Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 193 – 215
© TÜBİTAK
doi:10.55730/1300-0632.3978

A robust model for spot virtual machine bidding in the cloud market using information gap decision theory (IGDT)

Mona NAGHDEHFOROUSHHA[1], **Mehdi DEHGHAN TAKHT FOOLADI**[2], **Mohammad Hossein REZVANI**[1,*], **Mohammad Mehdi GILANIAN SADEGHI**[1]

[1]Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[2]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Received: 04.02.2022 • **Accepted/Published Online:** 13.12.2022 • **Final Version:** 19.01.2023

Abstract: The spot market is one of the most common cloud markets where cloud providers, such as Amazon EC2, rent their surplus computing resources at lower prices in the form of spot virtual machines (SVMs). In this market, which is often managed through an auction mechanism, users seek optimal bidding strategies for renting SVMs to minimize cost and risk. Uncertainty in the price of SVMs and their low availability/reliability is a challenging issue to bid on the user side. In this paper, we present a robust model for minimizing the cost of executing tasks by considering the uncertainty of the price of SVMs based on the Information Gap Decision Theory (IGDT). It evaluates the risk-aversion and the risk-seeker nature of the user's bidding strategy and measures the cost of risk/immunity. The main advantage of this method is that it formulates user decisions without the need for any presumption about the price distribution of SVMs. With this decision-support system, users can rely on the predicted confidence intervals to make the optimal bid for future time slots according to the selected risk level. The results are compared with Monte Carlo simulations and a scenario-based approach to evaluate the effectiveness of the proposed IGDT-based model. Evaluation results based on historical Amazon EC2 prices confirm the efficiency of the proposed method to handle the uncertain nature of the price of SVMs in terms of significant criteria such as robustness cost, opportunity cost, uncertainty budget, and execution time.

Key words: Cloud computing market, bidding strategy, price uncertainty, information gap decision theory (IGDT)

1. Introduction

Cloud service providers such as Amazon offer a variety of virtual machines (VMs) in the form of spot virtual machine (SVM) instances [1, 2]. The auction mechanism is the most common way to bid on the SVM. Amazon's price history shows that the price of SVMs fluctuates over time [3, 4]. The user's goal is to minimize overall processing costs by performing tasks on SVMs under a series of constraints. Usually, the user and the provider have conflicting interests regarding SVM prices [5–7]. According to Amazon policy, if the user's bid is higher than the price of the SVM, the requested SVM will be given to the user. Otherwise, the user will have to wait for the next bidding period [8–10]. However, deciding on the right bidding price for the user is a complicated task. If the user's bid is low, the probability of interruption in the running of the user's tasks increases, resulting in longer execution times and higher costs. On the contrary, if the user's bid is high, the provider may raise

*Correspondence: rezvani@qiau.ac.ir

the price, which eventually will increase the user's costs [1]. On the other hand, inherent uncertainties in the spot market may increase complexity. The main reasons for price uncertainty are frequent price fluctuations, unavailability of SVMs, and other users' pricing patterns.

So far, various methods for modeling uncertainty have been introduced in the literature. These include probabilistic methods, information gap decision theory (IGDT), robust optimization, stochastic programming, and time-based analysis [11]. In particular, IGDT is a powerful way to describe uncertainty. The advantage of IGDT over probabilistic methods such as Monte Carlo is that it does not require specific information about uncertain parameters. Some of this uncertain information is the probability density function, the fuzzy logic membership function, and the exact definition of scenarios. Moreover, compared to other methods, IGDT is computationally lightweight because it does not require any assumptions about the nature of the uncertainty. Conversely, in stochastic programming, several possible scenarios should be considered, which makes the optimization problem very complicated on a large scale. Additionally, these approaches suffer from a high computational burden. The IGDT does not require a probability density function for making robust decisions in severe uncertainties. Moreover, unlike robust optimization methods, the IGDT does not need to know the variation interval regarding uncertain parameters [11, 12]. The only necessary assumption is the predicted value of the uncertain input parameter because it only focuses on the difference between the actual value of the uncertain parameter and the predicted value. The IGDT seeks to determine the maximum allowable uncertainty bound for the uncertain parameter so that the objective function remains within the permissible range. It proposes a confidence interval to the decision-maker by considering a distance around the predicted value of the uncertainty parameter [12].

IGDT has recently been promoted to cope with severe uncertainties in power systems [13–17]. However, although the IGDT has been applied to uncertain microeconomic and operational parameters in energy systems [14], investigating its application on the SVM has remained untouched. To the best of our knowledge, this is the first research in which the uncertainty of influenced parameters on bid price (e.g., price of SVMs) is modeled with the IGDT method. In summary, the main contributions of this research are as follows:

1) We consider uncertainty for the price of SVMs in the proposed bidding problem.

2) We use the IGDT method for modeling the price uncertainty of SVMs.

3) Risk-averse and risk-seeker decision-making strategies are analyzed for the proposed bidding problem.

The remainder of this paper is organized as follows: Section 2 reviews the literature; Section 3 describes the problem formulation; Section 4 elaborates on the proposed method and describes the proposed IGDT-based algorithm for optimal bidding strategy; Section 5 evaluates the numerical results of applying the proposed model to the Amazon spot instance prices; and finally, Section 6 concludes the study and provides suggestions for future research.

2. Related work

Previous studies have used both deterministic and nondeterministic models to design bidding strategies for SVM. Deterministic models assume that the realization of model parameters in the future may be the same as their prediction. Therefore, predicting the price of SVM before submitting a bid is critical for the bidder. So far, many techniques for analyzing and predicting the price of SVM have been introduced, most of which are based on a combination of statistical and machine learning approaches. For example, in [18], the authors first analyzed the actual price distribution based on spot price history using a k-AMSE parameter. Then, they presented a prediction model based on the Gated Recurrent Unit (GRU) network. Their results show that the proposed method is more accurate than others. In [19], a utility-based strategy is presented in favor of user decision-making for the short-term trade-offs between the spot price and availability. The results have shown that this approach can provide efficient choices of VM instances with low bids and high availability. Khandelwal et al. [2] used a random regression forest model to predict the price of SVMs. They compared the proposed method with several machine learning algorithms such as support vector machine (SVM), neural network (NN), decision tree, and random forests. Moreover, a similar study was conducted by Al-Theiabat et al. [20], using the deep learning method and the use of long-term short-term memory (LSTM). They found that this method has less error than other machine learning approaches such as autoregressive integrated moving average (ARIMA). Liu et al. [3] designed a hidden Markov model (HMM) to predict spot prices. Their results showed that the proposed model could predict the spot price more accurately than regression-based forecasting methods. Moreover, Naghdehforoushha et al. [21] used Markov decision process (MDP) to model bidding and scheduling problems jointly. The proposed model works at two time scales on two levels. At the top level, it selects the most appropriate user bids and adjusts the spot price to minimize the cost of SVMs on the cloud provider side. At a low level, it decides to admit tasks to maximize user-side satisfaction. Their results showed that the proposed method minimizes cloud providers' costs and maximizes user gain more effectively than heuristic methods. The major disadvantage of the above research is not handling the uncertainties in the price of SVMs. This can increase the deviation of the predicted prices from their actual values in the next time slots [4, 22].

In contrast, nondeterministic models assume that the realization of model parameters in the future may be different from their predictions. This research uses various techniques to model uncertainty, such as probability distribution, fuzzy distribution/sets, and bounded intervals/sets. Their major difference lies in the description concerning the severity of uncertainty. Table 1 shows a brief comparison of uncertainty modeling techniques and their advantages and disadvantages. Interested readers can refer to [11, 23, 28] for further study.

Zhao et al. [30] formulated two deterministic and stochastic models for the problem of resource rental planning in the spot market. They showed that the stochastic optimization approach performs better than the prediction approach in terms of reducing the rental cost. However, it provides a heavyweight solution due to the use of a scenario tree. Zheng et al. [5] proposed an optimal bidding strategy for cloud users, which depends on the probability distribution function of prices. Mireslami et al. [24] proposed an algorithm for deploying a web application with two phases: resource reservation and dynamic bidding. During the reservation step, resources are reserved according to the expected service-level agreement (SLA). Then in the dynamic bidding step, the user demand is modeled as a random variable. In another study, Ivashko et al. [22] developed a model to find the optimal bid using a threshold-based strategy. Although their method minimizes the cost of renting SVMs, it requires prior knowledge of price probability distribution, an assumption that does not exist in the real world. Xie and Lui [25] used Q-learning techniques to deduce favorable prices from historical data. They first designed a dynamic discrete-time pricing scheme and formulated an MDP to describe price-dependent demands. Their results showed that the proposed dynamic model could lead to an increase in revenue of up to 20% compared to the static pricing approach.

Unlike previous research, our proposed IGDT-based approach is a nonprobabilistic nonfuzzy method with two significant advantages: Firstly, by solving the deterministic method, the minimum execution cost of each user is obtained based on the price history. In this way, the user can find a confidence interval for the price of SVMs close to the deterministic desired cost. Our model depends on the amount of money the user wants to spend to get the SVMs within the specified deadline. Secondly, it finds the minimum and maximum price (price

Table 1. A summary of uncertainty modeling attributes [28].

Group	**Main idea**	**Advantage**	**Disadvantage**
Stochastic optimization	Applying probability density functions such as scene method, Monte Carlo simulation, point estimation method, and chance constrained programming	Accurate, simulation of the real world, suitable for big and complex problems	It is time-consuming and complex. Moreover, its execution time depends on the number of uncertain variables.
Robust optimization	Using uncertain sets such as engineering game model two-stage robust optimization model and distributed robust optimization model	It is useful when just one interval exists.	Conservative decision, difficult to use in nonlinear problems
Interval optimization	Using intervals	It is useful when just one interval exists.	It cannot model the correlation between intervals.
Possibilistic method	Using fuzzy membership function	Ability to obtain the membership function of output variable	Time-consuming, cannot model correlation
Hybrid optimization	Modeling multiple uncertainties	Ability to model the real-world conditions and different uncertainties	Time-consuming
IGDT	Using forecasted values	Useful for decision making in severe uncertainties	Complexity

interval for SVMs) in both strategies based on IGDT. In this way, the user can perform the task cost-effectively. Moreover, the spot market is regulated. The user does not have to bid a high value to win. It is enough for the user to bid in the range where prices fluctuate. In other words, even if users bid high for SVMs over time, the provider can still set a reasonable price. Our evaluations show that the IGDT is a powerful mathematical tool for assessing risks and opportunities for uncertain prices.

3. Problem formulation

In this section, we present the formulation of the objective function. This function concerns the cost of processing the user tasks on obtained SVMs per time slot. Then, we will model the uncertainty of the price of SVMs using the IGDT method. Accordingly, we define robustness and functions based on the IGDT.

3.1. Assumptions

We consider the following assumptions in this study according to [6, 21]:

1) All SVMs are homogeneous and of the same type.
2) The user predicts the price of SVMs an hour ahead.
3) Each user considers only the processing costs and neglects the storage and data transfer costs.
4) Each user uses a checkpointing technique to store calculations before an out-of-bid event.
5) Every user acts as a price-taker agent. It means that the user's bid alone cannot influence the price of SVMs.

3.2. Variables

The decision variable is the bid price for the round $t+1$, which is denoted by b_{t+1}. The user must calculate this value for the next hour, considering the price of SVMs in the current hour. The set of bids submitted by the user is denoted by $B = [b_1, ..., b_T]$, in which T is the last round to bid. Table 2 shows the mathematical notations used in this paper.

Table 2. Mathematical notations.

Functions	**Explanation**
DCF	The objective function of the deterministic model
RCF	The objective function of the robust model
OCF	The objective function of the opportunity model
Acronyms	**Explanation**
DCF	Deterministic cost function
RCF	Robust cost function
OCF	opportunity cost function
Symbols	**Explanation**
h	Index of hours (1 *to* T)
i	Index of SVMs (1 *to* N_{SVM})
$prg(i,h)$	Execution progress time for $task_i$ on SVM_i at the period h (in hours)
$t^{deadline}(i)$	Deadline for completion of $task_i$ (in hours)
$t^{execution}(i)$	Run time of $task_i$ (in hours)
$t_{checkpointing}$	The time needed for check-pointing (in minutes)
t_{resume}	The time needed for resuming a work (in minutes)
N_{SVM}	Number of SVMs
T	Number of time slots
β	Budget of uncertainty
Ψ	Robust region of the uncertainty sources
$SVM = \{1, ..., N_{SVM}\}$	Set of all requested SVMs by the user
$H = \{0, ..., T\}$	Set of daily hours (time slots)
$MH(i,h)$	Runtime spent on each SVM (hour)
$p^{spot}(h)$	Price of SVMs for each time slot (\$/hour)
$\hat{p}^{spot}(h+1)$	The estimated price of SVMs for next time slot (\$/hour)
$b(h)$	The bid price of the user for the SVMs at time slot h (\$/hour)
$\hat{b}(h+1)$	The estimated bid price of the cloud user for the next time slot (\$/hour)
Δb	The amount of bid price changes
$\delta(h)$	Binary variable indicating whether requested SVMs are allocated to the user at the time slot h or not
$\hat{\delta}(h+1)$	Estimated binary variable indicating whether requested SVMs are allocated to the user at time slot $h+1$ or not
α	The envelope of the robust region of the uncertainty $p^{spot}(h)$
$\tilde{p}^{spot}(h)$	The predicted price of SVMs for time slot h (\$)
$p^{on-demand}$	Price of on-demand VMs (\$)

3.3. The deterministic model for the problem

From a user perspective, the optimization problem concerning the cost of processing tasks on SVMs during a T-hour period can be written as follows:

$$DCF = \min \sum_{i=1}^{N_{SVM}} \sum_{h=1}^{T} \delta(h) \times prg_i(h) \times f(p^{spot}(h)), \tag{1}$$

where $\delta(h)$ is a binary variable. It determines if the SVM is allocated to the user at time h. It works according to the current price and the bid price provided by the user. Moreover, $prg_i(h)$ denotes the progress of the $i-th$ task on the corresponding SVM at time h, and $f(p^{spot}(h))$ denotes the function of the price of SVMs at time h. Later in Section 5.1, we estimate the function $f(.)$ through the curve fitting technique for hourly prices concerning selected SVMs. The processing cost is the amount of money the user pays to the provider conditional on the successful bidding. Eq. (1) states that the user aims to minimize the sum of the costs associated with the progress of the tasks at the specified prices. It is worth noting that as the deadline of a task approaches, the user is willing to pay more. In this regard, an illustrative example is shown in Figure 1. As is evident in this figure, when the percentage of task completion changes from, for example, 18% to 28%, the amount of change in the processing cost of the user rises from 1% to 2%. This is the amount that the user pays to the provider.

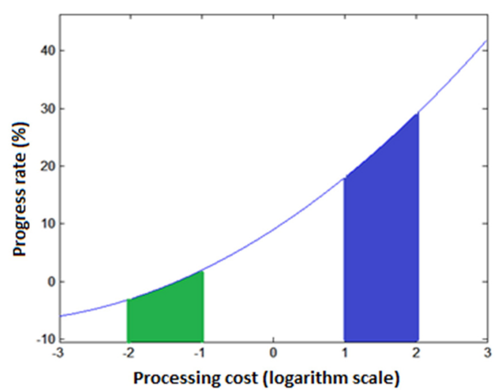


Figure 1. An illustrative example of a dramatic change in user costs by increasing the progress ratio of tasks.

According to the Amazon auction rule, if the bid price at the round h is higher than the price of SVMs at that round, the requested SVMs are allocated to the user. Otherwise, no SVM is allocated to the user. Accordingly, a binary variable, $\delta(h)$, indicates whether or not the requested SVMs are assigned to the user in time interval h. This has been shown in Eq. (2), mathematically.

$$\delta(h) = \begin{cases} 1 & \text{if } (b(h) \geq p^{spot}(h)) \\ 0 & \text{if } otherwise \end{cases} \tag{2}$$

The allocation decision variables in the previous, current, and next (estimated) rounds are denoted by $\delta(h-1)$, $\delta(h)$, and $\hat{\delta}(h+1)$, respectively. Depending on what these values are, the rate of progress of the $i-th$ task on the corresponding SVM at each hour is calculated as follows:

$$prg_i(h) = \begin{cases} 1 - t_{checkpointing} & \text{if } (\delta(h-1) = 1, \delta(h) = 1, \hat{\delta}(h+1) = 0) \\ 1 - t_{resume} & \text{if } (\delta(h-1) = 0, \delta(h) = 1, \hat{\delta}(h+1) = 1) \\ 1 - t_{checkpointing} - t_{resume} & \text{if } (\delta(h-1) = 0, \delta(h) = 1, \hat{\delta}(h+1) = 0) \\ 1 & \text{if } (\delta(h-1) = 1, \delta(h) = 1, \hat{\delta}(h+1) = 1) \\ 0 & \text{if } otherwise \end{cases} \tag{3}$$

As stated before, we can use a fault-tolerance mechanism such as checkpointing to store the system state when an out-of-bid failure occurs. For the estimation of $\hat{\delta}(h+1)$ in Eq. (3), the user must at first estimate the price of SVM ($\hat{p}^{spot}(h+1)$) for the next hour. Then, the user has to estimate the amount of bid price ($\hat{b}(h+1)$). The value of $\hat{b}(h+1)$ at the next time slot is calculated as follows:

$$\hat{b}(h+1) = b(h) \times (1 + \Delta b), \tag{4}$$

where Δb is obtained as follows:

$$\Delta b = \frac{p^{spot}(h) - p^{spot}(h-1)}{p^{spot}(h-1)} \qquad \forall h \in H \tag{5}$$

Here, Δb can be estimated from price fluctuations in two consecutive time slots. In this regard, the total progress ratio concerning each task on the dedicated SVM during all hours should not exceed a user-specified threshold. Thus:

$$\sum_{h=1}^{T} prg_i(h) \leq t^{deadline}(i) \qquad \forall i \in SVM, \forall h \in H \tag{6}$$

Moreover, it should be ensured that the deadline for executing tasks is not violated. In this regard, the sum of hours spent executing tasks on SVMs shall not exceed the user-specified deadline. Thus:

$$\sum_{h=1}^{T} MH(i,h) \leq t^{deadline}(i) \qquad \forall i \in SVM \tag{7}$$

It should also be noted that the price of SVMs in each hour should not exceed the price of on-demand VMs. If so, the user would prefer to use on-demand VMs with high reliability instead of unreliable SVMs. Therefore, the following constraint can be defined accordingly:

$$p^{spot}(h) < p^{on-demand} \qquad \forall h \in H \tag{8}$$

4. Proposed method

In this section, the IGDT method is used to inspect the price uncertainty of SVMs. Here, we map the elements of the IGDT model to the SVM bidding issue.

4.1. The IGDT background

Decision theory is used as a dynamic process to obtain the optimal set of decision variables. The idea of the IGDT was first introduced by Ben-Haim [12]. Due to a small number of assumptions, the IGDT is a powerful mathematical tool for decision-making in environments that suffer from uncertainty. It is an interval-based optimization approach that looks to optimize either robustness against incurring unexpected loss or the opportunity of gaining unexpected profit. Therefore, based on the risk management strategy, the IGDT can be formulated as either a risk-averse or risk-seeker framework. The risk-averse/seeker customer implements a robust/opportunistic function through which the maximum/minimum unfavorable/favorable uncertainty horizons are determined. Suppose the deviation of each uncertain parameter from its forecast value falls into its corresponding robust/opportunistic uncertainty horizon. In that case, the cost expectations of the risk-averse/risk-seeker user are guaranteed. Details of the IGDT method can be found in [12]. Generally, Each IGDT decision problem is characterized by three components: the system model, the uncertainty model, and the performance requirements, which will be described in this section.

A) The system model: Assume that λ and d denote the uncertainty parameters and the set of decision variables, respectively. The system response to decisions d and uncertain parameters λ is evaluated with an objective function $f(\lambda, d)$. This function describes the relationship between the final deciding factor, the decision variable(s), and the uncertain parameter(s). This paper considers the price of SVMs as an uncertain parameter. Moreover, the decision variable is the value of user bids. The system model is the processing cost function concerning the user tasks on SVMs.

B) The uncertainty model: The uncertainty model describes the distance between the known values (predicted data) and unknown values (real data) as a function of deterministic parameters. In this study, the price of SVMs is an unknown variable that directly impacts expected cost and the bidding strategy. There are different models to illustrate the uncertainty of parameters, which must be selected carefully depending on the type of uncertainty [26]. These are the energy-bound model, slope-bound model, and envelope-bound model [12]. In the energy-bound model, the uncertainty sets refer to phenomena that transiently deviate from their nominal value. In the slope-bound model, uncertainty sets refer to phenomena in which an uncertainty radius constrains the deviation rate from its nominal value. In most studies, the envelope-bound model is used, which involves the value prediction for nondeterministic variables. This model is adopted to represent the uncertainty of market prices. The related robustness and opportunity functions will be derived based on this model. The uncertainty model is represented as follows:

$$\psi(\lambda, \alpha) = \left\{ \lambda(t) : \left| \frac{\lambda(t) - \tilde{\lambda}(t)}{\tilde{\lambda}(t)} \right| \leq \alpha \right\}, \quad (9)$$

where $\psi(\lambda, \alpha)$ denotes the uncertainty function, λ denotes the uncertainty parameter (price of SVMs), and α denotes the length of the confidence interval (radius). Moreover, $\tilde{\lambda}(t)$ denotes the predicted value of the uncertain price.

Generally, the bigger the predicted value, the bigger will be the confidence interval. Figure 2 simply shows the concept of uncertainty modeling in the IGDT. **C) The performance requirements:** System expectations

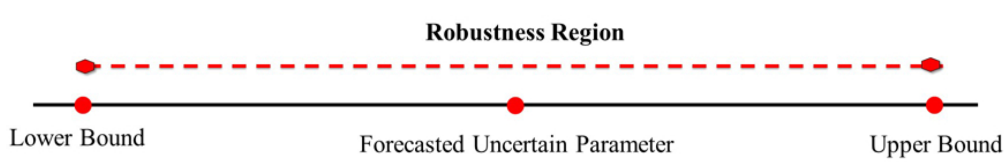


Figure 2. The concept of uncertainty modeling in the IGDT [12].

are described in terms of cost or other functions and can be evaluated in terms of robustness and opportunity. According to a risk-averse policy, the robustness function tries to find the maximum confidence interval so that, in the worst-case scenario, the objective function f does not exceed a critical value. The critical value specifies the maximum value allowed to increase the objective function f relative to its base value. The decision-maker often determines it. In this case, the best decision can be made robustly and conservatively by immunizing the objective function against the maximum uncertainty radius. Let us denote the predefined critical value for the objective function f by r_c. We can now define the robustness function as a mathematical optimization problem as follows:

$$\hat{\alpha}(r_c) = \max_{\alpha} \left\{ \alpha : \max_{d}(f(d,u)) \leq r_c \right\} \tag{10}$$

In Eq. (10), a large value of $\hat{\alpha}(r_c)$ represents a more resistant and more risk-averse decision. In contrast, a small value of $\hat{\alpha}(r_c)$ represents a weak and unstable decision. Here, our interest is to obtain the maximum value for $\hat{\alpha}(r_c)$. In contrast to the robustness function, the opportunity function under a risk-seeker policy tries to find a minimum confidence interval in such a way that, in the best case, the value of the objective function f does not exceed its critical value. A minimization problem can represent the opportunity function according to Eq. (11). Similarly, let us denote the predefined critical value for the objective function f by r_w, so that $r_w < r_c$. To show the immunization against the lost benefits, our interest is to obtain the minimum value for $\hat{\alpha}(r_w)$.

$$\hat{\beta}(r_w) = \min_{\alpha} \left\{ \alpha : \min_{d}(f(d,u)) \leq r_w \right\} \tag{11}$$

Given the above explanations, it can be concluded that the decision-maker can make the best decisions under conservative/opportunistic uncertainty using the IGDT. These goals are achieved through maximizing the robustness function in Eq. (10) and minimizing the opportunity function in Eq. (11) under risk-averse/risk-seeker policies.

4.2. Problem customization using the IGDT

In Section 3.3, we outlined the deterministic model of the problem where the uncertainty parameter is ignored. In this model, the optimal solution to that problem may be feasible only for the estimated value of the price of SVMs per hour $\tilde{p}^{spot}(h)$. In contrast, the IGDT method takes into account the uncertainty parameters which were not considered by the deterministic model at all. Hence, the IGDT aims to extend the robust regions of uncertain parameters considering a specific threshold for the non-deterministic objective function.

In Section, we present the uncertainty model and the robustness and opportunity functions for the minimization cost problem for the user-side. First, let us formulate the uncertainty model for the problem. We assume that the price of SVMs at each hour is unknown. It should be noted that the price of SVMs directly impacts the value of the bid offered by the user. Using the following fractional error, we model the price uncertainty of SVMs by considering the price prediction error as the input random variable:

$$\psi(\tilde{p}^{spot}(h), \alpha) = \left\{ \tilde{p}^{spot}(h) : \left| \frac{p^{spot}(h) - \tilde{p}^{spot}(h)}{\tilde{p}^{spot}(h)} \right| \leq \alpha \right\} \tag{12}$$

As stated earlier, the size of the gap between the predicted prices, $\tilde{p}^{spot}(h)$, and the actual unknown prices,

$p^{spot}(h)$, is modeled by the uncertainty parameter α so that the deviation of the price's fractional error from its predicted value should always be smaller than α.

We now proceed to formulate the robustness function for the problem. According to the IGDT concept, the maximum value of α is obtained by solving the robust problem in Eq. (13). The robustness function $\hat{\alpha}(r_c)$ determines the degree of robustness for processing costs against uncertainty. It is used to determine the amount of immunity against high prices. It maximizes the envelope of the robust region using Eq. (14). Here, the DCF denotes the deterministic cost function attained by Eq. (1), while the RCF denotes the robustness cost function in Eq. (15). The robust region of $\tilde{p}^{spot}(h)$ can be summarized by Eq. (16). Therefore, Eq. (2) and constraint in Eq. (8) can be rewritten as Eqs. (17)–(18) to obtain the worst case. The optimal value of the robustness function is obtained by solving the following optimization problem:

$$\hat{\alpha}(r_c) = \max \alpha_1 \tag{13}$$

s.t:

$$RCF \leq DCF \times (1 + \beta) \tag{14}$$

$$RCF = \sum_{i=1}^{N_{SVM}} \sum_{h=1}^{T} \delta(h) \times prg_i(h) \times f(\tilde{p}^{spot}(h)) \times (1 - \alpha_1) \tag{15}$$

$$\psi(\tilde{p}^{spot}(h), \alpha_1) = \left\{ \tilde{p}^{spot}(h) : \left| \frac{p^{spot}(h) - \tilde{p}^{spot}(h)}{\tilde{p}^{spot}(h)} \right| \leq \alpha_1 \right\} \tag{16}$$

$$\delta(h) = \begin{cases} 1 & \text{if } b(h) \geq \tilde{p}^{spot}(h) \times (1 - \alpha_1) \\ 0 & \text{if } otherwise \end{cases} \tag{17}$$

$$\tilde{p}^{spot}(h) \times (1 - \alpha_1) < p^{on-demand} \qquad \forall h \in H \tag{18}$$

$$The\ inherent\ limitations\ of\ the\ problem,\ i.e.\ Eqs.\ (6) - (8) \tag{19}$$

Obviously, the larger the value $\hat{\alpha}(r_c)$ is, the more desirable it is.

In the next step, we proceed to define the optimization problem regarding the opportunity function. In this paper, the opportunity function evaluates the potential benefits of experiencing decreased costs due to low prices. Our purpose is to find the best way of price fluctuation that could lead to a lower cost. The opportunity function $\hat{\alpha}(r_w)$ determines the opportunity to reduce costs. It provides a low price of SVMs to the user. Therefore, it is considered a measure to determine the windfall benefit. The optimal value of the opportunity function, $\hat{\alpha}(r_w)$, is obtained by solving the following optimization problem:

$$\hat{\alpha}(r_w) = \min \alpha_2 \tag{20}$$

s.t:

$$OCF \leq DCF \times (1 - \beta) \tag{21}$$

$$RCF = \sum_{i=1}^{N_{SVM}} \sum_{h=1}^{T} \delta(h) \times prg_i(h) \times f(\tilde{p}^{spot}(h)) \times (1 + \alpha_2) \tag{22}$$

$$\psi(\tilde{p}^{spot}(h), \alpha_2) = \left\{ \tilde{p}^{spot}(h) : \left| \frac{p^{spot}(h) - \tilde{p}^{spot}(h)}{\tilde{p}^{spot}(h)} \right| \leq \alpha_2 \right\} \tag{23}$$

$$\delta(h) = \begin{cases} 1 & \text{if}\, b(h) \geq \tilde{p}^{spot}(h) \times (1 + \alpha_2) \\ 0 & \text{if}\, otherwise \end{cases} \tag{24}$$

$$\tilde{p}^{spot}(h) \times (1 + \alpha_2) < p^{on-demand} \qquad \forall h \in H \tag{25}$$

$$The\ inherent\ limitations\ of\ the\ problem,\ i.e.,\ Eqs.\ (6) - (8) \tag{26}$$

Obviously, the smaller the value of $\hat{\alpha}(r_w)$ is, the more desirable it is.

The opportunity and robustness functions are mixed-integer nonlinear programming problems that can be solved using the BARON solver under GAMS [27].

4.3. The proposed algorithm for optimal bidding strategy

The user sends hourly bidding to the cloud provider to rent the needed SVMs. The above robustness and opportunity functions of the IGDT ensure the user makes appropriate decisions concerning the bid intervals. Figure 3 briefly shows the proposed algorithm's flowchart. Moreover, in the following, we explain the procedure of determining the bid price in each time slot:

1) Initially, the user finds the minimum cost function using Eqs. (1)–(8). The resultant cost value is, in fact, the expected minimum cost if prices of SVMs are equal to the predicted values per hour.

2) Next, the user finds the optimal value of the robustness function using Eqs. (13)–(19). As stated before, the value of the robustness function is less than the expected minimum cost obtained in step (1).

3) Similarly, the user finds the optimal value of the opportunity function using Eqs. (20)–(26).

4) For all robustness/opportunity cost function levels, the corresponding actual price values for each iteration s are obtained.

5) Considering the maximum/minimum values of the uncertainty radius, the user can decide on their optimum bid price.

5. Experimental evaluation

In this section, we perform a series of computational experiments to evaluate the proposed IGDT-based model described in Section 4. The experiments were performed on a computer running a 64-bit Microsoft Windows OS with a 2.2 GHz Intel 7-core processor. Two sets of experiments are applied to evaluate the proposed method's performance. The first set of experiments is related to a risk-neutral bidding strategy. It is the deterministic model without applying the proposed IGDT to it. In contrast, the second set of experiments evaluates the proposed risk-averse/risk-seeker strategy using the proposed IGDT method.

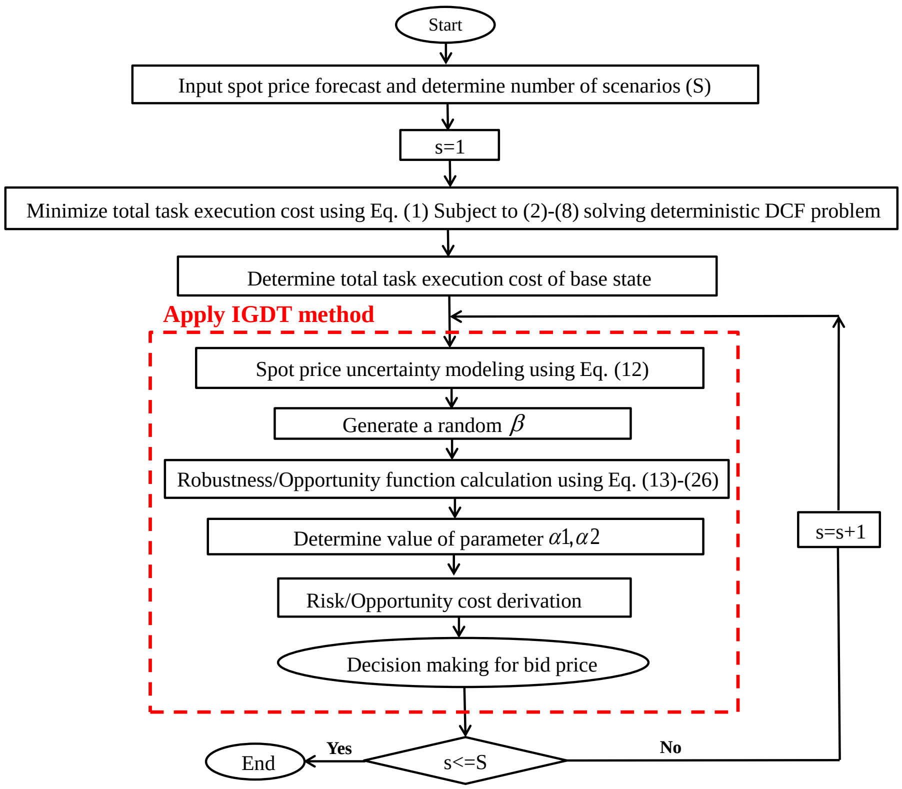


Figure 3. Flowchart of the proposed IGDT-based approach.

5.1. Input data

Each user participates in a next-hour market with a time horizon of 24 h. We evaluated our algorithm using one of the most popular public cloud providers, namely the Amazon EC2 [1]. Spot prices for 90 days are contained in a CSV file. This CSV file is the input dataset for the program code. We create a CSV file, a conversion from a JSON file, by running the AWS CLI command on the console. Retrieving real-time price history from AWS includes the following steps:

1) Provide all the necessary details to create an account on AWS.

2) Go to IAM (Identity and Access Management Console), once the AWS account is activated.

3) Go to the "User" tab on the left side of the IAM Console to build a new account.

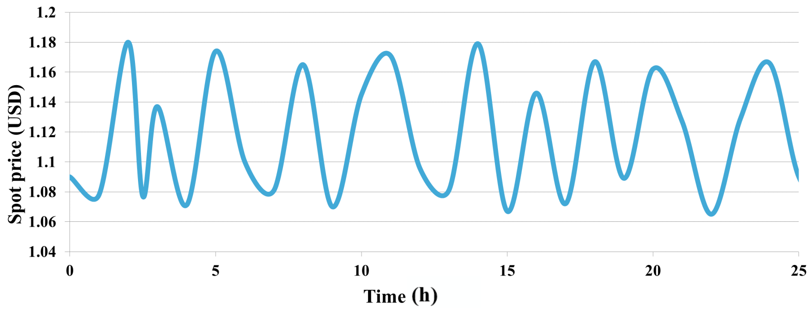
4) Access issues are often faced when the user is not made the admin. Attach a policy named "Administrator Access" to the user.

5) Upon effective account development, a different access ID and a hidden key are given per AWS user.

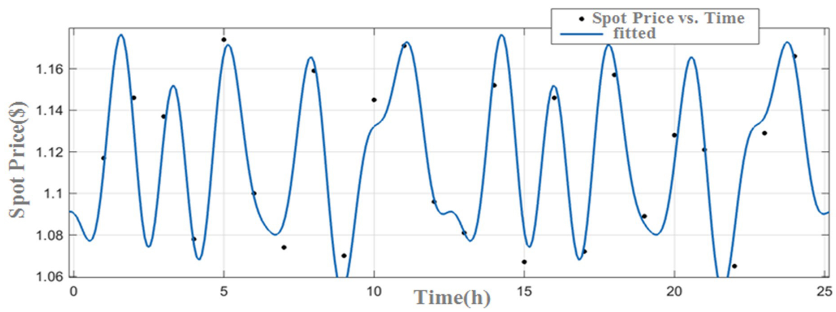
The installation of AWS CLI is the next step for fetching the history of spot prices. To fetch the data for the history of spot price into a JSON file format, the following command is run on the CLI [29]:

AWS ec2 describe-spot-price-history –instance-types m1.4xlarge –start-time 2016-02-17 T04:04:09 –end-time 2016-02-18 T04:04:09.

Figure 4a shows the pattern of price changes for the m1.4xlarge instance concerning the "us-east-1f" geographical zone for February 17, 2016. We obtained the price data using the instructions above. Interested readers can refer to [1] for more details.



The pattern of prices



Curve fitting for prices pattern

Figure 4. a) The pattern of prices, b) curve fitting concerning "m1.4xlarge" instance.

Next, using the MATLAB 2021a software, we proceed to find the best-fit curve concerning the price pattern of Figure 4a to predict future spot prices. The fitted curve of Eq. (27) is shown in Figure 4b. The details of the calculations are as follows:

$$\begin{aligned} f(x) = {} & a_0 + a_1 \times cos(x \times w) + b_1 \times sin(x \times w) + a_2 \times cos(2 \times x \times w) + b_2 \times sin(2 \times x \times w) \\ & + a_3 \times cos(3 \times x \times w) + b_3 \times sin(3 \times x \times w) + a_4 \times cos(4 \times x \times w) + b_4 \times sin(4 \times x \times w) \\ & + a_5 \times cos(5 \times x \times w) + b_5 \times sin(5 \times x \times w) + a_6 \times cos(6 \times x \times w) + b_6 \times sin(6 \times x \times w) \\ & + a_7 \times cos(7 \times x \times w) + b_7 \times sin(7 \times x \times w) + a_8 \times cos(8 \times x \times w) + b_8 \times sin(8 \times x \times w) \\ & a_0 = 1.117, a_1 = 0.002, b_1 = -0.0003, a_2 = -0.002, b_2 = -0.008, a_3 = -0.013, b_3 = -0.001, \\ & a_4 = -0.028, b_4 = -0.0007, a_5 = 0.006, b_5 = -0.012, a_6 = -0.015, b_6 = -0.016, a_7 = 0.006, \\ & b_7 = -0.006, a_8 = 0.018, b_8 = 0.009, w = 0.495 \end{aligned} \tag{27}$$

5.2. Risk-neutral results without IGDT (the results of the deterministic model)

In the deterministic approach, the proposed objective function is defined using Eq. (1) and constraints Eqs. (2)–(8). The minimum total processing cost for a user is equal to \$63.205, and its execution time is 0.492 s. To clarify the matter, we illustrate an exemplary execution scenario in Figure 5. Suppose a user wants to run a task with two VM instances in parallel. If this task requires 14 h, the workload in each SVM instance is 7 h. The total time available for execution is 9 h, from which 7 h are effective considering that the time of checkpointing is 0.5 h and the time of resume is 0.5 h. The spot price fluctuates during execution, and the bid price is \$0.4.

Hence, it goes out-of-bid for time intervals [5, 6, 9, 10, 11]. Altogether, the components of the total cost consist of a) 3 h of running at \$0.1, b) 4 h of running at \$0.2, and c) 2 h of running at \$0.3. Thus, the total cost is \$1.7. It is concluded that with two VM instances, the effective cost per instance is \$1.7/2 = 0.85. The average effective execution is 7/9 = 0.78 per hour. Subsequently, the allocation of SVMs to the user and the progress

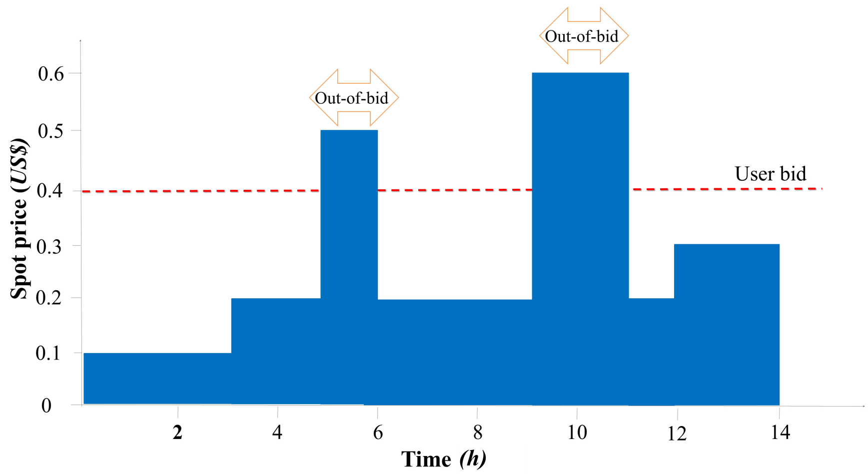


Figure 5. An exemplary execution scenario [17].

of tasks is examined. We have already stated that changes to the binary variable $\delta(h)$ at any time interval h show the allocation vector concerning the SVMs over 24 h. This vector is shown in Figure 6. Using Eq. (2), if the current bid is higher than the price of SVMs, the user will be able to use cheaper SVMs to perform user tasks.

As is evident in Figure 6, the bids offered by the consumer in periods 1, 4, 6, 7, 9, 12, 15, 17, 19, and 22 are higher than the spot prices. It means that during these periods, the user is allowed to perform tasks on the requested SVMs. The user will not be allowed to use the SVMs at other times due to the low bidding prices. Therefore, at those hours, the value of the variable $\delta(h)$ becomes zero. Figure 7 shows the progress of the user

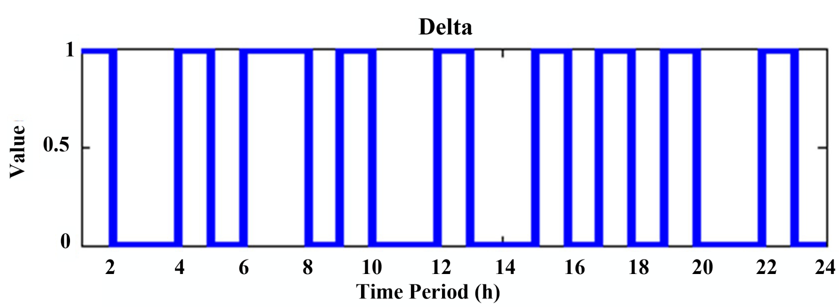


Figure 6. The value of the variable $\delta(h)$ for 24 h.

tasks on different SVMs. As stated earlier in Eq. (3), in each hour, if the deadline is not violated and the SVMs are assigned to the user, then the progress ratio of the tasks can be calculated. We assume that $t_{checkpointing}$ and t_{resume} is equal to 0.083 and 0.16, respectively. As previously stated in Eq. (3), if the SVMs have been allocated in the previous period, the rate of execution progress for the task, $prg_i(h)$, will vary at the current and next time slots. Based on this, five different cases for $prg_i(h)$ can be distinguished on each SVM at each

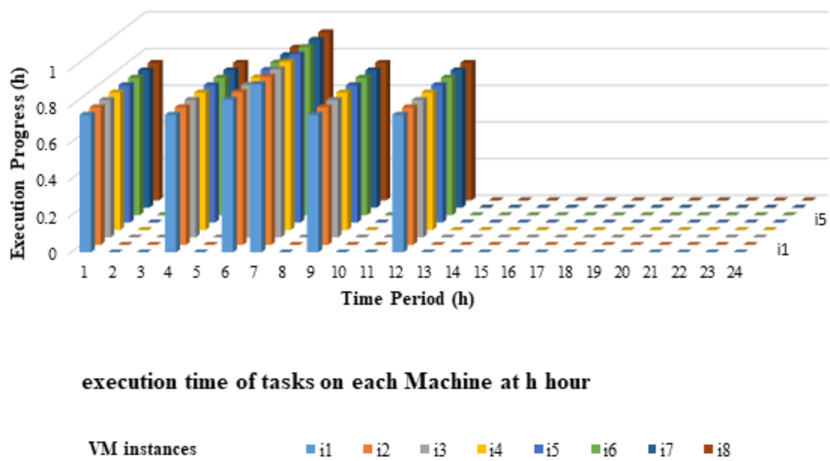


Figure 7. Execution progress ratio of task on SVMs for 24 h.

time slot:

1) This situation occurs at time slot 6 in Figure 7. According to the first condition of Eq. (3), the SVMs have been allocated to the user in the previous and current hours. It means that during these hours, the bids offered by the user were higher than the price of SVMs (according to Eq. (2)). However, this is not the case in the next hour. Therefore, the out-of-bid event will occur in the next hour. In this case, we use the checkpointing operation to prevent the loss of task results concerning the previous hours. To do this, a portion of the execution time, $t_{checkpointing}$, is spent on checkpointing. In this case, the value of the progress ratio is equal to $1 - t_{checkpointing}$.

2) This situation occurs at time slot 7 in Figure 7. According to the second condition of Eq. (3), the SVMs have been allocated to the user in the current and next hours. It means that during these hours, the bids offered by the user were higher than the price of SVMs, but this was not the case in the previous hour. Therefore, the out-of-bid event occurred during the last hour. In this case, we use the recovery operation to resume the task. To do this, a portion of the execution time, t_{resume}, is spent resuming. In this case, the value of the progress ratio is equal to $1 - t_{resume}$.

3) This situation occurs at time slots 1, 4, 9, and 12 in Figure 7. As stated in the third condition of Eq. (3), only in the current hour has the SVMs been allocated to the user. It means that merely during the current hour, the user bid was higher than the price of SVMs, but this was not the case in the previous and future hours. Therefore, the out-of-bid event has occurred in the previous and future hours. Thus, the previous results must be retrieved due to the out-of-bid occurrence in the previous hour. Moreover, the obtained results must be stored in the current hour due to the out-of-bid occurrence in the future hour. To do this, two portions of the execution time, namely $t_{checkpointing}$ and t_{resume}, are spent for checkpointing and resuming operations, respectively. In this case, the value of the progress ratio is equal to $1 - t_{checkpointing} - t_{resume}$.

4) This situation has not occurred in Figure 7 at all. According to the fourth condition of Eq. (3), the SVMs have been allocated to the user for three consecutive hours. It means that during these hours, the user bid was higher than the price of SVMs. Therefore, in this case, the value of the progress ratio is equal to one full hour.

5) When none of the above four situations occur, the value of the progress ratio at that time slot is equal to zero. This situation occurs at time slots 2, 3, 5, 8, 10, 11, and 13–24 in Figure 7.

It is also emphasized that in each of the above cases and at each hour, it must be checked that the deadline for the task is not violated.

5.3. Risk-based results using IGDT (IGDT-based approach)

In this case, to study the efficiency of the IGDT-based method, the value of the parameter β is chosen as a random number in the interval [0,1] according to the uniform distribution. The random values of β over 100 scenarios are shown in Figure 8.

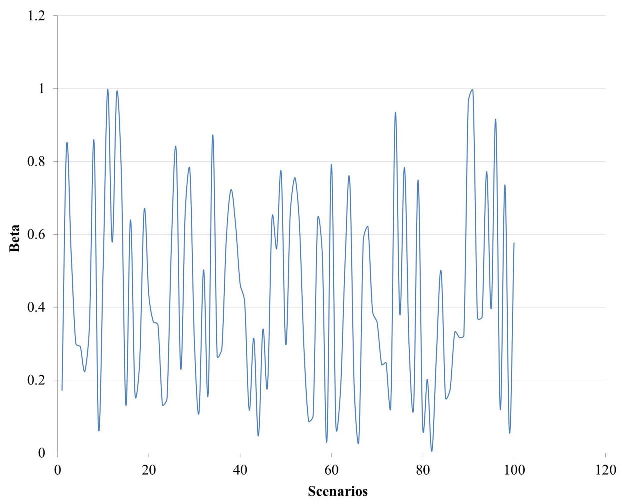


Figure 8. Random values generated for β in each execution scenario.

As explained in Section 4.2, in the IGDT-based approach, two robustness and opportunity strategies are implemented. In the risk-averse strategy, the user seeks less profit and takes less risk. On the contrary, in the risk-seeker strategy, the user seeks more profit with more risk [12]. The variations of the "tolerable uncertainty" versus the "objective cost" in both of these strategies are depicted in Figure 9. Note that in this figure, "Alpha1" and "Alpha2" represent $\hat{\alpha}(r_c)$ Eq. (13) and $\hat{\alpha}(r_w)$ Eq. (20), respectively. As is evident in Figure 5.3, the tolerable uncertainty increases from 1.04 ($\beta = 0.1717$) to 2 ($\beta = 0.5763$). This means that the objective cost increases from \$0.12 ($\beta = 0.1717$) to \$62.88 ($\beta = 0.5763$). As expected, when the objective cost increases, it shows more tolerance toward uncertainty. The algorithm's average calculation time in each scenario of the robustness strategy is equal to 0.2427 s. Moreover, it is observed from Figure 5.3 that the tolerable uncertainty decreases from –2.02 ($\beta = 0.1717$) to –2.42 ($\beta = 0.5763$). Moreover, the objective cost has increased from \$66.15 ($\beta = 0.1717$) to \$126.29 ($\beta = 0.5763$). It means that with increasing the objective cost, less tolerance toward uncertainty is observed. The algorithm's average calculation time in each scenario of the opportunity strategy is equal to 0.2409 s.

Figure 10 compares the "total cost" of task processing for both robustness and opportunity strategies for different scenarios. As is evident in the figure, the total cost of the robustness strategy is less than that of the opportunity strategy. Note that in the opportunity function, according to Eq. (21), the DCF coefficient must be increased so that the OCF value can be increased. Note that the DCF coefficient in Eq. (21) is $1 - \beta$. On the other hand, because the value of β is in the range [0,1], so with the decrease of β value, the value of $1 - \beta$ becomes larger. This, after multiplying by the DCF value in Eq. (21), leads to a larger OCF value. Thus, since the β value is falling, the risk-seeker policy (opportunity strategy) seeks to decrease costs.

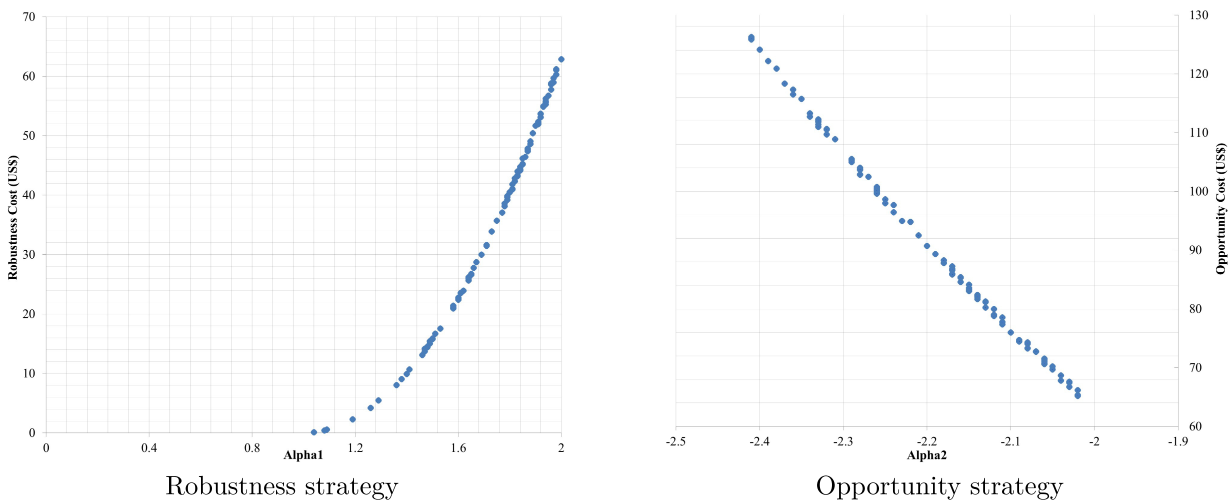


Figure 9. The variations of "tolerable uncertainty" vs. "objective cost" concerning a) robustness strategy and b) opportunity strategy.

Similar to the above explanations, it is obvious that this argument is reversed in the risk-averse policy (robustness strategy).

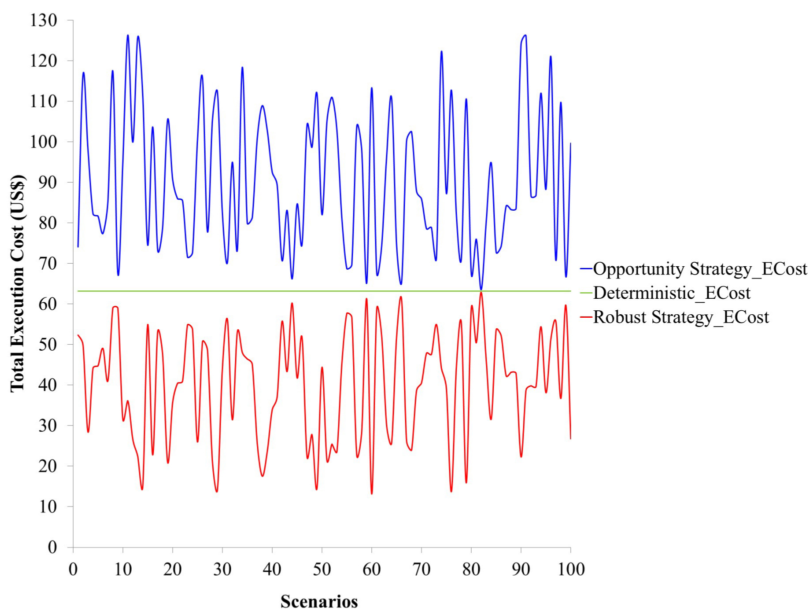


Figure 10. The "total processing cost" concerning the robustness and opportunity strategies as well as the deterministic model for different scenarios.

Moreover, our results on 100 different scenarios show that the minimum β value equals 0.05. This value corresponds to $\alpha 1 = 1.98$ and $\alpha 2 = -2.02$ in both the robustness and the opportunity strategies, respectively. In contrast, the maximum β value is equal to 1. It corresponds to $\alpha 1 = 1.04$ and $\alpha 2 = -2.41$ for both robustness and the opportunity strategies, respectively. Moreover, the execution times for both robustness and opportunity strategies for different scenarios are shown in Figure 11. Except for some points that have severe fluctuations, the execution time for both strategies is almost equal in other scenarios.

Figure 12 shows the user bids under the price uncertainty during 24 h. As seen from the figure, the price of SVMs increases at time slots 1-2, 4-5, 7-8, 9-11, 13-14, 15-16, 17-18, 19-20, and 22-24. Therefore, according to the IGDT-base method, user bids have been raised correspondingly at time slots 5-6, 8-9, 10-12, 14-15, 16-17, 18-19, 20-21, and 23-24. In contrast, the price of SVMs has been shrinking at the time slots 2-4, 5-7, 8-9, 11-13, 14-15, 16-17, 18-19, and 20-22. Accordingly, user bids at time slots 3-5, 6-8, 910, 12-14, 15-16, 17-18, 19-20, and 21-23 have experienced a fall.

Figure 11. The execution times concerning both robustness and opportunity strategies for different scenarios.

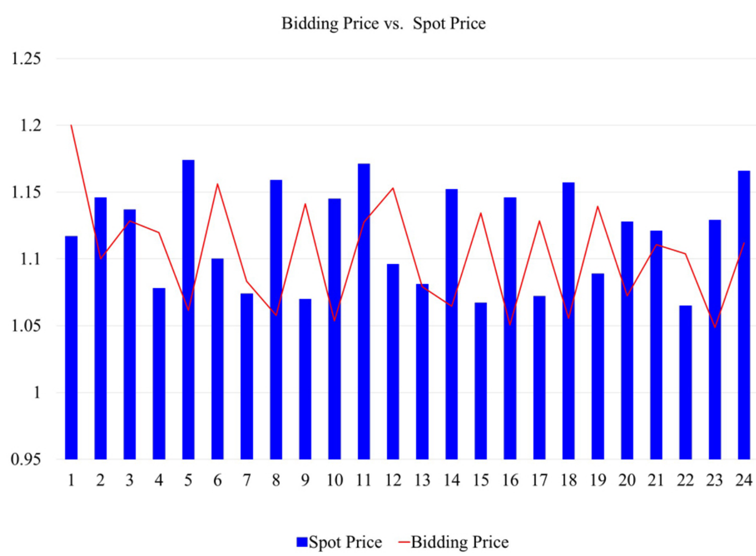


Figure 12. The user bids under price uncertainty for 24 h.

Figure 13a shows the variations in the uncertainty budget interval regarding the robustness strategy. As expected, as the uncertainty budget increases, the robustness interval also increases. According to Eq. (14), when the β value increases, because the β is in the range [0,1], the value of $1+\beta$ is greater than 1. Hence, the value of the expression on the right-side of Eq. (14) also increases. As β increases, a higher bound for costs are considered. It makes sense for the risk-averse user to incur less risk by spending more. Figure 13b plots the variations in uncertainty budget interval for the opportunity strategy. Unlike Figure 13a, here in Figure 13b, the opportunity interval shrinks with an increase in the uncertainty budget. Conversely, according to Eq. (21), when the β value increases, because the β is in the range [0,1], the value of $1-\beta$ is less than 1. Hence, the value of the expression on the right-side of Eq. (21) decreases. As β increases, a lower bound for costs are considered. It makes sense for the risk-seeker user to make the most profit and, at the same time, to spend the least cost.

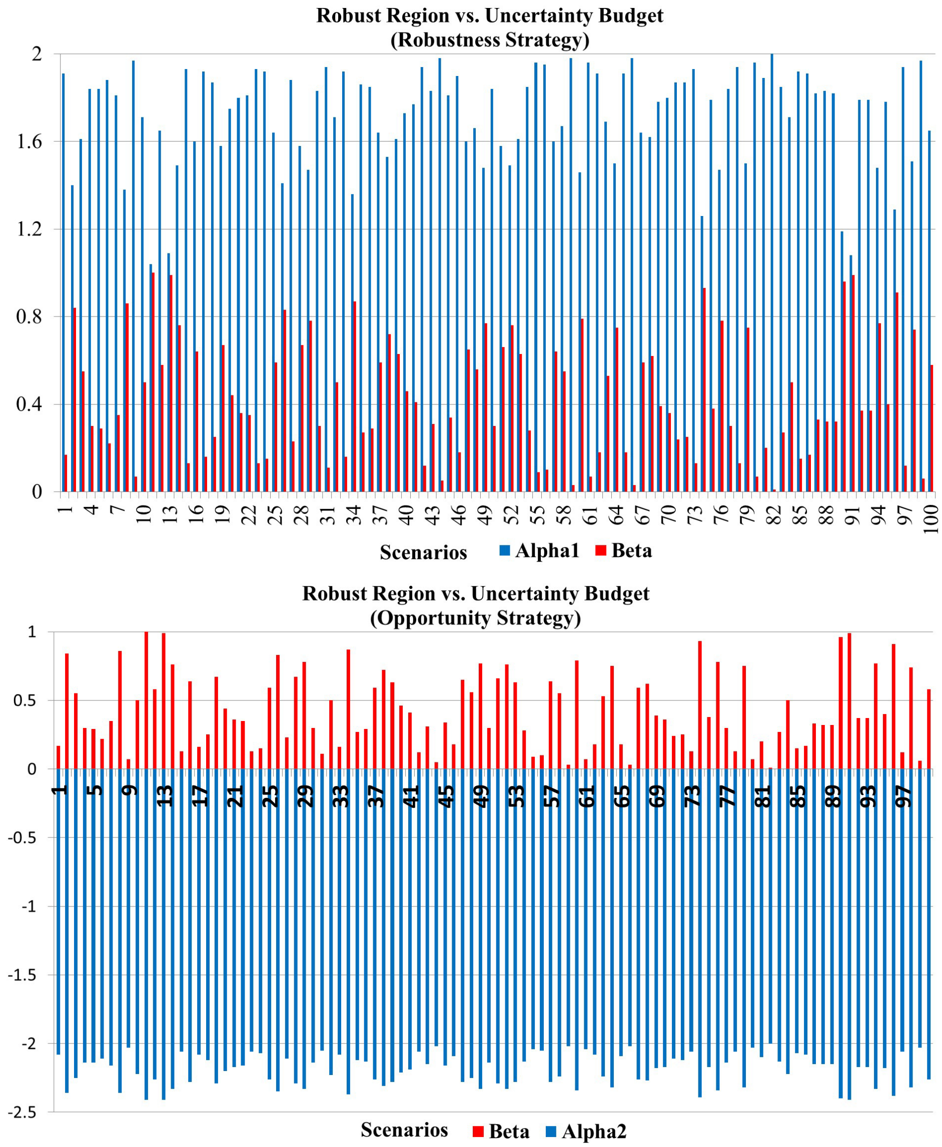


Figure 13. Variations in uncertainty budget interval for a) robustness strategy, b) opportunity strategy.

5.4. Evaluation of robustness in risk-averse (RA) strategy

This section compares the results of the RA strategy with Monte Carlo Simulation (MCS) and the SB approach. In both studies in this section, the tolerable cost is assumed to be 5% (i.e. $\beta = 0.05$).

5.4.1. Monte Carlo simulations

To evaluate the robustness of the RA strategy, the results are compared with MCS. This way, 1000 random values of spot prices are generated uniformly in the interval of $[(1-\alpha_1)\times \overset{\sim spot}{p}(h),\ p^{on-demand}]$ and the proposed model is solved for each random samples for the user. If the risk-averse strategy is robust for all values from the above set, the cost should be less than that of the risk-averse strategy. Simply speaking, for all samples, we must have $RCT \leq DCT \times (1+\beta)$.

The cost in the RA strategy is obtained using $(1-\alpha_1)\times \overset{\sim spot}{p}(h)$. The proposed RA strategy is robust conditional that the total processing cost is higher than that of the MCS in all scenarios.

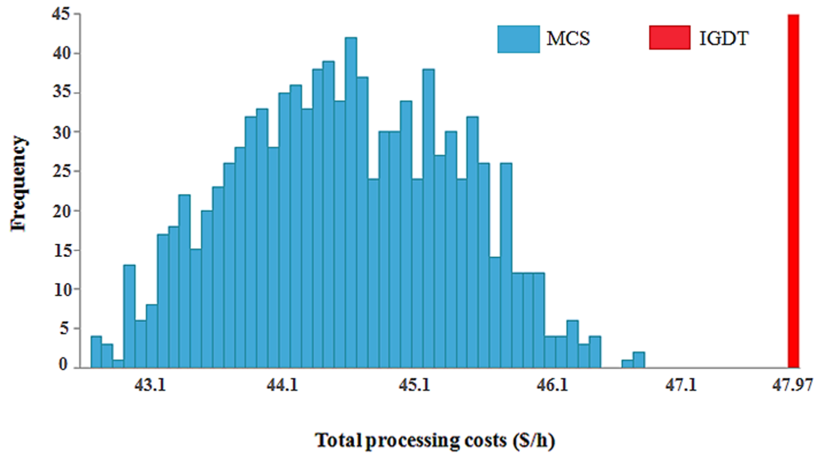


Figure 14. Distribution of total cost obtained by MCS and its comparison with RA strategy.

Figure 14 illustrates the distribution of total processing cost obtained by MCS and the RA strategy. The average cost obtained by MCS is $/h44.17, whereas the cost obtained by RA strategy is $/h47.97. As seen in this figure, the maximum cost obtained by the MCS is less than that of the RA strategy. It means that for any spot price value in the above interval, the cost will be less than the value attained by the RA strategy.

5.4.2. SB uncertainty modeling

In this section, the results of the RA strategy are compared with those obtained in the SB approach. We used the stochastic optimization solution to deal with spot price uncertainty for the resource rental scheme [30]. This way, the dynamic stochastic spot prices are represented in a multistage scenario tree. Moreover, the bid price is considered an approximation to the actual spot price.

Figure 15 shows the total cost in each scenario of the SB approach and the cost obtained by the RA strategy. The expected cost obtained by the SB approach is $/h48.29 with a standard deviation of $/h0.67, while the cost attained by the RA strategy is $/h47.97, which is lower than that of the SB approach. In some scenarios of the SB model, the user may face high levels of the total cost. Conversely, in the RA strategy, much lower costs can be obtained for some values of β. This confirms that IGDT can handle severe uncertainties of input parameters in decision problems. Moreover, the computational burden of the SB approach is much higher than that of the proposed RA method because the equality/inequality constraints must be considered simultaneously for all possible scenarios. Another concern of the SB approach is to generate accurate PDF price data for scenarios. Such a concern is not raised in the IGDT-based RA strategy.

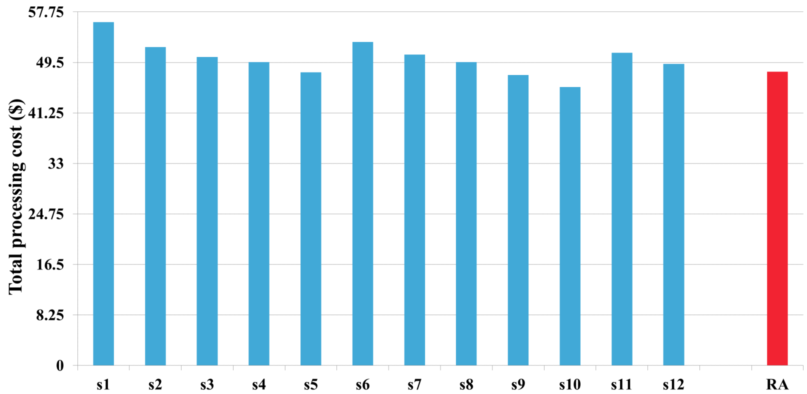


Figure 15. Distribution of total cost obtained by SB approach versus the RA strategy.

6. Conclusion and future trends

This paper used IGDT to develop RA/RS strategies for the user to find the maximum/minimum tolerable uncertainty in the spot price. The proposed model helps the user make the right decisions to cost-effectively determine the appropriate suggested prices. The diversity of costs in different scenarios of the SB model is considerable, and the user may face high levels of the total cost in some scenarios. Our experimental evaluations showed that the costs obtained in most scenarios of the SB approach are around the average cost. There are various costs for different values of β in the proposed IGDT-based model. Our evaluations showed that the average cost in the IGDT-based RA strategy is almost 1% better than the SB approach. Fortunately, in the IGDT-based method, for some β values, the cost is acceptably lower than the average. This is due to severe uncertainties in the input parameter of the problem. Unlike the SB approach, the IGDT does not require any information about the probability distribution function of the uncertain parameter (e.g., the spot price). It guarantees that the cost does not exceed a particular threshold value. Unlike MCS, the IGDT-based approach is computationally efficient because it does not require many iterations. Moreover, unlike the SB approach, there is no need to consider multiple scenarios for uncertain parameters. Our proposed mechanism can determine the level of robustness and opportunity to decide on the bid price.

The robustness of the result obtained in the RA strategy of the IGDT method was confirmed by Monte Carlo simulation. It was also shown that if the uncertainty in the spot price is not considered, the user may incur a higher cost in actual conditions than the corresponding amount in the RA strategy.

There exist important lines of research for future studies. Other mechanisms for checkpointing can be used to increase reliability. Adopting a low-risk bidding strategy while considering the user's future demand may lead to significant performance improvements. A combination of on-demand and spot pricing methods can also be used. Here, a procedure for determining the optimal number of required SVMs can be used by users. Another line of future research is multiobjective optimization using other uncertain parameters. It is worth noting that the proposed bidding strategy is not limited to Amazon and can be adopted by other cloud providers.

References

[1] Portella G, Rodrigues G, Nakano E, Melo A. Statistical Analysis of Amazon EC2 Cloud Pricing Models. Concurrency and Computation: Practice and Experience 2017; 31 (18): 1-16. doi: 10.1002/cpe

[2] Khandelwal V, Chaturvedi A, Gupta CP. Amazon EC2 SVMs' price prediction using regression random forests. IEEE Transactions on Cloud Computing 2017; 8 (1): 59-72. doi: 10.1109/TCC.2017.2780159

[3] Liu D, Cai Z, Li X. Hidden markov model based spot price prediction for cloud computing. In: IEEE 2017 International parallel and distributed processing with applications symposium and IEEE 2017 international ubiquitous computing and communications conference; Guangzhou, China; 2017. pp. 996-1003.

[4] Baughman M, Haas C, Wolski R, Foster I, Chard K. Predicting Amazon SVMs' prices with LSTM networks. In: ACM 2018 Scientific Cloud Computing Workshop; tempe, AZ, USA; 2018. pp. 1-7.

[5] Zheng L, Joe-Wong C, Tan CW, Chiang M, Wang X. How to bid the cloud. In: ACM 2015 Special Interest Group on Data Communication Conference; United Kingdom, London; 2015. pp. 71–84.

[6] Tang S, Yuan J, Wang C, Li XY. A framework for amazon ec2 bidding strategy under sla constraints. IEEE Transactions on Parallel and Distributed Systems 2014; 25 (1): 2-11. doi: 10.1109/TPDS.2015.2495120

[7] Mishra AK, Yadav DK, Kumar Y, Jain N. Improving reliability and reducing cost of task execution on preemptible VM instances using machine learning approach. The Journal of Supercomputing 2019; 75 (4): 2149-2180. doi: 10.1007/s11227-018-2717-7

[8] Chen Z, Ding W, Xu Y, Tian M, Zhong H. Fair auctioning and trading framework for cloud virtual machines based on blockchain. Computer Communications 2021; 171: 89-98. doi: 10.1016/j.comcom.2021.02.010

[9] Nanehkaran AB, Rezvani MH. An Incentive-Compatible Routing Protocol for Delay-Tolerant Networks Using Second-Price Sealed-Bid Auction Mechanism. Wireless Personal Communications 2021; 121: 1547-1576. doi: 10.1007/s11277-021-08684-w

[10] Besharati R, Rezvani MH, Gilanian Sadeghi MM. An Incentive-Compatible Offloading Mechanism in Fog-Cloud Environments Using Second-Price Sealed-Bid Auction. Journal of Grid Computing 2021; 19 (3). doi: 10.1007/s10723-021-09576-w

[11] Soroudi A, Amraee T. Decision making under uncertainty in energy systems: State of the art. Renewable and Sustainable Energy Reviews 2013; 28: 376–384. doi: 10.1016/j.rser.2013.08.039

[12] Ben-Haim Y. Information-gap Decision Theory: Decisions Under Severe Uncertainty. Academic Press Series on Decision and Risk. Academic Press 2001.

[13] Ahmadigorji M, Amjady N, Dehghan S. A robust model for multiyear distribution network reinforcement planning based on information-gap decision theory. IEEE Transactions on Power Systems 2018; 33 (2): 1339–1351. doi: 10.1109/TPWRS.2017.2732447

[14] Oconnell A, Soroudi A, Keane A. Distribution network operation under uncertainty using information gap decision theory. IEEE Transactions on Smart Grid 2018; 9 (3): 1848–1858. doi: 10.1109/TSG.2016.2601120

[15] Soroudi A, Rabiee A, Keane A. Information gap decision theory approach to deal with wind power uncertainty in unit commitment. Electric Power Systems Research 2017; 145: 137–148. doi: 10.1016/j.epsr.2017.01.001

[16] Salimi M, Nasr MA, Hosseinian SH. Information Gap Decision Theory-Based Active Distribution System Planning for Resilience Enhancement. IEEE Transactions on Smart Grid 2020; 11 (5): 4390-4402. doi: 10.1109/TSG.2020.2992642

[17] Nasr MA, Nasr-Azadani E, Rabiee A, Hosseinian SH. A risk-averse energy management system for isolated microgrids considering generation and demand uncertainties based on information gap decision theory. IET Renewable Power Generation, 2019; 13 (6): 940-951. doi: 10.1049/iet-rpg.2018.5856

[18] Kong D, Liu S, Pan L. Amazon Spot Instance Price Prediction with GRU Network. In: IEEE 2021 International Computer Supported Cooperative Work in Design Conference; Dalian, China; 2021.

[19] Portella JG, Rodrigues GN, Nakano EY, Boukerche A, Magalhaes MA. A Novel Statistical and Neural Network Combined Approach for the Cloud Spot Market. IEEE Transactions on Cloud Computing 2021. doi: 10.1109/TCC.2021.3091936

[20] Al-Theiabat H, Al-Ayyoub M, Alsmirat M, Aldwair M. A Deep Learning Approach for Amazon EC2 SVMs' price Prediction. In: IEEE/ACM 2018 International Computer Systems and Applications Conference; 2018. pp. 1-5.

[21] Naghdehforoushha M, Rezvani MH, Dehghan Takht-fooladi M, Gilanian Sadeghi MM. BLMDP: A New Bi-level Markov Decision Process Approach to Joint Bidding and Task-Scheduling in Cloud Spot Market. Turkish Journal of Electrical Engineering & Computer Sciences 2021; 30 (4): 1419-1438. doi: 10.55730/1300-0632.3857

[22] Ivashko EE, Ivashko AA, Safonov GR, Tchernykh A. Cost-Efficient Strategy in Clouds with spot Price Uncertainty. Automation and Remote Control 2020; 81 (4): 731-745. doi: 10.1134/S000511792004013X

[23] Izadi M, Hosseinian SH, Dehghan S, Fakharian A, Amjady N. A critical review on definitions, indices, and uncertainty characterization in resiliency-oriented operation of power systems. International Transactions on Electrical Energy Systems, 2021; 31 (4). doi: 10.1002/2050-7038.12680

[24] Mireslami S, Rakai L, Wang M, Far BH. Dynamic cloud resource allocation considering demand uncertainty. IEEE Transactions on Cloud Computin 2019; 9 (3): 981-994. doi: 10.1109/TCC.2019.2897304

[25] Xie H, Lui JC. A Reinforcement Learning Approach to Price Cloud Resources With Provable Convergence Guarantees. IEEE Transactions on Neural Networks and Learning Systems 2021; 1-13. doi: 10.1109/TNNLS.2021.3085088

[26] Song J, Guerin R. Pricing (and bidding) strategies for delay differentiated cloud services. ACM Transactions on Economics and Computation 2020; 8 (2): 1-58. doi: 10.1145/3381531

[27] Boisvert RF, Howe SE, Kahaner DK. GAMS: A framework for the management of scientific software. ACM Transactions on Mathematical Software 1985; 11 (4): 313-355. doi: 10.1145/6187.6188

[28] Fan H, Wang C, Liu L, Li X. Review of Uncertainty Modeling for Optimal Operation of Integrated Energy System. Frontiers in Energy Research 2022. doi: 10.3389/fenrg.2021.641337

[29] Malik M, Bagmar N. Forecasting Price of Amazon Spot Instances Using Machine Learning. International Journal of Artificial Intelligence and Machine Learning 2021; 11 (2): 71-82. doi: 10.4018/IJAIML.20210701.oa5

[30] Zhao H, Pan M, Liu X, Li X, Fang Y. Exploring Fine-Grained Resource Rental Planning in Cloud Computing. IEEE Transactions on Cloud Computing 2016; 3 (3): 304-317. doi: 10.1109/TCC.2015.2464799