Future Generation Computer Systems 113 (2020) 391–406

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Resource management in the federated cloud environment using Cournot and Bertrand competitions

Neda Khorasani [a], Saeid Abrishami [a,*], Mehdi Feizi [b], Mahdi Abolfazli Esfahani [c], Faeze Ramezani [a]

[a] Department of Computer Engineering, Ferdowsi University of Mashhad, Iran
[b] Department of Economics, Ferdowsi University of Mashhad, Iran
[c] Department of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 14 September 2019
Received in revised form 22 June 2020
Accepted 4 July 2020
Available online 14 July 2020

Keywords:
Federated cloud
Resource management
Game-theory
Cournot game
Bertrand game

ABSTRACT

Cloud federation helps cloud providers to scale up by renting resources from other providers when the workload increased. Moreover, cloud providers with idle or underutilized resources can sell their resources to others and earn revenue inside a cloud federation. One of the most critical problems in the federated cloud environment is the management of resources and cloud providers. The game theory seems to be an excellent way to model cloud federation. This paper introduces a new model for resource management between cloud providers in a centralized federated cloud environment, based on the well known Cournot and Bertrand games. To resolve the problem of heterogeneous resources, a physical resource unit which has specific computational features and can be shared between cloud providers is introduced in this paper. Besides, by introducing a new revenue-sharing approach between providers, this paper increased the collaboration of different providers. This model is implemented by the federated Cloudsim tool, and experiments show that the Cournot model outperforms others in terms of overall benefit and responsiveness. In other words, Cournot model can respond to an acceptable number of requests while making more profit. Besides, the reviews show that the proposed model works better than other methods in terms of time.

1. Introduction

Cloud computing is a flexible model for sharing distributed resources and services to the customer anywhere and anytime. In recent years cloud computing becomes more popular because of its scalable pool of resources and services for users, which gives users the ability to control their usage by themselves and only pay for the resources they used from the cloud [1]. Cloud computing is desirable for businesses with the possibility of outsourcing some of their operations, which results in a significant decrease in the cost of physical infrastructures. There are three types of cloud services which are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [2], and this paper focuses on IaaS model.

Because of increasing demands for cloud resources, in some cases, cloud providers cannot satisfy the requirements of their users alone. So, relying on a single cloud provider may prevent users from getting resources with high-quality whenever they need [3]. In this situation, cloud federation helps cloud providers to scale up by renting resources from other providers when the workload is increased. Moreover, cloud providers with idle or underutilized resources can sell their resources to others and earn revenue. The use of cloud federation can improve the quality of service, cost–benefit, and reliability of cloud providers [4]. There are two types of cloud federation: centralized and Peer-to-Peer. The centralized cloud federation has a central entity that manages operations between cloud providers; On the other hand, in the Peer-to-Peer model, cloud providers communicate and negotiate with each other directly [3]. In the centralized cloud federation, a third-party must manage the resources; The challenge is if this third-party fails, the whole federation fails. In the decentralized model, there is no single point of failure problem, but this model is more complicated than the centralized model. In the federated cloud, resource providers must communicate with each other. Besides, any cloud provider has its own methods and interfaces for providing various services. So, we need a third-party to facilitate communication between them. In this paper, a cloud federation manager as a third-party is responsible for managing and making communicate between cloud providers in the federation.

One of the most critical problems in the federated cloud environment is the management of resources and cloud providers.

* Corresponding author.
E-mail addresses: neda.khorasani@mail.um.ac.ir (N. Khorasani), s-abrishami@um.ac.ir (S. Abrishami), feizi@um.ac.ir (M. Feizi), mahdi001@ntu.edu.sg (M.A. Esfahani), fa.ramezani@mail.um.ac.ir (F. Ramezani).

https://doi.org/10.1016/j.future.2020.07.010
0167-739X/

Many studies have been done in this field, and they have introduced different models and algorithms for this problem [5–8].

Game-theory introduces a framework for situations of conflicting interests and has attracted lots of researchers in the area of cloud computing in recent years [9,10]. Game-theory is very suitable for the situation that we have the number of independent players that their decision affected by interactions of others' behavior with their own. In the federated cloud environment, several cloud providers need to decide how to share their resources in the federation so that they can both increase their own profits and avoid a lack of resources. The number and price of other providers' resources are also crucial to this decision. For this reason, game theory seems to be a good way to model cloud federation. Many studies have modeled cloud computing as a market and used game theory models in order to increase the profits of cloud providers, etc. Game theory-based resource management mechanisms have received a considerable amount of attention in the literature on federated cloud computing [11–15].

This paper introduces a new model for resource sharing and pricing between cloud providers in a centralized federated cloud environment, based on the well known Cournot and Bertrand games. In the Cournot game, providers compete for the quantity of their resources; on the other hand, in the Bertrand game, they compete for the price of their resources. The lack of existing a unique instance of virtual machines for switching between cloud providers has made it difficult for anyone to implement Cournot and Bertrand games in the federated cloud environment so far. To resolve this problem, this paper introduces a physical resource unit which has specific computational features and can be shared between cloud providers. Buyers can request for these physical units and then run any arbitrary virtual machine on top of these units. In this model, a time interval is defined so that cloud providers can manage and plan their resources during each time interval. Games are executed at the beginning of these intervals, and at the end of each time interval, the income of the federation has been divided between cloud providers based on the ratio of their shared resources to the total resources. This revenue-sharing model encourages cloud providers to cooperate in the federation because they earn revenue even if the federation does not use their resources. On the other hand, providers can sell their resources in the spot market and, when federation needed, retake the spot resources and provide them to the federation. Therefore, there is no reason for providers not to be honest about their free resources. The contributions of this study can be summarized as:

- Introducing the concept of the computational unit to share physical resources instead of virtual machines for eliminating the complexities of management of various virtual machine instances.
- Resource sharing based on a time interval to help cloud providers to plan their needs better and taking advantage of the difference between the time zone of cloud providers.
- Using game theory for resource discovery and pricing at the beginning of each time interval. So, during the time interval, the number and the price of federation resources are clear.
- Encourage cloud providers to cooperate with federation by introducing a revenue-sharing model between them at the end of any time interval, no matter how much of their resources have been sold.

The rest of this paper is organized as follows: Section 2 presents related works about cloud federation models. Section 3 reviews the system model of the proposed cloud federation. Section 4 elaborates on the proposed algorithms to solve the problem. Section 5 shows simulations and results to demonstrate the effectiveness of the proposed approach. Conclusion and future work are provided in Section 6.

2. Related work

Cloud federation has raised in recent years, and many studies have published on this topic. This section reviews related works in resource management in the federated cloud environments. Researches in this field were divided into two types: (1) formation of the federation and (2) management of resources in the federation. Resource management contains four parts, including resource discovery, selection, pricing, and allocation [4]. At each section, first, we review traditional methods, and afterward, game-theory based researches will be reviewed. Their benefits and limitations and comparison with this paper will also be discussed in detail.

2.1. Federation formation

In recent years, several studies have been done on the formation of the federation; for example, Ray et al. [16] tried to form a federation based on Multi-objective optimization problem that maximizes the profit of federation and also tries to create a balance between QoS and profit of members. Cloud providers send their requests to the federation broker with the name of the preferred service provider, and then the broker tries to select the best response to these requests. In Habibi et al. [17], similar to [16], cloud providers send their request to the federated broker, and then the federated broker distributes received requests among federation members based on different objective functions. Xu and Palanisamy [18] proposed a contracts-based resource sharing model for geo-distributed federated clouds. The proposed model permits service providers to settle resource sharing contracts and apply a cost-aware job scheduling for pre-defined time intervals during 24 h of a day. In our proposed method, similar to these researches, cloud providers send their requests to the third-party in the federation. But, the proposed method does not waste time finding the best resources, because the suitable resources are located and managed before, at the beginning of time intervals.

Chauhan et al. [19] proposed and described an overview of a central cloud broker in the federated cloud environments. However, they left elaboration on this research in more detail as the future works. The proposed broker is responsible for resource discovery, provisioning, scheduling, monitoring, cost estimation, and resource information service. The described broker in this work acts like the federation manager in our proposed model. These third-parties manages activities in the federation. It is essential to mention that this study has no implementation, and it just presents an idea. The only similarity of this work and ours is in the broker. For example, they did not discuss about time intervals or revenue sharing.

Some studies only focused on the game-theory for the formation of a federation. Some of these studies aimed at maximizing profit. For example, Mashayekhy et al. [2] introduced a formation game that considers the cooperation of the cloud providers in offering cloud IaaS services. The proposed model enables cloud providers to form a cloud federation by maximizing their profit. The aim of this research is the same as our model, but they have no time interval, and their revenue sharing is different from ours.

In addition to maximizing profits, Hassan et al. [20] paid attention to minimizing penalties. They proposed a formation mechanism by utilizing a trust-based cooperative game theory, which enables the cloud providers to dynamically form a federation based on profit maximization and penalty cost minimization. Minimizing penalty cost, with a trust model, helps to find the most reliable cloud providers. However, in some cases, a cloud provider that has agreed to deliver resources, rejects requests and denies to provide resources to the federation. Mashayekhy et al. [21] introduced a coalitional graph game for forming a federation of cloud providers with a high reputation. They guarantee

SLA based on past direct interaction between cloud providers. In contrast, in our proposed method resources are shared at the beginning of each time interval, so there is no penalty for not fulfilling a commitment. As a result, there are no concepts of reliability and reputation in our model.

Halabi et al. [22] used the hedonic coalitional game to create a federation based on the security level of cloud providers. In their model, Cloud providers are selfish players that try to form a federation, and they want to join federation based on the security risk levels and reputations of members. Besides, Ray et al. [23] used a hedonic coalition game to form a federation of trusted cloud providers, which restricts the untrusted cloud provider to be a member of the federation. Their research aims to maximize the overall profit and QoS, while maximizing the satisfaction level of each individual cloud provider. In their next article, Ray et al. [24] also consider the QoS violation in VM migration in the federation. Unlike these articles, security and trust are not one of our goals in the proposed method.

Chen et al. [25] studied the challenges of joint vertical and horizontal cloud federation (VHCF). They proposed a new game to model the cooperation and competition between clouds in the VHDF environment. They found optimal (equilibrium) workload factoring strategies of different coalitions by computing the least pessimistic core as a bilevel optimization problem. Based on this, they introduced the most stable coalition structure and payoff division.

Generally, in the mentioned studies, a provider earns revenue when it sells its resources to the other providers. Therefore if a provider shares its extra resources but cannot sells them, it earns nothing. This problem affects the cooperation of the providers with the federation. In our proposed model, the providers earn revenue by sharing resources in the federation, even their shared resources are not sold. Moreover, in the proposed mechanism, providers can sell their shared resources in the spot market and then retake whenever the federation requires more resources. So, there is no reason for providers to not cooperate in the federation.

2.2. Resource discovery and selection

In the field of resource discovery and selection, Saravanan et al. [6] proposed two different ranking mechanisms with an enhanced broker to help the customer to select the best service provider. They introduced a distributed ranking algorithm and sorted cloud providers based on their considered attributes [26]. Afterward, the probability-based Bayesian network ranking model is applied to rank the cloud service providers based on the actual value of the Service Measurement Index (SMI) attributes [27,28]. Ray et al. [29] also proposed a multi-criteria decision analysis for selecting the best federation based on three non-functional parameters, including price, Quality, and trust of federation due to different time periods.

Besides, Messina et al. [30] have proposed a fully decentralized trust-based approach for large-scale federations. Their method chooses the most encouraging collaborators, based on both direct and indirect reliable connections of a collaborator with others. However, Thomas and Chandrasekaran [31] selected the best partner in a cloud federation by ranking different cloud providers based on the Analytic Hierarchy Process (AHP) and considering the trust values of various cloud providers and user requirements. To solve the challenges of mapping applications to suitable resources based on their requirements, P. Wright et al. [7] proposed a two-phase constraints-based model to discover and select the most appropriate resources.

Some studies were done on discovering and selecting resources by using game-theory; for example, Do et al. [15] proposed a two-stage game for heterogeneous markets. In the first stage, Cloud providers play non-cooperatively to sell their services with the price that gains maximum revenue for them. Then, rational users can see the list of resources of each cloud provider and their prices; and then decide to select resources from a cloud provider that have the best performance and price.

In the mentioned approaches, a provider discovers resources by itself or with the help of a broker; when it is faced, or predicts the lack of resources. However, our method benefits from the definition of time intervals and runs game models at the beginning of each time interval, so no time is wasted to explore and select resources during a time interval.

Furthermore, one of the main challenges of resource selection is the heterogeneity of resources because different resource providers offer different virtual machines. Some researches consider only one type of virtual machine in their model for addressing this challenge. But, in our model, we have solved this problem by defining the concept of 'physical computational unit' which is the basis of resource sharing between providers. Afterward, the providers can sell arbitrary virtual machines to their clients by using the right number of computational units.

2.3. Resource pricing

In the field of resource pricing, Hongxing et al. [5] designed a double-auction-based approach that helps cloud providers to maximize their individual-profits. Their approach achieves asymptotic optimality in the social welfare of the cloud federation. Toosi et al. [32] introduced a financial option-based market for cloud providers that enables them to gain profit from their reserved resources. They can prevent of Service Level Agreement (SLA) violation by trading resources from other cloud providers in the federated cloud environments. Salma Rebai et al. [33] have proposed a linear model for optimizing profit in the federation. In their study, providers determine the number and price of their extra VMs for certain time intervals. The differences between this study and ours are in the revenue sharing model and heterogeneous resources.

A data communication model has proposed by Das et al. [34], which utilizes the usage pattern, types of requests, and infrastructure expenses to introduce a cost calculation model that helps to increase resource utilization and profit by facilitating the resource allocation process. Unlike the above mentioned articles, we have used game theory to determine the price of resources.

Other studies such as Roh et al. [14] focus only on game-theory for resolving the resource pricing issues. They proposed a concave game for solving the pricing problem between geo-distributed application service providers. In order to solve the problem of heterogeneous resources, they assumed that all compute resources have finite-sizes, and they are divisible., Unlike this article, we have used the concept of the physical computational unit to solve this problem.

Samaan et al. [13] proposed an economic model that helps providers to sell unused resources to spot market. They have introduced a time period for predicting unused capacity. Every provider knows the amount of guaranteed demand for each time period and can evaluate the unused resources. After that, every provider sets the number of resources that wants to share in the spot market. Finally, every provider gives the remaining resources to the federation. For resolving the problem of heterogeneous resources, they used only one type of virtual machine. This work is similar to our work in many aspects, but, the main differences are in the revenue-sharing model and type of resources. In this work, revenue-sharing is based on the used resources, while our model is based on the shared resources. Moreover, for resolving the problem of heterogeneous resources, they bring a constraint to their work by using only one type of VM.

As same as resource discovery, in most researches, a lot of time is wasted on price negotiation. But in our work, with the definition of time intervals and running games at the beginning of each time interval, the price is set at the beginning of each time interval and hence there is no price negotiation during a time interval.

2.4. Resource allocation

In the area of allocating resources, Aral and Ovatman [8] proposed a heuristic algorithm based on sub-graph matching for resource allocation in the federated cloud. The aim of this algorithm is serving more users without sacrificing the Quality of Services (QoS), while decreasing the cost that users have to pay for individual VMs. Lee et al. [35] focused on the problem of resource competition in the Distributed Resource Allocation (DRA) federated cloud and proposed a resource allocation model. The recommended model groups tasks based on their communications and dependencies to reduce communication latency between cloud providers in the first step. Then a pricing strategy based on the degree of competition, and a marginal cost applies to choose the best cloud provider for outsourcing requests. In another research, Ma et al. [36] worked on the challenge of support and management of QoS in resource allocation. They have studied this challenge from two aspects. At the first one, they implemented a multi-QoS task allocation model for allocating user's tasks to VMs. In another, they proposed a model for resource migrating in the federated cloud environments by considering cooperative and competitive models.

In case of minimizing the cost of cloud providers while satisfying SLA, Ardagna et al. [37] proposed a model for resource allocation in the geographically distributed cloud sites. In this model, the workload is predicted, and when it is too heavy for one location, it is dynamically redirected to another site. In [38], Liu et al. focused on the cost of communication between cloud providers in the federated cloud. They proposed an efficient method for resource allocation that decreases repacking overhead and increases system throughput. Unlike the above mentioned works, our method uses game theory to solve the resource allocation problem.

There are different studies in the VM migration field but the cost of network traffic in this field attracted limited attention. Zhang et al. [39] worked on network-aware VM migration (NetVMM) problem in an overcommitted cloud and they try to minimize the network traffic cost. They also proposed a model [40] to monitor and control memory when multiple virtual machines compete for memory. They used a balloon driver for server consolidation to optimize the performance of memory-intensive and disk-intensive applications.

There are also many game-theory based studies in this field. Ardagna et al. [12] worked on the resource allocation problem for SaaS providers that used resources of multiple IaaS providers. They assumed that the strategy of each player depends on the strategy of other players, and they proposed a Generalized Nash Equilibrium game. In this game, the SaaS providers aim to minimize the cost of resources they buy from IaaS providers and also the penalty of request execution failures; and IaaS providers try to maximize their revenue. Unlike this article, our work is concentrated on the collaboration between IaaS providers.

Hassan et al. [41] proposed centralized and distributed price-based resource allocation algorithms that guarantee mutual benefits; in this manner, the cloud providers are encouraged to form a Horizontal Dynamic Cloud Federation (HDCF) platform. They have studied cooperative and non-cooperative games to analyze the interaction among cloud providers in an HDCF environment. Their model motivates low-cost providers to have more contributions to an HDCF platform. Besides, they have shown that a non-cooperative game does not affect maximizing various properties, such as investments. Later, Hassan et al. [11] have studied the resource management issue from another aspect. They focused on minimizing energy and encouraged cloud providers to cooperate by increasing their long-term revenue in the federated cloud. They proposed a coalition game that chooses the set of low-energy-cost cloud providers for resource allocation. They claim their method is stable and fair.

In summary, there are three main differences between our proposed method and most of the previous works. First, most of the activities (resource discovery, selection, pricing, and allocation) are done at the beginning of each time interval. Therefore, there is no need to waste time during the time interval for these activities. Second, resource sharing is done using the concept of physical computational units, which solves the problem of resource heterogeneity. And third, its revenue-sharing is based on the amount of shared resources, instead of the amount of used (sold) resources.

3. System model

This paper introduces a new centralized cloud federation model. In the centralized cloud federation environments, cloud providers send their required resources to the federation, and a third-party in the federation tries to discover available resources for these requests. Three roles are defined in the proposed federation model, which are seller, buyer, and federation manager. Sellers are cloud providers who share their extra resources, while the buyers are the cloud providers who request these resources. The federation manager is a third-party that manages all of the interactions between sellers and buyers and other operations in the federation.

Two main concepts are used in this federation environment, which can be described as below.

Time Interval (TI): *TI* is a period in which cloud providers share their extra resources or buy resources from the federation if required. At the beginning of each interval, sellers declare their extra resources according to their predicted workload and current free resources at a specific price. After that, buyers can send their requests to the federation, and the federation manager answers to these requests if possible. Buyers can use the purchased resources until the end of the time interval. The concept of time interval has two benefits. First, the federation can take advantage of time zone differences between cloud providers. When it is night for a cloud provider, it can behave as a seller, while at the same time, another provider can act as a buyer because it is daytime for it. The second benefit is that the providers can plan their needs for the whole time interval.

Computational Unit (CU): *CU* is a bundle of physical computational resources such as processing cores, RAM, and storage, which is sold and bought as a whole package among cloud providers. At the beginning of each *TI*, sellers must specify the number of their idle *CU*s, and the buyers can request for these *CU*s instead of virtual machines (*VM*). Buyers can run any number of *VMs* on this *CU*s according to their needs, during the time interval.

At the beginning of each *TI*, the federation manager predicts the supply and demand of *CU*s in the current interval according to the history of similar *TI*s. According to this prediction, it creates a demand curve, which represents the amount of *CU*s that buyers are willing to purchase at various prices (See Fig. 1). Furthermore, the federation manager predicts two other factors, Willingness To Pay (*WTP*) and Willingness To Accept (*WTA*). *WTP* is the maximum price that the buyers are willing to pay for each *CU*,

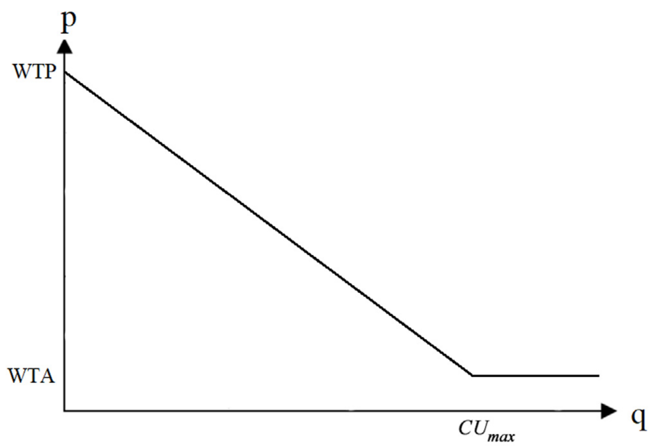


Fig. 1. The relation between price and demand.

and *WTA* is the minimum price that the sellers expect to receive for each *CU*. If the price reaches *WTP*, no one buys from the federation, and if it reaches below *WTA*, no one sells its resources to the federation.

During every *TI* four steps can happen in different orders: price determination, quota determination, resource allocation, and revenue sharing. These steps are elaborated in the following.

3.1. Price determination

The price of each physical resources, i.e. *CU*s, must be determined by the federation manager or sellers, at the beginning of each *TI*. This price will be called as the federation price ($Price^f$). Today, three types of resource pricing are usually used by the providers: Reserved, On-Demand, and Spot. Reserved instances are long-term contracts that are not related to the proposed model. Since providers typically sell their resources using the on-demand model, this paper considers it as the selling price for each provider. Therefore, if $Price_f$ exceeds the on-demand price for a buyer, then it has no motivation to buy expensive *CU*s. On the other hand, since the providers usually sell their extra resources as spot instances, sellers will not share their resources, if $Price^f$ falls below their predicted spot prices. Therefore, it is important to specify the maximum and minimum values for $Price^f$, and the federation manager can specify these values from Fig. 1. Assuming that *WTA* is the minimum spot price among sellers, and *WTP* is the maximum on-demand price among buyers, $Price^f$ satisfies:

$$WTA < Price^f < WTP \quad (1)$$

3.2. Quota determination

In this step, the federation manager must determine the number of *CU*s that each seller must share in the federation, which is called $CU_{quota}(i)$. To this aim, first, each seller i should specify the number of its extra *CU*s, which is called $CU_{extra}(i)$, according to its prediction of the users' demand in the current *TI* and its available resources. The sellers should note that if they encounter a lack of *CU*s during *TI*, they have to buy (their own) *CU*s from the federation at $Price^f$ (which is probably higher than their revenue for each *CU*). Therefore, they should predict their extra resources carefully. In addition, CU^f shows the total number of *CU*s ($\sum_{i=1}^{n} CU_{quota}(i)$) in the federation.

3.3. Resource allocation

During a *TI*, when a buyer needs resources, it sends a request to the federation manager. The federation manager knows the remaining quota of each seller and tries to allocate resources to the buyers according to some allocation algorithms, such as first-fit or best fit. Since some of the shared *CU*s may not be sold, the federation manager should allocate resources fairly according to the quota of each provider.

3.4. Revenue sharing

Two types of revenue are considered in this paper, which are expected and real revenue. The expected revenue (Rev^f_{expt}) is calculated by the federation manager at the beginning of each *TI*, according to its estimation of the buyer's demand, and $Price^f$ in the current *TI*. The real revenue (Rev^f_{real}) is calculated at the end of each *TI*, and demonstrates the income of the federation by selling *CU*s to the buyers. CU^f_{sold} and $CU_{sold}(i)$ show the number of sold resources by the federation and the ith seller, respectively. At the end of each *TI*, the federation manager should distribute Rev^f_{real} among the sellers.

In the next section, the proposed framework is introduced based on the concepts, roles, and steps in a federated environment studied in this section. The notations used in the paper are summarized in Table 1.

4. Proposed method

A large part of economic transactions takes place through markets, and firms have various strategic options to choose. In the most basic models, firms choose quantity or price as a strategic variable, and they are seen as quantity or price setters [42].

The objective of the proposed method is to encourage cloud providers to share their idle resources in the federation by introducing a new revenue-sharing approach. Moreover, this paper tries to decrease the time of resource discovery and pricing. To this aim, the basic marketing strategies can be used to model the federation. As previously mentioned, the extra resources of the providers are traded based on *CU*s in this paper, which makes all resources homogeneous. Therefore, we can apply the games of price and quantity with homogeneous resources (products) among cloud providers. In both games, the players are sellers, and it is assumed that they are rational and reliable, and all of them have the same constant marginal cost (c). In the first game, which is based on Cournot Competition, sellers compete on quantity and try to maximize their benefits by sharing the optimum amount of resources. In the second game, which is based on Bertrand competition, sellers compete on price to sell their idle resources. In this section, first, Cournot and Bertrand Games will be described, and then we will show how they can be applied to our system model.

4.1. Cournot game: Quantity competition

This section reviews the condition that firms determine the quantity, which is first analyzed by Cournot (1838) [42]. The price, p, determines the market situation, and therefore, it is equal to the reverse demand, $p = P(q)$, where q is the total quantity in the market, and P is the price function which is dependent on the total quantity. In the simple case of an oligopoly with linear demands, there is a homogeneous product market with n firms in which $firm_i$ sets its quantity q_i. So, the total quantity is $q = q_1 + q_2 + \cdots + q_n$. As previously mentioned, the price is calculated by linear reverse demand $p = P(q) = a - b \times q$ (with $a, b > 0$). Furthermore, it is assumed that all firms have the same

Table 1
Summary of notations.

Parameters	Description
a	The maximum price of CU in the federation (Equal to WTP)
b	The slope of the line (relation between price and the number of CUs)
c	The marginal cost of CUs
n	The number of seller in the federation
p	The price of resources
q	The quantity of resources
$\bar{q}$	The maximum number of firm's resources
TI	The Time Interval
CU	The computational unit
WTP	The maximum price of CU in the federation
WTA	The minimum price of CU in the federation
$Price^f$	The price of each CU in the federation
$Price_i$	The price of ith seller CUs in the Bertrand game
$CU_{extra}(i)$	The extra CUs of ith Seller
$CU_{quota}(i)$	The number of CUs that seller i shares in the federation
CU^f	Total number of CUs in the federation
CU_{max}	Maximum number of required CUs in the federation
$CU_{sold}(i)$	The number of sold CUs from ith seller in the federation
CU^f_{sold}	Total number of sold CUs by the federation
Rev^f_{expt}	The expected revenue of the federation
Rev^f_{real}	The real revenue of the federation
$Rev_{real}(i)$	The real revenue of ith seller from the federation
$Rev_{expt}(i)$	The expected revenue of ith seller from the federation
τ	The predicted workload
t	The actual workload

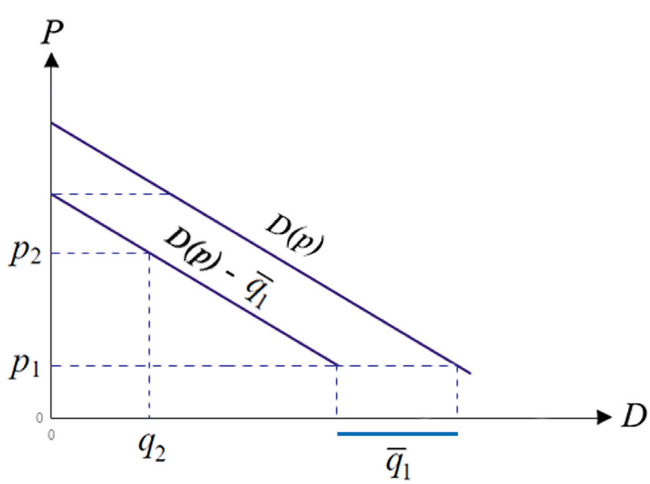


Fig. 2. Demand curve in Bertrand game with capacity constraint.

cost structure, and they have constant marginal cost, which is $c_i = c$ for all i. Now, the quantity produced by each firm at the Cournot equilibrium can be evaluated as [42]:

$$q_i^*(n) = \frac{a-c}{b(n+1)} \tag{2}$$

and the total quantity and price of the market are obtained as follow:

$$q^*(n) = \frac{n(a-c)}{b(n+1)} \tag{3}$$

$$p^*(n) = a - b \times q^*(n) = \frac{a+nc}{(n+1)} \tag{4}$$

In the equilibrium of this competition, all firms produce equal quantities.

4.1.1. Cournot competition in the federated cloud environment

In the Cournot Competition, sellers compete on the number of resources they will share, and the federation manager specifies the price of resources. After that, the federation manager knows the number and the price of idle resources for the current (TI). From now onward, buyers know the price of resources and just need to send their request for resources to the federation manager, and the federation manager answers to this request if there are enough resources in the federation.

Two different types of competitions are studied for this game. In the first competition, at the beginning of TI, the federation manager determines the optimum price of CUs based on the demand curve (Fig. 1), and then sellers specify the number of CUs that they will share in the federation. In the second competition, first, sellers specify the number of CUs that they want to share, and then, the federation manager determines the price based on the total CUs and demand curve. The formulation of this game can be described based on the following definitions:

- *Players:* Players of the game are sellers, i.e., providers which would like to share their extra resources with other providers.
- *Strategies:* The strategy of sellers is to maximize their benefits by sharing the appropriate number of CUs.
- *Payoffs:* Each seller earns from the revenue of federation based on the ratio of the number of its shared CUs to the total number of CUs shared in that TI.

4.1.2. Price-first competition

In this competition, at the beginning of TI, the federation manager evaluates the maximum revenue of selling CUs according to the history of TIs (demand curve in Fig. 1); By using this amount, federation can calculate $Price^f$; and by knowing this price, each seller determines the number of CUs that it intends to share with the buyers to maximize its profit. The four steps of the current TI are executed as follows:

4.1.2.1. Price determination. As previously mentioned, the federation manager predicts the demand curve (Fig. 1) of the current *TI*, based on the history of similar *TI*s. According to Fig. 1, the price of each *CU* in each *TI* can be calculated by Eq. (5):

$$Price^f = \begin{cases} a - b \times CU^f & \text{if } CU^f < CU_{max} \\ WTA & \text{otherwise} \end{cases} \tag{5}$$

where a is equal to *WTP* and b is the slope of the line which demonstrates the relation between price and the number of *CU*s and can be defined by Eq. (6):

$$b = \frac{WTP - WTA}{CU_{max}} \tag{6}$$

where CU_{max} is the number of resources that will not change $Price^f$ if the sellers share more than that. In fact, at the point of CU_{max}, the $Price^f$ is equal to the *WTA*. If we assume that in each *TI* all of the shared *CU*s will be sold out, the Rev^f_{expt} in *TI* can be calculated based on Eq. (7) as:

$$Rev^f_{expt} = Price^f \times CU^f \tag{7}$$

Now federation tries to estimate the optimal number of *CU*s by maximizing its income. Getting derivatives of Eq. (7) results in:

$$\frac{d}{d\,CU^f} \; Rev^f_{expt} = 0 \tag{8}$$

and after solving this equation we have:

$$CU^f = \frac{a}{2b} \tag{9}$$

It is essential to notice that the federation manager does not earn any benefits, and all of its income will be divided between the sellers. For this reason, the federation manager is not a player in this game. When the number of optimal resources is calculated, the federation price will be obtained based on Eq. (5).

4.1.2.2. Quota determination. Every seller determines the number of extra *CU*s that it wants to share and announces it to the federation manager. Having extra *CU*s of each seller i ($CU_{extra}(i)$), and the optimum number of *CU*s for the federation (CU^f), which is calculated by Eq. (9), the federation manager can specify the quota that each seller must share. For this reason, the federation manager tries to divide the number of required *CU*s among sellers equally. If a seller does not have enough resources to share in the federation, it shares all of its available resources, and its remaining quota will be divided among other sellers.

4.1.2.3. Resource allocation. This step is performed exactly as described previously in system model.

4.1.2.4. Revenue sharing. At the end of each *TI*, the federation manager must divide Rev^f_{real} among the sellers, in proportion to their quota, no matter how much of their quota has been sold in the current *TI*. The revenue of each seller i, $Rev_{real}(i)$, is determined by Eq. (10):

$$Rev_{real}(i) = Rev^f_{real} \times \frac{CU_{quota}(i)}{CU^f} - c \times CU_{sold}(i) \tag{10}$$

which makes cloud providers confident that if they share resources, they gain revenue (no matter their resources are used or not). Hence, this model encourages sellers to share their idle resources to the federation to earn income.

4.1.3. Quota-first competition

This competition is similar to the previous competition, except for the first two steps, which are swapped. In that sense, first, the sellers announce their number of extra *CU*s, and then the federation manager determines $Price^f$ for each *CU*, according to the total number of shared *CU*s. These two steps are elaborated in the following.

4.1.3.1. Quota determination. In this type of competition, sellers specify the number of *CU*s that they want to share in the first step. The sellers know the formulation of revenue sharing, which is defined in Eq. (10). Hence, they try to maximize their revenue by solving Eq. (11):

$$\frac{d}{d\,CU_{quota}(i)} \left(Rev^f_{expt} \times \frac{CU_{quota}(i)}{CU^f} - c \times CU_{quota}(i) \right) = 0 \tag{11}$$

After solving this equation, the quota of every seller will be cleared in the Nash equilibrium, and if this is greater than its available *CU*s, the seller must share current available *CU*s. In other words, the quota of every seller in the Nash equilibrium obtained from Eq. (12):

$$CU_{quota}(i) = \max\,\{\frac{a - c}{(n+1)b}\;,\; CU_{extra}(i)\} \tag{12}$$

4.1.3.2. Price determination. In this step, the federation manager knows the number of shared resources, and it can calculate the price of each *CU* for the current *TI*, according to Eq. (5).

4.2. Bertrand game: Price competition

This section reviews the Bertrand (1883) model where products are homogeneous, and firms have the same marginal cost, which is $c_i = c$. There are n firms, and all of them simultaneously try to set a price for maximizing their profit. The firm that sets a lower price will attract all demands $D(p)$, where p is the price. If all firms set the same price, demands will be divided between them equally [42]. The demand for the ith firm in this competition is as below:

$$D_i(p_i) = \begin{cases} D(p_i) & \text{if } p_i < p_j \\ \alpha_i D(p_i) & \text{if } p_i = p_j \\ 0 & \text{if } p_i > p_j \end{cases} \tag{13}$$

where for any (i, j) that satisfies ($0 < i, j < n$ *and* $i \neq j$), α_i is the ratio of the ith firm's production to other firms. In the equilibrium of this competition, all firms set their price equal to the marginal cost.

However, in the federated cloud environment, the number of resources of each cloud provider is limited. Therefore, we have considered a version of Bertrand competition with quantity constraints. Suppose there are two firms with homogeneous products but with limited production capacities (q_1 and q_2). If the first firm sets its price p_1 less than the other firm, i.e., $p_1 < p_2$, then the whole demand goes to the first firm. But if the first firm has not enough capacity, i.e., $\overline{q_1} < D(p_1)$ ($\overline{q_1}$ is the maximum number of 1th firm's product), the remaining demand is assigned to firm number 2 as following (see Fig. 2):

$$D_2(p_2) = \begin{cases} D(p_2) - \overline{q_1} & \text{if } D(p_2) > \overline{q_1} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

This competition can be extended to n firms, which will be applied to cloud federation in the next subsection.

4.2.1. Bertrand competition in the federated cloud environment

In the Bertrand competition, first, the sellers announce the number of their extra *CU*s to the federation manager, and the federation manager gives this information to the other sellers. Using this information, sellers can set the price for their *CU*s for the current *TI*. Now, the federation knows the quantity and price of extra *CU*s of the sellers at the beginning of the current *TI*. Once the federation manager receives a request from a buyer, it chooses the seller with the minimum price and enough available resources. The formulation of this game can be described based on the following definitions:

- *Players:* Players of the game are sellers, i.e., providers who would like to share their extra resources with other providers.
- *Strategies:* The strategy of the sellers is to set the price for their extra *CU*s, in order to maximize their benefits by selling them to the buyers.
- *Payoffs:* Each seller earns revenue from the federation based on the number of its *CU*s that are sold to the buyers in the current *TI*.

Every *TI* in this competition consists of four steps in the following order:

4.2.1.1. Quota determination. In this step, sellers predict their workloads and specify the number of extra *CU*s and announce it to the federation manager. In this game, the extra resources are equal to the quota of every sellers in the federation ($CU_{quota}(i) = CU_{extra}(i)$).

4.2.1.2. Price determination. Now, sellers must set the price for their *CU*s such that their revenue is maximized. The formulation of revenue can be calculated by Eq. (15):

$$Rev_{real}(i) = Price_i \times CU_{sold}(i) - c \times CU_{sold}(i) \tag{15}$$

where $Price_i$ is the price of *CU*s of the ith seller and $CU_{sold}(i)$ is the number of *CU*s that the ith seller can sell in the current *TI* with $Price_i$.

But the point is that any seller who announced a lower price could sell all of its resources sooner. So, a seller should offer a price in such that there will be demand for its resources. In the worst scenario, each seller assumes that all other sellers have suggested a lower price than it, and therefore their *CU*s will be sold sooner. In this case, the number of *CU*s that can be sold by the ith seller in the federation is obtained from Eq. (16).

$$CU_{sold}(i) = \begin{cases} D(Price_i) - \sum_{j=1}^{n} CU_{quota}(j) \\ \qquad \text{if } D(Price_i) > \sum_{j=1}^{n} CU_{quota}(j) \text{ and } i \neq j \\ 0 \qquad \text{otherwise} \end{cases} \tag{16}$$

and based on Fig. 1, $D(Price_i)$ is:

$$D(Price_i) = \frac{a - Price_i}{b} \tag{17}$$

where a is equal to *WTP* and b can be calculated by Eq. (6). So, the expected revenue of the ith seller that we name it as $Rev_{expt}(i)$ can be obtained from Eq. (18):

$$\begin{aligned} Rev_{expt}(i) = Price_i\left(\frac{a - Price_i}{b} - \sum_{j=1}^{n} CU_{quota}(j)\right) \\ - c\left(\frac{a - Price_i}{b} - \sum_{j=1}^{n} CU_{quota}(j)\right) \end{aligned} \tag{18}$$

Hence, sellers try to maximize their revenue by solving Eq. (19):

$$\frac{d}{d\, Price_i}\left(Rev_{expt}(i)\right) = 0 \tag{19}$$

and after solving this equation, the price of each seller will be cleared in the equilibrium and calculated based on Eq. (20):

$$Price_i = \frac{a - b\sum_{j=1}^{n} CU_{quota}(j) + c}{2} \tag{20}$$

4.2.1.3. Resource allocation. Now, the federation manager knows the amount and price of extra *CU*s of each seller for this period of *TI*. So, when a buyer sends a request to the federation manager, it just chooses *CU*s from a seller that has enough idle *CU*s with minimum price.

Table 2
The number of shared *CU*s by each seller in the federation.

	Seller 1	Seller 2	Seller 3	Seller 4	Seller 5	Seller 6
2 members	300	800	–	–	–	–
3 members	100	200	500	–	–	–
4 members	50	100	300	500	–	–
6 members	100	200	300	400	500	600

4.2.1.4. Revenue sharing. At the end of *TI*, every seller earns revenue from federation equal to the price of its *CU*s that are sold in the federation based on Eq. (15).

5. Experimental results

This section will present our evaluation methodology and simulation results to show the effectiveness of the proposed model for resource management in the federated cloud environment based on Couront and Bertrand games.

5.1. Experimental setup

To evaluate the proposed method, the federated cloud environment is simulated with the participation of two, three, four, and six independent cloud providers as sellers. This simulation was performed using the *FederatedCloudSim* tool that is a very flexible cloud simulation framework and can be used for different federated cloud scenarios [43]. Table 2, shows the average number of extra *CU*s of any seller when federation has two, three, four, and six members.

We conduct simulations using *FederatedCloudsim*'s workloads [44]. In order to be able to use these workloads, we have to integrate the workload traces and configure our scenario. Three different workloads are used in the experiments. The first and second workloads are for the duration of one month, and the third workload is for a period of two weeks. There are two types of evaluation in this paper. In the first one, one type of workload is used, and the number of providers is changed. In the second type, the number of providers is fixed, and experiments are performed by changing workloads.

The maximum price of the federation (*WTP*) is decided based on the price of resources of Amazon [45], Microsoft [46], and Google [47] cloud service providers; We set *WTP* as the price that was higher than the others. The minimum price (*WTA*) of *CU*s has also been set based on the minimum spot price of *VM*s offered by Amazon EC2 [45] in every *TI*. It is important to mention that since the proposed model simultaneously estimates the price and quantity for the sellers, using resources from different cloud services does not affect our system, and our system is only dependent on the *WTP* and *WTA* of the cloud services. In fact, we use WTP and WTA as an upper and lower price limit in our federation. For example, WTP is higher than any provider's prices in our federation, and if the federation price becomes more than WTP, demands become zero.

At the beginning of each *TI*, we need to predict the client's workload. There are many workload prediction and forecasting methods for cloud environment [48]. Among these methods, we have decided to use the simple method of Exponential Moving Average (EMA). EMA is computed using the following formula:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n \tag{21}$$

where τ_{n+1} is the predicted workload for next *TI*. t_n and τ_n are actual and predicted workload of last similar *TI*, respectively, and α is the smoothing factor that we set it to $(\frac{1}{2})$. Experiments show that predicting input workload with EMA has approximately 7% relative prediction error and a mean absolute prediction error of

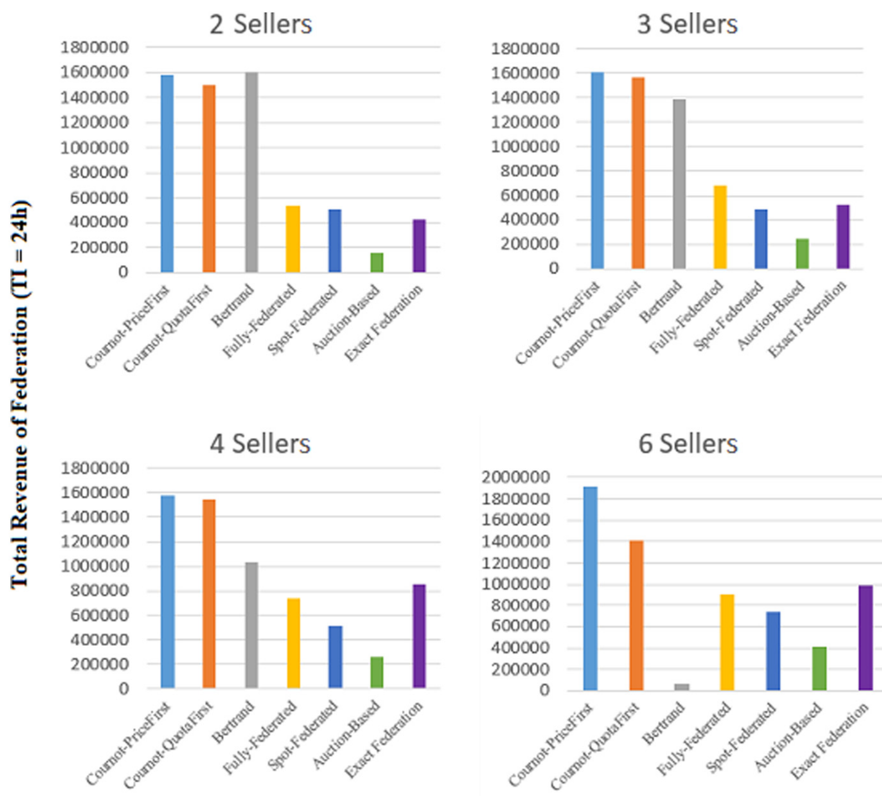


Fig. 3. The total revenue of the federation with two, three, four and six members and the Time Interval equals to 24 h.

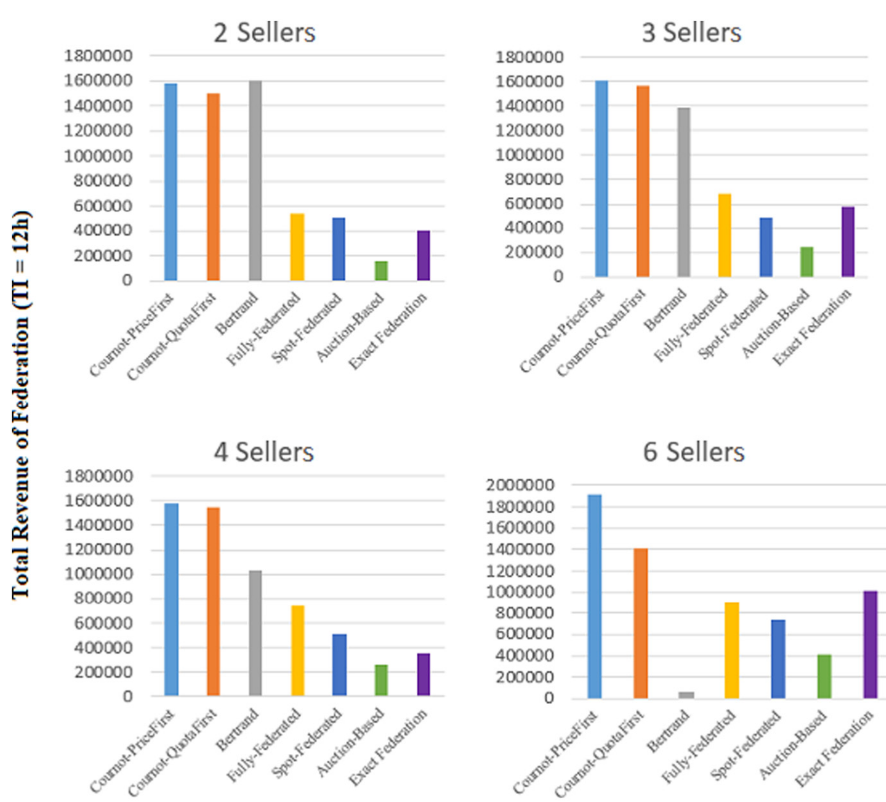


Fig. 4. The total revenue of the federation with two, three, four and six members and the Time Interval equals to 12 h.

2, which is acceptable. Having a robust linear model for predicting workloads can assist the proposed system in performing and generalizing well in the presence of different workloads. It is possible to use machine learning techniques to achieve a better workload prediction, but machine learning techniques usually do not have a high generalization ability and cannot perform well on

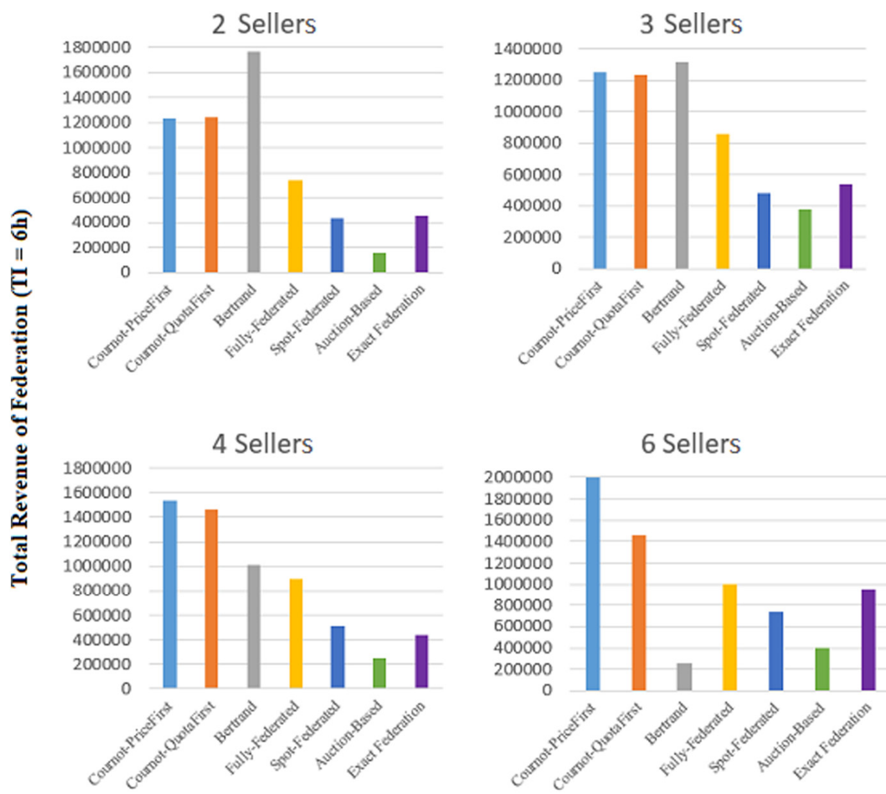


Fig. 5. The total revenue of the federation with two, three, four and six members and the Time Interval equals to 6 h.

Table 3
Comparison of various techniques in predicting workload.

Method	Mean absolute error
Bayesian Ridge Regression [49]	**1.483**
K-nearest neighbor regression [50]	28.471
Support vector regression [51]	41.363
Kernel ridge regression [52]	1.511
Decision tree regressor [53]	26.185
Exponential Moving Average (EMA)	2.00

different workloads. We have trained and tested various machine learning techniques, as shown in Table 3. In order to train machine learning methods, we have assumed that the system knows the exact workload in the last seven time intervals and wants to predict the workload for the next state. The models are trained on half of the workload and evaluated in the second half. As shown in Table 3, Bayesian Ridge Regression can achieve the lowest mean absolute error compared to others. It can be seen that the result of using EMA is acceptable compared to the others, and the error is not significant. EMA can work in different workloads without any training, which results in good generalization ability. On the other hand, a model trained on one workload, via machine learning approaches, cannot perform well on another one. Hence, we suggest using EMA when there is no prior information about the workload, and when the workload might change a lot over time. Nevertheless, Bayesian Ridge Regression can be used when a long history of workload is available to train a robust model.

The proposed model is compared with Samaan et al. [13], Xu et al. [18] and Rebai et al. [33] models. Samaan et al. have presented two models in their paper, which are: (1) *Fully − Federated*, that providers share all of the unused resources to the federation, and (2) *Spot − Federated*, that first providers set the number of spot resources and then share the remaining resources to the federation. Because *Auction − Based* methods are very common in the federated cloud resource management field, we also choose an auction-based model which is proposed by Xu et al. which is very close to ours because they have used the concept of time interval. The last model, which is called *Exact − Federation*, is proposed by Rebai et al. and it uses an integer linear programming formulation to optimize the profit in a cloud federation. The analysis of the results comes in the next subsection.

5.2. Analysis of results

Two models are considered for the input workload. In the first one, the workload is not changed base on price change, and so we have the same demands for each price of resources. In another model, workload changes based on the price of resources, and it is inversely related to price.

5.2.1. Fixed demands

In reality, the number of requests received by a cloud federation depends on the price of the federation. When the price is high, the number of requests reduces, and having a low price will increase the number of requests that a federation could have. If we consider a fixed workload that does not change based on the price, all providers just need to predict the input workload and share all of their unused resources at the highest price inside the federation. The federation will get the highest revenue, and all sellers would get high income.

5.2.2. Dynamic demands

In this model, we have two types of comparison. The first one shows when the workload is fixed, how the revenue varies with different number of providers, and the second one shows that when the number of providers is fixed, how the revenue varies with different workloads.

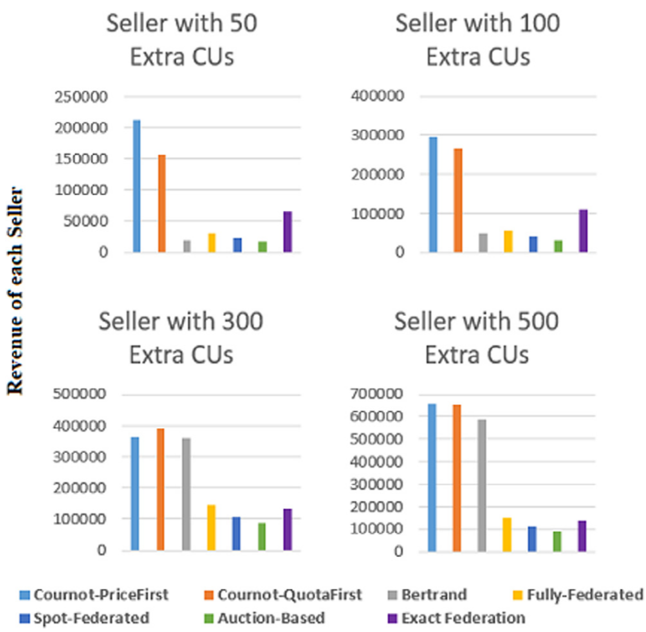


Fig. 6. The total revenue of each seller in the federation with four members after a month when the Time Interval was 6 h.

5.2.2.1. *One workload — various number of sellers.* In Figs. 3–5, we compare the total profit of the federation obtained by two models of Cournot competition (including *Price — First* and *Quota — First*) and Bertrand competition with the *Fully — Federated*, *Spot — Federated*, *Auction — Based* and *Exact — Federation* models. In all cases, Cournot competition yields a higher profit from the other models of federation. But, in the Bertrand game, when the number of cloud providers increases, and the difference between the number of their resources decreases, the profit drops significantly. This happens because in the Bertrand game the price is set by the sellers themselves, based on the total number of *CU*s of the other sellers. So, as the resources of other members increase, each seller tries to set a lower price to increase the probability of selling its resources. This will reduce the price of the federation and as a result the total revenue of the federation is decreased. But, the result of this low price is the higher demands, and since the sellers share all of their extra *CU*s, the federation manager can respond to more requests.

Figs. 6 and 7 show the individual revenue of each cloud provider in the federation with four and six sellers. The charts show that in the two models of Cournot competition, the individual revenue of each seller is more than the revenue of the same seller in the other models. In the Cournot competitions, sellers have benefited from the proportion of their shared *CU*s, and this is not related to the use of their *CU*s in the federation (sellers earn revenue based on the amount of the resources they have shared). However, in the *fully* and *spot* federation models, each seller gets revenue from the federation based on the number of *CU*s it has shared and is used in the federation. In the *Auction — Based* model, sellers earn money based on a pre-agreed price for the number of resources they provide to the buyers. The pre-agreed price is based on the buyers' bid, and therefore it is usually low, and for this reason, they earn lower revenue in comparison to our model. In the *Exact — Federation* model, each seller specifies the price of its resources itself. This price depends on the seller's extra *CU*s but their revenues depend on their sold *CU*s. For this reason, the seller with less *CU* defines a higher price. Furthermore, buyers preferred to outsource their requests to the sellers who have lower prices and sufficient resources. So, sellers with more resources can sell more resources with fewer prices, and sellers with fewer resources can sell fewer resources at higher prices. It can be assumed that the total revenue distributes between all sellers and so they earn less revenue compared to our model. As previously mentioned, in the Bertrand game the price is set by the sellers themselves, based on the total number of *CU*s of the other sellers. If the predicted workload exceeds all other seller's *CU*s, a seller with fewer extra *CU*s, tries to bid the lowest price (in order to increase the probability of selling its resources); In the same way, other sellers determine the price of their *CU*s, and eventually, the one with more resources sets a higher price. Otherwise, all sellers set the lowest price for their *CU*s in order to be able to sell their resources. As can be seen, in a federation with four members, with the presence of higher number of requests compared to the number of available *CU*s, the seller who has more extra *CU*s sets the higher price and earns more revenue; and the seller with fewer extra *CU*s sets the lower price and earns very little revenue. But in a six-member federation, where the number of *CU*s is more than the number of requests, all sellers set the lowest price. As a result, sellers' revenue has dropped dramatically.

Figs. 8–10 show the percentage of accepted requests to all available extra *CU*s in the federation. In the Bertrand game, where sellers share all their extra *CU*s in the federation with a low price, the demand is high and is close to the available extra *CU*s in the federation. So, a high percentage of requests are processed. Furthermore, we see that the *Auction — Based* model also responds to a high percentage of requests because of the pre-agreed low price. But, in the *Fully* and *Spot* federation, because of the high federation price, the amount of requests decreases and so the percentage of accepted requests is lower than the others. In the *Exact — Federation* model, the price of *CU*s is not very high and for this reason, the amount of requests does not drop down significantly. The percentage of accepted requests in the Cournot game is acceptable based on the price of resources. In other words, generally, we can see Cournot can make a balance between revenue and response rates; while it is responding to an acceptable amount of requests, it earns more revenue than the other models.

5.2.2.2. *Fixed number of sellers — various workloads.* In Figs. 11 and 12, we have evaluated our model for four sellers with three different workloads [44]. In Fig. 11, the total revenue of federation is compared between our models and *fully* and *spot* federated, *Auction — Based* and *Exact — Federation* models for three workloads. As can be seen, the proposed model can achieve a higher revenue compared to the others. Fig. 12 also shows the percentage of accepted requests to the available extra *CU*s in the federation.

Furthermore, since the proposed algorithm identifies the available resources and their prices at the beginning of each *TI*, therefore, when a request is sent to the federation, no time is spent discovering and pricing resources. So we can claim that our proposed method has a better response time to the requests than the other methods.

6. Conclusions and future work

This paper proposed a resource management model in the cloud federation environment to address some of the challenges in this area. By introducing physical units instead of virtual machines, we have unified the different types of provider's resources and based on this homogenization we have been able to introduce a solution to increase the profits, the number of shared resources and the number of accepted requests in the federation based on the Cournot and Bertrand games.

The proposed approach identifies the number of shared resources and their prices at the beginning of each time interval.

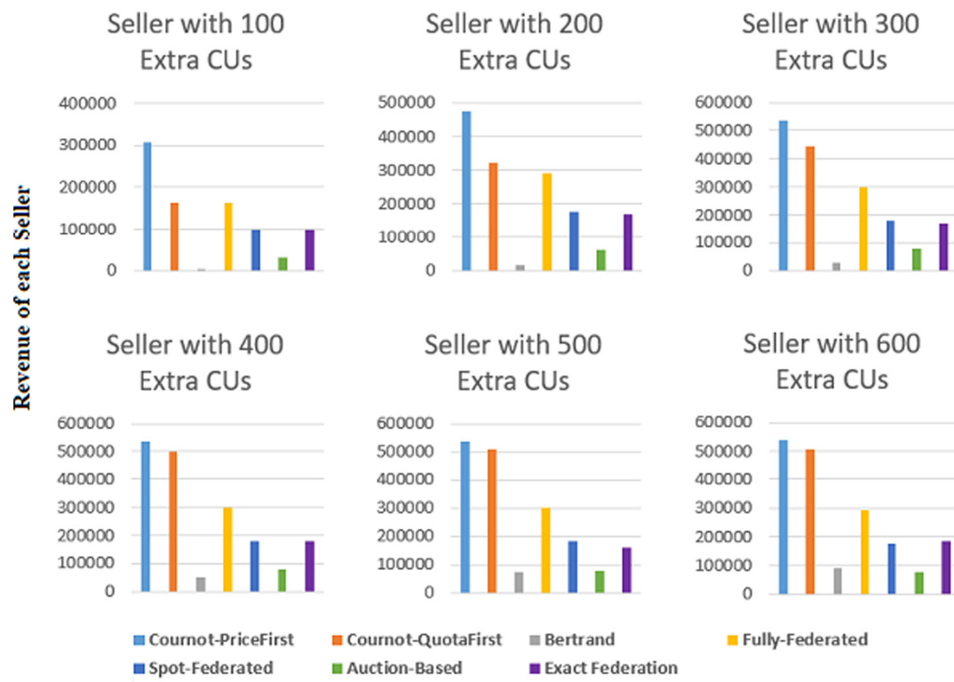


Fig. 7. The total revenue of each seller in the federation with six members after a month when the Time Interval was 6 h.

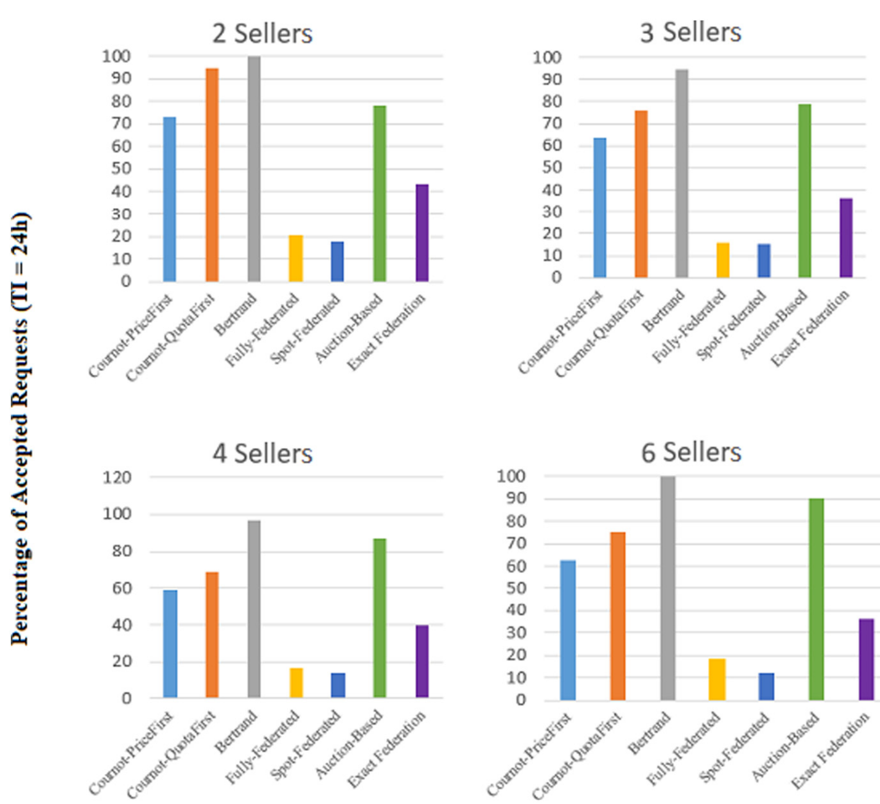


Fig. 8. The percentage of accepted requests in the federation with two, three, four and six members when Time Interval was 24 h.

Hence, when a request is received, the proposed model does not spend time for discovering and pricing resources. Therefore, the proposed model works better than other methods in terms of time. In addition, the results show that in our proposed model, providers can respond more favorably to requests as well and earn more revenue.

For future work, we can include probabilistic models in our approach. For example, when sellers are deciding on the number of resources that they want to share in the federation, they can consider the likelihood of the wrong prediction in their computation. This probability helps sellers as a risk factor, and they can risk in some situations, based on the state of their resources and

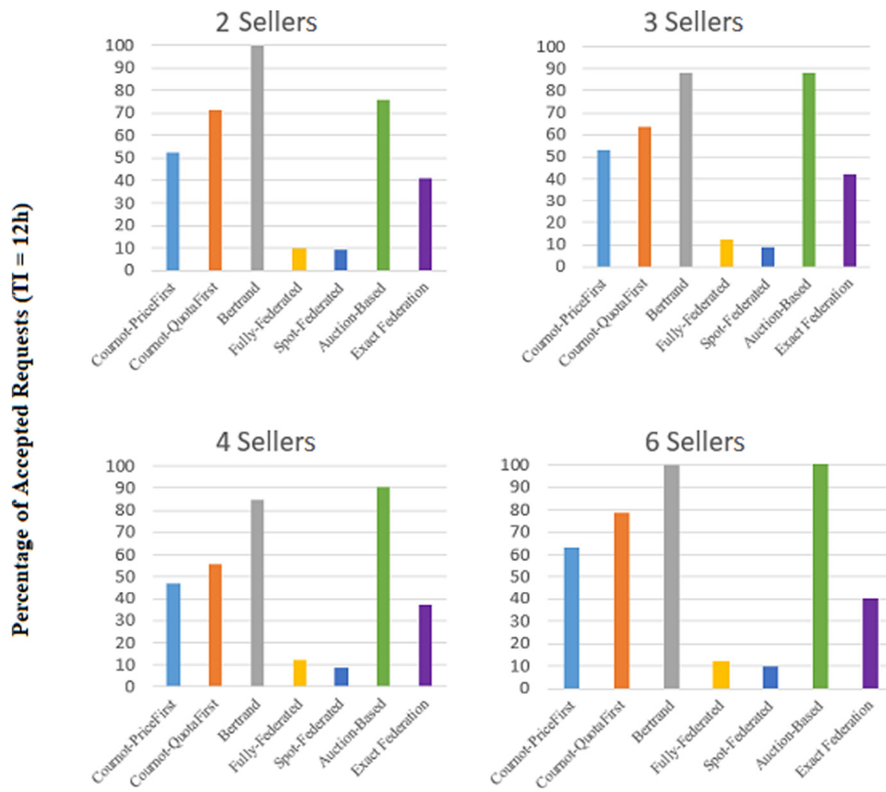


Fig. 9. The percentage of accepted requests in the federation with two, three, four and six members when Time Interval was 12 h.

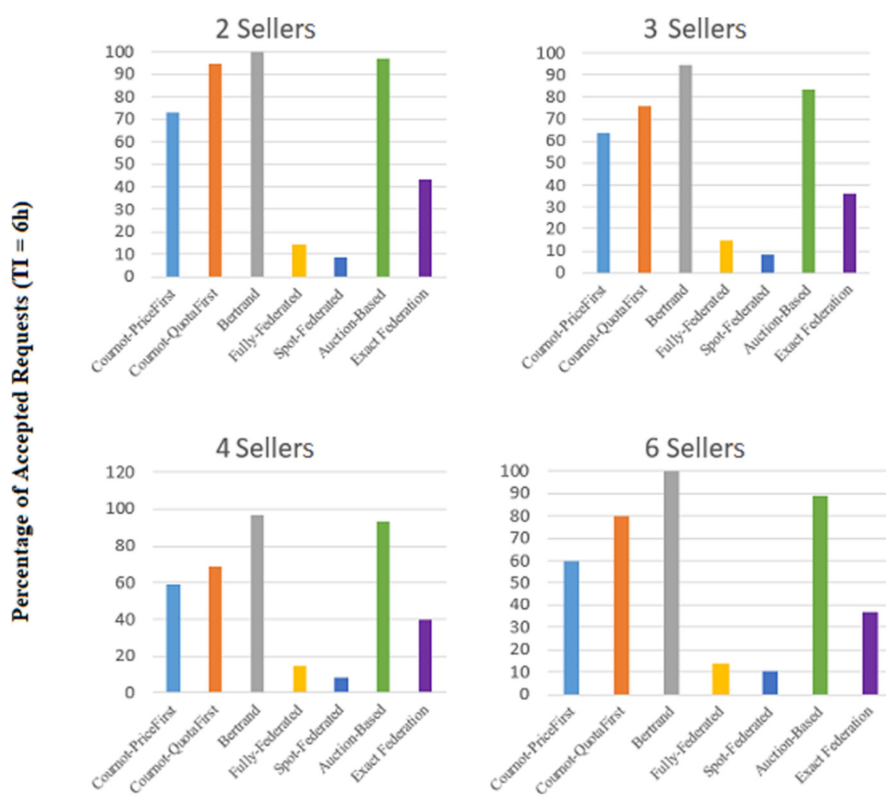


Fig. 10. The percentage of accepted requests in the federation with two, three, four and six members when Time Interval was 6 h.

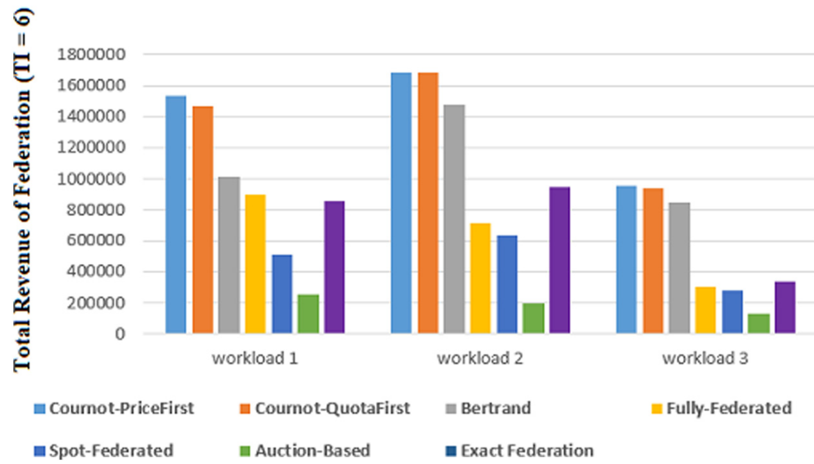


Fig. 11. The total revenue of the federation with three workloads when Time Interval was 6 h.

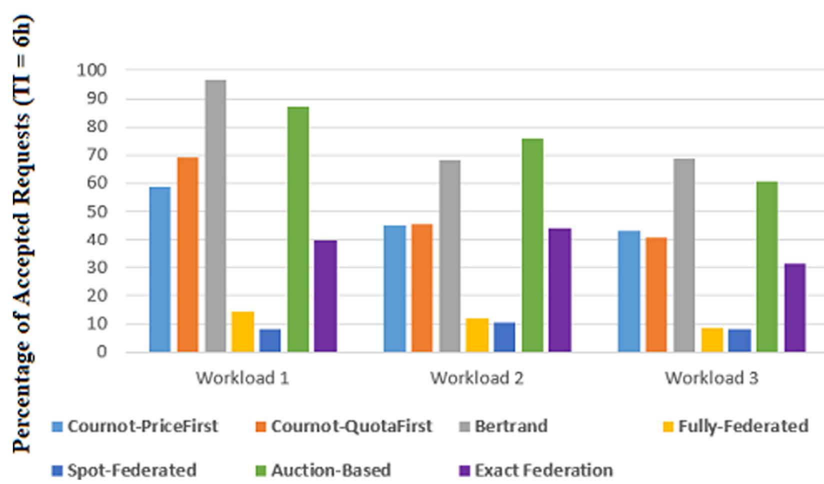


Fig. 12. The percentage of accepted requests in the federation with three different workloads when Time Interval was 6 h.

workload history, and earn more revenue by sharing resources more than their available extra resources. Furthermore, in the Cournot competition, the revenue of sellers is not related to the number of their sold resources. Hence, it is possible to implement a more accurate algorithm for allocating resource fairer.

CRediT authorship contribution statement

Neda Khorasani: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft, Visualization. **Saeid Abrishami:** Supervision, Writing - review & editing, Conceptualization, Methodology, Validation. **Mehdi Feizi:** Writing - review & editing, Conceptualization, Methodology, Validation. **Mahdi Abolfazli Esfahani:** Writing - review & editing. **Faeze Ramezani:** Conceptualization, Software.

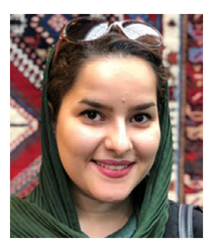
Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.D. Josep, R. Katz, A. Konwinski, L. Gunho, D. Patterson, A. Rabkin, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.
[2] L. Mashayekhy, M.M. Nejad, D. Grosu, Cloud federations in the sky: Formation game and mechanism, IEEE Trans. Cloud Comput. 3 (1) (2014) 14–27.
[3] N. Grozev, R. Buyya, Inter-cloud architectures and application brokering: Taxonomy and survey, Softw. - Pract. Exp. 44 (3) (2014) 369–390.
[4] M. Liaqat, V. Chang, A. Gani, S.H. Ab Hamid, M. Toseef, U. Shoaib, R.L. Ali, Federated cloud resource management: Review and discussion, J. Netw. Comput. Appl. 77 (2017) 87–105.
[5] H. Li, C. Wu, Z. Li, F.C. Lau, Virtual machine trading in a federation of clouds: Individual profit and social welfare maximization, IEEE/ACM Trans. Netw. 24 (3) (2016) 1827–1840.
[6] M. Saravanan, M. Aramudhan, S.S. Pandiyan, T. Avudaiappan, Priority based prediction mechanism for ranking providers in federated cloud architecture, Cluster Comput. (2018) 1–9.
[7] P. Wright, Y.L. Sun, T. Harmer, A. Keenan, A. Stewart, R. Perrott, A constraints-based resource discovery model for multi-provider cloud environments, J. Cloud Comput.: Adv. Syst. Appl. 1 (1) (2012) 6.
[8] A. Aral, T. Ovatman, Subgraph matching for resource allocation in the federated cloud environment, in: IEEE International Conference on Cloud Computing, IEEE, 2015, pp. 1033–1036.
[9] M. Taghavi, J. Bentahar, H. Otrok, Two-stage game theoretical framework for IaaS market share dynamics, Future Gener. Comput. Syst. 102 (2020) 173–189.
[10] B. Shen, Y. Shen, W. Ji, Profit optimization in service-oriented data market: A Stackelberg game approach, Future Gener. Comput. Syst. 95 (2019) 17–25.
[11] M.M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, B. Song, A. Alamri, Energy-aware resource and revenue management in federated cloud: A game-theoretic approach, IEEE Syst. J. 11 (2) (2015) 951–961.

[12] D. Ardagna, M. Ciavotta, M. Passacantando, Generalized nash equilibria for the service provisioning problem in multi-cloud systems, IEEE Trans. Serv. Comput. 10 (3) (2015) 381–395.
[13] N. Samaan, A novel economic sharing model in a federation of selfish cloud providers, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2013) 12–21.
[14] H. Roh, C. Jung, W. Lee, D.-Z. Du, Resource pricing game in geo-distributed clouds, in: 2013 Proceedings IEEE INFOCOM, IEEE, 2013, pp. 1519–1527.
[15] C.T. Do, N.H. Tran, E.-N. Huh, C.S. Hong, D. Niyato, Z. Han, Dynamics of service selection and provider pricing game in heterogeneous cloud market, J. Netw. Comput. Appl. 69 (2016) 152–165.
[16] B.K. Ray, A. Saha, S. Khatua, S. Roy, Toward maximization of profit and quality of cloud federation: Solution to cloud federation formation problem, J. Supercomput. 75 (2) (2019) 885–929.
[17] M. Habibi, M. Fazli, A. Movaghar, Efficient distribution of requests in federated cloud computing environments utilizing statistical multiplexing, Future Gener. Comput. Syst. 90 (2019) 451–460.
[18] J. Xu, B. Palanisamy, Cost-aware resource management for federated clouds using resource sharing contracts, in: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), IEEE, 2017, pp. 238–245.
[19] S.S. Chauhan, E.S. Pilli, R. Joshi, A broker based framework for federated cloud environment, in: 2016 International Conference on Emerging Trends in Communication Technologies (ETCT), IEEE, 2016, pp. 1–5.
[20] M.M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, S.M.M. Rahman, A. Alelaiwi, A. Alamri, M.A. Hamid, Qos and trust-aware coalition formation game in data-intensive cloud federations, Concurr. Comput. Pract. Exp. 28 (10) (2016) 2889–2905.
[21] L. Mashayekhy, M.M. Nejad, D. Grosu, A trust-aware mechanism for cloud federation formation, IEEE Trans. Cloud Comput. (2019) 1.
[22] T. Halabi, M. Bellaiche, A. Abusitta, A cooperative game for online cloud federation formation based on security risk assessment, in: 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), IEEE, 2018, pp. 83–88.
[23] B. Ray, A. Saha, S. Khatua, S. Roy, Quality and profit assured trusted cloud federation formation: Game theory based approach, IEEE Trans. Serv. Comput. (2018) 1.
[24] B.K. Ray, A. Saha, S. Roy, Migration cost and profit oriented cloud federation formation: Hedonic coalition game based approach, Cluster Comput. 21 (4) (2018) 1981–1999.
[25] H. Chen, B. An, D. Niyato, Y.C. Soh, C. Miao, Workload factoring and resource sharing via joint vertical and horizontal cloud federation networks, IEEE J. Sel. Areas Commun. 35 (3) (2017) 557–570.
[26] C.R. Choi, H.Y. Jeong, A broker-based quality evaluation system for service selection according to the QoS preferences of users, Inform. Sci. 277 (2014) 553–566.
[27] J. Siegel, J. Perdue, Cloud services measures for global use: The service measurement index (*SMI*), in: 2012 Annual SRII Global Conference, IEEE, 2012, pp. 411–415.
[28] D. Parmenter, Key Performance Indicators: Developing, Implementing, and using winning *KPI*s, John Wiley & Sons, 2015.
[29] B.K. Ray, A.I. Middya, S. Roy, S. Khatua, Multi-criteria based federation selection in cloud, in: 2017 9th International Conference on Communication Systems and Networks (COMSNETS), IEEE, 2017, pp. 182–189.
[30] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, G.M. Sarné, A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures, Future Gener. Comput. Syst. 56 (2016) 77–94.
[31] M.V. Thomas, K. Chandrasekaran, Dynamic partner selection in Cloud Federation for ensuring the quality of service for cloud consumers, Int. J. Model. Simul. Sci. Comput. 8 (03) (2017) 1750036.
[32] A.N. Toosi, R.K. Thulasiram, R. Buyya, Financial option market model for federated cloud environments, in: 2012 IEEE Fifth International Conference on Utility and Cloud Computing, IEEE, 2012, pp. 3–12.
[33] S. Rebai, M. Hadji, D. Zeghlache, Improving profit through cloud federation, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), IEEE, 2015, pp. 732–739.
[34] A.K. Das, T. Adhikary, M.A. Razzaque, E.J. Cho, C.S. Hong, A QoS and profit aware cloud confederation model for IaaS service providers, in: Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, ACM, 2014, p. 42.
[35] Y.-H. Lee, K.-C. Huang, M.-R. Shieh, K.-C. Lai, Distributed resource allocation in federated clouds, J. Supercomput. 73 (7) (2017) 3196–3211.
[36] K. Ma, A. Bagula, O. Ajayi, Quality of service (qos) modelling in federated cloud computing, 2019, arXiv preprint arXiv:1911.03051.
[37] D. Ardagna, S. Casolari, M. Colajanni, B. Panicucci, Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems, J. Parallel Distrib. Comput. 72 (6) (2012) 796–808.
[38] C.-Y. Liu, K.-C. Huang, Y.-H. Lee, K.-C. Lai, Efficient resource allocation mechanism for federated clouds, Int. J. Grid High Perform. Comput. 7 (4) (2015) 74–87.
[39] W. Zhang, S. Han, H. He, H. Chen, Network-aware virtual machine migration in an overcommitted cloud, Future Gener. Comput. Syst. 76 (2017) 428–442.
[40] W.-Z. Zhang, H.-C. Xie, C.-H. Hsu, Automatic memory control of multiple virtual machines on a consolidated server, IEEE Trans. Cloud Comput. 5 (1) (2015) 2–14.
[41] M.M. Hassan, M.S. Hossain, A.J. Sarkar, E.-N. Huh, Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform, Inf. Syst. Front. 16 (4) (2014) 523–542.
[42] P. Belleflamme, M. Peitz, Industrial Organization: Markets and Strategies, Cambridge University Press, 2015.
[43] A. Kohne, M. Spohr, L. Nagel, O. Spinczyk, Federatedcloudsim: A SLA-aware federated cloud simulation framework, in: Proceedings of the 2nd International Workshop on CrossCloud Systems, ACM, 2014, p. 3.
[44] Federated cloudsim download, 2017, https://ess.cs.uni-dortmund.de/Software/FederatedCloudSim/workloads.zip.
[45] Amazon EC2 pricing, 2019, http://aws.amazon.com/ec2/pricing/.
[46] Microsoft azure pricing calculator, 2019, https://azure.microsoft.com/en-us/pricing/calculator/.
[47] Google compute engine pricing, 2019, https://cloud.google.com/compute/all-pricing.
[48] M. Masdari, A. Khoshnevis, A survey and classification of the workload forecasting methods in cloud computing, Cluster Comput. (2019) 1–26.
[49] C. Robert, Machine Learning, A Probabilistic Perspective, Taylor & Francis, 2014.
[50] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Amer. Statist. 46 (3) (1992) 175–185.
[51] H. Drucker, C.J. Burges, L. Kaufman, A.J. Smola, V. Vapnik, Support vector regression machines, in: Advances in Neural Information Processing Systems, 1997, pp. 155–161.
[52] V. Vovk, Kernel ridge regression, in: Empirical Inference, Springer, 2013, pp. 105–116.
[53] W.-Y. Loh, Classification and regression trees, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 1 (1) (2011) 14–23.

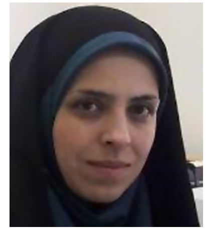


Neda Khorasani received her B.Eng. in Computer Engineering from Ferdowsi University of Mashhad (FUM) and started pursuing her M.Eng. degree directly without taking the university entrance exam, as a talented student, in the field of Software Engineering at FUM. Her research interests include cloud computing and game theory.

Saeid Abrishami received the M.Sc. and Ph.D. degrees in computer engineering from Ferdowsi University of Mashhad, in 1999 and 2011, respectively. Since 2003, he has been with the Computer Engineering Department, Ferdowsi University of Mashhad, where he is currently an assistant professor. During the spring and summer of 2009 he was a visiting researcher at the Delft University of Technology. His research interests focus on resource management and scheduling in the distributed systems, especially in Grids and Clouds.

Mehdi Feizi received his B.Sc. in Computer Engineering from Ferdowsi University of Mashhad (FUM) in 2004, M.Sc. in Socio-economic Systems Engineering from Institute for Management and Planning studies in 2006, and Ph.D. in economics from Goethe University Frankfurt in 2012. He is currently an Assistant Professor of Economics with the Department of Economics at FUM. His research interests broadly relate to economics theory, particularly game theory and mechanism design, and behavioral economics.

Mahdi Abolfazli Esfahani received his B.Eng. and M.Eng. in Computer Engineering, and Artificial Intelligence and Robotics from Ferdowsi University of Mashhad, Iran, respectively. He was the leader of the Nexus RoboCup soccer simulation team and achieved more than 20 national and international awards. He is currently pursuing a Ph.D. degree with the Department of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include deep learning, evolutionary computing, game theory and Robotics.



Faeze Ramezani received her B.Eng. in Computer Engineering from Ferdowsi University of Mashhad (FUM) in 2007, M.Sc. degrees in computer engineering from the University of Isfahan, in 2011 and she is currently Ph.D. Candidate of Computer Engineering — Software in FUM. Her research interests in resource management in cloud, game theory and mechanism design.