

# A Reservation Aggregation Framework Design for Demand Estimation

T. Vítek, D. Pachner

*Effective management practices in the tourism and hotel area have seldom been more important than at the present time. Pricing decisions cannot be taken without serious thought. IT has provided the opportunity for a customer to make a quick market search and it offers decision support systems that can be used in the hotel management. The heart of yield management system consists of the predicting machine, which estimates the number of incoming reservations. Incoming reservations arrive randomly in time. The time series calculi as well as the estimators known from control engineering require properly defined time rows (with a constant period). This requirement is usually not fulfilled, so the input data are not exploited properly. This paper outlines a procedure that aggregates the reservations into a time series that is useful for demand prediction. The algorithm prepares the data systematically for further processing. Any method that process time rows can be used for subsequent prediction: time series, linear models or time extrapolation.*

*Keywords: Yield management, demand prediction, time rows, data proceeding.*

## 1 Introduction

The Internet and other technologies enable a customer to find an appropriate substitution for a service, if he is not satisfied with the pricing and the quality of the given offer. Hence, the service provider must find the optimum market position. From the customer's point of view, price is often the essential feature of a particular. On the other hand, a hotel wants to keep its unit price at the highest possible level. The definition of the expedient price can vary from business to business. A hotel company rarely has a monopolistic or oligopolistic position. A competitive market should therefore be assumed.

The yield mechanism can be interpreted, from a technical point of view, as a decision support system that helps to find the optimum pricing level for a particular product. The greatest advance is that it suggests the pricing automatically. The hotel management can then either accept or modify the offered scenario. A yield management (YM) system architecture – historical data processing – estimation and modeling of the future demand and price optimization, needs to be developed. The new idea presented in this paper involves an analogy between the YM approach and the model-based predictive control approach. This new concept leads to more appropriate demand forecasting and control. While traditional YM demand predictions are based on historical demand data only, the new idea suggests compound predictions based on both demand and price data simultaneously. This new approach recovers hidden dependencies in the historical data and should provide better justified demand predictions.

## 2 Ordering of incoming reservations

The next section describes the booking system structure and the time dependency of incoming reservations is analyzed. Finally a structure that enables analysis of time trends is introduced. In the following text, the term “reservation” is often used. A reservation is basically a data record consisting of the date, for which the reservation is arranged, the number of persons, the agreed price for a person and the time at which the deal was arranged. In the next paragraph only the

date of stay, the number of persons and the arrangement date are to be considered. The price factor will be considered in a separate section.

## 3 Historical reservations

A reservation can appear at any time  $t$  or advance  $s$ . For a statistical row analysis we need the time row format. Therefore a discretisation system for time  $t$  and for the order of advance  $s$  must be stated. The task is not easy, as this is a two dimensional problem. First of all the sampling period for time  $t$  must be stated. The system is supposed to plan the prices for a night in a hotel. This implicitly defines the sampling period of time – the sampling moment is the check-in time. The sampling period is one day, as there is only one check-in per day. The reservations must be handled separately for each day in the calendar. The advance ordering period  $s$  is thus the time interval before one particular check-in time, as in Fig. 1. The advance is zero at check-in time and it is positive – the earlier the reservation the greater  $s$  will be.

As was mentioned above, the reservations for a particular day are discontinues function, not a smooth function. It is difficult to compare, analyze and build a model in such conditions. For this reason, the reservations must be aggregated in some well-selected intervals, which will enable us to compare and analyze trends. The user must define these bounds for intervals. The use of defined aggregation enables us to comparison to be made between two situations. An example of such an aggregation is shown in the Fig. 2.

The definition of aggregation can be intermediated by the introduction of indices  $r_i$ . These are the bounds of the intervals in which we want to sum the reservations. It is very important to assign the reservation to one specific index  $r_i$ . The appropriate assignment is shown in Fig. 2, where an arrowhead marks it. It represents the assignment of reservations in the interval between two successive indices. These intervals are also used in real booking system applications. The standard categories of reservations are as follows.

- 1) check-in on or even after day closure – directly on  $r_0$  or later that night
- 2) check-in between  $r_0 - r_1$  hours in advance

- 3) check-in between  $r_1-r_2$  hours in advance
- 4) check-in between  $r_{n-1}-r_n$  hours in advance
- 5) check-in before  $r_n$

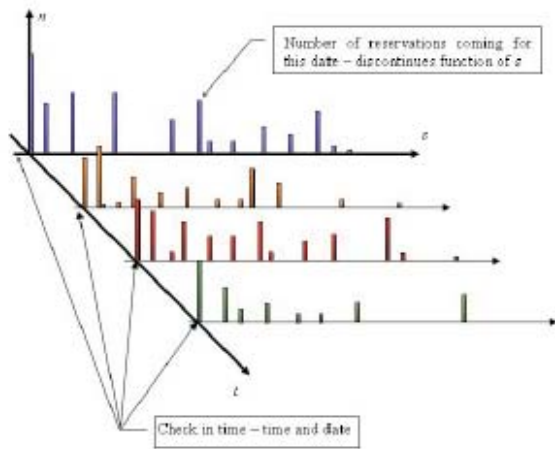


Fig. 1: Reservations coming for the days as a discontinuous function of  $s$

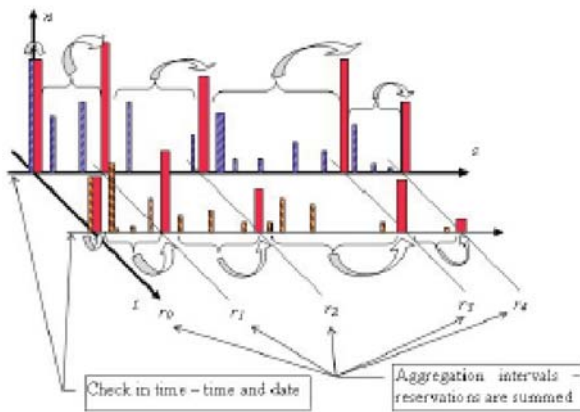


Fig. 2: Aggregation of reservations in time intervals defined in the  $s$  variable

To formulate the previous mapping into a suitable form, the indices  $r_i$  must be appropriately defined. The times defined by user represent the bounds of the interval. The set of times must be given in ascending order. The first index is assumed to be zero. In order to monitor all incoming reservation, one index  $r_{n+1}$  should be implicitly added to the end of this set. This index will contain all the reservations coming before the index  $r_n$ . This induces the following set of time indices

$$R_g = \{r_0, r_1, r_2, \dots, r_n, r_{n+1} \mid \forall i, j \in N, i < j \Rightarrow r_i < r_j\}. \quad (1)$$

An example of such a set is

$$R_g = \{0 \text{ hours}, 6 \text{ hours}, 12 \text{ hours}, 1 \text{ day}, 5 \text{ days}, 5+ \}. \quad (2)$$

All data coming after day closure should be assigned to index  $r_0$ . It should be noted that the maximum delay is 24 hours, because then it is already the next day. The previously-defined structure can be used for defining the projection of the reservations onto the set of indices.

### 4 Representative price for an interval

The aggregation procedure described in section 3 gathers a series of reservations from a particular interval for one value. This value quantifies the interest of customers in that particular period as a total number of buys. Each of these sales can generally have a different selling price. This algorithm will introduce a single representative price that will be attractive to an average buyer in this interval. Fig. 3 displays the situation between two time indices  $r_i$  and  $r_{i+1}$ . This implies that a price representing a time span should be counted, as some kind of weighed average throughout the interval. The yield from a given period is known and constant, as is the number of reservations, so the right representative is that, which preserves a constant yield from a given set of reservations.

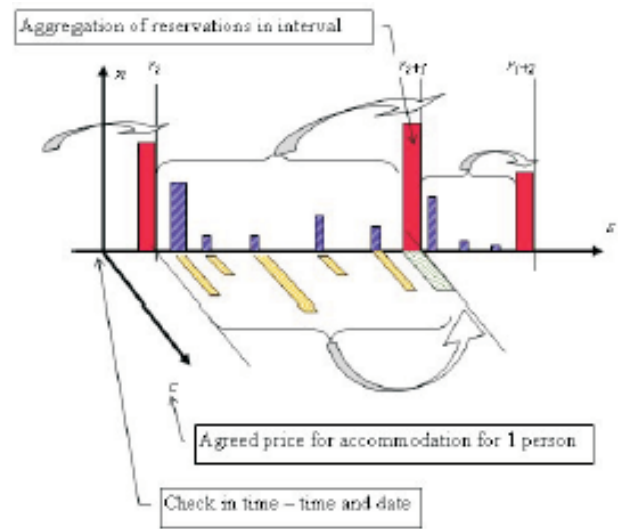


Fig. 3: Reservation as a combination of price and number of persons – visualization of aggregation in prices and reservations

This equation can be formulated in an integral or discrete version, as

$$\bar{c}(r_k) \int_{r_{k-1}}^{r_k} n(s) ds = \int_{r_{k-1}}^{r_k} c(s) n(s) ds. \quad (3)$$

The right side of the equation represents the overall revenue in the particular period. The left side is equal to the average price multiplied by the number of reservations in the period. Hence the equation expresses the balance between sales counted from the dates and sales counted from the interval representative price. This representative price can be further expressed from equation (3) as follows

$$\bar{c}(r_k) = \frac{\int_{r_{k-1}}^{r_k} c(s) n(s) ds}{\int_{r_{k-1}}^{r_k} n(s) ds}. \quad (4)$$

Equation (4) represents the weighted average of the prices, where the weight for each particular price is the number of reservations that were sold for this price.

### 5 Mapping reservations at the border of the future and the past

Section 3 considers that all reservations are available when they are to be analyzed. This kind of post processing enabled aggregation intervals of any size to be defined (1). These intervals reflect the specification of the hotel management concerning the moments, at which price can be changed. However the connected aggregation of reservations in these intervals supposes that of all data in particular interval is known. This system works well with historical data, where the reserved day is in the past, but it cannot be used unchanged when dealing with reservations for future days. First of all, it is necessary to recover what data is known at an arbitrary time denoted as  $\tau$ . Fig. 4 displays a time period where the data is already known, as is the future period in which the information about reservations and prices remains unknown. It can be

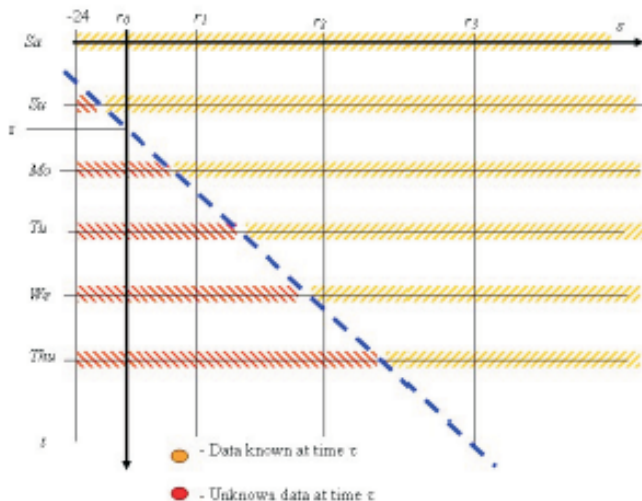


Fig. 4: Reservations known at time  $\tau$  – problem with interval limits

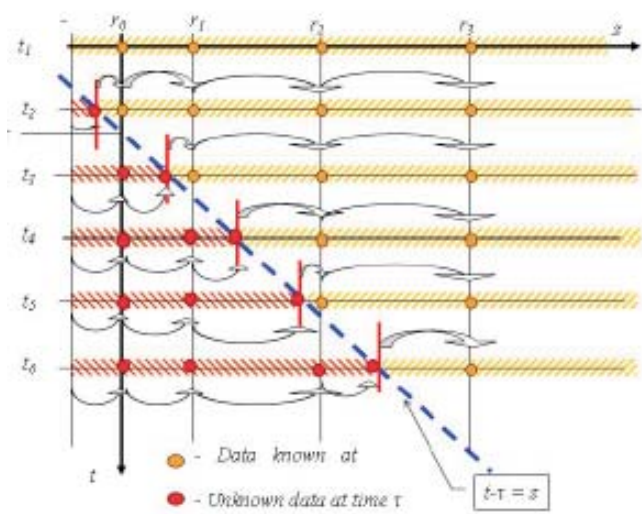


Fig. 5: The intersection of the actual time line and a particular date is a new interval limit

clearly seen that each of the days that border a time border has a different ratio of known and unknown history. Hence the desired set of time indices (1) cannot be used directly. It is necessary to define one extra index for each of rows different that will separate known past and unknown future data.

The situation is displayed in Fig. 5. As can be seen, the actual time  $\tau$  divides the time plane of variables  $t$  and  $s$  into two half planes. One of them is filled with known data, while the other remains unknown. The separating line is well known, determined by the current time  $\tau$ . The condition that determines the points on this line is

$$t - \tau = s. \tag{5}$$

The reservations for each of the days  $\{t_1, t_2, t_3, \dots\}$  is defined on the line that goes through the particular day ( $t_i$ ) and it is parallel to the  $s$  axis. The separating line intersects the line of an individual day at only one point. This is the point that separates known and future reservations. Therefore it can be an appropriate limit for an interval, at which the reservations can be aggregated.

The previous paragraph implies that each day should have its own set of indices. The user gives the original set of indices. One index is inserted later, and its position is dependent on the current time  $\tau$ . The intersection  $r_m$  can be computed, from current time  $\tau$  and the particular date for which a reservation should be made  $t_1$

$$r_m = t_i - \tau. \tag{6}$$

As shown in Fig. 5 the intersection  $r_m$  can be either in the past for  $t_i + 24 \leq \tau$ , or in the future. As shown in the figure, only the future reservation should be separated by the middle index  $r_m$ . For one particular date, the set of indices can be created as a union of the set of desired indices (1). For a particular date the set can be created by ordering the union

$$R_e = \{r_0, r_1, r_2, \dots, r_n, r_{n+1}\} \cup \{r_m\}. \tag{7}$$

The values are known in those intervals where the lower limit is greater or equal to  $r_m$ . This condition splits set defined by condition 7 into two sets. One of them  $R_p$  contains the limits to intervals with known data, while the other set  $R_f$  contains the interval limits for unknown data. These two sets can be created by ordering the following sets

$$R_f = \{r_i \in R_e \mid r_i \leq r_m\}, \tag{8}$$

$$R_p = \{r_i \in R_e \mid r_i \geq r_m\}. \tag{9}$$

The set of indices  $R_p$  enables aggregation of reservations in intervals of already known data. The set contains the limits of the intervals. Aggregation of the number of reserved places must be performed for all neighboring limits. The past indices induce the following occupation and price aggregations for all elements of the ordered set  $R_p$

$$\forall r_i, r_{i+1} \in R_p \quad r_i \leq r_{i+1},$$

$$\bar{n}(r_{i+1}) = \int_{r_i}^{r_{i+1}} n(s) ds,$$

$$\bar{c}(r_{i+1}) = \frac{\int_{r_i}^{r_{i+1}} c(s)n(s) ds}{\int_{r_i}^{r_{i+1}} n(s) ds}.$$

Future aggregations remain unknown. But they can be defined very similarly.

## 6 Conclusion

The outlined algorithm enables standard reservation data to be processed into a proper time row format. The data is commonly available in a hotel database. The idea must be transformed into an SQL macro that generates the time row for each required day. The algorithm can be successfully exploited by various estimation methods that compare the trends for individual days. The procedure enables a comparison of time trends. This requirement is essential for successful demand prediction.

## 7 Acknowledgments

The research described in the paper was funded by the Czech Ministry of Trade under grant No. FI-IM/189.

## References

- [1] Luciani, S.: Implementing Yield Management in Medium Sized Hotels: an Investigation of Obstacles and Success in Florence Hotels. *Hospitality management*, Vol. **18** (1999), p. 129–142.
- [2] Tallury, K. T., Van Rizin, G. J.: *The Theory and Practice of the Revenue Management*. Kluwer Academic Publishers, 2004. ISBN 1-4020-7933-8.
- [3] Donaghy, K., McMahon, U., McDowell, D.: Yield Management: an Overview. *Hospitality management*, Vol. **14** (1995), p. 139–150.
- [4] Kimes, E. S.: Revenue Management: A Retrospective. *Cornell Hotel and Restaurant Quarterly*, Vol. October–December (2003), p. 131–138.
- [5] Weatherford, L. R., Kimes, S. E., Scott, D. A.: Forecasting for Hotel Revenue Management Testing Aggregation against Disaggregation. *Cornell Hotel and Restaurant Administration Quarterly*, Cornell University Vol. **3** (2001), p. 53–64.
- [6] Francis, G., Fidato, A., Humphreys, I.: Airport-Airline Interaction: the Impact of Low-Cost Carriers on Two European Airports. *Journal of Air Transport Management*, Pergamon Vol. **9** (2003), p. 267–273.

---

Ing. Tomáš Vitek  
phone: +420 737 109 868  
e-mail: vitekt2@centrum.cz

Daniel Pachner  
phone: +420 224357206  
email: daniel.pachner@seznam.cz

Department of Control Engineering  
Czech Technical University  
Faculty of Electrical Engineering  
Technická 2  
166 27 Praha 6, Czech Republic