ISSN 0005-1179, Automation and Remote Control, 2020, Vol. 81, No. 4, pp. 731–745. © Pleiades Publishing, Ltd., 2020.
Russian Text © The Author(s), 2019, published in Matematicheskaya Teoriya Igr i Ee Prilozheniya, 2019, No. 3, pp. 5–30.

Cost-Efficient Strategy in Clouds with Spot Price Uncertainty

E. E. Ivashko[*,**,a], **A. A. Ivashko**[*,**,b], **G. R. Safonov**[**,c], **and A. Tchernykh**[***,d]

[*] *Institute of Applied Mathematical Research of Karelian Research Centre, Russian Academy of Sciences, Petrozavodsk, Russia*
[**] *Petrozavodsk State University, Petrozavodsk, Russia*
[***] *CICESE Research Center, Ensenada, Baja California, Mexico*
e-mail: [a]*ivashko@krc.karelia.ru,* [b]*aivashko@krc.karelia.ru,*
[c]*jiri.safonov@gmail.com,* [d]*chernykh@cicese.mx*

Received April 20, 2018
Revised July 25, 2019
Accepted November 28, 2019

Abstract—We address a cloud spot bidding problem for user cost optimization. We propose stochastic optimization model to minimize the expected resource rental cost in the presence of spot price uncertainty. The model is based on the well-known full-information best-choice problem. Based on the model, we derive the strategy for cloud spots bidding. The strategy allows to minimize the expected cost for a spot instance in a specific period of time with quality of service guarantee. Our simulation analysis based on realistic settings clearly demonstrates the advantages of the proposed optimization solutions.

Keywords: cloud computing, spot instance, mathematical modeling, full-information best-choice problem, Amazon EC2

DOI: 10.1134/S000511792004013X

1. INTRODUCTION

Cloud computing providing on-demand access to resources (servers, storage, applications, and services) becomes a crucial part of the IT industry [17]. Cloud providers reoffer the models of access and monetization of computing resources. These business-approaches require new studies to analyze the rational (or optimal) behavior both of customers and cloud providers.

One of the major trend-setters of the cloud computing market is Amazon. Its Elastic Compute Cloud (Amazon EC2) platform provides several options for virtual servers renting [4]. In addition to On-demand and Reserved Instances, in 2009, Amazon has introduced Spot Instances (SIs) [3]. It had become a new approach to pricing and new way to obtain compute capacity with reduced prices.

With SIs, customers can bid on unused Amazon EC2 compute capacity. The current spot price can be changed on periodic basis reflecting current supply and demand. SIs are aimed to non-interactive applications with a flexible start time, which could be easily paused. The examples of such applications are video rendering/converting, data analysis, scientific modeling, and so on. Comparing to on-demand prices, auctions on SIs allow to save 50–90% off the customers costs.

We consider the following general problem. A cloud provider uses a flat auction to sell computing instances. At discrete moments of time, he generates a spot price, and grants computing instances to all the customers whose price levels are higher than the spot price. The customers who granted computing instances pay the same value, which is the lowest winning bid price level. Our objective is to find a computing slot in a specific period of time to minimize the expected cost.

We reduce this problem to a well-known full-information best-choice problem. The solution of the problem is a set of thresholds. The threshold values depend on spot prices distribution and remaining period. This strategy guarantees to a customer minimization of the expected prices.

The rest of the paper is organized as follows. In Section 2, we review several papers dealing with cloud auctions. In Section 3, we briefly describe Amazon spot auctions mechanism. In Section 4, we propose a mathematical model of SI auctions, and derive the strategies to minimize the expected spot price. We validate the proposed model and our strategy through simulation in Section 5. Section 6 shows an example of deriving the strategy based on Amazon SI auctions data. Finally, in Section 7, we summarize our contribution, discuss the results and describe future improvements.

2. RELATED WORKS

Cloud computing becomes a vital part of the modern information industry. It has received much attention from the scientific community. Zheng Li et al. 2014 [16] provide an overview of the researches on cloud spot market. The more recent and extended survey is given by D. Kumar et al. 2018 [14]. The authors explore two main research directions: investigation of the existing pricing mechanisms based on real examples of cloud providers politics and proposing new mechanisms for pricing and resources provisioning. They highlight different models describing spot prices distribution. Among probabilistic models several researches use Markovian or semi-Markovian models, develop Price Transition Probability Matrix, while others assume that spot prices are independent identically distributed random variables having normal distribution or mixture of Gaussians. In the latter model, there are also different approaches: statistics-based, economic-inspired, etc. In our work, we assume spot prices are independent identically distributed random variables having known probability distribution.

One of the research questions is the most profitable pricing policy for cloud provider.

Abhishek et al. 2012 [1] discuss the choice between fixed and market-defined prices. The authors show that the dynamic price approach is flexible but more complex. Based on queueing theory, they propose models of spot and pay as you go markets. These models are used to provide theoretical analysis and simulation results, which show that isolated pay as you go market is more profitable for a cloud provider than mixed with spot market. But this is true only for a monopolistic provider.

Xu and Li, 2012 [30] describe the mathematical model aimed to maximization of revenue of a cloud provider. The authors consider a scenario where cloud provider affects on spots demand varying the price of its service. The demand itself is characterized by stochastic processes of arrivals and departures with underlying Poisson distributions with parameters depending on price. The system, in general, is considered as continuous time birth-death Markov process. The authors derive the optimal cloud provider pricing policy, which maximizes revenue in the infinite time horizon. It should be mentioned that based on the proposed model, in some cases, dynamic pricing is the most profitable.

Song et al. 2012 [22] develop a bidding strategy for cloud service brokers. A special profit aware dynamic bidding algorithm is proposed. The authors demonstrate that the algorithm shows a near optimal performance. It also can be used with other cost objective functions of the cloud service brokers.

As it is mentioned above, Amazon has introduced their SIs in 2009, and sells them at auctions, where customers make bids on desired resources. As announced, the actual price depends on the whole amount of available resources and demand. SIs attracted much interest from research community as a new price model. One of the most interesting issues of SI auctions is dynamic pricing paradigm. Its characterization is fundamental from the point of view of effective stochastic scheduling algorithms development and fault tolerant mechanisms.

Ben-Yehuda et al. 2011 [5] analyze real pricing based on historic data of Amazon auctions on SIs. They prove that spot prices are usually not market-driven, but instead, they are typically generated by Amazon randomly within a tight price interval via a dynamic hidden reserve price. The authors notice that prices on different types of instances show similar behavior. This conclusion is made by analysis of Amazon EC2 spot prices of first half of 2010.

Cheng et al. 2016 [7] study price differentials between West and East SIs markets of Amazon EC2, and pricing dynamics.

Javadi et al. 2013 [9] show that spot price dynamics of each SI is characterized by a mixture of Gaussian distribution with three or four components. The authors perform a comprehensive analysis of SIs based on one year price history in four data centers of Amazon EC2. They analyzed all different types of SIs and determined the time dynamics for spot price in hour-in-day and day-of-week. They highlighted bi-modality of spot price probability distribution functions, and propose to use a mixture of Gaussians distribution with three or four components to model spot price dynamics. Several methods are used to estimate the parameters of this distribution. The proposed model is validated through extensive simulations.

Wallace et al. 2013 [28] use neural networks to predict spot prices. For the experiments, they used spot price data points for seven months starting December 2009 and ending June 2010. To predict spot prices standard Multi-Layer Perceptron model with a back propagation error training algorithm and an adaptive learning rate is used. The authors perform simulations based on Amazon EC2 spot prices data. The simulations demonstrate very good results even for sudden price changes, and show that neural networks are well suited for prediction of spot prices. The results of these works can be used by customers to choose the best bidding strategy.

Time series-based analysis is a popular method to analyze spot prices. M.B. Chhetri et al. 2018 [8] first decompose the Spot price history into time series components; each component, which can exhibit deterministic or non-deterministic qualities, is then separately forecast using different standard forecasting techniques and look back periods; and finally, the individual forecasts are aggregated to form the Spot price forecast. S. Alkharif et al. 2018 [2] use LSTM model to forecast time series for EC2 cloud price.

Information on spot price dynamics is used to design bidding strategies, addressing, for example, cost minimization and reliability maximization.

Karunakaran et al. 2015 [12] compare four simple bidding strategies: (1) bidding close to the reserved instance price; (2) bidding above the average spot price, which is deduced from the spot-price history; (3) bidding close to the on-demand price; and (4) bidding above the on-demand price. The strategies are considered from the point of view of several customer performance metrics, which are (1) cost of job completion; (2) wait time; and (3) interruption during job execution. The authors conclude that first strategy is well-suited for price-sensitive users who have flexible job-completion times, and are capable of frequent checkpointing. The second and third strategies show good tradeoff between costs, waiting times, and interruption rates. Finally, fourth strategy demonstrates cost increase that does not accompanied with a significant reduction in wait times or interruption rates. Thus, bidding above on-demand, though attractive, is not useful.

Voorsluys, 2014 [27] proposes an approach to run deadline-constrained computational jobs on a pool of computational resources composed solely by low-cost SI. The author builds a resource management and scheduling policy system, which is called SpotRMS. The policy uses job runtime estimations to decide both the best types of instances to run each job, and when jobs should be run to meet their deadlines. In particular, the author studies techniques to run computational jobs on intermittent SI. These techniques include the use of different bidding strategies and fault tolerance techniques to tolerate unplanned unavailability of SI. The proposed approach is verified

Table 1. Related papers comparison

Paper/Parameter	Persp. (user/ cloud)	Mechanism (existing /new)	Theoret. analysis	Data (real/ synthetic)	Amazon SI	Prices charact.
V. Abhishek et al., 2012 [1]	cloud	new	+	synthetic		
H. Xu, B. Li, 2012 [30]	cloud	new	+	synthetic		
W. Voorsluys, 2014 [27]	cloud	new	+	synthetic		
O. Agmon Ben-Yehuda, et al., 2011 [5]		existing	+	real	+	+
B. Javadi, et al., 2013 [9]				real	+	+
R.M. Wallace et al., 2013 [28]		existing		real	+	+
S. Karunakaran, R.P. Sundarraj, 2015 [12]	user	existing		real	+	
K. Sowmya, R.P. Sundarraj, 2013 [23]	user	existing	+	real	+	
Y. Song, et al., 2012 [22]	cloud	new	+	real	+	
				synthetic	+	
S. Tang, et al., 2012 [25]	user	existing	+	real	+	+
B. Kaminski, P. Szufel, 2015 [10]				real	+	+
W. Wang, et al., 2013 [29]	cloud	new	+	synthetic		
A.N. Toosi, et al., 2016 [26]	cloud	new	+	synthetic		
I. Menache, et al., 2014 [18]	cloud	new	+			

on trace-driven simulation based on Amazon data and demonstrates price effectiveness. Petcu, 2014 [20] describes solutions to consume resources and services from multiple clouds.

Menache et al. 2014 [18] introduce a self-learning algorithm of costs optimization based on adaptive rent on-demand and spot instances. The authors introduce an algorithm for resource allocation to address the tradeoff between computation cost of on-demand instances and performance of SIs issues. The algorithm uses machine learning approaches to dynamically adapt resource allocation based on its performance on prior job executions, history of spot prices, and workload characteristics.

Kokkinos et al. 2014 [13] presented Cost and Utilization Optimization mechanism, formulated as an Integer Linear Programming problem, for optimizing the cost and the utilization of a set of running Amazon EC2 instances.

Sowmya and Sundarraj, 2013 [23] model the bidding strategies in a spot market as a well-known from game theory a prisoner dilemma game. The authors check their discoveries against real time data from Amazon EC2 spot market. Considering that most of the spot market bidders are repetitive bidders, the authors implement a Co-operation strategy which is in-line with the Iterated Prisoner Dilemma Game.

Tang et al. 2012 [25] propose bidding strategies to minimize the cost and provide good reliability. The authors use Constrained Markov Decision Process, Price Transition Probability Matrix, and linear programming to obtain an optimal randomized bidding strategy (which is called AMAZING). The authors evaluate the model and demonstrate how users should bid optimally on SIs to reach different objectives with desired levels of confidence.

Kamitski and Szufel, 2015 [10] introduce EC2 cloud pricing simulator. It operates on historical data and emulates the Amazon EC2 pricing mechanism, so that it could be used to evaluate different spot bidding strategies.

Wang et al. 2013 [29] and Toosi et al. 2016 [26] propose new auction mechanisms addressing truthfulness and revenue maximization.

Wang et al. 2013 [29] design a special type of dynamic auctions. This design determines the amount of instances to be auctioned in each period, as well as the underlying auction mechanisms based on dynamic payment schemes. The authors prove that the proposed design is two-dimensionally truthful and asymptotically optimal for high demands.

Toosi et al. 2016 [26] adapt the Consensus Revenue Estimate auction mechanism to the setting of a multi-unit online auction for cloud resources. They also combine it with a scheme for dynamically calculating reserve prices based on data center Power Usage Effectiveness and electricity costs. The authors notice that the final mechanism is envy-free, has a high probability of being truthful, and generates a near optimal profit for the provider. It also maximizes revenue without requiring prior knowledge on the bid distributions. Based on the simulations the authors show that the proposed mechanism outperforms the uniform price auction.

In Table 1, we summarize related papers considering their main characteristics: user or cloud provider oriented study; existing or new mechanism are used, theoretical or experimental analysis is conducted, real or synthetic data are applied, Amazon Spot Instance auction or others are considered; prices characterization is included or not.

3. AMAZON SPOT AUCTIONS

Amazon offers its customers three renting mechanisms: reserved instances, on-demand instances, and spot auctions. They provide different assurances regarding when instances can be launched and terminated, and with what costs. Reserved instance gives a client ability to launch reserved instance whenever they wish. A client can purchase an on-demand instance when he needs it, at a higher hourly fee, but with no guarantee that launching will be possible at any given time. Both reserved and on-demand instances remain active until terminated by the client.

The third type is a Vickrey-style auction on unused capacity. It provides no guarantee regarding either launch time or termination time. Clients bid the maximum hourly price they are ready to pay for a SI. The request on the instance is granted, if the bid is higher than the spot price, otherwise, it waits. All winning customers pay the same price, which is equal to the value of the lowest winning bid. The spot price changes periodically based on supply and demand. The instance runs until the client terminates it, or the spot price increases above clients maximum price.

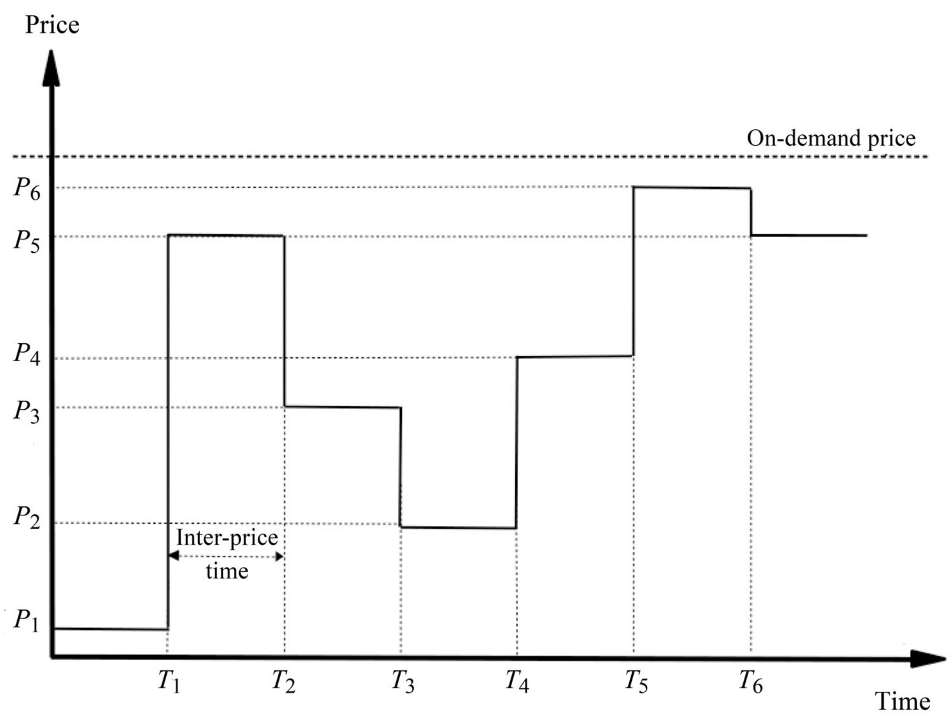

Fig. 1. Spot pricing structure.

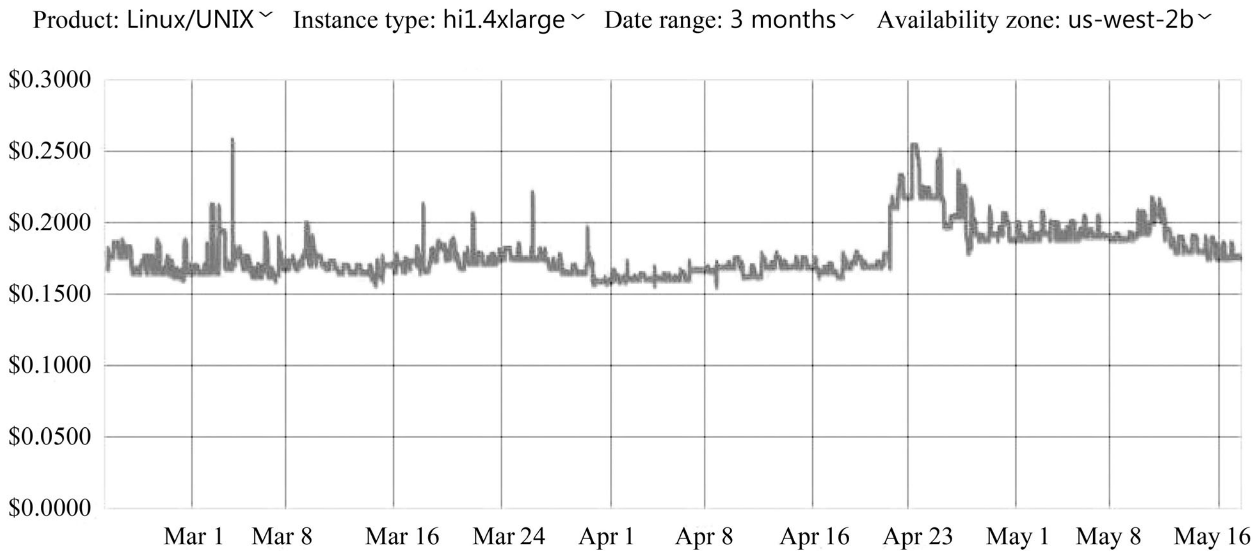

Fig. 2. Spot pricing dynamics.

Amazon EC2 is charging based on full hours, unless the instance is terminated due to a spot price change. In this case, the last fraction of an hour is free of charge.

Figure 2 shows an example of three months spot price dynamics (see Amazon Spot Instance Pricing History [24]).

Clearly, lower bidding prices usually provide lower cost. However, they degrade other metrics such as job completion time and number of interruptions. Therefore, spot auction is a trade-off between reliability of service and cost of SIs.

4. PROPOSED STRATEGY

4.1. Mathematical Model

There is a following well-known best-choice problem proposed by Cayley [6]. Moser, 1956 [19] reformulated the Cayley's problem as follows: a user observes, one by one, random variables $X_1, \ldots, X_k$ known to be independent identical distributed (iid) from a uniform distribution on the interval $(0, 1)$. If he stops after observing X_j, then he receives X_j as a reward. The aim is to maximize the reward.

It is proved that the optimal strategy is to stop when there are m observations left, if the value of the present observation is greater than E_m, where the E_m is defined recursively by

$$E_n = 0 \quad \text{and} \quad E_m = \frac{(1 + E_{m+1}^2)}{2}, \quad m = 1, \ldots, n-1.$$

We will call thresholds values E_m used for the such kind of strategy.

The sequence of thresholds is decreasing as it shown in Fig. 3.

We use these results to reformulate previously described problem of optimal bidding strategy at SIs auction.

4.2. Solution

As one can see, the best-choice problem has clear similarities with the best-bidding problem at cloud auction. We assume that a client would like to rent an instance in a specific period

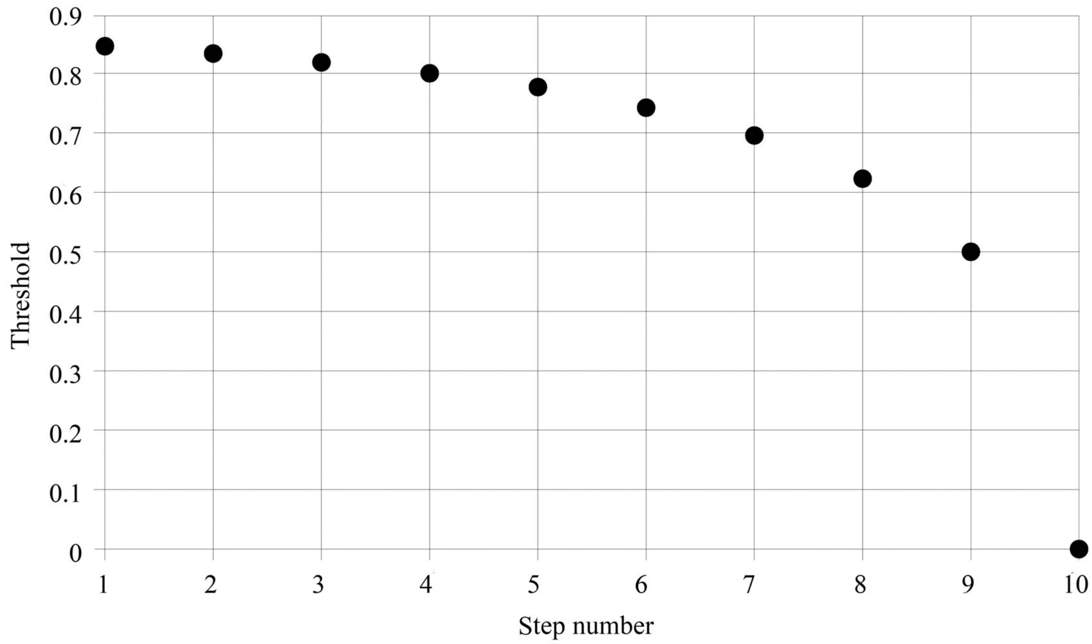

Fig. 3. The sequence of thresholds.

of time. As in the best-choice problem, the client observes one by one cloud spot prices which are distributed from 0 to the maximum price, $p_{\max}$ value. Therefore he can use a threshold-based strategy to minimize the expected cost for a spot instance. Assuming that the prices are independent identically distributed with a known continuous probability function, the best strategy could be derived using the same procedure as is used in the best-choice problem.

Let us give the formal definition of the problem. There is a spot auction, where spot prices are independent identical distributed (iid) random variables $x_1, x_2, x_3, \ldots$ with a continuous probability function (pdf) $F(x)$ on the interval $[0, p_{\max}]$. $p_{\max}$ corresponds to a maximal possible SI price at the auction. Obviously, it is equal to on-demand price. A customer has n periods to win a SI. If he fail to get a spot earlier, he has to buy on-demand instance on the maximum price $p_{\max}$ on the last period.

Let the customer sets his bid τ_i before the ith SIs auctioning. He aims to minimize expected spot price rent with a given period of time. The considered problem is a well-known full-information best-choice problem solved by the method of backward induction [19]. Failing to win a SI at the period n, the customer have to buy it on the maximum price $p_{\max}$. So, any price less or equal of $p_{\max}$ is good for the customer: $\tau_n = p_{\max}$.

Taking into account that spot prices are iid random variables with continuous pdf $F(x)$, the expected cost at the period n is equal to $\int_0^{\tau_n} x dF(x)$, Lebesgue–Stieltjes integral with respect to the distribution function $F(x)$, see [11].

At step $n-1$, the customer price level is the minimum from the current SI price, and expected cost at the nth period:

$$\tau_{n-1} = \mathbf{E}\big[\min\{\tau_n, x\}\big] = \int\limits_0^{p_{\max}} \min\{\tau_n, x\} dF(x) = \int\limits_0^{\tau_n} x dF(x) + \int\limits_{\tau_n}^{p_{\max}} \tau_n dF(x).$$

Here $\mathbf{E}[X]$ denotes an expectation of random variable X.

Continuing the backward induction, we get the following solution as a system of recurrent equations:

$$\begin{cases} \tau_n = p_{\max} \\ \tau_i = \mathbf{E}[\min\{\tau_{i+1}, x\}] = \int\limits_0^{\tau_{i+1}} x dF(x) + \int\limits_{\tau_{i+1}}^{p_{\max}} \tau_{i+1} dF(x). \end{cases} \tag{4.1}$$

Using formula (4.1), one can get the optimal bid τ_i value at any step i for a given period n. At first step, a customer should use the first (the lowest) price. If this bid τ_1 does not win, the next bid τ_2 should be used at the next step. Following this process, the customer is guaranteed to win an instance in a period n with the lowest possible expected spot price.

5. EXPERIMENTAL ANALYSIS

As it was mention above, formula (4.1) describes the solution for any continuous pdf $F(x)$. However, according to many researches the real distribution of Amazon spot prices is composite. For example, following Javadi et al. 2013 [9], spot prices has a mixture of Gaussian distributions with three or four components. However, it is known that any continuous distribution can be reduced to uniform one on $[0,1]$ by appropriate scaling (see, for example, Law and Kelton. 1999 [15]). For standard uniform distribution, formula (4.1) is transformed to the simpler one:

$$\begin{cases} \tau_n = 1 \\ \tau_i = \mathbf{E}[\min\{\tau_{i+1}, x\}] = \int\limits_0^{\tau_{i+1}} x dx + \int\limits_{\tau_{i+1}}^{1} \tau_{i+1} dx, \quad i = 1, \ldots, n-1. \end{cases} \tag{5.1}$$

Based on formula (5.1), Table 2 and Fig. 4 show the optimal bid values for the case of 10 steps period. We perform the numerical simulation to compare proposed strategy (Strategy 1) with three others:

- Strategy 2: constant strategy—a customer defines a constant value and bids it at every step. If he loses all n bids, he buys a spot for on-demand price.
- Strategy 3: random strategy—a customer simulates a random variable uniformly distributed on $[0,1]$ at each step and uses it as a threshold. If he loses all n bids, he buys a spot for on-demand price.
- Strategy 4: linear strategy—a customer uses thresholds defined by formula

$$\tau = kn + b,$$

  where n is a step number and coefficients k and b depend on the overall steps number.

We enumerate all the possible events to discover the best constant threshold. Given a specific threshold x the expected spot price is defined by following formula:

$$\begin{aligned} P(x) &= \int\limits_0^x y dy + \int\limits_x^1 dy \int\limits_0^x y dy + \ldots + \underbrace{\int\limits_x^1 dy \ldots \int\limits_x^1 dy}_{n-1} \int\limits_0^x y dy + \underbrace{\int\limits_x^1 dy \ldots \int\limits_x^1 dy}_{n} \\ &= (1-x)^n + \frac{x^2}{2} \sum_{i=0}^{n-1} (1-x)^i. \end{aligned} \tag{5.2}$$

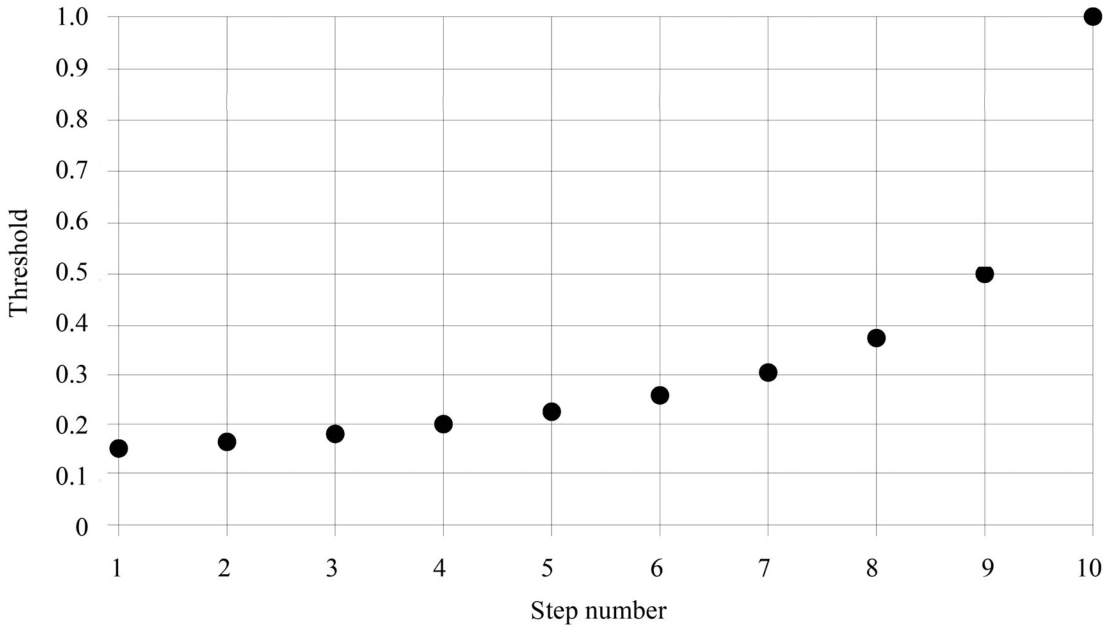

Fig. 4. The optimal bid values for the case of 10 steps period and standard uniform distribution.

Table 2. Optimal bid values, $n = 10$

step	1	2	3	4	5	6	7	8	9	10
price	0.150	0.164	0.180	0.20	0.225	0.258	0.305	0.375	0.5	1.0

Here, first additive component corresponds to an event of appearing spot price lower than x at first step; second component—at second step; and so on. Finally, the last component corresponds to an event of missing low spot price.

Obviously, the greater x the more chance to rent a spot, but the expected price of the rented spot is higher. The lower x, the lower price of the rented spot, but at the same time the less chance to hire it. So, the best constant threshold for the case of uniform on $[0, 1]$ prices distribution is defined by the following formula:

$$x^* = \operatorname{argmin}\left((1-x)^n + \frac{x^2}{2}\sum_{i=0}^{n-1}(1-x)^i\right). \tag{5.3}$$

For the case of $n = 10$, formulae (3) and (4) give threshold value $x^* = 0.275$ and mean price $P(x^*) = 0.17$.

The best coefficients k and b of the linear strategy are derived from the numerical simulation; they are $k = 0.04$ and $b = 0.09$ for the case of $n = 10$ steps, and $k = 0.001$ and $b = 0.01$ for the case of $n = 100$.

The thresholds for all four strategies are given in Table 3, and shown in Fig. 5. Note that Strategy 3 gives only an example of the threshold values. As Strategy 3 uses random threshold, Table 3 and Fig. 5 show pseudo random numbers generated with uniform distribution on $[0, 1]$ interval. The same is also true for the case of $n = 100$.

The simulation results are given in Table 4.

Spot prices are generated as pseudo random values uniform on $[0, 1]$. At the second step, the spot price is lower than threshold of Strategy 2. So, using Strategy 2 at step 2 one could be

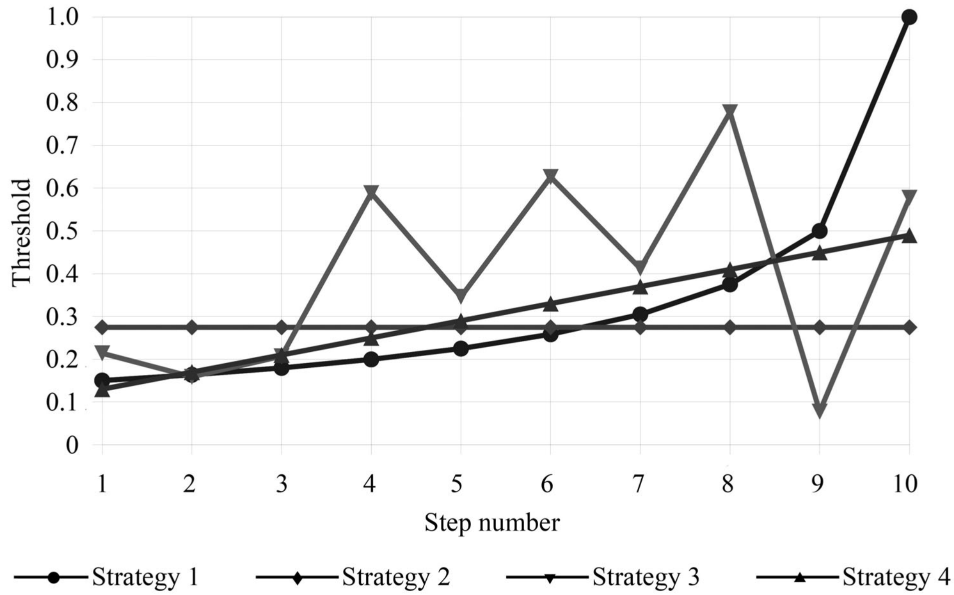

Fig. 5. Bid values generated by strategies for 10 steps period.

able to get an instance at price 0.253. Using Strategy 3, one could be able to get an instance at step 4, when pseudorandomly generated threshold become higher than spot price (so, acceptance step is 4 and spot price is 0.302). Spot price appeared lower than the linear strategy's threshold at step 5 (acceptance step is 5 and spot price is 0.283). Finally, the last winning bid corresponds to

Table 3. Bid values generated by strategies for 10 steps period

Threshold/ strategy	1	2	3	4	5	6	7	8	9	10
Strategy 1	0.15	0.164	0.18	0.2	0.225	0.258	0.305	0.375	0.5	1.0
Strategy 2	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275	0.275
Strategy 3	0.214	0.159	0.207	0.589	0.346	0.626	0.412	0.777	0.078	0.577
Strategy 4	0.13	0.17	0.21	0.25	0.29	0.33	0.37	0.41	0.45	0.49

Table 4. Simulation results ($n = 10$)

Step Strategy	1	2	3	4	5	6	7	8	9	10	Acceptance step	Spot price
Strategy 1	0.150	0.164	0.18	0.2	0.225	0.258					6	0.161
Strategy 2	0.275	0.275									2	0.253
Strategy 3	0.214	0.159	0.207	0.589							4	0.302
Strategy 4	0.13	0.17	0.21	0.25	0.29						5	0.283
spot price	0.359	0.253	0.721	0.302	0.283	0.161						

Table 5. Strategies comparison ($n = 10$)

	Mean acceptance step	Mean spot price
Strategy 1	4.604	0.139
Strategy 2	3.546	0.185
Strategy 3	1.999	0.334
Strategy 4	4.365	0.148

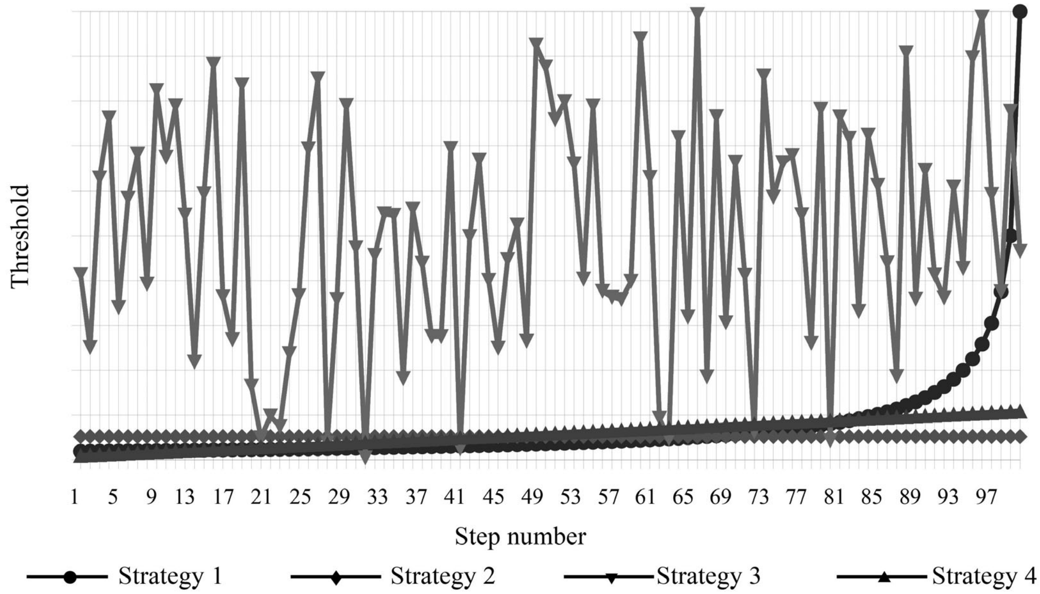

Fig. 6. Bid values generated by strategies for 100 steps period.

Table 6. Strategies comparison ($n = 100$)

	Mean acceptance step	Mean spot price
Strategy 1	35.294	0.019
Strategy 2	19.182	0.031
Strategy 3	2.0	0.333
Strategy 4	31.127	0.022

Strategy 1. The threshold value which exceeded spot price is 0.258. So, acceptance step is 6 and spot price is 0.161.

Note, that Table 4 presents only an example of stochastic process realization. To estimate the expected values, we performed numerical simulation 10^6 times experiments. Mean acceptance step and mean spot price are given in Table 5.

As one can see, Strategy 1 defined by recurrent formula (2) provides better results than three other strategies from the point of view of mean spot price.

The same experiments are performed for a period of 100 steps. The strategies are visualized in Fig. 6. Table 6 gives the mean acceptance steps and mean spot prices comparison.

Strategy 3 does not depend on step number and, in the most cases, gives much higher thresholds, than three other strategies. It is obvious that following this strategy one gets much higher spot prices and rents an instance much earlier comparing to the results of three other strategies. Following the problem statements, a client has a specific period to rent an instance. So, if the given period is long enough, it is more wisely to hold a low threshold relying on low spot price.

As one can see, our recurrent thresholds minimize the expected spot price. Also, the longer period promises the more profit in spot price.

6. AMAZON SPOT INSTANCES BIDDING MODELLING

In this section, we discuss a method to derive the optimal strategy using the real Amazon spot prices data [24]. We build a prices distribution function and derive the optimal thresholds using formula (4.1).

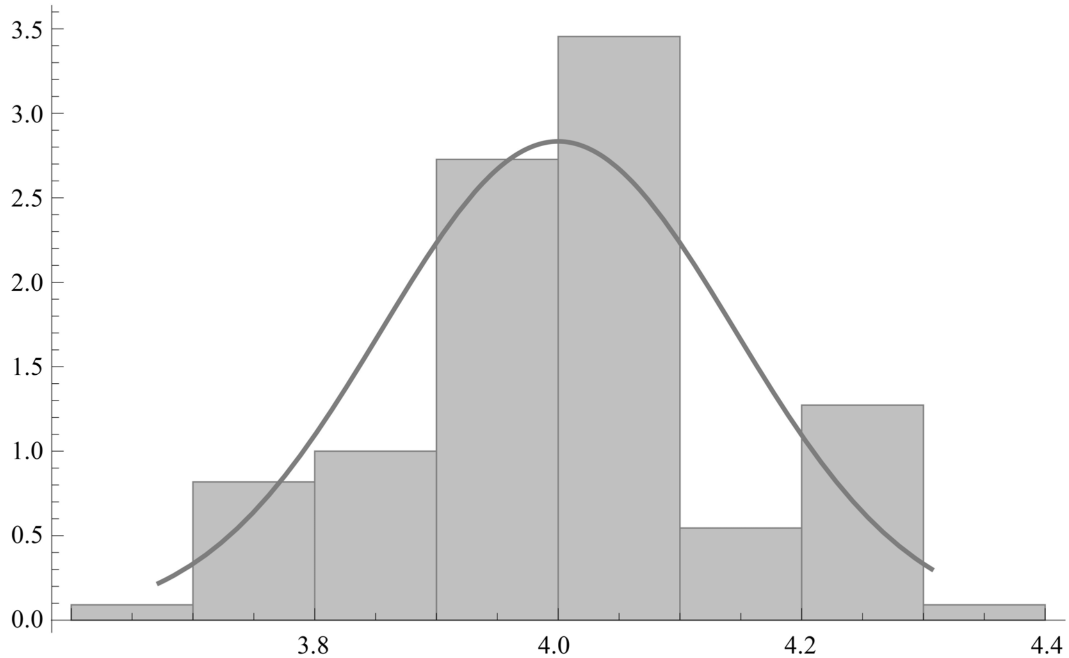

Fig. 7. Histogram and distribution function of spot prices.

First, using statistics of spot prices construct a prices distribution function. For example, we use statistics of `Linux p3.8xlarge` instance in a period of February 1–22, 2019. Having the data we construct a histogram of prices. Analyzing the data one can construct the prices distribution function (using the appropriate statistical methods [21]), see Fig. 7. We used the following truncated Gaussian distribution:

$$f_{b_1,b_2}(x) = \begin{cases} \dfrac{C}{\sqrt{2\pi}\sigma} e^{\frac{-(x-a)^2}{2\sigma^2}}, & x \in [b_1, b_2] \\ 0, & x \notin [b_1, b_2], \end{cases} \tag{6.1}$$

here

$$C = \frac{1}{\int\limits_{b_1}^{b_2} \dfrac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-a)^2}{2\sigma^2}} dx},$$

$b_1 = p_{\min} = 3.672$, $b_2 = p_{\max} = 4.307$ and $a = 4.002$, $\sigma = 0.141$.

Using formula (4.1) we derive the optimal thresholds τ_i, $i = 1, \ldots, n$ (see Table 7).

We perform the numerical simulation to compare proposed strategy (Strategy 1) with three others described in Section 5 (their parameters are adopted to the distribution (6.1), see Table 8).

As one can see, Strategy 1 (the optimal one) gives the lowest prices of an instance. The same procedure can be used for any other time period where the prices have a probabilistic continuous distribution.

7. CONCLUSION AND DISCUSSION

We consider a spot bidding problem for cost minimization of an instance renting from a client point of view. We suppose that the client does not have to rent an instance immediately, and

Table 7. Bid values generated by optimal strategy τ_i for 10 steps period

i	1	2	3	4	5	6	7	8	9	10
τ_i	3.838	3.846	3.855	3.866	3.879	3.895	3.916	3.947	4.0	4.307

Table 8. Results of the numerical simulations, $n = 10$

Step	1	2	3	4	5	6	7	8	9	10	Mean step	Mean price
Strategy 1	3.838	3.846	3.855	3.866	3.879	3.895	3.916	3.947	4.000	4.307	8.364	3.990
Strategy 2	3.916	3.916	3.916	3.916	3.916	3.916	3.916	3.916	3.916	3.916	7.364	4.094
Strategy 3	3.862	3.936	3.890	3.913	3.873	3.895	4.051	3.950	3.973	4.013	7.273	4.073
Strategy 4	3.819	3.843	3.866	3.890	3.914	3.938	3.962	3.985	4.010	4.033	7.727	4.018

has a specific period to make a decision. We demonstrate similarities between cloud SIs auctions and well-known full-information best-choice problem. Based on this analysis, we propose a novel threshold-based bidding strategy, which minimizes the expected cost of an instance renting.

We perform a joint analysis of the proposed strategy with three heuristic strategies. Corresponding results indicate that our algorithm provide the best cost and waiting time trade-offs. We also propose a method to derive the optimal strategy using the real Amazon spot prices data.

However, further study is required to assess its actual efficiency and effectiveness on other spot price distributions. This will be subject of future work requiring a better understanding of the impact of a probability distribution on effectiveness of bidding strategies. The distribution is complex, its parameters are changing over the time. Its form is dependent on instance type, region, seasonality and so on. Our model can be integrated with dynamic analysis of the current spot prices probability distribution.

Other direction of research is to take into account possible interruption of the instance. When we renting an instance, there is no guarantee that the instance will not be interrupted, and the client will complete his task. This can be achieved by a special set of thresholds. These thresholds should be increasing in time to guarantee instance renting in a specific period of time. From the other side, the thresholds should be low enough to provide low expected instance renting cost. But low thresholds result to high probability of the instance interruption. If it is important to take into an account the probability of the instance interruption in mathematical model.

FUNDING

This work was partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnologa, Mexico), grant no. 178415.

REFERENCES

1. Abhishek, V., Kash, I.A., and Key, P., Fixed and Market Pricing for Cloud Services, *Proc. of the 7th Workshop Economics of Networks System Computer (NetE-con 2012)*, IEEE Computer Society, Orlando, 20 March 2012, pp. 157–162.
2. Alkharif, S., Lee, K., and Kim, H., LSTM Model to Forecast Time Series for EC2 Cloud Price, *16th IEEE Int. Conf. on Dependable, Autonom. and Secure Computing, 16th IEEE Int. Conf. on Pervasive Intelligence and Computing, 4th IEEE Int. Conf. on Big Data Intelligence and Computing, 3rd IEEE Cyber Science and Technology Congress (DASC/PICOM/DATACOM/CYBERSCITECH)*, 2018, pp. 1085–1088. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00067
3. *Amazon EC2 Spot Instances.* http://aws.amazon.com/ec2/spot/
4. *Amazon Elastic Compute Cloud (Amazon EC2)*, Amazon Inc. http://aws.amazon.com/ec2

5. Ben-Yehuda, A.O., Ben-Yehuda, M., Schuster, A., and Tsafrir, D., Deconstructing Amazon EC2 Spot Instance Pricing, *2011 IEEE Third Int. Conf. on Cloud Computing Technology and Science (CloudCom)*, Athens, 2011, pp. 304–311. https://doi.org/10.1109/CloudCom.2011.48
6. Cayley, A., Mathematical Questions with Their Solutions, *Educational Times*, 1875, no. 23, pp. 18–19. 1875.
7. Cheng, H.K., Li, Z., and Naranjo, A., Cloud Computing Spot Pricing Dynamics: Latency and Limits to Arbitrage, *Inf. Syst. Res.*, 2016, vol. 27, no. 1, pp. 145–165.
8. Chhetri, M.B., Lumpe, M., Quoc, B.V., and Kowalczyk, R., On Forecasting Amazon EC2 Spot Prices Using Time-Series Decomposition With Hybrid Look-Backs, *2017 IEEE 1st Int. Conf. on EDGE Computing (IEEE EDGE)*, Goscinski, A.M. and Luo, M., Eds., 2017, pp. 158–165. https://doi.org/10.1109/IEEE. EDGE.2017.29
9. Javadi, B., Ruppa K. Thulasiram, Rajkumar Buyya Characterizing Spot Price Dynamics in Public Cloud Environments, *Future Generation Comput. Syst.*, 2013, vol. 29, no. 4, pp. 988–999. http://dx.doi.org/10.1016/j.future.2012.06.012
10. Kaminski, B. and Szufel, P., On optimization of Simulation Execution on Amazon EC2 Spot Market, *Simulat. Modelling Pract. Theory*, 2015, vol. 58, pp. 172–187.
11. Karr, A.F., *Probability*, New York: Springer-Verlag, 1993.
12. Karunakaran, S. and Sundarraj, R.P., Bidding Strategies for Spot Instances in Cloud Computing Markets, *IEEE Internet Comput.*, 2015, vol. 19, no. 3, pp. 32–40. https://doi.org/10.1109/MIC.2014.87
13. Kokkinos, P., Varvarigou, T.A., Kretsis, A., Soumplis, P., and Varvarigos, E.A., SuMo: Analysis and Optimization of Amazon EC2 Instances. *J. Grid Comput.*, 2014, vol. 13, pp. 255–274.
14. Kumar, D., Baranwal, G., Raza, Z., and Vidyarthi, D.P., A Survey on Spot Pricing in Cloud Computing, *J. Network Syst. Manage.*, 2018, pp. 809–856. https://doi.org/10.1007/s10922-017-9444-x
15. Law, A.M. and Kelton, D.M., *Simulation Modeling and Analysis*, Boston: McGraw-Hill Higher Education, 1999, 3rd ed.
16. Li, Z., O'Brien, L., Ranjan, R., Flint, S., and Zomaya, A., On Cloud Spot Market: An Overview of the Research, *First Int. Conf. on Intelligent Cloud Computing (ICC)*, Al-Saidi, A., Fleischer, R., Maamar, Z., and Rana, O.F., Eds., Muscat, Oman, 2015, pp. 46–61.
17. Mell, P. and Grace, T., *The NIST Definition of Cloud Computing*, NIST Special Publication 800145, 2011. http://dx.doi.org/10.6028/NIST.SP.800-145
18. Menache, I., Shamir, O., and Jain, N., On-demand, Spot, or Both: Dynamic Resource Allocation for Executing Batch Jobs in the Cloud, *11th Int. Conf. on Autonomic Computing (ICAC 14)*, 2014, pp. 177–187. www.usenix.org/conference/icac14/technical-sessions/presentation/menache
19. Moser, L., On a Problem of Cayley, *Scripta Math.*, 1956, vol. 22, no. 5, pp. 289–292.
20. Petcu, D., Consuming Resources and Services from Multiple Clouds, *J. Grid Comput.*, 2014, vol. 12, no. 2, pp. 321–345.
21. Rohatgi, V.K., *An Introduction to Probability Theory and Mathematical Statistics*, New York: Wiley, 1976.
22. Song, Y., Zafer, M., and Lee, K.-W., Optimal Bidding in Spot Instance Market, *Proc. IEEE INFOCOM*, 2012, art. no. 6195567, pp. 190–198.
23. Sowmya, K. and Sundarraj, R.P., Strategic Bidding for Cloud Resources under Dynamic Pricing Schemes, *Proc. Int. Symp. on Cloud and Services Computing (ISCOS)*, 2012, art. no. 6481231, pp. 25–30.
24. *Spot Instance Pricing History Amazon Elastic Compute Cloud.* http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html
25. Tang, S., Yuan, J., and Li, X.Y., Towards Optimal Bidding Strategy for Amazon EC2 Cloud Spot Instance, *2012 IEEE 5th Int. Conf. on Cloud Computing (CLOUD)*, Honolulu, 2012, pp. 91–98. https://doi.org/10.1109/CLOUD.2012.134
26. Toosi, A.N., Vanmechelen, K., Khodadadi, F., and Buyya, R., An Auction Mechanism for Cloud Spot Markets, *ACM Trans. Autonom. Adapt. Sys.*, 2016, vol. 11, no. 1, art. no. 2, pp. 1–33.
27. Voorsluys, W., *Resource Provisioning in Spot Market-Based Cloud Computing Environments*, Melbourne: Univ. Melbourne, Australia, 2014.

28. Wallace, R.M. et al., Applications of Neural-Based Spot Market Prediction for Cloud Computing, *IEEE 7th Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Berlin, 2013, pp. 710–716. https://doi.org/10.1109/IDAACS.2013.6663017

29. Wang, W., Liang, B., and Li, B., Revenue Maximization with Dynamic Auctions in IaaS Cloud Markets, *IEEE/ACM 21st Int. Symp. on Quality of Service (IWQoS)*, Montreal, 2013, pp. 1–6. https://doi.org/10.1109/IWQoS.2013.6550265

30. Xu, H. and Li, B., Maximizing Revenue with Dynamic Cloud Pricing: The Infinite Horizoncase, *IEEE Int. Conf. on Communications (ICC)*, Ottawa, 2012, pp. 2929–2933. https://doi.org/10.1109/ICC.2012.6364013