



HotelSimu: Simulation-Based Optimization for Hotel Dynamic Pricing

Andrea Mariello, Manuel Dalcastagné^(✉), and Mauro Brunato

Department of Information Engineering and Computer Science, University of Trento,
Via Sommarive 9, 38123 Povo, TN, Italy
andrea.mariello@alumni.unitn.it,
{m.dalcastagne,mauro.brunato}@unitn.it

Abstract. Exact and approximated mathematical optimization methods have already been used to solve hotel revenue management (RM) problems. However, to obtain solutions which can be solved in acceptable CPU times, these methods require simplified models. Approximated solutions can be obtained by using simulation-based optimization, but existing approaches create empirical demand curves which cannot be easily modified if the current market situation deviates from the past one. We introduce HotelSimu, a flexible simulation-based optimization approach for hotel RM, whose parametric demand models can be used to inject new information into the simulator and adapt pricing policies to mutated market conditions. Also, cancellations and reservations are interleaved, and seasonal averages can be set on a daily basis. Monte Carlo simulations are employed with black-box optimization to maximize revenue, and the applicability of our models is evaluated in a case study on a set of hotels in Trento, Italy.

Keywords: Revenue management · Dynamic pricing · Simulation-based optimization

1 Introduction

Information technology drastically changed how people plan travels and accommodations. In fact, tools such as online travel agencies or price comparison websites are now extensively used [31], and hotels are no longer forced to sell their rooms only through traditional intermediaries. Also, many hotels have already adopted RM techniques to manage their availability of rooms, in order to maximize their revenue.

Optimization problems related to hotel RM are usually expressed following two approaches: capacity control [3, 5, 8, 10, 11, 15, 21, 22], where the decision variable is the amount of offered supply, and dynamic pricing [4, 6, 20, 34, 35], where the price is the decision variable. In both cases, several mathematical optimization methods have already been proposed to maximize revenue [12, 19, 28]. Many of these formulations assume that demand is independent from the chosen policy. More complex scenarios, where demand is influenced by other factors (e.g.,

price), are more difficult to handle and closed-form solutions are rarely available [9]. Demand is usually considered as a known deterministic function or as a stochastic function following a known distribution family with unknown parameters. Also, if stochastic cancellations are considered, the CPU time for solving the problem tends to grow exponentially and approaches like dynamic programming are effective only in specific cases [23]. A possible solution to mitigate the complexity of the model is approximated dynamic programming [3, 8, 35], where the problem is partitioned into simpler subproblems. Nonetheless, approximated models cannot provide exact solutions for realistic scenarios because of the large number of possible states [27].

The approximate maximization of revenue can be achieved using simulation-based optimization [7, 13]. The analytical model is substituted with a simulator of many inter-related processes like reservations, cancellations, no-shows, walk-ins. Then, black-box optimization is used to find the policy which maximizes the revenue. An effective technique to maximize revenue and simulate different stochastic aspects of the hotel booking scenario is Monte Carlo simulation [26]. The generation of reservations and cancellations leads to a distribution of possible revenues, and the expected value of the distribution is considered as the variable to be maximized. For example, in [6, 34] a Monte Carlo approach is employed to simulate demand as the result of many stochastic processes, and in [6] the effect that price has on demand is also considered.

In this paper, we present a flexible simulation-based optimization approach for hotel RM based on dynamic pricing. We simulate demand using a novel set of parametric models based on the RIM quantifiers [32], whose parameters are daily statistics which can be estimated from data. Our models allow to change the curves parametrically, redistributing demand along the booking horizon, without requiring any change of advance historical data. In fact, bookings and cancellations associated to each day are distributed along the booking horizon with a non-homogeneous Poisson process, where demand expectations of each day are defined by our parametric models. The hotel manager can inject new information in the system, adapting pricing policies to the mutated conditions of the market. For the optimization, we use an efficient implementation of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES)[16]¹. We position after the work of [6, 34], on which we build, to provide a simpler way for the hotel manager to run *what-if* analyses. Furthermore, reservation requests and cancellations are not grouped into disjoint sets of events like in [34], but occur in an interleaved way. The structure of the remainder of this paper is as follows. Section 2 describes HotelSimu, and defines more in detail the parametric models used for the simulations. Section 3 provides some details about the optimization algorithm, and Sect. 4 shows the applicability of our models to a set of hotels in Trento, Italy. Results show that our approach leads to an average revenue increase similar to that of other dynamic pricing strategies, even though only aggregated data has been used. Finally, Sect. 5 provides the main implications of our work for the hotel manager and briefly describes possible extensions.

¹ Code available at <http://beniz.github.io/libcmaes>.

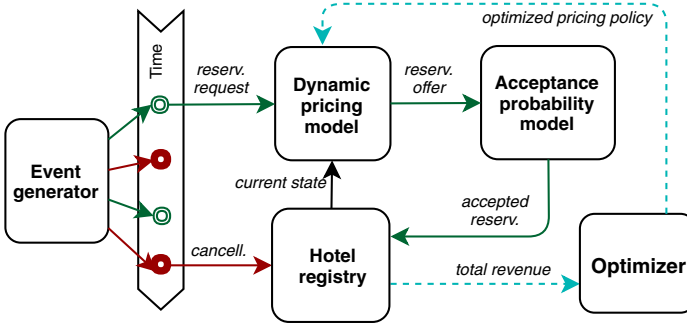


Fig. 1. HotelSimu overview. Reservation requests and cancellations are interspersed. The state of the hotel after one complete simulation is used by the optimizer to compute the total revenue and adjust the pricing policy.

2 Simulation Methods

The main components of HotelSimu are shown in Fig. 1. An event generator simulates the reservation requests and the cancellations. A registry stores the information about the state of the hotel, in particular accepted reservations and room availability. A dynamic pricing model proposes an offer for each reservation request, and an acceptance probability model simulates the stochastic process by which customers accept or discard reservation offers. An optimizer searches for the optimal pricing policy to maximize revenue.

2.1 Definitions

Let us now define the main concepts and the notation used throughout the paper.

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, which depends on the features of RR .

Definition 3. An accepted reservation or simply reservation (R) is a reservation offer accepted by the customer. It is registered on the hotel registry and it effectively changes room availability.

Definition 4. The acceptance probability of a reservation offer ($\text{Pr}_{\text{accept}}(RO)$) 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 are in the registry at time t .

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 $R_{\text{T TA}} = d(R_{\text{res}}, R_{\text{arr}})$. If $R_{\text{T TA}} = 0$, a customer makes a reservation on the arrival day or arrives at the hotel with no reservation and we refer to the customer as a walk-in user.

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

Definition 9. A cancellation (C) is characterized by the cancellation day (C_{day}), which is the day the event occurs, and 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 $t \in [0, R_{\text{T TA}}]$. According to this definition, the probability that R is canceled within its lifetime is

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

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.

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

2.2 Simulation of Reservation Requests

Let \mathcal{R}_r^a , $r \in \text{RH}$, $a \in \text{AH}$, 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:

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

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:

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

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:

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

with $i = 0, 1, \dots, \text{BH}$, $a \in \text{AH}$, and for any parameter $\alpha > 0$. The expression of $Q_\alpha(i, \text{BH})$ is similar to that of the RIM quantifiers proposed in [32], after reflection and translation. We use $Q_\alpha(i, \text{BH})$ because:

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

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

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 constant curve ($\alpha = 1$) to increasing and decreasing curves (see Fig. 2);
- 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, $Q_\alpha(0, \text{BH})$.

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

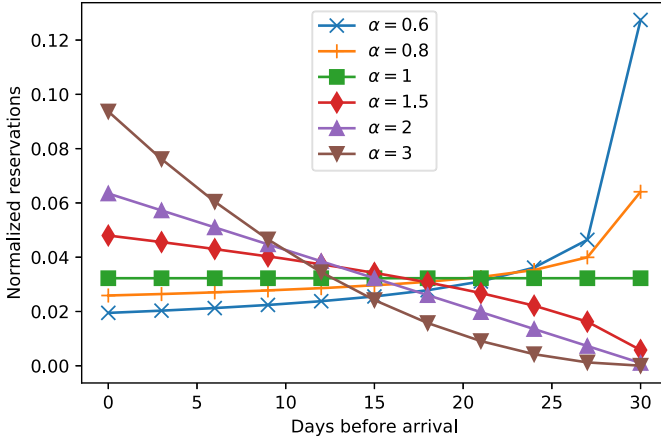


Fig. 2. $Q_\alpha(i, BH)$ for $BH = 30$ and for different values of α .

parametric model, so $\mathcal{R}^a \sim \text{Poisson}(\Lambda(a))$. Therefore, reservation requests are generated for each simulated day according to the following model:

$$\begin{cases} \mathcal{R}_r^a \sim \text{Poisson}(\lambda_\alpha(i, a)) & \text{if } i \leq BH, \\ \mathcal{R}_r^a = 0 & \text{otherwise.} \end{cases} \quad (5)$$

Poisson processes are usually chosen to model arrival processes [14] and, in our context, they can represent the arrival of reservation requests with a minimum set of parameters. In [34], 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 ($\mathcal{R}_{\text{accept}}^a$). 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 ($\mathcal{R}_{\text{cancel}}^a$). $\mathcal{R}_{\text{accept}}^a$ can be approximated by the expected number of arrivals, while $\mathcal{R}_{\text{cancel}}^a$ can be seen as the expected number of cancellations.

HotelSimu includes also a model of the acceptance probability $\text{Pr}_{\text{accept}}(RO)$. 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 $\mathbb{E}[\text{Pr}_{\text{accept}}(RO)]$ 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:

$$\mathbb{E}[\mathcal{R}^a] = \Lambda(a) \approx \frac{\mathcal{R}_{\text{accept}}^a + \mathcal{R}_{\text{cancel}}^a}{\mathbb{E}[\text{Pr}_{\text{accept}}(RO)]}. \quad (6)$$

2.3 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:

$$\text{Pr}(X - 1 = k) = \int_{\frac{k}{\text{max}(X)}}^{\frac{k+1}{\text{max}(X)}} \frac{(1-x)^{\frac{\text{max}(X)}{\text{avg}(X)-0.5}-2}}{\text{B}(1, \frac{\text{max}(X)}{\text{avg}(X)-0.5} - 1)} dx, \quad (7)$$

where X is the number of nights/rooms, $X - 1$ is the number of additional nights/rooms, $\text{max}(X)$ is either max-nights^a or max-rooms^a , and $\text{avg}(X)$ is either nights^a or rooms^a . $k = 0, 1, \dots, \text{max}(X) - 1$, and $\text{B}(\alpha, \beta)$ 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 $\beta = \frac{\text{max}(X)}{\text{avg}(X)-0.5} - 1$. 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 [34]. β is chosen so as to have an expected value approximately equal to $\text{avg}(X) - 1$. 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 $\frac{\text{avg}(X)-0.5}{\text{max}(X)}$. 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 $\text{max}(X) = 5$ is at most 0.33, for expected values equal to $0, 0.1, 0.2, \dots, \text{max}(X) - 1$.

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 $\text{avg}(X)$. By following (7), an instance of the random variable X , which is either RR_{los} or RR_{size} , is generated as $X = 1 + \lfloor Y \times \text{max}(X) \rfloor$, where $Y \sim \text{Beta}(1, \frac{\text{max}(X)}{\text{avg}(X)-0.5} - 1)$.

2.4 Simulation of Cancellations

Under the same assumptions of Sect. 2.2, 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:

$$\Pr_{\text{cancel}}(R) = \Omega(a) = \sum_{i=0}^{R_{\text{TTA}}} \omega(i, a), \quad (8)$$

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:

$$\begin{aligned} \omega_{\alpha}(i, a) &= \Omega(a) \times Q_{\alpha}(i, R_{\text{TTA}}) \\ &= \Omega(a) \times \left(\left(\frac{R_{\text{TTA}} + 1 - i}{R_{\text{TTA}} + 1} \right)^{\alpha} - \left(\frac{R_{\text{TTA}} - i}{R_{\text{TTA}} + 1} \right)^{\alpha} \right), \end{aligned} \quad (9)$$

with $i = 0, 1, \dots, R_{\text{TTA}}$, $a = R_{\text{arr}}$, 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 ($Q_{\alpha}(0, R_{\text{TTA}})$), which includes the so-called *no-shows*. $\Omega(a)$ can be estimated as follows:

$$\Omega(a) \approx \frac{\mathcal{R}_{\text{cancel}}^a}{\mathcal{R}_{\text{cancel}}^a + \mathcal{R}_{\text{accept}}^a}, \quad (10)$$

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

3 Optimizing the Noisy Simulator Function

Since Monte Carlo simulation employs stochastic processes, the performance of each solution corresponds to a distribution of results. The expected value of the distribution is used as an approximation of the objective function to be optimized, so the optimization operates in the presence of noise.

In the literature, multiple works tested diverse heuristic algorithms on noisy functions, and they have shown that population-based approaches like CMA-ES are a good choice to optimize noisy functions [1, 2, 24, 25]. In fact, instead of relying only on a single solution, at each iteration CMA-ES combines a subset of its candidate solutions in order to direct the search in the most promising direction. By combining multiple solutions located in a restricted area of the search space, the impact of noise is decreased due to an *implicit averaging* effect [1, 33]. Moreover, to further reduce the effect of noise on the optimization, we compute the performance of each solution as the mean of the outcome of multiple simulations. From probability theory, one knows that the effect of noise can be reduced by evaluating multiple times each solution [33]. More precisely, CMA-ES is an evolutionary optimization algorithm in which a multivariate normal distribution $N(\mu_t, M_t)$ is used to sample solutions, where t defines the iteration

of the algorithm. At each iteration, the mean μ_t defines the center of the distribution, while the covariance matrix M_t determines shape and orientation of the ellipsoid corresponding to $N(\mu_t, M_t)$. Also, a step size σ_t controls the spread of the distribution as a percentage of the search space. Iteratively, CMA-ES follows the following steps. First, a population of λ solutions is sampled from $N(\mu_t, M_t)$. Second, candidates are evaluated and ranked according to the respective evaluations. Third, the best $\lfloor \frac{\lambda}{2} \rfloor$ results are used to update μ_t and M_t , in order to move the search towards the most promising search direction. Fourth, σ_t is increased or decreased according to the length of the so-called evolution paths. Evolution paths are weighted vector sums of the last points visited by the algorithm. They provide information about the correlations among points, and they are used to find the direction recently followed by the optimization. If consecutive steps are going in the same direction, the same distance could be covered by longer steps and the current path is too long. If consecutive steps are not going in the same direction, single steps tend to cancel each other out and so the current path is too short.

CMA-ES is executed with $\lambda = 4 + \lfloor 3 \log d \rfloor$ and $\sigma = 0.5$, where d is the dimensionality of the objective function and $\sigma \in (0, 20]$. The size of the population is the one suggested by the authors of [16], who have also tested that CMA-ES with this population size is a robust and fast local search method [18]. We experimented even other parameter settings of the algorithm, but because of space limitations of this publication we present only the preliminary results obtained with the mentioned settings. Also, all the standard stopping criterias of CMA-ES are active [17]. Each time a stopping criteria is triggered, the algorithm is restarted from another randomly generated point in the search space, with a new population of the same size.

4 Results

In the following experiments, we show how our models can be used to search for the optimal pricing policies that maximize the total revenue of a set of hotels of different sizes. We assume there is only one category of rooms, and that at least historical data about final demand is available. However, if advance historical data is also available and empirical demand curves can be estimated, our models can be calibrated using optimization algorithms [29].

4.1 Setup of the Experiments

We consider a monotonically decreasing reservation curve with 40% of the customers treated as walk-in users, calibrating our models according to the reservation models estimated from historical data in [34]. The goodness of this choice is also confirmed by data collected by the Italian Institute of Statistics (Istat) on the features of trips², which show that approximately 40% of the interviewed

² <http://dati.istat.it/?lang=en;section:Communication,culture,trips/Trips/Tripsandtheircharacteristics>.

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. 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 [6], which is based on a set of multipliers that leads to an increase or decrease in the average price according to the features of a reservation request. The multipliers vary around 1, and each multiplier changes the reference price according to the value it assumes: a value lower than 1 corresponds to a discount and a value larger than 1 is a price increase. We assume that RO_{price} corresponds to the unit price for 1 room and 1 night, where the unit price proposed to the customer is computed as follows:

$$RO_{\text{price}} = \text{price}^a \cdot \xi(RR_{\text{TTA}}, RR_{\text{los}}, RR_{\text{size}}, S, \Delta, \eta), \quad (11)$$

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 $[(1 - \Delta)\text{price}^a, (1 + \Delta)\text{price}^a]$, with a slope proportional to η :

$$\begin{aligned} \xi(RR_{\text{TTA}}, RR_{\text{los}}, RR_{\text{size}}, S, \Delta, \eta) &= \xi(t, l, s, S, \Delta, \eta) = \\ &= (1 - \Delta) + 2\Delta \cdot \Phi(\eta \cdot (M_T(t)M_L(l)M_S(s)M_C(S) - 1)). \end{aligned} \quad (12)$$

$\Phi(\cdot)$ 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. As concerns the parameters of the multipliers, we set $T_0 = 30$ and $C_0 = L_0 = G_0 = 1.6$. Also, $\eta = 3$ and $\Delta = 0.6$ in order 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 [34]. 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, called *price elasticity*, is modeled by the following function:

$$\text{Pr}_{\text{accept}}(RO) = 1 - \Phi(\rho \cdot (RO_{\text{price}} - \text{price}^a)), \quad (13)$$

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%.

Table 1. Characteristics of hotels used for the tests, and results. Arrivals, occupancy (as room-nights) and revenue after optimization are expressed as percentage increase, where maximum and minimum values are in bold. Optimization total CPU time and single-run simulation CPU time are defined in seconds.

Hotel	Rooms	Price (€)	Arrivals	Occupancy	Revenue	Optimization	Simulation
01	52	120.00	48.2 ± 0.5	47.6 ± 0.6	18.4 ± 0.6	11000 ± 109.0	2.0 ± 0.012
02	34	69.50	50.6 ± 0.6	50.6 ± 0.7	20.4 ± 0.7	10800 ± 82.33	1.91 ± 0.006
03	136	290.00	51.6 ± 0.3	52.6 ± 0.3	21.6 ± 0.3	18400 ± 258.5	3.54 ± 0.030
04	46	153.33	44.0 ± 0.5	44.2 ± 0.6	17.8 ± 0.6	9910 ± 97.61	1.78 ± 0.008
05	113	136.675	55.5 ± 0.3	55.2 ± 0.4	23.1 ± 0.4	16700 ± 241.2	3.19 ± 0.031
06	37	74.00	46.9 ± 0.6	45.8 ± 0.6	17.8 ± 0.6	9570 ± 87.87	1.69 ± 0.012
07	9	39.00	38.2 ± 1.1	37.7 ± 1.3	12.8 ± 1.2	7370 ± 17.70	1.25 ± 0.010
08	22	216.50	43.2 ± 0.7	42.0 ± 0.8	18.0 ± 0.8	7870 ± 35.67	1.35 ± 0.004
09	14	66.50	42.0 ± 0.9	41.8 ± 1.0	17.7 ± 1.0	7740 ± 32.26	1.3 ± 0.010
10	19	82.67	41.8 ± 0.8	40.5 ± 0.9	17.3 ± 0.9	8650 ± 46.21	1.49 ± 0.008

We empirically show the applicability of HotelSimu to 10 hotels in Trento, Italy. We selected representative hotels from the official open data of the Province of Trento³, as reported in Table 1. The information on the average arrivals and the average number of nights per reservation is taken from the Statistics Institute of the Province of Trento (Ispat)⁴. 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 usually register more arrivals than smaller hotels. We use real aggregated data on tourists and different hotels to simulate time series of reservations and cancellations, and we consider these time series as a baseline to be compared to the outcome of the optimization.

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.

The optimization has a budget of 300 iterations (for a maximum running time of 5/6 h). 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. The optimization is repeated 10 times. Each experiment is started from an initial solution which has been generated by a uniform distribution defined over the search space. Tests have been run on

³ <http://dati.trentino.it/dataset/esercizi-alberghieri>.

⁴ <http://www.statistica.provincia.tn.it>, section “Annuari del Turismo”.

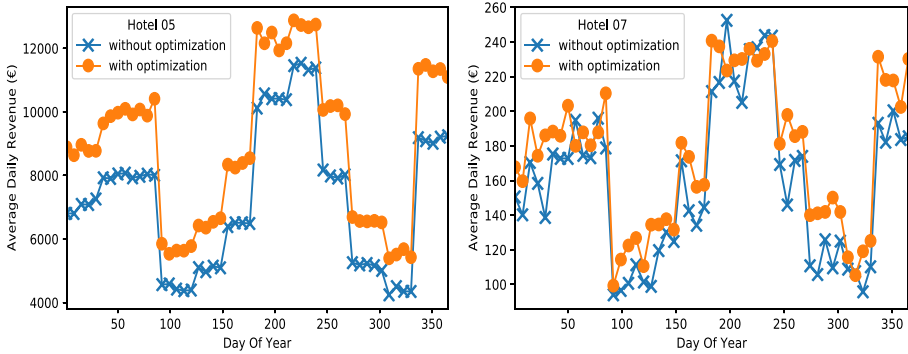


Fig. 3. Average daily revenue for Hotel 05 and Hotel 07 (one value per week).

virtual machines using a KVM hypervisor (1 per hotel), each one with 512 MB of RAM and 1 CPU (1 core) at 2.1 GHz.

4.2 Results on Arrivals, Occupancy and Revenue

In Table 1 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. Results are expressed in terms of averages and standard errors, and they are statistically significant according to the two-tailed unequal variances t -test [30], with a significance level $\alpha = 0.01$. A unit of occupancy corresponds to the so-called *room-night*, which is a room occupied for one night.

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 Fig. 3. 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 Fig. 4. These results are in accordance with the expected behavior of non-homogeneous Poisson distributions, whose coefficient

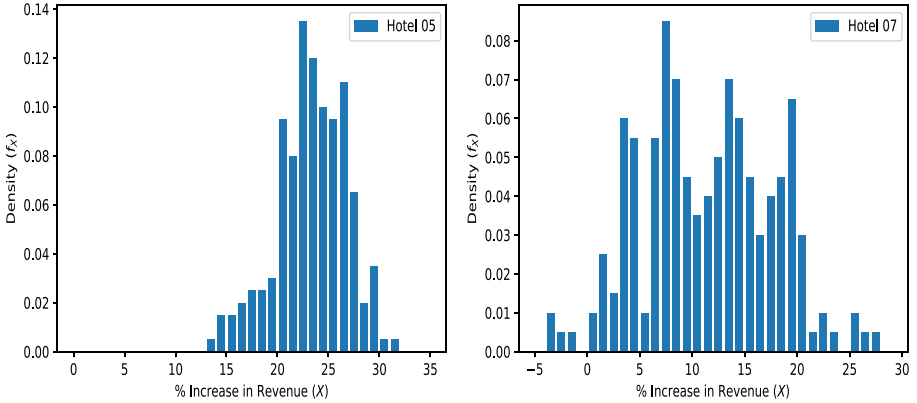


Fig. 4. Estimated distributions of increase in revenue after optimization for Hotel 05 and Hotel 07.

of variation decreases as the expected value increases. In the context of hotel demand, this property implies that for smaller hotels, which can accommodate a limited number of guests and therefore are characterized by less arrivals, the coefficient of variation is higher than that of large hotels. As a consequence, the increased variability for small hotels leads to higher risk of losses, as empirically shown by our results.

5 Conclusions

In this work we proposed HotelSimu, a flexible simulation-based optimization approach which can be used for maximizing the revenue of hotels. Since the output of the simulations is noisy, we optimized the noisy simulator function by using CMA-ES, a population-based algorithm which has already been studied in the literature and proved to be effective in noisy scenarios. Furthermore, we aggregated the outcome of multiple simulations in order to use the expected value to further reduce the effects of the noise on the optimization.

HotelSimu models stochastic arrivals and cancellations in an interleaved fashion, considering several characteristics of reservation requests in order to propose dynamic prices. Furthermore, it models the effect that price variations have on demand (price elasticity). Our models, based on the RIM quantifiers, allow the hotel manager to adapt pricing policies to dynamic market conditions, and to analyze different booking scenarios by changing a compact set of meaningful parameters. 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 and to consider monthly as well as weekly seasonal effects.

The case study shows that our parametric models lead to results similar to other dynamic pricing models in the literature, while relying only on aggregated data. The average revenue increase is $\approx 19\%$ with respect to the original pricing

policies, and the risk of losses is absent for medium-big hotels and limited for small hotels, with a maximum loss probability of ≈ 0.03 . Moreover, experiments show that HotelSimu can simulate one year and a half in ≈ 2 s on average on a low-end machine. Also, a complete optimization can be run within one night.

Acknowledgements. A. Mariello would like to thank H. T. Nguyen for the useful advice on the RIM quantifiers.

References

1. Arnold, D.V.: Noisy Optimization with Evolution Strategies, vol. 8. Springer, Cham (2012). <https://doi.org/10.1007/978-1-4615-1105-2>
2. Arnold, D.V., Beyer, H.G.: A comparison of evolution strategies with other direct search methods in the presence of noise. *Comput. Optim. Appl.* **24**(1), 135–159 (2003)
3. Aydin, N., Birbil, S.: Decomposition methods for dynamic room allocation in hotel revenue management. *Eur. J. Oper. Res.* **271**(1), 179–192 (2018)
4. Aziz, H.A., Saleh, M., Rasmy, M.H., Elshishiny, H.: Dynamic room pricing model for hotel revenue management systems. *Egypt. Inform. J.* **12**(3), 177–183 (2011)
5. Baker, T.K., Collier, D.A.: A comparative revenue analysis of hotel yield management heuristics. *Decis. Sci.* **30**(1), 239–263 (1999)
6. Bayoumi, A.E.M., Saleh, M., Atiya, A.F., Aziz, H.A.: Dynamic pricing for hotel revenue management using price multipliers. *J. Revenue Pricing Manag.* **12**(3), 271–285 (2013)
7. Bertsimas, D., De Boer, S.: Simulation-based booking limits for airline revenue management. *Oper. Res.* **53**(1), 90–106 (2005)
8. Bertsimas, D., Popescu, I.: Revenue management in a dynamic network environment. *Transp. Sci.* **37**(3), 257–277 (2003)
9. Bitran, G., Caldentey, R.: An overview of pricing models for revenue management. *Manuf. Serv. Oper. Manag.* **5**(3), 203–229 (2003)
10. Bitran, G.R., Mondschein, S.V.: An application of yield management to the hotel industry considering multiple day stays. *Oper. Res.* **43**(3), 427–443 (1995)
11. Choi, T.Y., Cho, V.: Towards a knowledge discovery framework for yield management in the Hong Kong hotel industry. *Int. J. Hosp. Manag.* **19**(1), 17–31 (2000)
12. Denizci Guillet, B., Mohammed, I.: Revenue management research in hospitality and tourism: a critical review of current literature and suggestions for future research. *Int. J. Contemp. Hosp. Manag.* **27**(4), 526–560 (2015)
13. Figueira, G., Almada-Lobo, B.: Hybrid simulation-optimization methods: a taxonomy and discussion. *Simul. Model. Pract. Theory* **46**, 118–134 (2014)
14. Grube, P., Núñez, F., Cipriano, A.: An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network. *Simul. Model. Pract. Theory* **19**(1), 393–405 (2011)
15. Guadix, J., Cortés, P., Onieva, L., Muñuzuri, J.: Technology revenue management system for customer groups in hotels. *J. Bus. Res.* **63**(5), 519–527 (2010)
16. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) *Towards a New Evolutionary Computation. Studies in Fuzziness and Soft Computing*, vol. 192, pp. 75–102. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-32494-1.4>

17. Hansen, N.: 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, pp. 2389–2396. ACM (2009)
18. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**(2), 159–195 (2001)
19. Ivanov, S.: *Hotel Revenue Management: From Theory To Practice*. Zangador (2014)
20. Kleywegt, A.J.: An optimal control problem of dynamic pricing. School of Industrial and Systems Engineering, Georgia Institute of Technology 2001 (2001)
21. Lai, K.K., Ng, W.L.: A stochastic approach to hotel revenue optimization. *Comput. Oper. Res.* **32**(5), 1059–1072 (2005)
22. Liu, S., Lai, K.K., Wang, S.: Booking models for hotel revenue management considering multiple-day stays. *Int. J. Revenue Manag.* **2**, 78–91 (2008)
23. McGill, J.I., Van Ryzin, G.J.: Revenue management: research overview and prospects. *Transp. Sci.* **33**(2), 233–256 (1999)
24. Nissen, V., Propach, J.: On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Trans. Evol. Comput.* **2**(3), 107–119 (1998)
25. Rana, S., Whitley, L.D., Cogswell, R.: Searching in the presence of noise. In: Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature - PPSN IV*, vol. 1141, pp. 198–207. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61723-X_984
26. Stefanovic, D., Stefanovic, N., Radenkovic, B.: Supply network modelling and simulation methodology. *Simul. Model. Pract. Theory* **17**(4), 743–766 (2009)
27. Subulan, K., Baykasoglu, A., Eren Akyol, D., Yildiz, G.: Metaheuristic-based simulation optimization approach to network revenue management with an improved self-adjusting bid-price function. *Eng. Econ.* **62**(1), 3–32 (2017)
28. Vives, A., Jacob, M., Payeras, M.: Revenue management and price optimization techniques in the hotel sector: a critical literature review. *Tourism Econ.* **24**, 720–752 (2018)
29. Vock, S., Enz, S., Cleophas, C.: Genetic algorithms for calibrating airline revenue management simulations. In: 2014 Proceedings of the Winter Simulation Conference, pp. 264–275 (2014)
30. Welch, B.L.: The generalization of student's problem when several different population variances are involved. *Biometrika* **34**(1/2), 28–35 (1947)
31. Xiang, Z., Magnini, V.P., Fesenmaier, D.R.: Information technology and consumer behavior in travel and tourism: insights from travel planning using the Internet. *J. Retail. Consum. Serv.* **22**, 244–249 (2015)
32. Yager, R.R.: Quantifier guided aggregation using OWA operators. *Int. J. Intell. Syst.* **11**(1), 49–73 (1996)
33. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. *IEEE Trans. Evol. Comput.* **9**(3), 303–317 (2005)
34. Zakhary, A., Atiya, A.F., El-Shishiny, H., Gayar, N.E.: Forecasting hotel arrivals and occupancy using Monte Carlo simulation. *J. Revenue Pricing Manag.* **10**(4), 344–366 (2011)
35. Zhang, D., Weatherford, L.: Dynamic pricing for network revenue management: a new approach and application in the hotel industry. *INFORMS J. Comput.* **29**(1), 18–35 (2016)