HotelSimu: a Parametric Simulator for Hotel Dynamic Pricing

Andrea Mariello*, Manuel Dalcastagnè, Mauro Brunato, Roberto Battiti

Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9 I-38123, Povo (TN), Italy

Abstract

In Hotel Revenue Management, an optimal pricing policy is crucial for maximizing the profit. However, the complexity of the involved processes makes the definition of effective and efficient models a challenging task. We propose an efficient hotel simulator, HotelSimu, which generates events by using Monte Carlo simulations. Differently from previous works, cancellations and reservations are interleaved, and they are generated according to parametric models that cover several scenarios. Seasonal averages can be set on a daily basis. The hotel registry changes after each single event, and the price of each reservation is set dynamically. The optimal pricing policy is retrieved by using black-box optimization. The applicability of the simulator is evaluated in a real scenario involving 10 structures in Trento, Italy. The adoption of optimized pricing policies based on our simulator leads to an average revenue increase of $\approx 19\%$ with respect to policies with fixed prices.

Keywords: Revenue Management, Dynamic Pricing, Hotel Simulation, Simulation-based Optimization

1. Introduction

Information Technology (IT) drastically changed how people plan and manage travels. The Internet revolutionized the way travelers can get information about trips and hotels. Tools such as search engines, online travel agencies or price comparison websites are now used for travel planning by people of different generations [1]. As a consequence, potential tourists can take decisions which are much more informed than before, and companies need to be competitive in order to survive. In fact, many organizations in tourism use IT to propose more convenient prices online, thus reducing commissions that would be given

*Corresponding author

Email addresses: `andrea.mariello@unitn.it` (Andrea Mariello), `m.dalcastagne@unitn.it` (Manuel Dalcastagnè), `mauro.brunato@unitn.it` (Mauro Brunato), `roberto.battiti@unitn.it` (Roberto Battiti)

to intermediaries [2]. Proposing the right price to potential customers is a very important aspect, with a direct impact on the revenue of companies.

Within the hospitality realm, any enhancement that can increase revenues might yield huge effects on profit. Indeed, many works in the literature focus on revenue management (RM), which considers diverse possibilities for optimizing revenue, from marketing to price control. Optimization problems related to RM are usually expressed following two main approaches [3]: capacity control [4, 5, 6, 7] and dynamic pricing [8, 9].

In capacity control, the decision variable is the amount of offered supply. In contrast, in dynamic pricing the decision variable is price. Recent works mainly focus on dynamic pricing, to propose different price offers for a limited set of products with fixed capacities. In this paper, we present a novel approach for the maximization of revenue based on dynamic pricing. We follow this method to simulate what happens on online booking platforms, where the same room can be sold at different prices according to factors like the length of stay, the time to arrival, the season and the occupancy level.

Dynamic pricing problems have been solved using various strategies [10, 11], including rule-based frameworks [12], linear programming [13], integer and dynamic programming [14].

The demand is considered to be deterministic [15] or stochastic [14]. However, many techniques adopt the simplifying assumption that the demand is independent from the chosen policy. More complex scenarios, where demand can be influenced by other factors (e.g. price), are more difficult to handle and closed-form solutions are rarely available [16]. In addition, if sold rooms become available again before they are consumed by the customer (e.g. via stochastic cancellations), the dimension of the optimization problem grows exponentially and approaches like dynamic programming are effective only in specific cases [17]. A possible solution to mitigate the complexity of the model is given by approaches based on approximated dynamic programming [14, 18]. To reduce the computational cost, a linear relaxation of the original problem is considered. However, such approximated models are not flexible enough to include stochastic cancellations interspersed with reservations.

If problems related to hotel RM scenarios are computationally too complex to be solved exactly, the approximate maximization of revenue can be achieved by using simulation-based optimization [19, 20]. In this way, the analytical model can be substituted with a simulator of many inter-related processes like reservations, cancellations, no-shows and walk-in customers. To define simulators of complex systems like a hotel booking system, an effective technique is Monte Carlo simulation [21]. Generated events such as reservations and cancellations lead to a distribution of possible revenues. The expected value of that distribution is then considered as the variable to be maximized.

In [22, 8], a Monte Carlo approach is employed to simulate the demand. New reservations are forecast according to the distribution learned from history and by means of additive/multiplicative pick-up [23] or exponential smoothing [24, 25]. A similar approach can be used also with other forecasting models, such as regression or moving average models [26].

In [8], the effect of the price on the demand is also considered: lower and higher prices are associated with higher and lower acceptance probabilities, respectively. The pricing policy is modeled by a set of multipliers which can increase or decrease the price of a simulated reservation with respect to the average price learned from data. The parameters of the pricing model are optimized by using CMA-ES [27, 28].

However, the aforementioned Monte Carlo method is based only on historical data, and it does not provide a simple way for the user to run *what-if* analyses. In real scenarios, it is very common that the distributions characterizing the demand change in response to mutated environment conditions. For example, it is highly probable that the pricing policy of a hotel should change when massive events like concerts, exhibitions or sport events happen nearby. The hotel manager would benefit from an automated system that exploits historical data whenever possible, and at the same time allows to readily inject new valuable information as soon as it becomes available in order to test different scenarios. Another limitation of [22, 8] is that cancellations are modeled as a set of events that are independent from reservation requests. In fact, cancellations are realized before the generation of new reservations. Consequently, the state of the hotel registry (e.g. the room availability) does not change dynamically after each event. The first-generated reservation would see an availability higher than the one that would be provided if all events are interspersed, as in a real booking scenario. Moreover, one cannot simulate the case of a customer booking a room and then canceling later during the same day (or even immediately after).

We propose a system that simulates the hotel booking process by considering interspersed reservations and cancellations, and that defines arrivals using parametric models. In more detail, the main advantages of our simulator include the following.

- The arrivals of new events, reservation requests as well as cancellations, are defined as parametric models. The parameters can be estimated from data or given directly by the hotel manager. We define models that can describe different realistic scenarios, and at the same time allow straightforward *what-if* analyses.
- The reservation requests and cancellations are not grouped into disjoint sets of events, but occur in an interleaved way. This procedure simulates the reservation process as it happens on online booking platforms, where reservations are not buffered but generated and finalized immediately.
- The exact dates to be simulated as well as the opening dates of the hotel can be chosen without any fixed time windows. As an example, a user can simulate a specific opening period, and reservations for dates completely or partly out of the imposed bounds are rejected.
- The seasonal averages for reservations, cancellations, nights, rooms and prices can be set on a day-by-day basis. Our simulator is not tied to fixed seasonality schemes, which most of the times contemplate only two

seasons (high and low) or three seasons (high, medium and low). This allows the hotel manager to define a pricing policy that adapts to different time frames during the year, including specific sets of days associated with special events.

- The core modules of the simulator are implemented in C++, and they are characterized by a limited memory footprint. Experiments covering more than one year and a half of reservations show that optimized policies can be retrieved in less than 6 hours on low-end machines for a medium-size hotel.

The structure of the remainder of this paper is as follows. Section 2 describes our system and its main components. Section 3 presents in more detail the proposed parametric models of reservations and cancellations. Section 4 shows the applicability of our simulator to a set of structures in Trento, Italy. Results show that our approach leads to an average revenue increase of $\approx 19\%$ with respect to policies with fixed prices.

2. Dynamic pricing based on fine-grained simulation-based optimization

In hotel revenue management, it can be difficult to model all the inter-related processes that are present in a real scenario. Therefore, most research has focused on simplified and more tractable views of the world. If one is not bound to use models that provide an exact solution as with the traditional dynamic programming approach, more realistic and fine-grained models can be defined by using simulation-based optimization. Even though only an approximated solution can be found, it is evident that any improvement in flexibility and performance, with respect to traditional approaches, can better support the decision making of the hotel manager.
We propose a simulator (HotelSimu) based on fine-grained discrete event simulation that can be effectively employed for optimizing a hotel dynamic pricing policy to maximize revenue. It is based on a set of parametric models, which can be used even by non-experts for the analysis of different scenarios. The optimization problem is formulated according to a dynamic pricing approach. A price is proposed for each reservation request, and the customer is free to accept or reject the offer.

2.1. Definitions

Let us now fix the notation and define the main concepts, before the general description of the system in Section 2.2.

Definition 1. A *reservation request* (RR) is an event characterized by the following features:

- the *reservation day* (RR_{res}), which is the day the request occurs;

- the *arrival day* (RR_{arr}), which is the day the customer arrives at the hotel;
- the *length of stay* (RR_{los}), which is the number of nights reserved;
- the *size* (RR_{size}), which is the number of rooms reserved.

Definition 2. A *reservation offer* (RO) is an admissible reservation request (for which there is room availability) characterized by the *price* (RO_{price}) proposed by the hotel.

RO_{price} depends on several factors, including the features of RR, the time a request arrives, the presence of extra services or the price proposed by competitors.

Definition 3. An *accepted reservation* or simply *reservation* (R) is a reservation offer accepted by the customer.

An accepted reservation is registered on the hotel registry, thus effectively changing the room availability.

Definition 4. The *acceptance probability* of a reservation offer ($\Pr_{\text{accept}}(RO)$) is the probability that a customer accepts RO and the proposed price, and therefore is equal to the probability that RO is registered on the book.

Definition 5. The *state of the hotel* $S(t)$ is defined as the state of the booking registry at time t, which corresponds to the historical records up to t as well as the set of reservations (for future arrival days) that appear in the registry at time t.

Before introducing formally the concept of booking horizon, we need to define the concepts of distance between two days and of time-to-arrival.

Definition 6. Given two days identified by $i, j \in \{0, 1, 2, \dots\}$, the number of days between i and j, or their *distance*, is:

$$d(i, j) = d(j, i) = |i - j| \geq 0.$$

Definition 7. Given a reservation R, the *time-to-arrival* of R is:

$$R_{\text{TTA}} = d(R_{\text{res}}, R_{\text{arr}}).$$

A similar definition is also valid for reservation requests and offers.
$R_{\text{TTA}} = 0$ can represent two possible events. One is the event of a customer that books one or more rooms for one or more nights starting from the night of the same day of the reservation. The other is the event of a customer that arrives at the hotel with no reservation and asks for one or more rooms. In the literature, the customer associated with the second event is usually called a *walk-in* user. In HotelSimu we do not distinguish the customers associated with the previous events, and we refer to them as *walk-in* users.

Definition 8. The *booking horizon* (*BH*) is the maximum time-to-arrival allowed by the hotel.

One of the main distinctive features of our approach is the management of cancellations, which are defined as follows.

Definition 9. A *cancellation* (C) is an event characterized by the following features:

- the *cancellation day* (C_{day}), which is the day the event occurs;
- the *reservation* (C_{res}), which is the reservation on the book that is canceled by the customer.

When a reservation is canceled, it is removed from the hotel registry, and the associated rooms can be booked by other customers.

Definition 10. The *cancellation probability*, t days before arrival of a reservation R ($\text{Pr}_{\text{cancel}}(R,t)$), is the probability that the customer associated with R cancels it exactly t days before arrival, with $t \in [0, R_{\text{TTA}}]$.

According to the previous definition, the probability that R is canceled within its lifetime is

$$\text{Pr}_{\text{cancel}}(R) = \sum_{t \in [0, R_{\text{TTA}}]} \text{Pr}_{\text{cancel}}(R,t). \tag{1}$$

Our simulator also allows one to consider opening and closing periods, and to simulate reservations accordingly.

Definition 11. The *reservation requests horizon* (RH) is the set of all the reservation days to be simulated. It corresponds to the values that each R_{res} can assume during the simulation.

Definition 12. The *arrivals horizon* (AH) is the set of all possible arrival days. It corresponds to the values that each R_{arr} can assume during the simulation.

Definition 13. The *optimization horizon* (OH) is the set of arrival days for which there is the need of an optimal dynamic pricing policy to maximize revenue.

Most hotel managers are interested in maximizing the total profit and not the revenue. In this paper, we assume that the hotel has only fixed costs, and therefore maximizing the revenue corresponds to maximizing the profit. The simulator can include variable costs, but its analysis is out of the scope of this paper.
AH represents the opening period of the hotel, and reservation requests including days outside its bounds are rejected. RH and AH can overlap (entirely or partly) or be disjoint. OH is instead required to be a subset or to be equal to AH. The total revenue that drives the optimization procedure is evaluated on OH only. This provides more flexibility to the hotel manager, which is able to simulate and compare different scenarios, including different opening periods and booking horizons.
The main models and components defined above are illustrated in Figure 1.

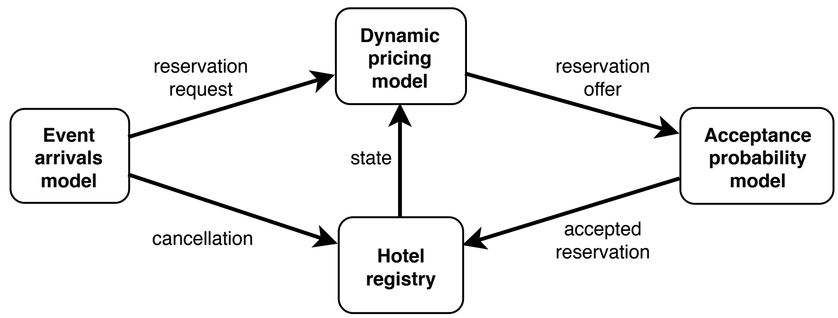

Figure 1: Dependency graph of the hotel registry and of the models used by HotelSimu.

2.2. System overview

HotelSimu includes several components that simulate the hotel booking process, as described in the previous Section, and that define and optimize the pricing policy.

As depicted in Figure 2, the system includes:

- an event generator, which simulates reservation requests created by the customers as well as cancellations;
- a registry, which stores the information about the state of the hotel, in particular accepted reservations and room availability;
- a dynamic pricing model, which proposes an offer for each reservation request;
- an acceptance probability model, which simulates the stochastic process by which customers accept or discard reservation offers;
- an optimizer, which searches for the optimal pricing policy to maximize revenue.

For each simulated reservation day $r \in \mathrm{RH}$, a random sequence of $\mathcal{C}_r$ cancellations and $\mathcal{R}_r$ reservation requests is generated. Each reservation request is associated with an arrival day $a \in \mathrm{AH}$ following or coinciding to r ($a \succeq r$). Each cancellation is instead associated with a specific registered reservation.

The proposal of a price depends on a reservation request and on the state of the hotel at the moment the event occurs. Since reservation requests and cancellations are interspersed, our procedure covers also the case of a customer reserving a room and, later on the same day, deciding to cancel it, maybe because of an error or because a better offer has been found.

Once a price has been proposed to the customer, a reservation is accepted according to the acceptance probability model. It is then registered into the hotel registry and, if a cancellation does not occur until the end of the simulation, it is considered in the evaluation of the total revenue to be passed to the optimizer.

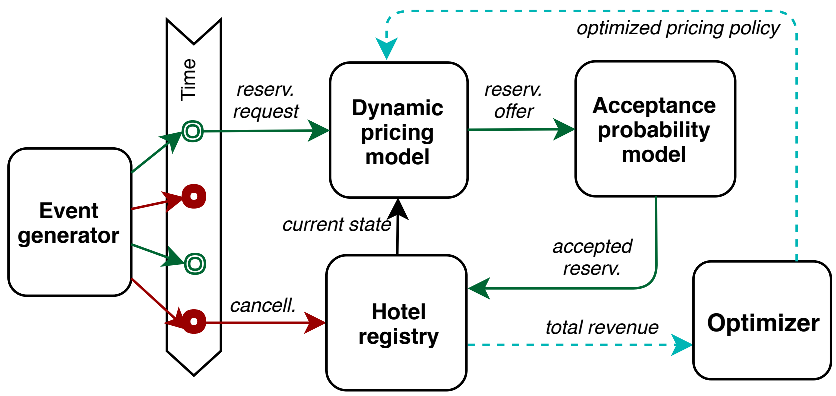

Figure 2: HotelSimu overview. Reservation requests and cancellations are interspersed. The state of the hotel after one complete simulation is used by the optimizer to compute the total revenue and adjust the pricing policy.

As concerns the optimization, one objective function evaluation corresponds to the average total revenue of several simulation runs, with respect to the reservations recorded in the registry within the OH.

3. Parametric models of reservations and cancellations

The simulator in [22] does not make any assumption on the distribution of reservation requests and cancellations. Reservation and cancellation models are learned completely from data. The hotel manager can only provide information about seasonality, in terms of duration and average prices.

To simplify the model in a way that it can be effectively used by hotel managers even when historical data are not available or difficult to retrieve, we propose a parametric approach, by which the knowledge of domain experts can be exploited by using only a limited set of synthetic indicators. While it is almost impossible for the hotel manager to provide the entire reservation model without reverting to automated solutions relying on history, it is a more viable option to ask for a synthetic indicator like the average number of reservation requests arriving in a certain period. The hotel industry often has aggregated information to be exploited to maximize revenue. A parametric approach can greatly simplify the use of a simulator by people that are not researchers in the revenue management community.

We propose a set of parametric models that can be used starting from the expected number of reservations and cancellations, expected length of stay and number of rooms. They are flexible enough to cover several, and sometimes very different, scenarios, with parameters that can be fit to data or passed directly by the hotel manager, as described in the following sections.

3.1. Simulation of reservation requests

Let $\mathcal{R}_r^a$, $r \in \mathrm{RH}, a \in \mathrm{AH}$, be the number of reservation requests generated on day r that are associated with arrival day a.
The total number of requests generated within RH and associated with one arrival day is therefore given by:

$$\mathcal{R}^a = \sum_{\substack{r \in \mathrm{RH} \\ r \preceq a}} \mathcal{R}_r^a, \tag{2}$$

where $\preceq$ describes the relation *precedes or coincides to*.
The expected total number of reservation requests associated with one arrival day can be seen as the result of several independent processes, which occur on each simulated day within the BH of an arrival day:

$$\mathbb{E}[\mathcal{R}^a] = \Lambda(a) = \sum_{i=0}^{\mathrm{BH}} \lambda(i, a), \tag{3}$$

where $\lambda(i, a)$ is the expected number of reservation requests occurring i days before the arrival day a.
If historical data are available, one can estimate directly $\lambda(i, a)$ for each i and a. To avoid the computational load of a point-wise estimation, and to facilitate what-if analyses, we define each $\lambda(i, a)$ by the following parametric model:

$$\begin{aligned} \lambda_\alpha(i,a) &= \Lambda(a) \times Q_\alpha(i, \mathrm{BH}) \\ &= \Lambda(a) \times \left(\left(\frac{\mathrm{BH}+1-i}{\mathrm{BH}+1} \right)^\alpha - \left(\frac{\mathrm{BH}-i}{\mathrm{BH}+1} \right)^\alpha \right), \end{aligned} \tag{4}$$

with $i = 0, 1, \ldots, \mathrm{BH}$, $a \in \mathrm{AH}$, and for any parameter $\alpha > 0$.
The expression of $Q_\alpha(i, \mathrm{BH})$ is similar to that of the RIM quantifiers proposed in [29], after reflection and translation. We use $Q_\alpha(i, \mathrm{BH})$ because:

- they define a function with discrete domain and continuous values;
- they sum up to 1:

  $$\sum_{i=0}^{BH} Q_\alpha(i, \mathrm{BH}) = 1,$$

  for any $\alpha > 0$ and therefore can represent a discrete probability distribution or a normalized curve;
- they can model different reservation scenarios through α, from a linear curve ($\alpha = 1$) to an exponential-decay curve with different decay factors (see Figure 3);
- they provide a simple way of finding α from the ratio of walk-in users with respect to the total number of reservations, that is, $Q_\alpha(0, \mathrm{BH})$.

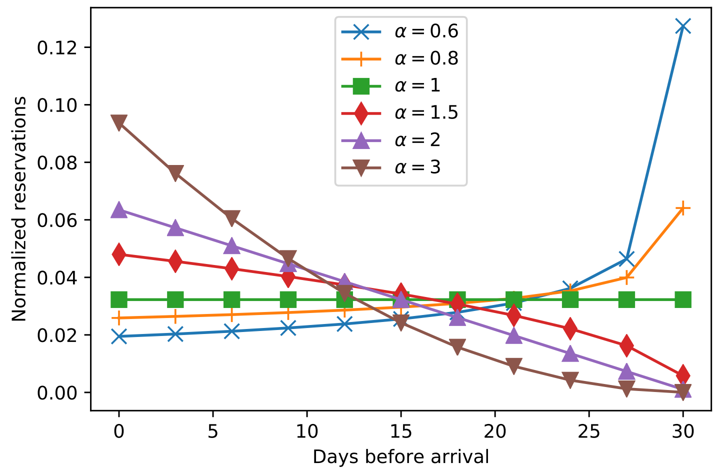

Figure 3: $Q_\alpha(i, \mathrm{BH})$ for $\mathrm{BH} = 30$ and for different values of α.

In our current implementation, we assume that the reservation requests follow a non-homogeneous Poisson process with an expected value given by our parametric model:

$$\mathcal{R}^a \sim \mathrm{Poisson}(\Lambda(a)). \tag{5}$$

Therefore, reservation requests are generated for each simulated day according to the following model:

$$\begin{cases} \mathcal{R}^a_r \sim \mathrm{Poisson}(\lambda_\alpha(i, a)) & \text{if } i \leq \mathrm{BH}, \\ \mathcal{R}^a_r = 0 & \text{otherwise.} \end{cases} \tag{6}$$

Poisson processes are usually chosen to model arrival processes [30] and, in our context, they can represent the arrival of reservation requests with a minimum set of parameters. In [22], a binomial distribution is used, with additional constraints on the variance of samples in order to set the success probability and the number of trials. However, a binomial distribution converges to a Poisson distribution when the number of trials (e.g. customers generating requests) grows. Removing the limit on the pool of customers that can generate new reservations makes the model more realistic, since the number of possible customers is usually unbounded and independent from the capacity of the hotel.

For the estimation of $\Lambda(a)$, we assume that it is possible to estimate the expected number of reservation requests for a specific arrival day that are accepted by the customers and not canceled ($\mathcal{R}^a_{\text{accept}}$). Similarly, we assume that one has access to the expected number of reservation requests for a specific arrival day that are accepted by the customers and canceled ($\mathcal{R}^a_{\text{cancel}}$). $\mathcal{R}^a_{\text{accept}}$ can be approximated by the expected number of arrivals, while $\mathcal{R}^a_{\text{cancel}}$ can be seen as the expected number of cancellations. Note that both of those quantities can be estimated easily from historical data or by exploiting the experience of the

hotel manager.
Our simulator includes also a model of the acceptance probability $\Pr_{\text{accept}}(RO)$. A model of probabilities (possibly one for each admissible input) can be estimated from data retrieved by an online booking platform, where one can keep track of users that search for a room and decide to finalize the reservation or leave the website. One can also estimate the expected acceptance probability $\mathbb{E}[\Pr_{\text{accept}}(RO)]$ as the expected fraction of reservation requests that are finalized by the users after the search.

Therefore, the expected total number of reservation requests (accepted or rejected) associated with one arrival day can be estimated as follows:

$$\mathbb{E}[\mathcal{R}^a] = \Lambda(a) \approx \frac{\mathcal{R}^a_{\text{accept}} + \mathcal{R}^a_{\text{cancel}}}{\mathbb{E}[\Pr_{\text{accept}}(RO)]}. \tag{7}$$

3.2. Simulation of nights and rooms

Let *nights*^{a} be the expected number of nights for a reservation associated with an arrival day a. Analogously, *rooms*^{a} is the expected number of rooms. *max-nights*^{a} and *max-rooms*^{a} represent the limits imposed by the hotel manager. Since each reservation request includes at least one night and one room, we model the discrete probability distribution of the number of additional nights/rooms as follows:

$$\Pr(X-1=k) = \int_{\frac{k}{\max(X)}}^{\frac{k+1}{\max(X)}} \frac{(1-x)^{\frac{\max(X)}{\operatorname{avg}(X)-0.5}-2}}{\mathrm{B}(1, \frac{\max(X)}{\operatorname{avg}(X)-0.5}-1)} dx, \tag{8}$$

where X is the number of nights/rooms, $X-1$ is the number of additional nights/rooms, $\max(X)$ is either *max-nights*^{a} or *max-rooms*^{a}, and $\operatorname{avg}(X)$ is either *nights*^{a} or *rooms*^{a}. $k = 0, 1, \ldots, \max(X)-1$, and $\mathrm{B}(\alpha, \beta)$ is the Beta function with parameters α and β.
The previously defined distribution is a discrete analogue of a (continuous) Beta distribution with $\alpha = 1$ and $\beta = \frac{\max(X)}{\operatorname{avg}(X)-0.5} - 1$. The value of α is chosen so as to have a distribution with an exponential-decay profile, which is similar to the distribution seen in [22]. β is chosen so as to have an expected value approximately equal to $\operatorname{avg}(X)-1$. This is achieved by imposing the equality of the expected value of the (continuous) Beta distribution, which is $\frac{\alpha}{\alpha+\beta}$, to the expected number of additional nights/rooms rescaled to $[0,1]$, which is $\frac{\operatorname{avg}(X)-0.5}{\max(X)}$. Note that we consider a correction of 0.5 to account for the discretization error and to position rescaled expected values in the middle of the discretization interval. Experiments show that the maximum error between the expected values and the empirical averages of the discrete analogue with $\max(X)=5$ is at most 0.33, for expected values equal to $0, 0.1, 0.2, \ldots, \max(X)-1$.
Even though modeling the length of stay or the number of rooms as Bernoulli or Poisson processes provides a simple and exact way of imposing the expected value, it is not applicable to our context, which cannot be reduced to a coin toss or to an arrival process. In the literature, the Beta distribution is often used to

model unknown probability distributions, with shapes that can be controlled by the parameters α and β. By building a discrete analogue of a Beta distribution, we can exploit its macroscopic features and to obtain a realistic model of the variable of interest. A similar model can be defined also for *group* reservations, which usually follow a different distribution from that of the length of stay of *normal* reservations. This can be easily achieved by considering a different value for $\mathrm{avg}(X)$.

By following (8), an instance of the random variable X, which is either RR_{los} or RR_{size}, is generated as follows:

$$X = 1 + \lfloor Y \times \max(X) \rfloor, \tag{9}$$

where $Y \sim \mathrm{Beta}(1, \frac{\max(X)}{\mathrm{avg}(X)-0.5} - 1)$.

3.3. Simulation of cancellations

Under the same assumptions of Section 3.1, and by analogy to (1), the probability that a reservation is canceled during its lifetime can be seen as the summation of the probabilities that a reservation is canceled exactly on a specific day within its lifetime:

$$\mathrm{Pr}_{\mathrm{cancel}}(R) = \Omega(a) = \sum_{i=0}^{R_{\mathrm{TTA}}} \omega(i,a), \tag{10}$$

where $\omega(i,a)$ is the probability that R is canceled exactly i days before the arrival day a, with i within its lifetime.
We define each $\omega(i,a)$ by the following parametric model:

$$\begin{aligned} \omega_\alpha(i,a) &= \Omega(a) \times Q_\alpha(i, R_{\mathrm{TTA}}) \\ &= \Omega(a) \times \left(\left(\frac{R_{\mathrm{TTA}} + 1 - i}{R_{\mathrm{TTA}} + 1} \right)^\alpha - \left(\frac{R_{\mathrm{TTA}} - i}{R_{\mathrm{TTA}} + 1} \right)^\alpha \right), \end{aligned} \tag{11}$$

with $i = 0, 1, \ldots, R_{\mathrm{TTA}}$, $a = R_{\mathrm{arr}}$, and for any parameter $\alpha > 0$.
In this context one can also find α from the fraction of cancellations that occur on the last day ($Q_\alpha(0, R_{\mathrm{TTA}})$), which includes the so-called *no-shows*.
$\Omega(a)$ can be estimated as follows:

$$\Omega(a) \approx \frac{\mathcal{R}^a_{\mathrm{cancel}}}{\mathcal{R}^a_{\mathrm{cancel}} + \mathcal{R}^a_{\mathrm{accept}}}, \tag{12}$$

with an arrival day $a = R_{\mathrm{arr}}$.
In HotelSimu, different stochastic cancellation scenarios can be simulated by changing $\omega_\alpha(i,a)$ through $\Omega(a)$ and α.

4. A case study with aggregated data

In the following experiments, we search for the optimal pricing policies that maximize the total revenue of a set of hotels.

We assume that there is only one category of rooms. Even though this assumption is strict, the inclusion of the details of each room would require the definition of more complex choice models that are out of the scope of this paper. The reservations are not tied to specific rooms. As an example, if a customer books 1 room for 3 days, the reservation is proposed to the customer for acceptance if for each single day there is at least 1 room available. This includes also the case of different physical rooms for different days for the same customer/reservation.
Our simulations include walk-in users as well as those customers booking a room on the same day of arrival (e.g. in the morning for the night).

4.1. Setup of the experiments

We consider a monotonically decreasing reservation curve with 40% of the customers treated as walk-in users, similarly to the reservation models estimated from historical data in [22]. The goodness of this model is also confirmed by data collected by the Italian Institute of Statistics (Istat) on the features of trips[1], which show that approximately 40% of the interviewed people travel without booking. As a consequence, it is reasonable to assume that the remaining 60% of the reservations is monotonically distributed in the BH in a decreasing fashion as moving away from the walk-in day. We also assume that the maximum number of cancellations occurs on the last day, and we fix this number to 40% of the total number of cancellations. This number can be different in presence of penalties, which can reduce late cancellations. Since we are interested in the applicability of the simulator to worst-case scenarios for the hotel manager, we do not consider penalties.
BH is fixed to 180 days, the maximum number of nights for one reservation to 10, and the maximum number of rooms to 4.
As concerns the pricing policy, we use the model proposed in [8], which is based on a set of multipliers that lead to an increase or decrease in the average price according to the features of a reservation request. We assume that RO_{price} corresponds to the unit price for 1 room and 1 night.
The unit price proposed to the customer is computed as follows:

$$RO_{\text{price}} = \mathit{price}^a \cdot \xi(RR_{\text{TTA}}, RR_{\text{los}}, RR_{\text{size}}, S, \Delta, \eta), \tag{13}$$

where price^a is the expected unit price for customers arriving on day a, and $\xi(\cdot)$ is a function of the reservation request features and of the hotel registry, with average value equal to 1. This function smoothly adjusts the price within the interval $[(1-\Delta)\mathit{price}^a, (1+\Delta)\mathit{price}^a]$, with a slope proportional to η:

$$\begin{aligned} &\xi(RR_{\text{TTA}}, RR_{\text{los}}, RR_{\text{size}}, S, \Delta, \eta) = \xi(t, l, s, S, \Delta, \eta) = \\ &= (1-\Delta) + 2\Delta \cdot \Phi(\eta \cdot (M_T(t) M_L(l) M_S(s) M_C(S) - 1)). \end{aligned} \tag{14}$$

[1]http://dati.istat.it/?lang=en; section: Communication, culture, trips/Trips/Trips and their characteristics; last access: October, 2018

$\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and $M_T(\cdot)$, $M_L(\cdot)$, $M_S(\cdot)$ and $M_C(\cdot)$ are functions (or *multipliers*) of the time-to-arrival, the length of stay, the number of rooms and the remaining hotel capacity at the moment the reservation request is generated, respectively. All the multipliers are defined so as they have an average value equal to 1. As concerns the parameters of the multipliers, we set $T_0 = 30$ and $C_0 = L_0 = G_0 = 1.6$. All the remaining parameters are instead optimized by the system in order to maximize the total revenue. For a more detailed description of the parameters see [8].
η is fixed to 3, while Δ is fixed to 0.6, so as to propose prices with a maximum increase/decrease of 60% with respect to $price^a$.

The effect on the room demand of changing the unit price is modeled by the acceptance probability, which we define similarly to [22]. When the proposed price is equal to the average price of reservations with the same arrival day, the acceptance probability is set to 0.5, to model the absence of any preference about accepting or rejecting the reservation. With prices fixed to the average values, the expected number of accepted reservations is equal to half of the total number of reservation requests. The expected percentage of accepted reservations increases when the price decreases and decreases otherwise.
This phenomenon, also called *price elasticity*, is modeled by the following function:

$$\Pr_{\text{accept}}(RO) = 1 - \Phi(\rho \cdot (RO_{\text{price}} - price^a)), \tag{15}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and ρ is a parameter that controls the slope of the function and allows us to consider different price elasticity scenarios. In the experiments, ρ is chosen so that $\Pr_{\text{accept}}(RO) \approx 1$ when there is a discount of at least 50% and $\Pr_{\text{accept}}(RO) \approx 0$ when the price increases of at least 50%. In this way we are able to simulate a realistic scenario, in which the ranges of acceptable prices for customers and hotel managers differ. In our case, customers can accept maximum variations of $\pm 50\%$, while hotel managers allow maximum variations of proposed prices of $\pm 60\%$ (through Δ in (13)).

We empirically show the applicability of HotelSimu to 10 structures in Trento, Italy. In order to simulate multiple scenarios, we selected (and anonymized) representative hotels from the official open data of the Province of Trento[2], with different capacities, stars and average prices, as reported in Table 1.

The information related to the average arrivals and the average number of nights per reservation is taken from the Statistics Institute of the Province of Trento (Ispat)[3]. Only monthly data up to 2016 are available. No information is available about the average number of rooms per reservation, so we assumed it to be equal to 1. We disaggregated data on arrivals and mapped them onto each hotel according to their capacity, under the assumption that bigger hotels

[2]http://dati.trentino.it/dataset/esercizi-alberghieri (last access: October, 2018).

[3]http://www.statistica.provincia.tn.it, section "Annuari del Turismo" (last access: October, 2018).

Table 1: Hotels used for the tests.

Hotel ID	Stars	Rooms	Average price (€)
01	3	52	120.00
02	2	34	69.50
03	4	136	290.00
04	4	46	153.33
05	3	113	136.67
06	3	37	74.00
07	1	9	39.00
08	4	22	216.50
09	2	14	66.50
10	3	19	82.67

usually register more arrivals than smaller hotels.

Since we do not have access to data related to revenues, we are not able to evaluate our model with respect to a ground truth. However, we use real aggregated data on tourists and different structures to simulate time series of reservations and cancellations, and we consider these (stochastic) time series as a baseline to be compared to the outcome of the optimization. Moreover, since we do not have access to the complete history of each hotel, we assume that the data of 2016 are valid also in the following years.

In the experiments RH starts on July 1st, 2017, and ends on December 31st, 2018. AH starts on July 1st, 2017, and ends on January 31st, 2019. OH starts on January 1st, 2018, and ends on December 31st, 2018. To avoid bias given by strict truncation, we maximize the total revenue in one year, after a transient period of 6 months and for customers arriving up to one month later than the end of the year of interest.

For the optimization, we use an efficient implementation of CMA-ES[4], with a step size of 0.5 (for an optimal exploration-exploitation balance) and a budget of 300 iterations (for a maximum optimization time of 5/6 hours). Each iteration retrieves the total revenue as the average on 20 simulation runs, all with the same parameter configuration, for a total of 6000 simulations within one optimization run. For each hotel, the optimization process is repeated 10 times.

For the tests, we used virtual machines on a KVM hypervisor (1 per hotel), each one with 512 MB of RAM and 1 CPU (1 core) at 2.1 GHz. In this way, one can evaluate the efficiency of our implementation within a controlled environment that is characterized by limited resources. In addition, this provides an upper bound for the execution time of the optimization on the majority of current consumer machines.

[4]Code available at http://beniz.github.io/libcmaes (last access: October, 2018).

Table 2: Percentage increase in arrivals, occupancy (as room-nights) and revenue after optimization. Maximum and minimum values are in bold.

Hotel ID	Arrivals	Occupancy	Revenue
01	48.2±0.5	47.6±0.6	18.4±0.6
02	50.6±0.6	50.6±0.7	20.4±0.7
03	51.6±0.3	52.6±0.3	21.6±0.3
04	44.0±0.5	44.2±0.6	17.8±0.6
05	**55.5±0.3**	**55.2±0.4**	**23.1±0.4**
06	46.9±0.6	45.8±0.6	17.8±0.6
07	**38.2±1.1**	**37.7±1.3**	**12.8±1.2**
08	43.2±0.7	42.0±0.8	18.0±0.8
09	42.0±0.9	41.8±1.0	17.7±1.0
10	41.8±0.8	40.5±0.9	17.3±0.9

4.2. Results on arrivals, occupancy and revenue

In Table 2 we report the results on customer arrivals, occupancy and total revenue as the percentage increase led by the optimized pricing model with respect to the configuration with the multipliers equal to 1. A unit of occupancy corresponds to the so-called *room-night*, which is a room occupied for one night. Results are expressed in terms of averages and standard errors, and they are statistically significant according to the two-tailed unequal variances t-test [31], with a significance level $\alpha = 0.01$. Results are promising for all the hotels, with a minimum of 12.8% increase in revenue, 37.7% in occupancy and 38.2% in arrivals. The maximum increase in revenue is reached for Hotel 05, with a value of 23.1%. The minimum values are reached for small hotels, where the limited number of rooms leads to fewer arrivals and then relatively low revenues. In this context, there is also more variability, since the hotel can become full with few reservations, thus leading to the rejection of more requests. Experiments suggest that higher revenues can be obtained for medium and big hotels, where the system exploits the capacity of the hotel to increase the number of arrivals. The time series of the average daily revenue during the year of interest for the best and worst scenarios are reported in Figure 4. For Hotel 05, it is evident that the time series produced by the optimized model is significantly higher than that produced without optimization. In this case, there is less chance of having a loss in revenue because of an optimistic configuration found during the optimization process. For Hotel 07, the two time series are not significantly different because of the higher uncertainty caused by the small dimension of the hotel. This leads to higher risk and to the possibility of having a loss (with probability ≈ 0.03), as it is evident from the distribution of the increase in revenue in Figure 5.

4.3. Results on CPU time

The CPU time to complete one optimization run on a low-end machine is at most 5 hours and 7 minutes, as reported in Table 3.

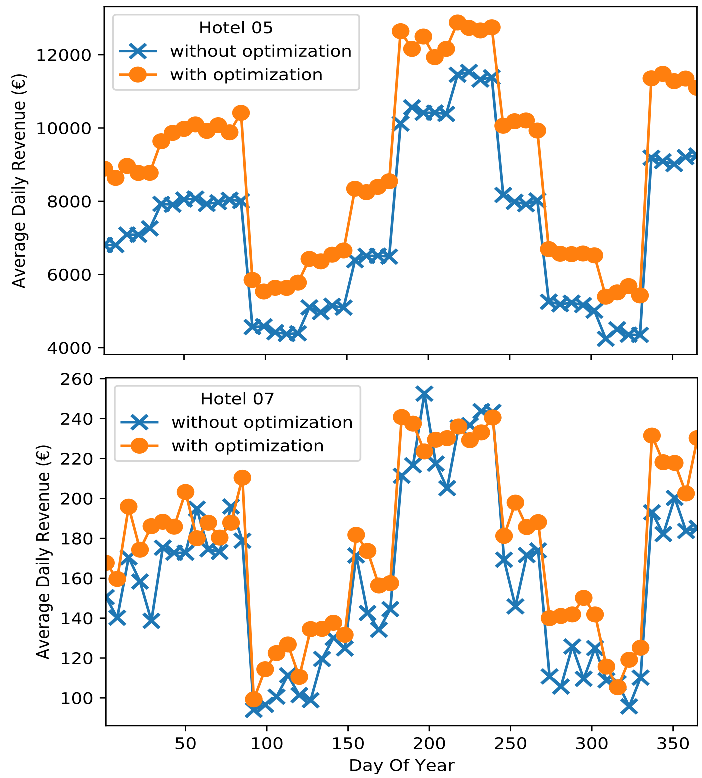

Figure 4: Average daily revenue for Hotel 05 and Hotel 07 (one value per week).

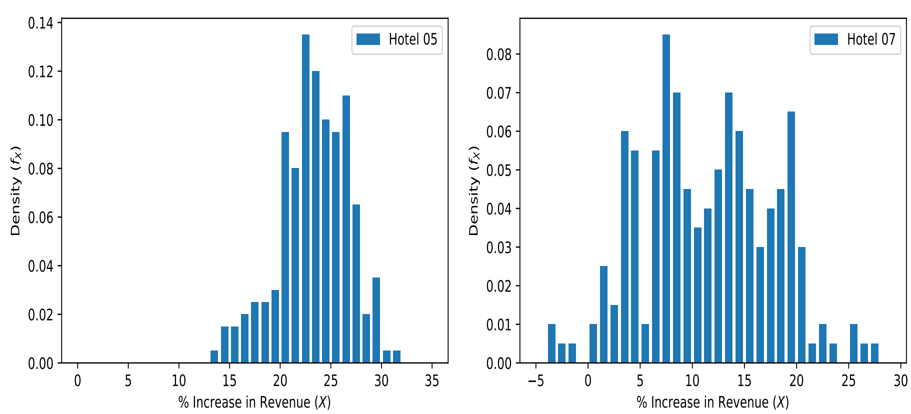

Figure 5: Estimated distributions of increase in revenue after optimization for Hotel 05 and Hotel 07.

Table 3: Optimization total CPU time (in seconds), single-run simulation CPU time (in seconds) and single-run simulation generated events.

Hotel ID	Optimization time	Simulation time	Events
01	11000±109.0	2.0 ±0.012	20300±63.60
02	10800±82.33	1.91±0.006	13200±25.72
03	18400±258.5	3.54±0.030	53300±69.77
04	9910±97.61	1.78±0.008	17900±49.47
05	16700±241.2	3.19±0.031	44400±101.7
06	9570±87.87	1.69±0.012	14400±60.86
07	7370±17.70	1.25±0.010	3450±7.195
08	7870±35.67	1.35±0.004	8510±17.37
09	7740±32.26	1.3 ±0.010	5380±15.76
10	8650±46.21	1.49±0.008	7320±20.52

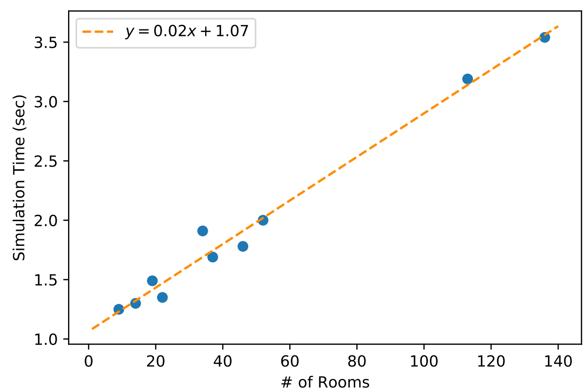

Figure 6: Single-run simulation CPU time against hotel size. The dashed line is the learned regression model.

This allows the optimization to be run during the night, so that the pricing model can be tuned every day, with the inclusion of new records into the hotel registry.
Experiments show that one year and a half is simulated in at most 3.54 seconds, with more than 53000 events generated for Hotel 03. As expected after the setup of Section 4.1, the CPU time of a single run of simulation depends linearly on the size of the hotel, as illustrated in Figure 6. This is explained by the way we map the total number of arrivals in the Trento area to each hotel. Hotels with higher capacity register more arrivals, and they are characterized by more generated events during the simulation.

5. Conclusion

In this work we proposed HotelSimu, an efficient hotel simulator that can be used for maximizing the total revenue. Our approach is parametric and allows the hotel manager to analyze different scenarios by changing a limited set of meaningful parameters. Differently from previous studies, reservations and cancellations occur in an interleaved way and they are not grouped into disjoint sets of events. One can also choose the exact dates to be simulated as well as the opening dates of the hotel, with a better management of out-of-bounds reservations. In addition, one is not tied to fixed seasonality schemes. Seasonal averages can be set even on a day-by-day basis, thus allowing the hotel manager to adapt the pricing policy to special events.

Experiments on aggregated data show that the implemented system can simulate one year and a half in ≈ 2 seconds on average on a low-end machine, and one complete optimization to find the best pricing policy can be run within one night. The average revenue increase is $\approx 19\%$ with respect to the original pricing policies. By using our simulator and an evolutionary approach for optimization as CMA-ES, we also show that the risk of losses is absent for medium-big hotels and limited for small hotels, with a maximum loss probability of ≈ 0.03.

Acknowledgements

This work was supported by the University of Trento. The research of Roberto Battiti was supported by the Russian Science Foundation, project no. 15–11–30022 “Global optimization, supercomputing computations, and applications”. A. Mariello would like to thank H.T. Nguyen for the useful advice on the RIM quantifiers.

References

References

[1] Z. Xiang, V. P. Magnini, D. R. Fesenmaier, Information technology and consumer behavior in travel and tourism: Insights from travel planning using the internet, Journal of Retailing and Consumer Services 22 (2015) 244–249.

[2] D. Buhalis, R. Law, Progress in information technology and tourism management: 20 years on and 10 years after the internet – the state of etourism research, Tourism management 29 (4) (2008) 609–623.

[3] K. T. Talluri, G. J. Van Ryzin, The theory and practice of revenue management, Vol. 68, Springer Science & Business Media, 2006.

[4] G. R. Bitran, S. V. Mondschein, An application of yield management to the hotel industry considering multiple day stays, Operations research 43 (3) (1995) 427–443.

[5] T. K. Baker, D. A. Collier, A comparative revenue analysis of hotel yield management heuristics, Decision Sciences 30 (1) (1999) 239–263.

[6] K.-K. Lai, W.-L. Ng, A stochastic approach to hotel revenue optimization, Computers & Operations Research 32 (5) (2005) 1059–1072.

[7] J. Guadix, P. Cortés, L. Onieva, J. Muñuzuri, Technology revenue management system for customer groups in hotels, Journal of Business Research 63 (5) (2010) 519–527.

[8] A. E.-M. Bayoumi, M. Saleh, A. F. Atiya, H. A. Aziz, Dynamic pricing for hotel revenue management using price multipliers, Journal of Revenue and Pricing Management 12 (3) (2013) 271–285.

[9] H. A. Aziz, M. Saleh, M. H. Rasmy, H. Elshishiny, Dynamic room pricing model for hotel revenue management systems, Egyptian Informatics Journal 12 (3) (2011) 177–183.

[10] S. Ivanov, Hotel revenue management: From theory to practice, Zangador, 2014.

[11] B. Denizci Guillet, I. Mohammed, Revenue management research in hospitality and tourism: A critical review of current literature and suggestions for future research, International Journal of Contemporary Hospitality Management 27 (4) (2015) 526–560.

[12] T. Y. Choi, V. Cho, Towards a knowledge discovery framework for yield management in the Hong Kong hotel industry, International Journal of Hospitality Management 19 (1) (2000) 17–31.

[13] S. Liu, K. K. Lai, S. Wang, Booking models for hotel revenue management considering multiple-day stays 2 (2008) 78–91.

[14] D. Bertsimas, I. Popescu, Revenue management in a dynamic network environment, Transportation science 37 (3) (2003) 257–277.

[15] A. J. Kleywegt, An optimal control problem of dynamic pricing, School of Industrial and Systems Engineering, Georgia Institute of Technology (2001).

[16] G. Bitran, R. Caldentey, An overview of pricing models for revenue management, Manufacturing & Service Operations Management 5 (3) (2003) 203–229.

[17] J. I. McGill, G. J. Van Ryzin, Revenue management: Research overview and prospects, Transportation science 33 (2) (1999) 233–256.

[18] D. Zhang, L. Weatherford, Dynamic pricing for network revenue management: A new approach and application in the hotel industry, INFORMS Journal on Computing 29 (1) (2016) 18–35.

[19] D. Bertsimas, S. De Boer, Simulation-based booking limits for airline revenue management, Operations Research 53 (1) (2005) 90–106.

[20] G. Figueira, B. Almada-Lobo, Hybrid simulation–optimization methods: A taxonomy and discussion, Simulation Modelling Practice and Theory 46 (2014) 118–134.

[21] D. Stefanovic, N. Stefanovic, B. Radenkovic, Supply network modelling and simulation methodology, Simulation Modelling Practice and Theory 17 (4) (2009) 743–766.

[22] A. Zakhary, A. F. Atiya, H. El-Shishiny, N. E. Gayar, Forecasting hotel arrivals and occupancy using Monte Carlo simulation, Journal of Revenue and Pricing Management 10 (4) (2011) 344–366.

[23] A. Zakhary, N. El Gayar, A. F. Atiya, A comparative study of the pickup method and its variations using a simulated hotel reservation data, ICGST international journal on artificial intelligence and machine learning 8 (2008) 15–21.

[24] E. S. Gardner Jr, Exponential smoothing: The state of the art—part ii, International journal of forecasting 22 (4) (2006) 637–666.

[25] R. R. Andrawis, A. F. Atiya, A new Bayesian formulation for Holt's exponential smoothing, Journal of Forecasting 28 (3) (2009) 218–234.

[26] L. R. Weatherford, S. E. Kimes, A comparison of forecasting methods for hotel revenue management, International journal of forecasting 19 (3) (2003) 401–415.

[27] N. Hansen, The CMA evolution strategy: a comparing review, in: Towards a new evolutionary computation, Springer, 2006, pp. 75–102.

[28] N. Hansen, Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, ACM, 2009, pp. 2389–2396.

[29] R. R. Yager, Quantifier guided aggregation using OWA operators, International Journal of Intelligent Systems 11 (1) (1996) 49–73.

[30] P. Grube, F. Núñez, A. Cipriano, An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network, Simulation Modelling Practice and Theory 19 (1) (2011) 393–405.

[31] B. L. Welch, The generalization of student's problem when several different population variances are involved, Biometrika 34 (1/2) (1947) 28–35.