Journal of Computer & Robotics 16 (1), Winter and Spring 2023, 75-87

Computer & Robotics

Non-Deterministic Optimal Pricing of VMs in Cloud Environments: An IGDT-Based Method

Mona Naghdehforoushha[a], Mehdi Dehghan Takht Fooladi[b], Mohammad Hossein Rezvani[a,*], Mohammad Mehdi Gilanian Sadeghi[a]

[a] Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran
Received 19 December 2022 ; Accepted 27 December 2022

Abstract

Today, cloud markets, especially Amazon, have attracted a lot of attention from users due to the provision of Spot Virtual Machines (SVMs). It has several advantages for both sides of the market. On the one hand, Amazon can generate revenue from its underutilized virtual machines. On the other hand, the customer can get the SVM as needed at a dynamic price through an auction method. Providing optimal bidding strategies in such a market is a crucial challenge. The bidding price is affected by uncertain parameters such as the price of SVMs, the number of available SVMs, the number of current customers, and their bidding values. In this paper, we use Information Gap Decision Theory (IGDT) to determine the best bidding strategy. Our proposed method includes both risk-averse and risk-neutral strategies. The evaluation results based on historical Amazon EC2 prices confirm the effectiveness of the proposed method in the presence of uncertain prices. It has high performance compared to the baseline methods in terms of robustness cost, uncertainty budget, and execution time.

Keywords: Cloud spot market; Bidding strategy; Uncertainty; Information Gap Decision Theory (IGDT)

1-Introduction

Cloud computing markets provide users with a large number of virtual resources in the form of Virtual Machine (VM) instances [1]. For example, the Amazon marketplace currently offers several types of Spot Virtual Machines (SVMs) by auction [2]. Auction has proven to be an effective mechanism for trading cloud services. It not only allows the customer to obtain the requested resources at lower prices but also allows the cloud provider to increase its profit [3].

We consider an Amazon-like auction market with a set of users and a set of homogeneous SVMs. In this market, any user can bid for SVM instances [4]. According to Amazon's policy, if the user's bid exceeds the price of the SVM, it will be leased to the user. Otherwise, the user must wait for the next bidding period [5]. Achieving SVMs significantly reduces users' computational costs for their jobs [6].

The constantly changing price of SVMs makes it difficult to decide on a bid price. Therefore, in most cases, making suggestions to the user is a challenging issue. If the user input is low, the probability of interrupting the execution of user tasks increases, which in turn leads to longer execution times and higher costs. Conversely, if the user's bid is high, the provider may increase the price. In this case, the cloud provider may lose its potential user and lose financially [7]. However, there is a financial benefit in calculating the bid price by taking into account the associated risks, especially the cash price. In addition, other uncertainties inherent in the cash market may also add to the complexity. Some of the most important uncertainties are the unavailability of SVMs, the future demand of users, and the patterns suggested by other users. Therefore, providing a suitable offer price requires the use of optimal bidding strategies.

The purpose of this paper is to design a robust strategy against momentary price fluctuations using Information Gap Decision Theory (IGDT) technique. IGDT is one of the powerful tools to find robust solutions with different levels of security against the uncertainty of input parameters. It has been used by previous researchers in other applications such as stock markets, electricity, etc. [8].

* Corresponding Author. Email: mhossein.rezvani@gmail.com

We use IGDT for user bidding concerning SVMs under spot price uncertainty. The use of IGDT has recently been promoted to deal with severe uncertainties in power systems [9]-[13]. To the best of our knowledge, no research has been conducted regarding the use of IGDT to model uncertainty in the cloud market. This is the first research in which the uncertainty of the parameters affecting the bid price (for example, SVM price) is modeled by the IGDT method.

In summary, the most important contributions of this article are as follows:

- Presenting a method for robust bidding decision-making for SVMs considering lower and upper limits for market prices
- Risk-constrained bidding decision-making for SVMs in the presence of spot price uncertainty to minimize the total processing cost of the user
- Obtaining appropriate bidding strategies by using IGDT

The rest of this paper is organized as follows: Section 2 reviews the related works; Section 3 presents the uncertainty modeling technique; Section 4 presents the formulation of the problem using two risk-averse and risk-neutral strategies; Section 5 provides the input data, results, and assessments in detail; Finally, the conclusion is provided in Section 6.

2- Related Works

Optimal bidding strategies without considering uncertainties (risk-neutral models) have been reviewed based on a combination of statistical and forecasting methods. For example, in [16], first, authors 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 the baseline ones. In [17], a utility-based strategy was proposed in favor of user decision-making for the short-term trade-offs between the spot price and availability. The results show that this solution can provide efficient choices of SVM instances, with low bids and high availability. Khandelwal et al. [18] 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. Also, a similar study was conducted by Al-Theiabat et al. [19], this time using the deep learning method and the use of Long-term Short-term Memory (LSTM). They found that this method has less error compared to other machine learning solutions such as Auto-regressive Integrated Moving Average (ARIMA). Liu et al. [20] designed a Hidden Markov Model (HMM) to predict spot prices. Their results show that the proposed model can predict the spot price more accurately compared to regression-based forecasting methods. The major disadvantage of the above-mentioned research is not handling the uncertainties in the price of SVMs. This can increase the deviation of the predicted prices from their real values in the next time slots.

Risk-based models of bidding problems mainly have been investigated based on probabilistic methods. Although there are various methods for handling uncertainty, only probabilistic methods have been applied to the bidding problem of SVMs. For example, Zheng et al. [21] proposed an optimal bidding strategy for cloud users which depends on the probability distribution function of prices. Mireslami et al. [22] 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. Also, Naghdehforoushha et al. [7] 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 lower level, it decides to admit tasks to maximize user-side satisfaction. Their results show that the proposed method manages to minimize cloud providers' costs and maximize user gain more effectively compared to heuristic methods. In another study, Ivashko et al. [23] developed a model to find the optimal bid using a threshold-based strategy. Although their method minimizes the cost of renting SVMs, it does require prior knowledge of price probability distribution, an assumption that does not exist in the real world at all. Xie and Lui [24] 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 show that the proposed dynamic model can lead to an increase in revenue of up to 20% compared to static pricing.

Unlike previous studies, our proposed IGDT-based method is a non-probabilistic method with two major advantages: First, 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. Regarding the user, our model depends on the amount of money of user wants to spend to get the SVMs within the specified deadline. Second, it finds the minimum and maximum price (price interval) for SVMs in a risk-averse strategy based on the IGDT. In this way, not only the user can cost-effectively perform the task, but also 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 the other words, even if users bid a high price for SVMs over time, the provider is still able to set a reasonable price. Our evaluations show that the IGDT is a robust mathematical tool for assessing risks in presence of uncertain prices.

3- Uncertainty Modeling

There are different models for dealing with existing uncertainties in the literature. Also, different methods have been created to model uncertainty from a wide range of methods, including stochastic programming, robust optimization, fuzzy, and IGDT [14, 15]. These methods differ in using different solutions to define the uncertainty of the input parameters. For example:

- In stochastic programming, the probability density function of the uncertain input parameter(s) must be known.
- In robust optimization, the uncertainty set or uncertainty radius must be known.
- In fuzzy methods, the membership function of the uncertain input parameter(s) must be known. Besides, working with fuzzy numbers is not easy.
- In the IGDT method, there is no need to have specific information about the uncertain parameter.

Since IGDT is an efficient method for modeling the uncertainty with unknown practical knowledge about their behavior, we use it to model the uncertainty of the spot price and investigate its effect in the presence of bid prices.

3-1- Principles of IGDT

IGDT is a decision-making method that tries to maximize system robustness in the face of severe uncertainties. The advantage of the IGDT compared to the stochastic programming method is related to the dependency of output variables from probabilistic scenarios. Moreover, IGDT is computationally lightweight. This means that it does not require any assumptions about the nature of the uncertain parameter. The only necessary assumption is the predicted value of the uncertain parameter so focus on the difference between the actual value of the uncertain parameter and the predicted value. It seeks to determine the maximum allowable bound of uncertainty for the uncertain parameter. In this case, the objective function remains within the allowable range. Also, the result of IGDT is accurate and efficient. The advantage of the IGDT compared to the robust optimization is related to the ability to model the worst and good realizations of uncertain parameters based on the user's strategy. The advantage of IGDT over probabilistic methods such as Monte Carlo is that it does not require specific information about uncertain parameters (e.g., probability density function, the fuzzy logic membership function, and the exact definition of scenarios) [8].
Despite the above-mentioned advantages of IGDT, there are also some disadvantages. It does not use historical data in uncertainty modeling. This can lead to more conservative results for IGDT. In the following, we will briefly explain this method.
As an optimization problem, this method includes objective function, equality, and inequality constraints, as follows:

$$of = \min_{X} f(X,\gamma) \tag{1}$$

$$H_i(X,\gamma) = 0 \tag{2}$$

$$G_j(X,\gamma) \le 0 \tag{3}$$

$$\gamma \in \Gamma \tag{4}$$

, where Γ denotes the set of uncertain parameters. Here, X represents decision variables. Also, γ is the uncertainty set describing the uncertain parameters, and G/H is the set of inequalities/equalities for the set of decision variables X .

One of the key points for IGDT is Info-Gap modeling. There are several types of models for uncertain parameters according to their attributes. Here, the envelope-bound model is used to represent the prior information about the uncertain input parameters γ [11]. Eq. (5) shows the mathematical description of the uncertainty set as an info-gap fractional error model in the IGDT.

$$\forall \gamma \in \Gamma(\bar{\gamma}, \alpha) = \left\{ \gamma : \left| \frac{\gamma - \bar{\gamma}}{\bar{\gamma}} \right| \le \alpha \right\} \tag{5}$$

, where $\bar{\gamma}$ is the predicted value of the uncertain parameter γ. Also, α is the maximum allowable deviation of the original realization of the uncertain parameter from its forecasted value. This is also called the unknown radius of uncertainty.

4- Problem Formulation

This section consists of two parts; first, in Section 4-1, the base case model is introduced, considering the objective function along with all constraints. Then, in Section 4-2, the risk-averse model is introduced, considering the objective function and constraints.

To evaluate the impact of uncertainty using the IGDT, it is necessary that obtain the solution in a Base Case (BC) according to the forecasted value of input parameters. BC is a deterministic model of the optimization problem by which a risk-neutral strategy is obtained. In other words, this strategy is the basic form of the proposed model without considering uncertainty. Then, in the second level, the optimization of different decision-making strategies is followed.

The BC optimization problem is described by Eqs. (6)-(8) and assumes that the uncertain parameter has no deviation from its predicted value.

$$of_b = \min_X f(X, \bar{\gamma}) \tag{6}$$

$$H_i(X, \bar{\gamma}) = 0 \tag{7}$$

$$G_j(X, \bar{\gamma}) \le 0 \tag{8}$$

Assuming that the unknown parameter is exactly equal to the predicted value, the BC value of the objective function is obtained based on the output obtained from Eq. (6)-(8). If the uncertain parameter differs from its predicted value, the decision-maker is encountered with two different strategies: *risk-averse* strategy and *risk-seeker* strategy. The decision-making in a risk-averse strategy is undertaken pessimistically. Here, the decision-maker assumes that uncertainty has an undesirable effect on the objective function. On contrary, the decision-making in a risk-seeker strategy is undertaken optimistically. Here, the decision-maker assumes that the uncertainty not only may not adversely affect the objective function but also can help achieve a better objective function compared to the BC value.

For VM instances that are affected by real-time price fluctuations, optimistic decision-making can cause irreparable financial losses to the user. To overcome this problem, we adopt the RA strategy to increase the robustness of the bidding strategy against uncertainty. It makes the objective function resistant to the possibility of error in the prediction of the uncertain parameter. To address uncertainty, a risk aversion strategy can be applied by the decision-maker with a risk-neutral strategy. Now we proceed to formulate it.

4-1- Risk-neutral Problem Formulation

In this section, we present the deterministic formulation for the optimization problem concerning the cost of processing tasks on SVMs during a *T*-hour period by Eq. (9). The base system model and related formulation are given in [25]. We consider the following assumptions in this study according to [25]:

- Every user, requesting a spot service, participates in the spot market to submit user tasks.
- Every user acts as a price-taker agent, in the sense that user bids alone cannot influence the price 0f SVMs.
- In each hour, the user predicts the price of SVMs an hour ahead.
- Each user considers only the processing costs and neglects the storage and data transfer costs.
- Since the price of SVMs depends on the supply-demand pattern in a geographical area, we limit ourselves to a predefined geographical area.
- The bid price dynamically changes each hour.
- The purchased SVMs are homogeneous and are only of the Spot type.

- Each user uses a *checkpointing* technique to store calculations before the out-of-bid event.

The decision variable is the bid price for the round t+1, which is denoted by b_{t+1}. This value must be calculated by the user for the next hour considering the price of SVMs of 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 1 shows the mathematical notations used in this paper.

This risk-neutral term is the deterministic form of the base problem.

$$DECF = Min\sum_{i=1}^{N_{VM}}\sum_{h=1}^{T}\delta(h)\cdot prg_i(h)\cdot f(p^{spot}(h)) \quad (9)$$

, where $\delta(h)$ denotes the binary variable associated with allocating SVMs to the user, $prg_i(h)$ 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 Eq. (28), we will estimate the function *f(.)* through the curve fitting technique for the hourly price of SVMs. The processing cost is the amount of money that is paid by the user for using the SVMs conditional on the successful bidding by the user. Eq. (9) simply states that the user's objective is to minimize the sum of the costs associated with the progress of the tasks at the specified prices. This cost will be paid by the user to the provider for the entire business hours.

According to Eq. (10), if the bid price at round *h* is higher than the price of SVMs at that round, the requested SVMs are allocated to the user, and the value of the binary variable $\delta(h)$ at that round is 1; otherwise, no SVM is allocated to the user:

$$\delta(h)=\begin{cases}1 & b(h)\geq p^{spot}(h)\\ 0 & else\end{cases} \quad (10)$$

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 *i*-th task on the corresponding SVM at each hour can be calculated as follows:

$$prg_i(h)=\begin{cases}1-t_{checkpoinitng} & if\ (\delta(h-1)=1\wedge\delta(h)=1\wedge\hat{\delta}(h+1)=0)\\ 1-t_{resume} & if\ (\delta(h-1)=0\wedge\delta(h)=1\wedge\hat{\delta}(h+1)=1)\\ 1-t_{checkpoinitng}-t_{resume} & if\ (\delta(h-1)=0\wedge\delta(h)=1\wedge\hat{\delta}(h+1)=0)\\ 1 & if\ (\delta(h-1)=1\wedge\delta(h)=1\wedge\hat{\delta}(h+1)=1)\\ 0 & otherwise\end{cases} \quad (11)$$

As stated before, we can use a fault-tolerance mechanism such as *checkpointing* to store the system state to avoid the resulting loss when an out-of-bid failure occurs [25].

For the estimation of $\hat{\delta}(h+1)$ in Eq. (11), one must at first estimate the price of SVMs $\hat{p}^{spot}(h+1)$ for the next hour, and then estimate the amount of bid price $\hat{b}(h+1)$. The value of $\hat{b}(h+1)$ at the next time slot is calculated using the following equation:

$$\hat{b}(h+1)=b(h)\times(1+\Delta b) \quad (12)$$

, in which Δb is obtained as follows:

$$\Delta b=\frac{p^{spot}(h)-p^{spot}(h-1)}{p^{spot}(h-1)} \qquad \forall h\in H \quad (13)$$

The Δb can be estimated from price fluctuations in two consecutive time slots. In this regard, the total progress rate concerning each task on a dedicated SVM during all hours should not exceed the user-specified time. This constraint is stated as follows:

$$\sum_{h=1}^{T}prg\ (i,h)\leq t^{execution}(i) \qquad \forall i\in VM, h\in H \quad (14)$$

Also, it should be ensured that the deadline for executing the user’s tasks is not violated. In this regard, the sum of hours spent executing tasks on each SVM shall not exceed the user-specified deadline for that task. So, we write the following constraint:

$$\sum_{h=1}^{T}MH(i,h)\leq t^{deadline}(i) \qquad \forall i\in VM \quad (15)$$

Table 1
Mathematical Notations

$DECF$	Deterministic ECost Function
$RECF$	Robust ECost Function
$DECF$	The objective function of the deterministic model
$RECF$	The objective function of the robust model
h	Index of hours (1 to T)
i	Index of SVMs (1 to N_{VM})
$SVM = \{1,..., N_{SVM}\}$	Set of all requested SVMs by the user
$H = \{0,\dots,T\}$	Set of daily hours (time slots)
$prg(i,h)$	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 the result (in minutes)
N_{VM}	number of SVMs
T	number of time slots
β	Budget of uncertainty
Ψ	Robust region of the uncertainty sources
$MH(i,h)$	Runtime spent on each machine (hour)
$p^{spot}(h)$	Price of SVMs for each time slot ($/hour)
$\hat{p}^{spot}(h+1)$	The estimated price of SVMs for the next time slot ($/hour)
$b(h)$	The bid price of the user for an SVM type at a 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 change in the bid price
$\delta(h)$	Binary variable indicating whether requested SVM instances 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 the time slot $h+1$ or not
α	The envelope of the robust region of the uncertainties $p^{spot}(h)$
$\tilde{P}^{spot}(h)$	The predicted price of SVMs for h time slots ($)
$p^{on-demand}$	price of on-demand VMs ($)

It should also be noted that the price of SVMs per hour can not exceed the price of on-demand instances. Otherwise, the user would prefer to use on-demand instances with high reliability instead of unreliable SVMs. So the following constraint can be defined accordingly:

$$p^{spot}(h) < P^{on-demand} \qquad \forall h \in H \tag{16}$$

4-2- Risk-Averse Problem Formulation using IGDT

As mentioned earlier, different methods, including stochastic optimization, robust optimization, probabilistic methods, and IGDT, are used in the literature to model uncertainties in optimization problems [26]-[29]. In particular, IGDT is a powerful way to describe uncertainty. This technique is an interval optimization method that optimizes the objective under uncertainty so that it does not require uncertain parameter historical data in the modeling. IGDT models the uncertainty by an interval around the predicted value of the uncertainty. It also controls the risk of prediction by guaranteeing a predetermined level of objective and introducing the maximum level of the confidence interval around the predicted value. The IGDT seeks to determine the maximum allowable bound of uncertainty for the uncertain parameter so that the objective function remains within the allowable range. It proposes a confidence interval to the decision-maker by considering a distance around the predicted value of the uncertainty parameter. According to the decision maker’s risk preference, IGDT can provide robust strategies corresponding to cost expectations for random variables located within a given interval. As widely accepted, cloud users are risk-averse. Hence, this method is the best option for conservative decision-makers. This strategy occurs if an uncertain parameter increases the objective function of cost. Therefore, this strategy seeks to find the maximum value of the uncertain parameters for the worst predetermined amount of the objective function relative to its base value. In this strategy, by solving the robustness function, the objective function is resisted against uncertain parameter deviation. The robustness function means that the decision-maker was assured of a deviation in the uncertain parameter, in which the objective function value would not be increased than the predetermined value. Therefore, the robustness function is applied in this paper so that we maximize the level of uncertainty within the user’s tolerance while maintaining a certain cost level for the user [11]. Hence a maximized allowable deviation for the price of SVMs from predicted values will be derived using the IGDT technique [8]. The robust function is utilized to maximize the risk level of the SVMs' price that the user can bear for an expected level of cost. By detecting the worst case, the user's bidding strategy optimization problem is solved as an MINLP. The following mathematical relationships describe this strategy:

$$\hat{\alpha} = \max_{\alpha, X} \alpha \tag{17}$$

$$H_i(X, \gamma) = 0, \quad i \in \Gamma_{equality} \tag{18}$$

$$G_j(X, \gamma) \leq 0, \quad j \in \Gamma_{inequality} \tag{19}$$

Finally, the risk-averse strategy used in our research is applied as follows:

$$\hat{\alpha}_1 = \max_{\alpha, X} \alpha_1 \tag{20}$$

St:

$$RECF \leq DECF.(1+\beta) \tag{21}$$

$$RECF = \sum_{i=1}^{N_{VM}} \sum_{h=1}^{T} \delta(h) \cdot prg(i,h) \cdot f(\tilde{p}^{spot}(h).(1+\alpha_1)) \tag{22}$$

$$\gamma\left(\tilde{p}^{spot}(h), \alpha\right) = \left\{ \tilde{p}^{spot}(h) : \frac{p^{spot}(h) - \tilde{p}^{spot}(h)}{\tilde{p}^{spot}(h)} \leq \alpha_1 \right\} \tag{23}$$

$$\delta(h) = \begin{cases} 1 & b(h) \geq \tilde{p}^{spot}(h).(1+\alpha_1) \\ 0 & else \end{cases} \tag{24}$$

$$\tilde{p}^{spot}(h) \cdot (1+\alpha_1) < p^{on-demand} \qquad \forall h \in H \tag{25}$$

The inherent limitations of the problem, i.e. Eqs. (14-16) (26)

According to the IGDT concept, the maximum value of α_1 is obtained by solving the single-objective robust problem in Eq. (20). The IGDT method takes into account the uncertainty parameters which were not considered by the deterministic method at all. It maximizes the envelope of the robust region using Eq. (23). Note that the *DECF* denotes the deterministic cost function attained by Eq. (9), while the *RECF* denotes the robustness cost function in Eq. (22). The robust region of $\tilde{p}^{spot}(h)$ can be summarized by Eq. (23). Therefore, Eq. (10) and constraint in Eq. (16) can be rewritten as Eqs. (24)-(25) to obtain the worst case. This considered model is a Mixed-Integer Non-linear Programming (MINLP), which is solved by the GAMS optimization software under the BARON solver [30].

At the end of this section, we briefly explain the proposed algorithm for the optimal bidding strategy. The cloud user sends hourly bids to the cloud provider to rent his/her needed computing resources. The above-mentioned *robustness* function of the IGDT, ensure the user makes appropriate decisions concerning the bidding interval. In the following, we explain the procedure for determining the bid value in each time slot (an hour):

(1) Initially, the user finds his/her minimum cost function using Eqs. (9)-(16). The resultant cost value is, in fact, the *expected minimum cost* if market prices are equal to the predicted values per hour.

(2) Now, the user finds his/her optimal value of the *robustness* function using Eqs. (20)-(26). As stated before, the value of the robustness function is less than the expected minimum cost obtained in step (1).

(3) For all levels of robustness cost (the confidence level) (α_1), the actual values of the price of SVMs for each iteration s are obtained.

(4) Considering the robustness and maximum values of α_1, the user calculates the optimum bid price as follows:

$$b(h+1) = \hat{b}(h+1) \times (1+\alpha_1) \tag{27}$$

Fig. 1 shows the flowchart of the proposed method in brief. Later in Section 5, we will provide more explanations about evaluation scenarios.

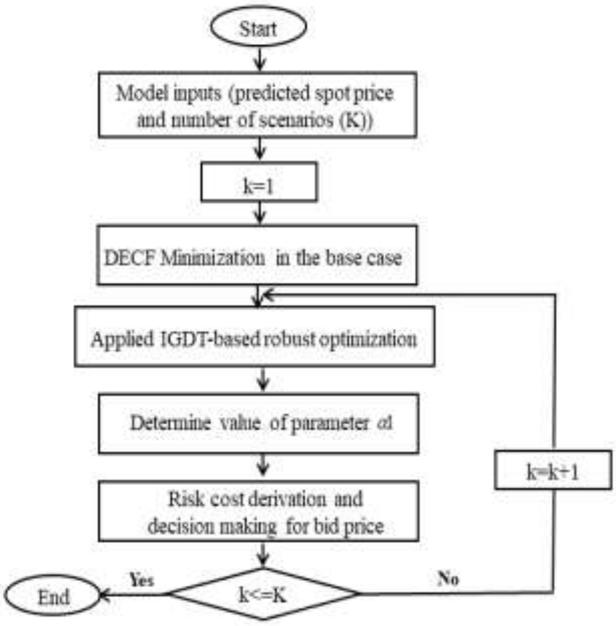


Fig 1. The flowchart of the proposed method

5- Performance Evaluation

To analyze the performance of the proposed method, we run extensive experiments on a 64-bit Intel® Core i7-2670QM processor with 6 MB Cache, 2.2 GHz CPU frequency, and 8 GB RAM. We evaluated our algorithm using one of the most popular public cloud providers, namely Amazon EC2 [31]. It is responsible for providing computing capacity that can scale in the Amazon Web Services (AWS) cloud [32]. Amazon offers its surplus computing capacity under a spot pricing scheme to customers, e.g., organizational tenants. Each user participates in a next-hour market with a time horizon of 24 hours. The resource pricing policy is imposed by Amazon EC2 based on "on-demand" and "spot" pricing schemes. The pricing procedure is performed for general-purpose SVMs, for example, *t2.small*. Note that our proposed method is not limited to Amazon services and can be generalized to any other type of SVMs from any cloud provider.

5-1- Data

Because Amazon has only provided 90-day spot prices historical to users for a variety of SVMs [31], we use them as input data. Fig. 2 shows, for example, the pattern of spot price changes for the specified VM instance (*m2.4xlarge*) concerning the *"us-east-1f"* geographical zone for February 17, 2017.

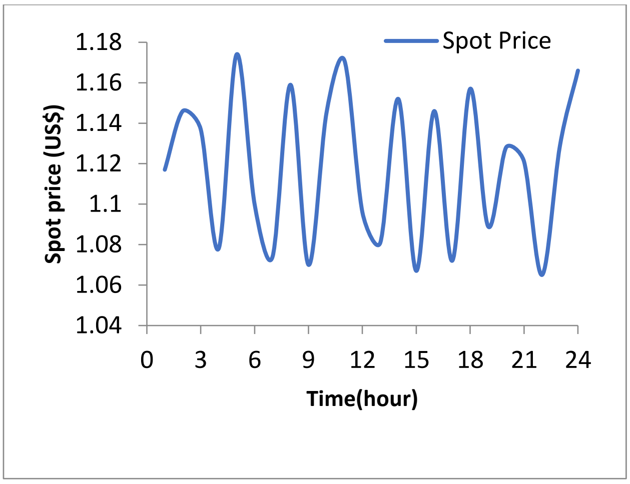
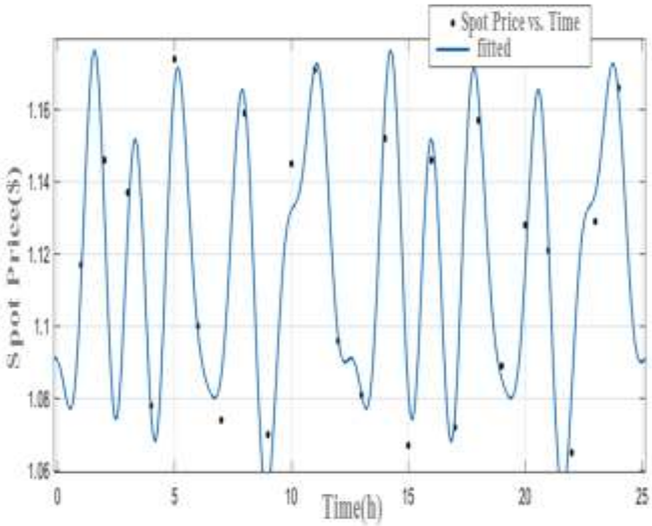


Fig 2. price fluctuations of SVMs concerning *"m2.4xlarge"* instances during different hours [31].

Now, using the MATLAB 2020a software, we proceed to find the best-fit curve concerning the price pattern of Fig. 3. The details of the calculations



are as follows:

Fig 3. Curve fitting for hourly prices pattern of Fig.2

$$
\begin{aligned}
& f(x) = a_0 + a_1 \times \cos(x \times w) + b_1 \times \sin(x \times w) + a_2 \times \cos(2 \times x \times w) + \\
& b_2 \times \sin(2 \times x \times w) + a_3 \times \cos(3 \times x \times w) + b_3 \times \sin(3 \times x \times w) + \\
& a_4 \times \cos(4 \times x \times w) + b_4 \times \sin(4 \times x \times w) + a_5 \times \cos(5 \times x \times w) + \\
& b_5 \times \sin(5 \times x \times w) + a_6 \times \cos(6 \times x \times w) + b_6 \times \sin(6 \times x \times w) + \\
& a_7 \times \cos(7 \times x \times w) + b_7 \times \sin(7 \times x \times w) + a_8 \times \cos(8 \times x \times w) + \\
& b_8 \times \sin(8 \times x \times w) \\
& a_0 = 1.117,\ a_1 = 0.002, b_1 = -0.0003, a_2 = -0.002, \\
& b_2 = -0.008, a_3 = -0.013, b_3 = -0.001, a_4 = -0.028, \\
& b_4 = 0.0007, a_5 = 0.006, b_5 = 0.012, \\
& a_6 = -0.015, b_6 = -0.016, a_7 = 0.006, \\
& b_7 = -0.006, a_8 = 0.018, b_8 = 0.009, w = 0.495
\end{aligned}
\tag{28}
$$

The fitted curve of Eq. (28) is shown in Fig. 3.

Two cases are represented to clarify the effectiveness of the proposed bidding strategy and to reveal the results of using IGDT optimization, as follows:

Case I: Without considering IGDT as a risk-neutral strategy

Case II: Considering IGDT as a risk-averse strategy

5-2- Evaluation Without Uncertainty

We will examine the allocation of spot VM instances to the consumer, as well as the execution progress rate of the tasks. The total cost, in this case, is $ 63.205 and its execution time is 0.492 seconds.

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 hours. This vector is shown in Fig. 4. According to Eq. (10), if the current bid is higher than the price of SVMs, the consumer will be able to use cheap SVMs to perform his/her tasks at that time. As is evident in Fig. 4, 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. This 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 resources for the rest of the time due to the low bid offered by him/her. So at those hours, the value of the variable $\delta(h)$ is zero.

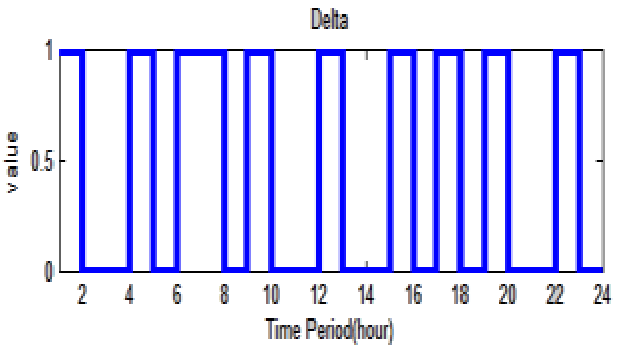


Fig 4. The value of the variable $\delta(h)$ for 24 hours

Fig. 5 shows the progress of the user's task on different SVMs. As stated earlier in Eq. (11), in each hour, if the deadline is not violated and the VM is assigned to the consumer, then the progress rate of the task can be calculated.

As stated before in Eq. (11), if the SVM is allocated during the previous time slot, the rate of execution progress for the task, $prg_i(h)$, at the current and next time slots will be different. Based on this, five different cases for $prg_i(h)$ can be distinguished on each VM at each time slot:

Case 1: This situation occurs at time slot 7 in Fig. 5. As stated in the first condition of Eq. (11), in the previous and current hours, the SVM has been allocated to the consumer. This means that during these hours, the bids offered by the user were higher than the price of SVMs (according to Eq. (10)). But this is not the case in the next hour. Therefore, the out-of-bid event will take place 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 rate is equal to $1-t_{checkpointing}$.

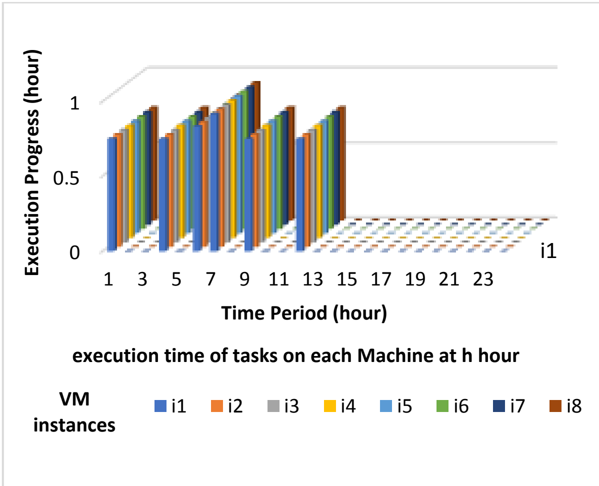


Fig 5. Execution progress rate of task on spot VMs for 24 hours

Case 2: This situation occurs at time slot 6 in Fig. 5. As stated in the second condition of Eq. (11), in the current and next hours, the spot VM has been allocated to the consumer. This means that during these hours, the bids offered by the consumer were higher than the spot price. But this is not the case in the previous hour. Therefore, the out-of-bid event has been taken place in the previous 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 rate is equal to $1-t_{resume}$.

Case 3: This situation occurs at time slots 1, 4, 9, and 12 in Fig. 5. As stated in the third condition of Eq. (11), only in the current hour, the spot VM has been allocated to the consumer. This means that merely during the current hour, the consumer bid was higher than the spot price. But this is not the case in the previous and next hours. Therefore, the out-of-bid event has taken place in the previous and next hours. For this reason, due to the out-of-bid occurrence in the previous hour, the previous results must be retrieved. Also, due to the out-of-bid occurrence in the next hour, the obtained results must be stored in the current 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 rate is equal to $1-t_{checkpointing}-t_{resume}$.

Case 4: This situation has not occurred in Fig. 5 at all. As stated in the fourth condition of Eq. (11), in three consecutive hours, the spot VM has been allocated to the consumer. This means that during these hours, the consumer bid was higher than the spot price. Therefore, in this case, the value of the progress rate is equal to one full hour.

Case 5: When none of the above four situations occur, the value of the progress rate at that time slot is equal to zero. This situation occurs at time slots 2, 3, 5, 8, 10, 11, and 13-24 in Fig. 5.

It is also emphasized that in each of the above cases and at each hour, it must be checked that the task deadline is no more than the current hour. Because in this case, the deadline for the task has ended and its continuing will no longer have worth to the consumer.

5-3- Evaluation in the Presence of Uncertainty

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 Fig. 6.

In the risk-averse strategy, the cost is minimized from the perspective of the decision variable. It simultaneously performs the maximization of the objective function from the perspective of the uncertainty variable. Thus, the best decision is made against the most pessimistic occurrence of uncertainty. In this case, the maximum cost will be less than a critical cost. As mentioned earlier, β denotes the critical cost deviation coefficient in deterministic mode. We consider 100 scenarios in the simulations. In each scenario, a random number in the interval [0, 1] is generated for β. Then, for each generated β, a confidence interval around the predicted value for the uncertainty parameter (spot price) is obtained. In addition, the total cost of executing the tasks for the robustness function in the risk-averse strategy is obtained.

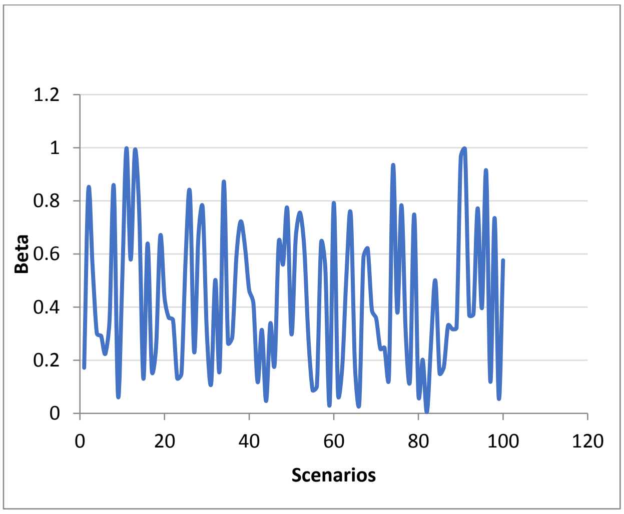


Fig 6. Random values generated for β in each execution scenario

As explained in Section 4-2, in the IGDT-based method, the robustness strategy is implemented. The variations of the *"tolerable uncertainty"* versus the *"objective cost"* in this strategy are depicted in Fig. 7. Note that in the figure, *"Alpha1"* represents $\hat{\alpha}_1$ (Eq. (20)). As is evident in Fig. 7, 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 (base mode) to $62.88 ($\beta=0.5763$). As expected, when the objective cost increases, it shows more tolerance toward uncertainty. This fact is evident in Fig. 9. The average calculation time of the algorithm in the robustness strategy in each scenario is equal to 0.2427 seconds.

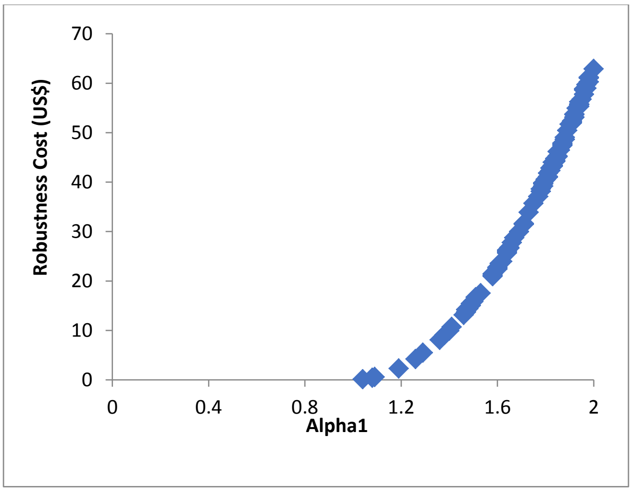


Fig 7. The variations of "tolerable uncertainty" vs "objective cost" concerning the robustness strategy

Fig. 8 shows a comparison of the "total cost" of task processing concerning *robustness* strategy for different scenarios.

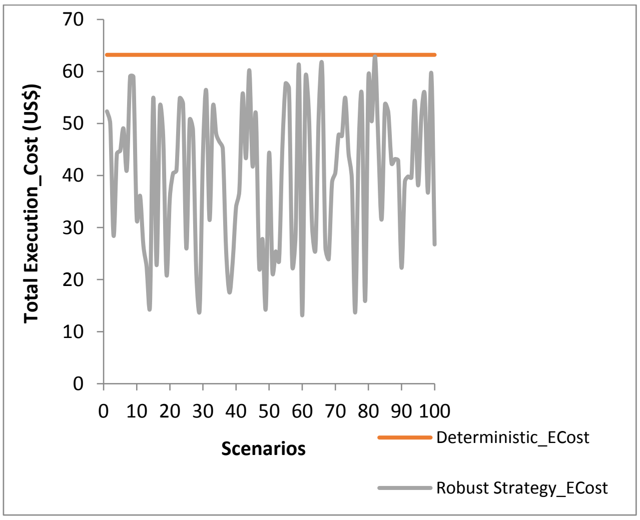


Fig 8. The "total cost" of task processing concerning robustness and deterministic strategies for different scenarios

Also, our results on 100 different scenarios show that the minimum β value is equal to 0.05. This value corresponds to $\alpha_1 = 1.98$ in the robustness strategy. In contrast, the maximum β value is equal to 1. This value corresponds to $\alpha_1 = 1.04$ in the robustness strategy. Due to space limitations, we omit the table of 100 scenario executions. Moreover, the execution time concerning the robustness strategy for different scenarios is shown in Fig. 9.

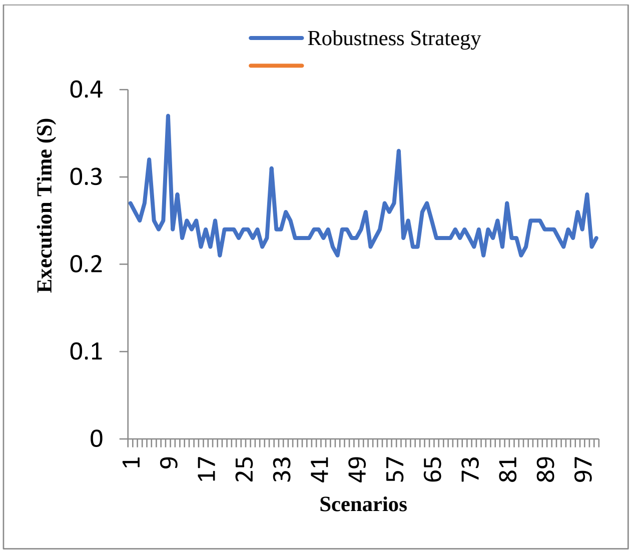


Fig 9. The execution time concerning *robustness* strategy for different scenarios

Fig. 10 shows consumer bids based on the IGDT algorithm versus spot prices during 24 hours. As can be seen from the figure, the spot price 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 algorithm, consumer 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 spot price has been decreasing at the time slots 2-4, 5-7, 8-9, 11-13, 14-15, 16-17, 18-19, and 20-22. Accordingly, consumer bids at time slots 3-5, 6-8, 9-10, 12-14, 15-16, 17-18, 19-20, and 21-23 have experienced a fall. As is evident, the consumer's bid is always one time slot, behind the trend of rising/falling the price of SVMs.

Fig 10. The user bids based on the IGDT algorithm vs spot prices during 24 hours

Fig. 11 shows the variations in the uncertainty budget interval for the robustness strategy. As expected, with an increase in the uncertainty budget, the robustness interval increases too. In the risk-averse policy (robustness strategy), the decision-maker aims to make the least possible profit by spending the most. Needless to say, such an agent must take the least risk. According to Eq. (21), when the β value increases, because the beta is in the range [0, 1], the value of 1+β is greater than 1 and is incremental. Mathematically, when this incremental expression is multiplied by the *DECF* value, it increases the distance between the *RECF* and the DECF values and thus increases the cost.

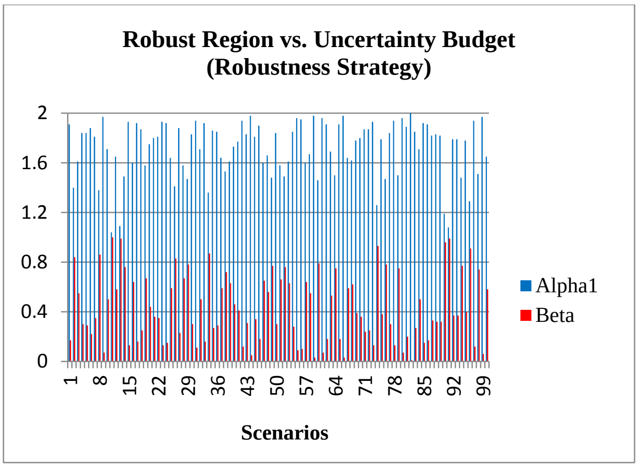


Fig 11. Variations in uncertainty budget interval for the robustness strategy

6- Conclusion

In this paper, a bidding strategy for SVMs using IGDT was presented. We targeted a cost-effective dynamic bidding strategy for spot pricing schemes in cloud marketplace environments. In this regard, the most important issue is to minimize the overall processing costs of the consumer by taking into account the QoS constraints (e.g., an upper bound on the execution time) while taking into account the uncertainty in the spot prices. We formulated the problem in both deterministic (without uncertainty modeling) and non-deterministic (with uncertainty modeling) settings. To handle the uncertainties, we used the IGDT method, which can significantly approximate the near-optimal solution. It uses a risk-averse strategy and determines the strength level to decide on the bid price. As a result, it helps the user to make the right decisions to set the right bid price.
Numerical analyzes on different scenarios using the Amazon spot price dataset proved that the proposed IGDT-based algorithm can reasonably balance monetary costs and reliability. Since the proposed bidding strategy is not dependent on a specific cloud provider, it can be used for all Infrastructure-as-a-Service (IaaS) environments.
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. Also, a procedure for determining the optimal number of required VM instances can be used by users. Another line of future research is multi-objective optimization using other uncertain parameters.

References

[1] Zhu, Z., Peng, J., Liu, K. and Zhang, X., 2020. A game-based resource pricing and allocation mechanism for profit maximization in cloud computing. Soft Computing, 24(6), pp.4191-4203.

[2] Chen Z., Ding W., Xu Y., Tian M., and Zhong, H., Fair auctioning and trading framework for cloud virtual machines based on blockchain. Computer Communications, 171, pp. 89-98, (2021).

[3] Song J., and Gu´erin R., Pricing (and bidding) strategies for delay differentiated cloud services. ACM Transactions on Economics and Computation (TEAC), 8(2), pp. 1-58, (2020).

[4] 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.

[5] Kumar, D., Baranwal, G., Raza, Z. and Vidyarthi, D.P., 2018. A survey on spot pricing in cloud computing. Journal of Network and Systems Management, 26(4), pp.809-856.

[6] Song, J. and Guérin, R., 2020. Pricing (and bidding) strategies for delay differentiated cloud services. ACM Transactions on Economics and Computation (TEAC), 8(2), pp.1-58.

[7] Naghdehforoushha M., Rezvani M.H., Dehghan Takht-fooladi M., and Gilanian Sadeghi M.M., BLMDP: A New Bi-level Markov Decision Process Approach to Joint Bidding and Task-Scheduling in Cloud Spot Market. Turkish Journal of Electrical Engineering and Computer Sciences, (2022).

[8] Ben-Haim Y., Information-gap Decision Theory: Decisions Under Severe Uncertainty. ser. Academic Press Series on Decision and Risk. Academic Press, (2001).

[9] Ahmadigorji M., Amjady N., and Dehghan S., A robust model for multiyear distribution network reinforcement planning based on information-gap decision theory, 33(2), pp. 1339–1351, (2018).

[10] Oconnell A., Soroudi A., and Keane A., Distribution network operation under uncertainty using information gap decision theory. IEEE Transactions on Smart Grid, 9(3), pp. 1848–1858, (2018).

[11] Soroudi A., Rabiee A., and Keane A., Information gap decision theory approach to deal with wind power uncertainty in unit commitment. Electric Power Systems Research, 145, pp. 137–148, (2017).

[12] Salimi, M., Nasr, M.A., and Hosseinian S.H., Information Gap Decision Theory-Based Active Distribution System Planning 35 for Resilience Enhancement. IEEE Transactions on Smart Grid, (2020).

[13] Nasr, M.A., Nasr-Azadani, E., Rabiee A., and Hosseinian S.H., A risk-averse energy management system for isolated micro37 grids considering generation and demand uncertainties based on information gap decision theory. IET Renewable Power Generation, (2019).

[14] Soroudi A., and Amraee T., Decision making under uncertainty in energy systems: State of the art. Renewable and Sustainable Energy Reviews, pp. 376–384, (2013).

[15] Izadi M., Hosseinian S.H., Dehghan S., Fakharian A., and 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).

[16] Kong D., Liu S., and Pan L., Amazon Spot Instance Price Prediction with GRU Network. IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), (2021).

[17] Portella J.G., Rodrigues G.N., Nakano E.Y., Boukerche A., and Magalhaes Melo A.c., A Novel Statistical and Neural Network Combined Approach for the Cloud Spot Market. IEEE Transactions on Cloud Computing, (2021).

[18] Khandelwal V., Chaturvedi A., and Gupta C.P., Amazon EC2 SVMs' price prediction using regression random forests. 1 IEEE Transactions on Cloud Computing, (2017).

[19] Al-Theiabat H., Al-Ayyoub M., Alsmirat M., and Aldwair M. A., Deep Learning Approach for Amazon EC2 SVMs' price Prediction. IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1-5, (2018).

[20] Liu D., Cai Z., and Li X., Hidden markov model based spot price prediction for cloud computing. IEEE International symposium on parallel and distributed processing with applications and IEEE international conference on ubiquitous computing

[21] Zheng L., Joe-Wong C., Tan C.W., Chiang M., and Wang X., How to bid the cloud. ACM Conference on Special Interest Group on Data Communication, pp. 71–84, (2015).

[22] Mireslami S., Rakai L., Wang M., and Far B.H., Dynamic cloud resource allocation considering demand uncertainty. IEEE Transactions on Cloud Computing, (2019).

[23] Ivashko E.E., Ivashko A.A., Safonov G.R., and Tchernykh A., Cost-Efficient Strategy in Clouds with spot Price Uncertainty. Automation and Remote Control, 81(4), (2020).

[24] Xie H., and Lui J.C., A Reinforcement Learning Approach to Price Cloud Resources With Provable Convergence Guarantees. IEEE Transactions on Neural Networks and Learning Systems, (2021).

[25] Tang S., Yuan J., Wang C., and Li X.Y., A framework for amazon ec2 bidding strategy under SLA constraints. IEEE Transactions on Parallel and Distributed Systems, 25(1), pp. 2-11, (2013).

[26] Santhosh A., Farid A.M., and Youcef-Toumi K., Real-time economic dispatch for the supply side of the energy-water nexus. Appl. Energy 122, pp. 42–52 Jun, (2014).

[27] Nourollahi R., Mohammadi-Ivatloo B., and Zare K., Hybrid Stochastic/Robust Offering Strategy for Coordinated Wind Power and Compressed Air Energy Storage in Multielectricity Markets. IEEE Syst. J., pp. 1–8, (2021).

[28] Nourollahi R., Salyani P., Zare K., and Mohammadi-Ivatloo B., Resiliency-oriented optimal scheduling of microgrids in the presence of demand response programs using a hybrid stochastic-robust optimization approach. Int. J. Electr. Power Energy Syst., (2021).

[29] Soroudi A., and Amraee T., Decision making under uncertainty in energy systems: State of the art. Renewable and Sustainable Energy Reviews, pp. 376–384, (2013).

[30] Boisvert, R.F., Howe, S.E. and Kahaner, D.K., 1985. GAMS: A framework for the management of scientific software. ACM Transactions on Mathematical Software (TOMS), 11(4), pp.313-355.

[31] Amazon EC2 spot instances (2017) [online]. Website https://aws.amazon.com/ec2/spot/.

[32] Gharehpasha, S. and Masdari, M., 2021. A discrete chaotic multi-objective SCA-ALO optimization algorithm for an optimal virtual machine placement in cloud data center. Journal of Ambient Intelligence and Humanized Computing, 12(10), pp.9323-9339.