreviated as LoS.
- The number of rooms to be reserved (group size).

The final price is given by the product of the reference price and the multipliers, as follows

Price = Seasonal Reference Price*Time multiplier*Capacity multiplier*LoS multiplier*Group Size multiplier

The resulting price will reflect the discounts/premiums resulting from the different values of the influencing variables

A. Components

Initially, Monte Carlo room demand simulator is used. Its function is to simulate all the hotel’s processes, such as reservations arrivals, cancellations, duration of stay, no shows, group reservations, seasonality, trend, etc, as faithfully as possible, projecting current and future room demand. This simulation is based on a probabilistic or stochastic modeling of each of the processes, and all parameters of these processes are estimated from the hotel’s historical data. This simulator will yield a forecast of future arrivals and occupancy. Based on the simulator’s forecasts, the expected total future revenue is estimated, and is passed forward to the optimization module that attempts to maximize this total revenue. The parameters of the price multipliers (e.g. the line slopes or positions) is estimated from the hotel’s historical data are the optimization variables. Once they are obtained, they in turn will determine the suggested price through the

price multiplier formula(1). To determine how the new suggested price will influence demand, a price elasticity function is estimated from historical data. Competitor's prices can be factored in as well. From this relation a new demand factor will be obtained for the price suggested by the optimization algorithm. This demand factor could be greater than 1, representing an excess demand if the suggested price is lower than the reference price, or smaller than 1, representing a reduction demand if the suggested price is higher than the reference price. The new demand will therefore be amplified or attenuated according to this demand factor. Subsequently, the new demand, represented in proportionally higher or lower reservation rate will be taken into account in a new run of the Monte Carlo simulator. This, in turn will produce a new forecast of the future demand, and hence also the expected total revenue. The optimization algorithm will test another set of price parameters, and we continue another iteration of the whole loop (shown in Figure 1). We keep looping around in this optimization/simulator framework for a few iterations, until we end up with hopefully the best parameter selections that lead to maximum total revenue. Once we find the optimal parameters, the final output is the price of every incoming reservation.

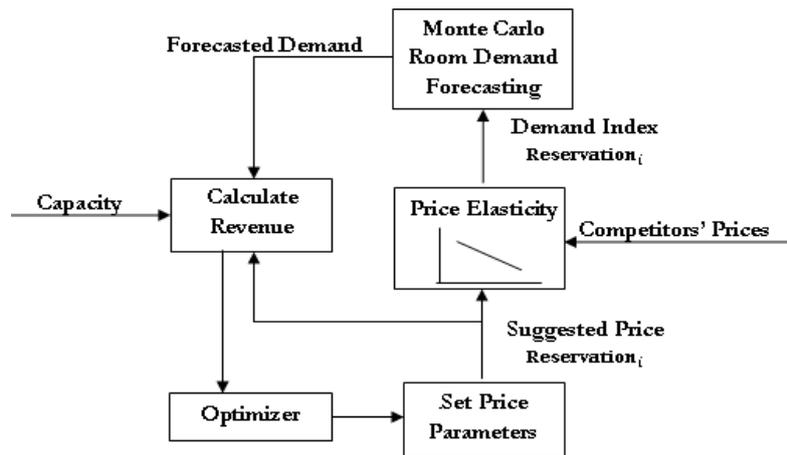


Figure 1: Framework Overview

B. Price Multipliers

The following are the four multipliers that determine the final price. These multipliers are usually taken as linear or piecewise linear functions of the influencing variables, whose levels and slope are determined by the optimization algorithm.

- 1) **Time Multiplier:** The time from the reservation date till the arrival date can be a very important controlling variable for the room's price. Figure 2 shows price multiplier (y-axis) against the time remaining till the arrival (the x-axis, in units of days). At the beginning of the booking horizon room prices should be low in order to quickly fill up the rooms, so the time multiplier will start from a low level. This can be considered as an "early bird" discount. As time goes by and arrival day becomes closer, the discount is gradually lifted, so the multiplier's value increases. This is until a few days before arrival. At this point we are confronted with several vacant rooms that have little prospect to be filled up in the remaining short time. For such situation, it is prudent to price them low, in order to avoid them going non-booked as the target date comes. So the multiplier's value decreases until it reaches its minimum value y_1^T on the arrival day.

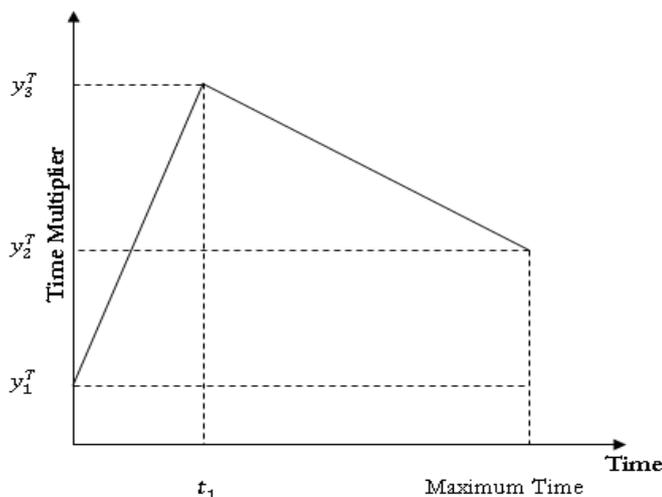


Figure 2: Time Multiplier Curve

- 2) **Capacity Multiplier:** The x-axis represents the total number of vacant rooms for the target date, and the y-axis represents the value of the multiplier. If there are many vacant rooms, then one has to offer some incentives, so the

capacity multiplier is at its lowest value y_1^C . The multiplier's value increases as the remaining capacity of the hotel is decreased, till it reaches the maximum value y_2^C when there are no remaining rooms to be sold. The capacity multiplier has the downward sloping linear shape, as shown in Figure 3:

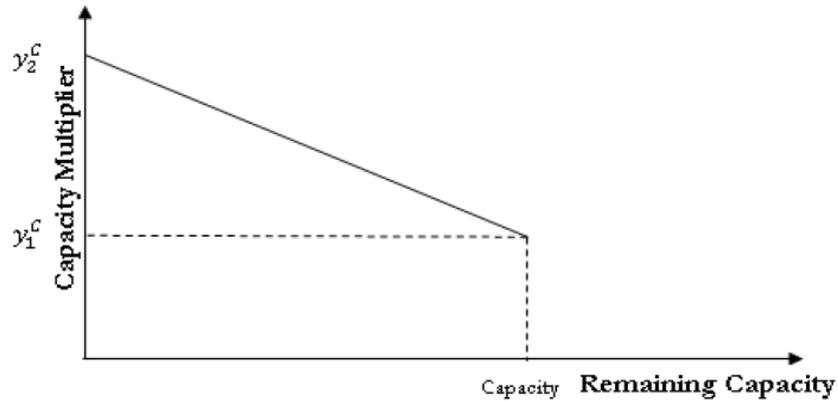


Figure 3: Capacity Multiplier Curve

- 3) *Length of Stay (LoS) Multiplier*: The pricing should be monotonically decreasing with the length of stay, in order to attract longer stays. Figure 4 shows the suggested relation for the multiplier as a function of the length of stay.

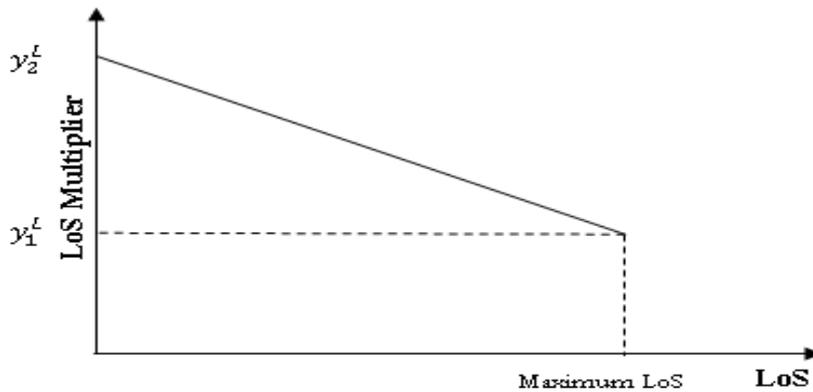


Figure 4: Length of Stay Multiplier Curve

- 4) *Group Size Multiplier*: Figure 5 shows the effect of group size on pricing. It considered the multiplier function which varies linearly from a peak value of y_2^G in case of no group (i.e. a single reservation) to a smaller value of y_1^G for the typical maximum size of a group.

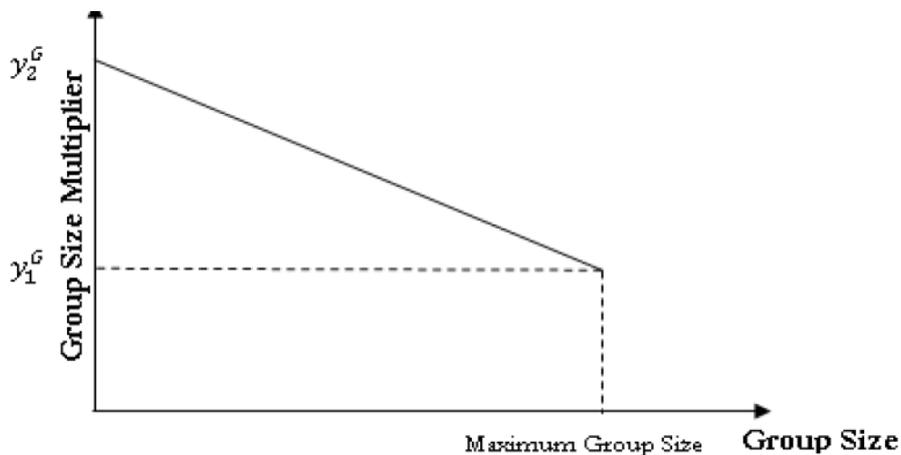


Figure 5: Group Size Multiplier Curve

C. Optimization Variables and Constraints

The optimization variables are essentially the following: *time multiplier-based*: y_1^T , y_2^T & t_1 ; *Capacity Multiplier* y_2^C ; *LoS Multiplier*: y_2^L ; *Group Size Multiplier*: y_2^G . These are the variables that the optimization algorithm will determine such that the revenue is maximized. These relations are the following:

$$y_3^T = 2 - \frac{t_1 * y_1^T + y_2^T * (Maximum\ Time - t_1)}{Maximum\ Time} \quad (2)$$

We have inequality constraints that will guarantee that the multiplier functions are well-behaved and produce logically accepted relations:

$$\begin{aligned} \text{Time Multiplier:} & \quad 0 \leq y_1^T \leq y_2^T \leq y_3^T \\ & \quad 0 \leq t_1 \leq T_0 \\ \text{Capacity Multiplier:} & \quad 1 \leq y_2^C \leq C_0 \\ \text{LoS Multiplier:} & \quad 1 \leq y_2^T \leq L_0 \\ \text{Group Size Multiplier:} & \quad 1 \leq y_2^G \leq G_0 \end{aligned}$$

D. Hotel Simulator Forecasting

To obtain the future revenue that is to be optimized, we need to forecast the future reservations. For this purpose we apply the hotel simulator. In this model we simulate all the processes of the hotel. There are essentially a number of components in this system, which includes Reservations, Seasonality, Cancellations, Length Of Stay(LoS), Group Reservations, Forecasting, Parameter Estimation, Monte Carlo Simulation.

E. Price Elasticity

The optimization procedure obtains the price multipliers, which in turn determine the overall price. This price affects the demand through the price elasticity relation. We use a probit function, to account for the saturation effects for extreme price levels. However, for most of the price range, we are operating in the linear portion of the probit function. The price elasticity relation is given by:

$$\text{Demand Index} = \text{probit}(\text{normalised price} - 1 \div a) + 0.5$$

where normalized price means normalized with respect to the reference price, and demand index is the relative demand (with demand index = 1 corresponding to the mean current demand that we observe when the price equals the reference price). The main price sensitivity variable is the scaling parameter a . It is the parameter affecting the slope of the demand-price relation, and as such it should be negative.

F. Optimization

The optimization has embedded in it the Monte Carlo simulator. The starting point is to consider the hotel's past and present reservations, and apply the hotel simulator forecasting model to generate future reservations. The room prices for these reservations are initially assumed to follow the seasonal average prices (or reference prices) obtained from the historical reservations. Subsequently, the optimization algorithm tests different multiplier values, which will give new room prices.

However, when adding up the individual revenue of each reservation, we obtain the collective effect for the specific choice of multipliers on total revenue. The optimization algorithm keeps searching for different sets of multipliers until it reaches the set that optimizes total revenue.

III. PROGRAM

It includes C programming which can be used by hotels to practice dynamic pricing so as to maximise total revenue. Four functions are used which provide value of each multiplier. These multiplier play an important role in calculation of normalised price. Seasonal reference price are generally fixed by hotel manager and when these multiplier are multiplied with it, a new price is obtained. On execution of this program it calculates revenue each time, and results in maximum revenue. Various variables are used which can be already explained above.

```

#include<stdio.h>
#include<conio.h>
#include<math.h>

```

```

float timeMultiplier();
float capacityMultiplier();
float losMultiplier();
float groupsizeMultiplier();
void main()
{

```

```

int res, t1, i, t_d, t_m, t_y, c_d, c_m, r=50, max_time=30;
float N_P, demand_ind, R_P, fitness ,a, y1l, y2l, y1t, y2t, y3t, y1c, y2c, y1g, y2g, T_R=0.0;

```

```
printf("Enter seasonal reference price, t1, a, y1t, y2t, y2c, y2l, y2g");
scanf("%f %d %f %f %f %f %f %f", &R_P, &t1, &a,&y1t,&y2t,&y2c,&y2l,&y2g");

y3t= 2-(t1*y1t*y2t*(max_time-t1))/max_time;
y1c= 2-y2c;
y1l= 2-y2l;
y1g= 2-y2g;

printf("Enter target date, target month, target year");
scanf("%d %d %d",&t_d,&t_m,&t_y);

for(i=0;i<r;)
{
    printf("enter no of rooms to be reserved");
    scanf("%d", &res);

    if (i+res<=r)
    {
        N_P=R_P* timeMultiplier() *capacityMultiplier() *losMultiplier() *groupsizeMultiplier();
        printf("price is %f,&N_P");
        res_rev= res*N_P;
        printf("total price is %f",&res_rev);
        T_R= T_R+ res_rev;
        i= i+ res;
    }

    else
        printf("not enough rooms, remaining rooms are %d",&(r-i));
}

fitness= T_R;
printf("fitness is %f", &fitness);
return();
}

float capacityMultiplier()
{
    float w;
    w= (y1c-y2c/r)*(r-i)+y2c;
    return(w);
}

float losMultiplier()
{
    float x, los, m_l=10;
    printf("Enter length of stay");
    scanf("%d",&los);
    x= (y1l-y2l/m_l)*los+y2l;
    return(x);
}

float groupsizeMultiplier()
{
    float y,m_g=5;
    y=(y1g-y2g/m_g)*res+y2g;
    return(y);
}

float timeMultiplier()
{
    float z, diff, t_d, c_d, t_m, c_m,t1;
    printf("enter current month, current date");
    scanf("%d %d", &c_m,&c_d);
```

```
if (t_m==c_m)
    diff = t-d-c_d;
else
{
    switch (c_m)
    {
        case 4||6||9||11:
            diff = (30-c_d)+t_d;
            break;
        case 2:
            if (t_y/4==0)
                diff = (29-c_d)+t_d;
            break;
        else
            diff= (28-c-d)+t_d;
            break;
        default:
            diff = (31-c_d)+t_d;
    }
}
if (diff<t1)
    z= (y3t-y1t/t1)*diff + y1t;
else
    z= (y2t-y3t/m_t-t1) * diff;

return(z);
}
```

Figure 6: C program of fitness function

IV. Conclusions

In this work, we explained hotel room dynamic pricing system. This model uses the concept of price multipliers that provide a varying discount/premium within some bands over some seasonal reference price. The transparent way of designing such system, including the knowledge of the variables that affect the pricing, will allay the hotel's concerns regarding the uncertainty of system's outcome. Moreover, some of the relations regarding the four influencing variables (hotel capacity, time till arrival, length of stay, and group size) can be adjusted or removed, according to the hotel's request.

References

- [1] Andrawis, R. and Atiya, A. (2009) A new Bayesian formulation for Holt's exponential smoothing. *Journal of Forecasting*. 28(3): 218-234.
- [2] Auger, A. and Hansen, N. (2005) A restart CMA evolution strategy with increasing population size. In: *Evolutionary Computation, 2005. The 2005 IEEE Congress on.*, pp.1769--1776.
- [3] Fibich, G., Gavious, A. and Lowengart, O. (2005) The dynamics of price elasticity of demand in the presence of reference price effects. *Journal of the Academy of Marketing Science*. 33(1): 66-78.
- [4] Nagrath, I.J. and Gopal, M. (2005) *Control systems engineering*. New Age International
- [5] Zakhary, A., Atiya, A., El-Shishiny, H. and El-Gayar, N. (2009) Forecasting hotel arrivals and occupancy using Monte Carlo simulation. *Journal of Revenue & Pricing Management*. 10(4): 344-366.