



# A novel optimized approach for resource reservation in cloud computing using producer–consumer theory of microeconomics

Alireza Mohammadi<sup>1</sup> · Mohammad Hossein Rezvani<sup>1</sup> 

Published online: 15 July 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Designing economic pricing mechanisms have recently attracted a great deal of attention in the context of cloud computing. We believe that microeconomics theory is a good candidate to model the resource reservation operations in cloud networks. Producer–consumer theory of microeconomics guarantees the maximization of social welfare of the customers, conditional that the particular consideration concerning customers and producers are met. As is the case in real-world cloud datacenters, the workload associated with each user is fed into the system and then the user is bound to a virtual machine (VM). In this research, we propose a microeconomic-inspired resource reservation scheme for cloud computing. The designed mechanism includes two steps: in the first step, we seek to find a Pareto efficient reservation set concerning bandwidth of VMs, and in the second step, our goal is to place VMs' reserved bandwidth rates on physical hosts. In our modeling, VMs and the cloud network are considered as consumers and producers of the market, respectively. Also, the bandwidth of requested services is considered as commodity. As is the case in microeconomics, we prove that the aggregation of users' utilities (users' social welfare in microeconomics terminology) could reach to global maximum, known as Pareto efficiency. After finding the best set of reserved bandwidth rates in the first step of mechanism, in the second step, the mechanism seeks to find the best placement for VMs on physical hosts. The placement operation is performed in such a way that results in minimization of total consumed power in datacenter. Since the VM placement problem has been proven to be NP-hard, we use a metaheuristic cuckoo search optimization approach to solve the optimization problem. Simulation results, obtained through the CloudSim framework, established the robustness of the proposed method in terms of significant criteria such as users' welfare, consumed power and Pareto optimality.

**Keywords** Cloud computing · Resource reservation · Virtual machine placement · Pricing mechanism · Producer–consumer theory

---

Extended author information available on the last page of the article

## 1 Introduction

Recently, the most organizations and corporations have started to invest on cloud computing, leveraging the advantages of service virtualization [1]. In fact, the aim of cloud computing is to optimize the aggregation of resources based on the concept of infrastructures as a service or in abbreviation IaaS. On the other hand, service virtualization is used as a supplementary technology in datacenter (DC) in order to reach a better utilization of network resources. Using virtualization approaches, the resources locating on physical hosts could serve multiple virtual machines (VMs). In fact, virtualization is the heart of cloud computing and has a lot of benefits such as resource utilization, system security capability, near to full efficiency, simple management, and so on [2]. In cloud-based DCs, the controlling operation in order to associate VMs to the appropriate physical hosts is carried out by a central entity, namely broker. Also, the information concerning physical hosts is stored in another entity, namely cloud service provider (CSP). This resource management process is called VM placement and is one of the most crucial issues in context of cloud computing. Interested readers can refer to [3] for a comprehensive survey on cloud brokerage.

Resource management in a typical cloud network includes two major steps: The first step comprises scheduling and workflow operations concerning the task which are submitted by users. As is the case in real-world cloud datacenters, the workload originated from each user is fed into the DC. Each user (known as cloudlet) is bound to a VM. This operation is done by cloud broker. Our research in this paper is not relevant to scheduling step. Interested readers can refer to [4, 5] to study more on this topic. The second step, itself comprises “*resource reservation*” and “*VM placement*” operations in which potential players are VMs and physical hosts. Concerning the *resource reservation* step, a large number of research activities have been carried out in the literature. The goal of many of these researches is to maximize the utilization of resources [6]. Recently, pricing approaches have attracted significant attention of researchers for managing the resource reservation step [7–17]. Any economic-inspired mechanism should address the way in which users’ demands (equivalently VMs’ demands) are met. Here, the goal is to provide maximum welfare for VMs, taking into account minimum power consumption in the DC. The most significant metrics in the context of pricing approaches are economic parameters such as revenue, cost, profit, fairness, economic equilibrium, and so on.

Concerning the *VM placement* step, significantly, heuristic and metaheuristic approaches have attracted the most attention of researchers [7–9, 18–22]. The VM placement is a crucial task and if is done improperly, may degrade the performance of the cloud network. So, almost all researchers have investigated the effects of their modeling on key performance indicators of the DC such as the level of consumed power, latency, CPU usage, memory, bandwidth, and so on.

Actually, the scope of our research in this paper is the second step of cloud computing, namely *resource reservation* and *VM placement*. We believe that the mathematical theories of microeconomics are good candidates in order to model the dynamics of VMs in the resource reservation step. To this end, we model the

interactions between users and the DC using a pricing approach leveraging the theory of producer–consumer and particularly, the concept of Walrasian equilibrium.

Our contribution is two-fold: First, subject to the resource reservation step, we design a price-based microeconomic mechanism which maximizes the social welfare of VMs by applying the theory of producer–consumer. In the proposed approach, users (or equivalently, VMs associated with users) are considered as consumers of the economic market, and the physical hosts are considered as producers. Also, the bandwidth of requested services plays the role of market commodities. Roughly speaking, by assuming the players as price-taker agents and also by equating the amount of demand and supply, we get the amount of resource (bandwidth) which should be reserved for each VM. In Sect. 3, we will prove that the aggregated utility of all users (the users' social welfare in microeconomics terminology) could reach to global maximum, known as Pareto efficient point. Then, in the second step, leveraging the identified reserved bandwidth rates in the first step, VMs should be placed on appropriate physical hosts. The placement is done with the aim of minimizing the consumed power. Since the VM placement problem has been proven to be NP-hard, we have used metaheuristic cuckoo search optimization (CSO) algorithm to optimize the placement problem.

To the best of our knowledge, this research is the first work in which the theory of producer–consumer (and the concept of Walrasian equilibrium) has been used in order to optimize the reserved bandwidth in cloud networks. We summarize the main contributions of our research as following:

- We model the resource reservation phase of cloud network based on producer–consumer theory of microeconomics. All reserved allocations have Walrasian equilibrium property which will be discussed in depth in Sect. 3.
- For each VM, we consider real-world necessary conditions in order to attain Pareto efficiency concerning the reserved resources.
- Based on identified reserved resources, our algorithm finds a low-power low-overhead placement solution for VMs.

Due to self-organizing nature of our proposed approach, from industry point of view, it can be used by the owners of cloud providers, especially for long-live services. For example, the users in cloud gaming can be charged based on this approach. Note that in cloud gaming, each gamer session may last for tens of minutes.

The organization of this paper is as follows: In Sect. 2, significant studies related to priced-based approaches and also placement algorithms are described; in Sect. 3, the proposed mechanism is presented and discussed in detail; in Sect. 4, the simulation of this mechanism on the CloudSim testbed and the evaluation results are elaborated; finally, Sect. 5 provides the conclusions and sketches future research directions.

## 2 Related researches

So far, many studies have been performed on resource reservation and VM placement in cloud networks. In this section, at first, we will review research works related to the subject of resource reservation, and then we will review the works related to VM placement. Some research works have tried to apply pricing approaches, taking into account CSPs and also the geographical location of DCs [23, 24]. Also, interested readers can refer to [7, 14, 15] for comprehensive surveys on pricing approaches in cloud computing. Generally, these pricing mechanisms fail to consider significant factors such as external stakeholders of the market, pricing strategies of other DCs, and the payment tendencies of clients. In cloud networks, “*differential pricing*” approach is used for allocating bandwidth among some groups of users with different demands in order to maximize profit [8]. This approach is unfair because some users might pay more than the others. Dabbagh et al. [9], proposed a pricing scheme in which the goal is to maximize the profit of a given VM subject to the available budget, while minimizing energy consumption. The proposed approach suffers the lack of considering market competition for determining the price. Wanis et al. [25] used *Ramsey* pricing to maximize the social welfare of users with respect to this limitation that the profit should not be less than a predetermined threshold value. This pricing scheme has been proposed for allocating virtualized resources in DCs. In some researches like [18], non-cooperative game theory approaches, for example *auctions*, has been used for maximizing the profit of CSPs. Since Nash equilibrium prerequisite is not guaranteed in all times, investigating the essential conditions of such non-cooperative games is a challenging issue. In the literature, other game-theoretic approaches such as “*Stackelberg*” games and “*bargaining*” games have been used for allocating the bandwidth to VMs and satisfying users’ demands [19, 26]. In these categories of games, various bidding approaches such as English, Dutch and Vickrey–Clarke–Groves (VCG) auctions have been used for resource management. The most important disadvantage of these auction-based methods is that they merely consider the satisfaction of one player against the other players, without taking into account the network capacity constraints! Mohammadi and Rezvani [20] are the first who applied the microeconomics theory to model the interactions between VMs and the physical hosts in cloud networks. Their idea, in contrast to game-theoretic approaches, falls into the category of non-strategic economic mechanisms in the sense that the actions of VMs associated with the users do not affect by the actions of other VMs. One of the significant works in the context of pricing is presented by Wu et al. [13]. They proposed a value-based pricing model, namely hedonic pricing model which takes into account both imposed service cost to CSP and willingness of customers to pay. Interested readers can refer to [11, 12, 16, 17] to see other researches in the context of cloud pricing schemes.

Now, we proceed to review the major related works subject to VM placement. The ever-increasing growth of modern DCs turns the VM placement problem a fundamental challenge in the context of cloud computing. According to the previous researches, the VM placement problem falls into the category of NP-hard problems. In the literature, there exist numerous heuristic and metaheuristic approaches for

solving these types of problems. For example, in first-fit decreasing (FFD) algorithm, which is presented in [27], the VMs are ordered in descending manner based on requirements and then the host with the least consumed power is selected for placing VMs on it. Jamali et al. [28] used imperialist competitive algorithm to place VMs. In this approach, any possible solution is considered as a country. Imperialist countries will overcome the colonies according to the intermediate countries. The most challenging disadvantage of this metaheuristic scheme is its complexity for selecting initial possible solutions and also the complexity for setting up parameters. Also, Tavakoli-Someh and Rezvani [29] proposed a VM placement solution based on non-dominated sorting genetic algorithm II (NSGA-II) to maximize utilization and minimize energy. As stated in the previous section, the CSO algorithm is an interesting metaheuristic approach which is used for solving VM placement problem. In general, the performance of CSO is better than those of particle swarm optimization (PSO) and genetic algorithm (GA) [27]. Sait et al. [30] used the CSO algorithm for reducing the consumed power and improving the resource utilization of physical hosts. Inspiring Levy flight of cuckoos, the VMs located on physical hosts may migrate to new locations in a best-fit (BF) manner. However, a major weak point of the CSO scheme is its slow convergence speed to find the optimal solution. In order to mitigate this limitation, we have combined CSO with simulated annealing (SA) approach. We will explain details of this, later in Sect. 3.

In the literature, some authors have paid attention to the role of energy considerations in solving the VM placement problem. However, they have not been discussed here in detail due to space limitations. Interested readers can refer to [31–34] to study more on this topic.

Eventually, it is worth mentioning that in this paper, we have not pay attention to security issues at all. Clearly, the mechanism could be enhanced if such real-world requirements are considered. For example, Feng et al. [35] proposed a joint pricing and security model for insurance cloud networks. Also, Negi et al. [36] proposed a filtering method for cloud computing environments to prevent some kinds of attacks. Interested readers can refer to [37–41] to study more on this topic.

## 3 Proposed method

### 3.1 VM bandwidth reservation problem

In this section, we propose the microeconomics-based modeling of interactions in the DC. We use one of the most significant theories in “*competitive markets*” of microeconomics, known as “*producer–consumer theory*.” The main characteristic of the competitive economy is that the behaviors of players of the market, i.e., producers and consumers, are influenced by their selfishness. The consumers in this economy have no predetermined tendency to buy the commodities. In other words, the market players accept the price of commodities as they are. It means that they cannot change the prices themselves. In microeconomics terminology, this fact can be expressed as follows: “in competitive economic, the potential players of the market are price-takers not price-setters [42].” Clearly, the competitive economy model

is disturbed in nature. As stated before, the producer–consumer theory falls into the category of “*non-strategic*” mechanisms in the sense that each consumer has complete knowledge about current prices and ignores the demand rates of the other consumers. In other words, the consumer pursues only his/her own optimal demand according to his/her budget constraints. In fact, consumers assure that the commodities are sufficiently available in the market. Similarly, producers have complete knowledge of current prices concerning commodities and are ignoring the demand rate issued by the other producers. Roughly speaking, the producers only choose the levels of production which may create maximum profit for them. The producers believe that commodities will eventually be sold. Regard to the aforementioned discussions, it can be said that the competitive economy has a distributed nature, so that any agent looks for his/her desires and ignores others preferences. In this research, we model interactions of the DC as a competitive economy. In our modeling, physical hosts of the DC are considered as producers of the market. In a similar manner, users (or equivalently, VMs associated with the users) are considered as consumers and the bandwidth unit (bps) of the requested services is considered as the commodities. We name our proposed microeconomic mechanism as “*Cloud Network with Competitive Economic*,” or in abbreviation CNCE. We summarize the main assumptions of our model as following:

- Upon joining the network, each customer is endowed a fix amount of budget.
- The cloud network provides multiple services simultaneously.
- For each service, the price of bandwidth unit is a predetermined value. These prices are set by service owners.
- The physical hosts of the DC play the role of producers.
- The users (VMs) play the role of consumers.
- The bandwidth unit of each service plays the role of commodity.

Table 1 shows the description of mathematical symbols and notations which are used in this section. We denote the set of VMs by  $VM = \{vm_1, vm_2, \dots, vm_{N_{VM}}\}$  in which  $N_{VM}$  represents the total number of VMs. Similarly, the set of physical hosts is denoted by  $H = \{h_1, h_2, \dots, h_{N_H}\}$  in which  $N_H$  represents the total number of physical hosts. Also, the set of services provided in CNCE is denoted by  $S = \{s_1, s_2, \dots, s_{N_S}\}$  in which  $N_S$  represents the total number of services. Also, function  $u_{vm_i}$  denotes the preferences of every VM in domain  $\mathbb{R}_+^N$ . The  $u_{vm_i}$  function reveals the utility of virtual machine  $vm_i$  concerning its demanded services. This function meets the conditions in Assumption 1.

**Assumption 1** (*Social welfare of VMs*) Function  $u_{vm_i}$  is continuous, strictly increasing and quasi-concave [42].

In order to meet conditions of Assumption 1, we have designed  $u_{vm_i}$  as a logarithmic function. This type of utility function is another form of increasing and concave function called Cobb–Douglas function. Cobb–Douglas function is the most commonly used function in microeconomics and is influenced by demanded bandwidth

**Table 1** Mathematical symbols and notations used in reservation step

Symbol	Symbol description
$N_{VM}$	Total number of virtual machines
$N_H$	Total number of physical machines (hosts)
$N_S$	Total number of services which are provided by DC
$VM = \{vm_1, vm_2, \dots, vm_{N_{VM}}\}$	The set of virtual machines
$H = \{h_1, h_2, \dots, h_{N_H}\}$	The set of hosts
$S = \{s_1, s_2, \dots, s_{N_S}\}$	The set of services provided by DC
$u_{vm_i}$	Utility function associated with virtual machine $vm_i$
$d_{vm_i}^{s_j}$	The rate of the demanded service $s_j$ which is requested by virtual machine $vm_i$
$d_{max}^{s_j}$	Maximum allowed rate of service $s_j$ for each VM
$d_{min}^{s_j}$	Minimum allowed rate of service $s_j$ for each VM
$b_{vm_i}$	Budget associated with virtual machine $vm_i$
$p_{s_j}$	The price of service $s_j$
$\mathbf{P} = (p_{s_1}, p_{s_2}, \dots, p_{s_{N_S}})$	Vector of prices associated with network services
$p_{s_j}^*$	Walrasian equilibrium price for service $s_j$
$g_{h_l}^{s_j}$	The rate of service $s_j$ which is provided by physical host $h_l$
$g_{h_l}^{max}$	The maximum rate (capacity) provided by physical host $h_l$
$\mathbf{P}^* = (p_{s_1}^*, p_{s_2}^*, \dots, p_{s_{N_S}}^*)$	Vector of Walrasian equilibrium price
$ED_{s_j}(\mathbf{P}), \quad 1 \leq j \leq N_S$	Excess demand of service $s_j$
$ED_{s_j}(\mathbf{P}^*)$	Excess demand of service $s_j$ at Walrasian equilibrium price

rate of each consumer. The utility function of virtual machine  $vm_i$  is in the form of Eq. (1). It is equal to weighted sum of utilities resulted from reservation rates of virtual machine  $vm_i$  subject to its demanded services:

$$u_{vm_i} = \sum_{j=1}^{N_S} w_{vm_i}^{s_j} \times \text{Ln}(1 + d_{vm_i}^{s_j}), \quad 1 \leq i \leq N_{VM} \tag{1}$$

In the above equation,  $w_{vm_i}^{s_j}$  denotes the weight (importance) of service  $s_j$  for each virtual machine  $vm_i$ . These weights are determined by the network owner and are generally application-specific. Clearly, for each virtual machine  $vm_i$ , it must be that  $\sum_{j=1}^{N_S} w_{vm_i}^{s_j} = 1$ . Also,  $d_{vm_i}^{s_j}$  denotes the rate of service  $s_j$  which is demanded by virtual machine  $vm_i$ .

As mentioned above, each physical host  $h_l$  plays the role of producer in CNCE. Let  $g_{h_l}^{s_j}$  denote the rate of service  $s_j$  which is generated by physical host  $h_l$ . It is clear that the domain of generated services for each host  $h_l$  is  $\mathbb{R}^{N_S}$ . Also, we assume that each host  $h_l$  has a possible service set. The characteristics of possible service set will be introduced later in Assumption 2.

Now, we proceed to formulate the optimization problem of VMs in the CNCE. As explained in Eq. (2), our problem is to maximize the aggregation of utilities concerning all VMs. In microeconomics terminology, it means that we want to

maximize the social welfare of VMs. This problem is expressed in Eq. (2) subject to constraints of Eqs. (3)–(5):

$$\text{Max } f_1 = \sum_{i=1}^{N_{\text{VM}}} u_{\text{vm}_i}, \quad (2)$$

$$\text{s.t. } \sum_{j=1}^{N_S} \left( d_{\text{vm}_i}^{s_j} \times p_{s_j} \right) \leq b_{\text{vm}_i}, \quad 1 \leq i \leq N_{\text{VM}} \quad (3)$$

$$d_{\text{min}}^{s_j} \leq d_{\text{vm}_i}^{s_j} \leq d_{\text{max}}^{s_j}, \quad 1 \leq j \leq N_S \quad (4)$$

$$\sum_{i=1}^{N_{\text{VM}}} d_{\text{vm}_i}^{s_j} \leq \sum_{l=1}^{N_H} g_{h_l}^{s_j}, \quad 1 \leq j \leq N_S \quad (5)$$

The expression in Eq. (3) represents the budget constraint of virtual machine  $\text{vm}_i$ . The condition in left-hand side of Eq. (3) represents expenditure of virtual machine  $\text{vm}_i$  which is the multiplication of rates of its demanded services by the prices of those services. This equation simply states that the expenditure of virtual machine  $\text{vm}_i$  should not exceed its budget,  $b_{\text{vm}_i}$ . For each virtual machine  $\text{vm}_i$ , the budget  $b_{\text{vm}_i}$  and the demand bandwidth rate  $d_{\text{vm}_i}^{s_j}$  are determined a priori. These two parameters are determined by the users in the cloud network. Equation (4) is an application-specific constraint. It states that for each virtual machine  $\text{vm}_i$ , rates of the demand for each service  $s_j$  should be in a range between a minimum and maximum. Equation (5) states that for each service  $s_j$ , total demands issued by VMs (actually the users) should not exceed the total supply of the hosts. Clearly, the possible areas concerning constraints in Eqs. (3)–(5) are compact. Therefore, the above nonlinear optimization problem will have a global optimum solution. The solution of optimization problem of Eq. (2) is the optimum rates of the demanded services for all virtual machines, i.e.,  $\{d_{\text{vm}_i}^{s_j}, 1 \leq j \leq N_S\}$ . These types of problems can be easily solved by optimization approaches such as gradient methods using Kahn–Tucker conditions and by means of common mathematical software tools such as JOM [43].

Now, we proceed to explain the producer problem. The characteristics of possible service set were introduced in Assumption 2.

**Assumption 2** (*Characteristics of possible service set*) Let  $g_{h_l}^{s_j}$  denote the rate of service  $s_j$  which is provided by physical host  $h_l$ . Then, the possible service supply,  $g_{h_l}^{s_j}$ , should have properties as following [42]:

- $\mathbf{0} \in g_{h_l}^{s_j} \subseteq \mathbb{R}^{N_S}$
- $g_{h_l}^{s_j}$  is closed and bounded.
- $g_{h_l}^{s_j}$  is convex.



The condition (a) guarantees the boundedness of service production rate from below, i.e., zero boundedness. The condition (b) guarantees the continuity of service production by physical host  $h_l$ . The condition (c) assures that the provided service from physical host  $h_l$  is unique. Therefore, every physical host  $h_l$  forms a convex solution area.

As stated before,  $\sum_{l=1}^{N_H} g_{h_l}^{s_j}$  denotes the total supply of hosts concerning the service  $s_j$  in which  $N_H$  denotes the number of physical hosts and  $1 \leq j \leq N_S$ . Now, let us formulate the optimization problem of physical hosts, as producers of the market:

$$\text{Max Rev} = \sum_{j=1}^{N_S} \sum_{l=1}^{N_H} \left( g_{h_l}^{s_j} \times p_{s_j} \right) \tag{6}$$

$$\text{s.t. } \sum_{j=1}^{N_S} g_{h_l}^{s_j} \leq g_{h_l}^{\text{max}} \tag{7}$$

The expression in Eq. (6) represents the total revenue earned by producers concerning  $N_S$  services. Equation (7) states that for each physical host  $h_l$ , the total production bandwidth rate of all services should not exceed the host's maximum permissible production,  $g_{h_l}^{\text{max}}$ . Here,  $g_{h_l}^{\text{max}}$  denotes uploading bandwidth of host  $h_l$ . The solution for optimization problem in Eq. (6) represents the optimum bandwidth rates concerning the generated services by all hosts, i.e.,  $\{g_{h_l}^{s_j}, 1 \leq j \leq N_S\}$ .

As stated before, in equilibrium point of competitive economy, the demands of consumers are met and also the supplied commodities are sold completely. Therefore, for every agent, movements of the other agents can be ignored, and hence the only information the consumer and the producer need is the current price of the market. This fact is the core of Walrasian equilibrium theory. Leon Walras (1834–1910), a French mathematical economist, presented the Walrasian supply and demand equilibrium theory for the first time. The general equilibrium theory attempts to find a vector of prices at which the total demand rates of each commodity is equal to its total supply rates. In the following, we will prove that any reservation in Walrasian equilibrium is a Pareto efficient solution for CNCE and, hence may lead to optimization of VMs' rates [42].

It is common in microeconomics to show the difference between total demand and total supply of any commodity by a function, named “*excess demand*.” Before proceed, let us denote the vector of service prices by  $\mathbf{P} = (p_{s_1}, p_{s_2}, \dots, p_{s_{N_S}})$ . As mentioned before, the number of services provided by the cloud is denoted by  $N_S$ . Clearly, the price vector  $\mathbf{P}$  is  $N_S$ -dimensional. Now, the *excess demand* of service  $s_j$  can be represented as following:

$$\text{ED}_{s_j}(\mathbf{P}) = \sum_{i=1}^{N_{VM}} d_{vm_i}^{s_j}(\mathbf{P}) - \sum_{l=1}^{N_H} g_{h_l}^{s_j}(\mathbf{P}), \quad 1 \leq j \leq N_S \tag{8}$$

If  $ED_{s_j}(\mathbf{P}) > 0$ , then we have extra demand for bandwidth of service  $s_j$  and if  $ED_{s_j}(\mathbf{P}) < 0$ , then we will have extra supply for bandwidth of service  $s_j$ . In microeconomic, extra demand and extra supply are considered as instability of market. The Walrasian equilibrium price,  $\mathbf{P}^*$  occurs when  $ED_{s_j}(\mathbf{P}^*) \gg 0$  for all  $1 \leq j \leq N_s$ . Here, the notation  $\gg$  means “equal or very close to.” According to Assumptions 1 and 2, the existence of equilibrium price vector  $\mathbf{P}^*$  is guaranteed by the following theorem [42]:

**Theorem 1** (Walrasian equilibrium in CNCE) *If any virtual machine  $vm_i$  meets Assumption (1) and any physical host  $h_l$  meets conditions in Assumption (2), then there exists at least one price vector  $\mathbf{P}^*$  in which  $ED_{s_j}(\mathbf{P}^*) \gg 0$  for all  $1 \leq j \leq N_s$ .*

**Proof** Before we proceed, let us emphasize on two important facts. The first is that although Cobb–Douglas utility function is not strongly increasing on  $\mathbb{R}_+^{N_s}$ , it guarantees the existence of Walrasian equilibrium. Second, when utilities satisfy Assumption 1, the excess demand vector will be homogeneous of degree zero. Interested readers can refer to pages 204–212 of [42] to see the proof of these theories. The key importance of homogeneity is that only relative prices matter in VMs’ choices. Thus, subject to homogeneity property, we will have  $ED_{s_j}(\mathbf{P}^*) = ED_{s_j}(\lambda \cdot \mathbf{P}^*) = \mathbf{0}$  for all  $\lambda > 0$ . So, should there exist some set of prices at which all markets clear. Also, any price vector which is obtained from multiplying that price by  $\lambda$  will also clear the CNCE market. This fact can simplify the calculations.  $\square$

**Example 1** Let us consider a two-person two-service CNCE and solve it to get a Walrasian equilibrium. We assume that VMs (users)  $vm_1$  and  $vm_2$  have identical utility functions,

$$u_{vm_i}(d_{vm_i}^{s_1}, d_{vm_i}^{s_2}) = (d_{vm_i}^{s_1 \rho} + d_{vm_i}^{s_2 \rho})^{\frac{1}{\rho}} \quad i = 1, 2, \tag{9}$$

where  $0 < \rho < 1$ . In microeconomics, the above function is known as constant elasticity of substitution (CES) utility function. It can easily be verified that Eq. (9) is strictly monotonic and strictly convex. Suppose there be 1 unit of each service and each consumer owns all of one service, so initial endowments concerning the consumers are  $\mathbf{e}_{vm_1}(1, 0)$  and  $\mathbf{e}_{vm_2}(0, 1)$ . Because the total endowment of each service is strictly positive and Eq. (9) is strongly increasing and strictly quasi-concave on  $\mathbb{R}_+^{N_s}$  when  $0 < \rho < 1$ , the requirements of Theorem 1 are satisfied, so we make sure that a Walrasian equilibrium exists in this economy. Now in order to obtain the demand functions concerning users, let us define the following optimization problem inspiring from Eqs. (2)–(3):

$$\text{Max } u_{vm_i}(d_{vm_i}^{s_1}, d_{vm_i}^{s_2}) = (d_{vm_i}^{s_1 \rho} + d_{vm_i}^{s_2 \rho})^{\frac{1}{\rho}} \quad i = 1, 2 \tag{10}$$

$$\text{s.t. } d_{vm_i}^{s_1} \cdot p_{s_1} + d_{vm_i}^{s_2} \cdot p_{s_2} \leq b_{vm_i} \quad i = 1, 2 \tag{11}$$

To solve the above nonlinear optimization problem, at first, we define Lagrangian function:

$$L(d_{vm_i}^{s_1}, d_{vm_i}^{s_2}, \lambda) = (d_{vm_i}^{s_1 \rho} + d_{vm_i}^{s_2 \rho})^{\frac{1}{\rho}} - \lambda \cdot (d_{vm_i}^{s_1} \cdot p_{s_1} + d_{vm_i}^{s_2} \cdot p_{s_2} - b_{vm_i}) \quad (12)$$

Now, we form the first-order Lagrangian Kuhn–Tucker conditions as following:

$$\frac{\partial L}{\partial d_{vm_i}^{s_1}} = (d_{vm_i}^{s_1 \rho} + d_{vm_i}^{s_2 \rho})^{\frac{1}{\rho}-1} \cdot d_{vm_i}^{s_1 \rho-1} - \lambda \cdot p_{s_1} = 0, \quad (13)$$

$$\frac{\partial L}{\partial d_{vm_i}^{s_2}} = (d_{vm_i}^{s_1 \rho} + d_{vm_i}^{s_2 \rho})^{\frac{1}{\rho}-1} \cdot d_{vm_i}^{s_2 \rho-1} - \lambda \cdot p_{s_2} = 0, \quad (14)$$

$$\frac{\partial L}{\partial \lambda} = p_{s_1} \cdot d_{vm_i}^{s_1} + p_{s_2} \cdot d_{vm_i}^{s_2} - b_{vm_i} = 0, \quad (15)$$

As is evident, we have three equations in three unknowns. After rearranging and simplifying Eqs. (13)–(15), we have:

$$d_{vm_i}^{s_1} = d_{vm_i}^{s_2} \cdot \left( \frac{p_{s_1}}{p_{s_2}} \right)^{\frac{1}{\rho-1}}, \quad (16)$$

$$b_{vm_i} = p_{s_1} \cdot d_{vm_i}^{s_1} + p_{s_2} \cdot d_{vm_i}^{s_2}, \quad (17)$$

After substituting Eq. (16) in Eq. (17), we have:

$$\begin{aligned} b_{vm_i} &= p_{s_1} \cdot d_{vm_i}^{s_1} \cdot \left( \frac{p_{s_1}}{p_{s_2}} \right)^{\frac{1}{\rho-1}} + p_{s_2} \cdot d_{vm_i}^{s_2}, \\ &= d_{vm_i}^{s_2} \left( p_{s_1}^{\frac{\rho}{\rho-1}} + p_{s_2}^{\frac{\rho}{\rho-1}} \right) \cdot p_{s_2}^{\frac{-1}{\rho-1}} \end{aligned} \quad (18)$$

By solving Eq. (18) for  $d_{vm_i}^{s_2}$ , we have:

$$d_{vm_i}^{s_2} = \frac{p_{s_2}^{\frac{1}{\rho-1}} \cdot b_{vm_i}}{p_{s_1}^{\frac{\rho}{\rho-1}} + p_{s_2}^{\frac{\rho}{\rho-1}}}, \quad (19)$$

Now, substituting from (19) into (16) gives us  $d_{vm_i}^{s_1}$ :

$$d_{vm_i}^{s_1} = \frac{p_{s_1}^{\frac{1}{\rho-1}} \cdot b_{vm_i}}{p_{s_1}^{\frac{\rho}{\rho-1}} + p_{s_2}^{\frac{\rho}{\rho-1}}}, \tag{20}$$

Equations (19) and (20) are known as Marshallian demand functions. By defining  $r = \rho/(\rho - 1)$ , we can rewrite the final Marshallian demands regard to two VMs, namely  $vm_1$  and  $vm_2$  as:

$$d_{vm_i}^{s_1}(\mathbf{P}, b_{vm_i}) = \frac{p_{s_1}^{r-1} \cdot b_{vm_i}}{p_{s_1}^r + p_{s_2}^r}, \tag{21}$$

$$d_{vm_i}^{s_2}(\mathbf{P}, b_{vm_i}) = \frac{p_{s_2}^{r-1} \cdot b_{vm_i}}{p_{s_1}^r + p_{s_2}^r}, \tag{22}$$

As can be seen from the above equations, the solution to the problem of each consumer  $vm_i$  in CNCE only depends on the prices of services, namely  $p_{s_1}$ ,  $p_{s_2}$ , and also the initial budget of customer, namely  $b_{vm_i}$ . Now, let us assume that the budget of each user is equal to the market value of the endowment, so  $b_{vm_1} = \mathbf{P} \cdot \mathbf{e}_{vm_1} = p_{s_1}$  and  $b_{vm_2} = \mathbf{P} \cdot \mathbf{e}_{vm_2} = p_{s_2}$ . Recall from Theorem 1 that there exists an equilibrium in which all prices are strictly positive. Taking this into account and also regard to this fact that only *relative* prices matter, leads us to choose a convenient normalization to simplify calculations. Let  $\bar{\mathbf{P}} \equiv (1/p_{s_2}) \cdot \mathbf{P}$ . Here,  $\bar{p}_{s_1} \equiv p_{s_1}/p_{s_2}$  and  $\bar{p}_{s_2} \equiv 1$ . So,  $p_{s_1}$  is just the relative price of the service  $s_1$ . Because each consumer's demand at  $\mathbf{P}$  is the same as the demand at  $\bar{\mathbf{P}}$ , our problem is equivalent with problem of finding an equilibrium set of relative prices,  $\bar{\mathbf{P}}$ . Now, consider the market for service  $s_1$ . Assuming an interior solution, the necessary condition to establish the equilibrium is that the total supply equates the total demand at price  $\bar{\mathbf{P}}^*$ . In other words, we must have

$$d_{vm_1}^{s_1}(\bar{\mathbf{P}}^*, \bar{\mathbf{P}}^* \cdot \mathbf{e}_{vm_1}) + d_{vm_2}^{s_1}(\bar{\mathbf{P}}^*, \bar{\mathbf{P}}^* \cdot \mathbf{e}_{vm_2}) = e_{vm_1}^{s_1} + e_{vm_2}^{s_1}, \tag{23}$$

Considering Eqs. (21)–(22) and the above discussion, this requires

$$\frac{\bar{p}_{s_1}^{* r-1} \cdot \bar{p}_{s_1}^*}{\bar{p}_{s_1}^{* r} + 1} + \frac{\bar{p}_{s_1}^{* r-1}}{\bar{p}_{s_1}^{* r} + 1} = 1, \tag{24}$$

Solving, we obtain  $\bar{p}_{s_1}^* = 1$ . We conclude that any vector  $\mathbf{P}^*$  where  $p_{s_1}^* = p_{s_2}^*$ , equates demand and supply in market  $s_1$ . By Walras' law, those same prices must equate demand and supply in market  $s_2$ , so we are done.  $\square$

Based on above explanations, in order to reach the Walrasian equilibrium price vector  $\mathbf{P}^*$ , the controller of CNCE should increase the price of each service  $s_j$  gradually when  $ED_{s_j}(\mathbf{P}) > 0$ . In contrast, the controller should decrease the price of each service  $s_j$  gradually when  $ED_{s_j}(\mathbf{P}) < 0$ . In this way, after finite number of iterations the market will reach to equilibrium price vector  $\mathbf{P}^*$  in which  $ED_{s_j}(\mathbf{P}^*) \gg 0$  for all  $1 \leq j \leq N_S$ .

**Definition 1** (Walrasian Equilibrium Allocations (WEAs)) Let  $\mathbf{P}^*$  be a Walrasian equilibrium for CNCE with initial endowments  $\mathbf{e}$ , and let

$$\mathbf{d}(\mathbf{P}^*) \equiv \left( \mathbf{d}_{vm_1}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_1}), \mathbf{d}_{vm_2}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_2}), \dots, \mathbf{d}_{vm_{N_{VM}}}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_{N_{VM}}}) \right), \tag{25}$$

where each element  $\mathbf{d}_{vm_i}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_i})$  denotes the demand of virtual machine  $vm_i$  subject to  $N_S$  services at prices  $\mathbf{P}^*$ . The term  $\mathbf{d}(\mathbf{P}^*)$  in Eq. (25) denotes a *Walrasian equilibrium allocation*, or *WEA*.

Now, let us assume that CNCE is a competitive economy with initial endowments  $\mathbf{e}$  as following:

$$\mathbf{e} \equiv (\mathbf{e}_{vm_1}, \mathbf{e}_{vm_2}, \dots, \mathbf{e}_{vm_{N_{VM}}}) \tag{26}$$

Also, let  $F(\mathbf{e})$  denote feasible allocations in CNCE:

$$F(\mathbf{e}) \equiv \left\{ \mathbf{d} \mid \sum_{i=1}^{N_{VM}} \mathbf{d}_{vm_i} = \sum_{i=1}^{N_{VM}} \mathbf{e}_{vm_i} \right\} \tag{27}$$

Now, we outline two significant lemmas without proof. These lemmas will be used in proof of Pareto optimality later.

**Lemma 1** Let  $\mathbf{P}^*$  be a Walrasian equilibrium for CNCE with initial endowments  $\mathbf{e}$  as defined in Eq. (26). Also, let  $\mathbf{d}(\mathbf{P}^*)$  be the WEA of CNCE as defined in Eq. (25). Then  $\mathbf{d}(\mathbf{P}^*) \in F(\mathbf{e})$ .

**Lemma 2** Suppose the demand function of virtual machine  $vm_i$ , namely  $u_{vm_i}$ , is well-defined at  $\mathbf{P} \geq \mathbf{0}$  and is strictly increasing on  $\mathbb{R}_+^{N_S}$ . Also suppose that it is equal to  $\hat{\mathbf{d}}_{vm_i}$ , and that  $\mathbf{d}_{vm_i} \in \mathbb{R}_+^{N_S}$ .

1. if  $u_{vm_i}(\mathbf{d}_{vm_i}) > u_{vm_i}(\hat{\mathbf{d}}_{vm_i})$ , then  $\mathbf{P} \cdot \mathbf{d}_{vm_i} > \mathbf{P} \cdot \hat{\mathbf{d}}_{vm_i}$ ,
2. if  $u_{vm_i}(\mathbf{d}_{vm_i}) \geq u_{vm_i}(\hat{\mathbf{d}}_{vm_i})$ , then  $\mathbf{P} \cdot \mathbf{d}_{vm_i} \geq \mathbf{P} \cdot \hat{\mathbf{d}}_{vm_i}$ ,

In microeconomics, the WEAs include allocations of services to users that lie on the *core* of those economies. Our goal at this moment is to prove that WEAs have this property in arbitrary economies such as CNCE. Let  $C(\mathbf{e})$  denote the set of allocations in the core.

**Theorem 2** (Core and Equilibria in Competitive CNCE Economies) *Consider an exchange economy  $(u_{vm_i}, \mathbf{d}_{vm_i})$ ,  $1 \leq i \leq N_{VM}$ . If each user’s utility function,  $u_{vm_i}$ , is strictly increasing on  $\mathbb{R}_+^{N_S}$ , then every WEA is in the core. Formally speaking,*

$$W(\mathbf{e}) \subset C(\mathbf{e}) \tag{28}$$

**Proof** Simply speaking, the theorem says that if  $\mathbf{d}(\mathbf{P}^*)$  is a WEA for equilibrium prices  $\mathbf{P}^*$ , then  $\mathbf{d}(\mathbf{P}^*) \in C(\mathbf{e})$ . To prove it, we use proof by contradiction. So, let us suppose  $\mathbf{d}(\mathbf{P}^*)$  is a WEA, and  $\mathbf{d}(\mathbf{P}^*) \notin C(\mathbf{e})$ . Since  $\mathbf{d}(\mathbf{P}^*)$  is a WEA, from Lemma 1 it must be that  $\mathbf{d}(\mathbf{P}^*) \in F(\mathbf{e})$ , so  $\mathbf{d}(\mathbf{P}^*)$  is feasible. However, because  $\mathbf{d}(\mathbf{P}^*) \notin C(\mathbf{e})$ , we can find a coalition  $S$  and a new allocation  $\mathbf{d}'$  such that

$$\sum_{i \in S} \mathbf{d}'_{vm_i} = \sum_{i \in S} \mathbf{e}_{vm_i} \tag{29}$$

And,

$$u_{vm_i}(\mathbf{d}'_{vm_i}) \geq u_{vm_i}(\mathbf{d}_{vm_i}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_i})), \quad i \in S \tag{30}$$

Equation (29) implies

$$\mathbf{P}^* \cdot \sum_{i \in S} \mathbf{d}'_{vm_i} = \mathbf{P}^* \cdot \sum_{i \in S} \mathbf{e}_{vm_i} \tag{31}$$

Now from Eq. (30) and Lemma 2, we know that for each  $i \in S$ , we must have

$$\mathbf{P}^* \cdot \mathbf{d}'_{vm_i} \geq \mathbf{P}^* \cdot \mathbf{d}_{vm_i}(\mathbf{P}^*, \mathbf{P}^* \cdot \mathbf{e}_{vm_i}) = \mathbf{P}^* \cdot \mathbf{e}_{vm_i} \tag{32}$$

Integrating over all users in  $S$ , we have

$$\mathbf{P}^* \cdot \sum_{i \in S} \mathbf{d}'_{vm_i} > \mathbf{P}^* \cdot \sum_{i \in S} \mathbf{e}_{vm_i} \tag{33}$$

Equation (33) contradicts Eq. (31). Thus, assuming the proposition to be false leads to a contradiction. Finally, it is concluded that  $\mathbf{d}(\mathbf{P}^*) \in C(\mathbf{e})$  and the theorem is proved.  $\square$

Now, we proceed to state the most important theory which proves that the competitive outcomes, such as CNCE, are Pareto efficient! In microeconomics, this is known as *First Welfare Theorem*.

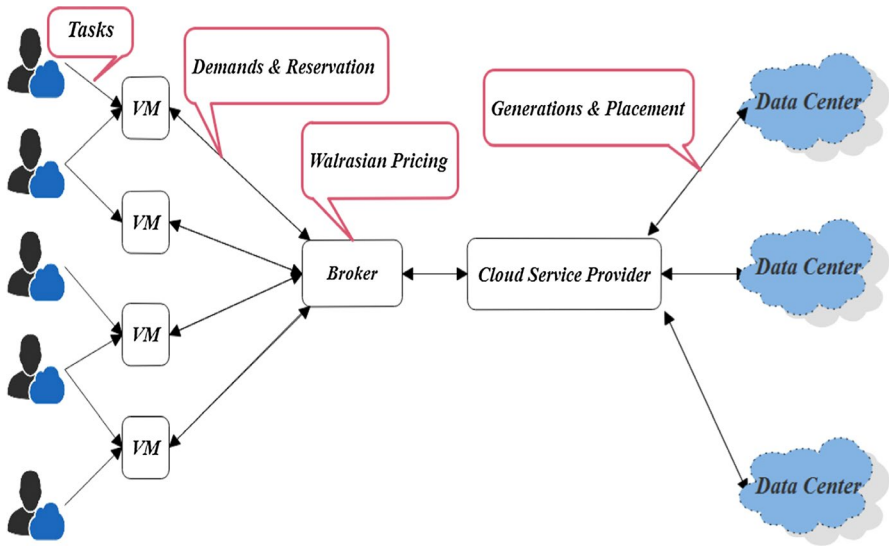


Fig. 1 Major activities in the CNCE architecture

**Theorem 3** (First Welfare Theorem) *Under the hypotheses of Theorem 2, every WEA is Pareto efficient.*

**Proof** In Theorem 2, we proved that every WEA is in the core. Taking this into account and also with respect to this fact that all core allocations are Pareto efficient, immediately proves the theorem!□

Figure 1 demonstrates the CNCE architecture along with the main activities in microeconomic-based bandwidth reservation scheme. As is evident in Fig. 1, in order to control the CNCE market, we have used an entity, named “broker.” The broker node is responsible for computing Walrasian equilibrium price and informing that price to all physical hosts and all VMs. To this end, the broker at first gathers the required information concerning VMs and hosts. This information includes the service demand, budget, and downloading capacity of VMs. Note that the information concerning physical hosts is typically stored on broker. After calculating Walrasian equilibrium Pareto efficient bandwidth rates, the broker sends the demand reservation rates back to VMs.

We now proceed to explain the messaging of the proposed algorithm. Figure 2 shows the primitives and transferred messages between network entities in resource reservation step of CNCE (see upper half of the figure). In summary, it can be said that in the first step of mechanism, the optimal reserved bandwidth rate for every VM is obtained based on the concept of Walrasian general equilibrium. Table 2 represents the bandwidth reservation messages in CNCE with their description and direction among VMs, broker and physical hosts. Also contents of messages are shown in Table 3. As stated before, every virtual machine has a demand rate function which is based on service price. Every virtual machine  $vm_i$  sends a VJR

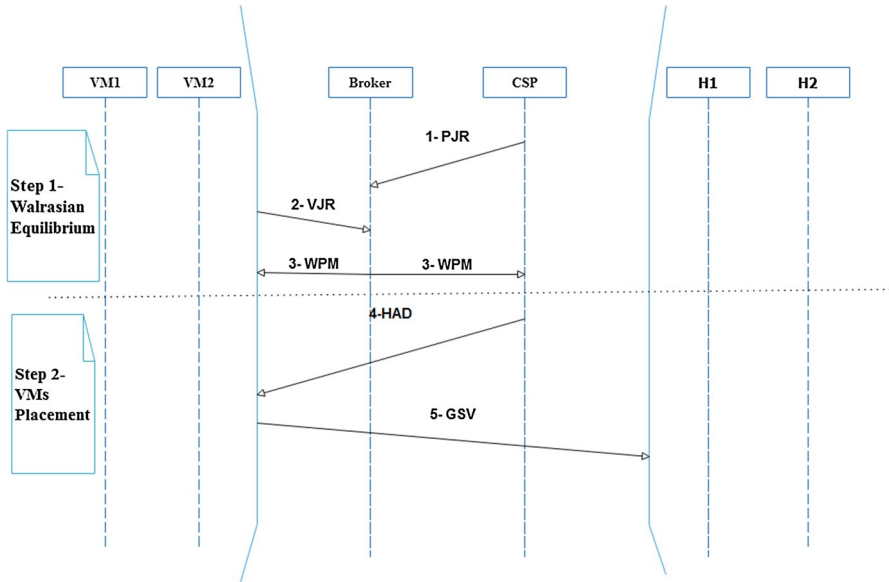


Fig. 2 Primitives of CNCE in both steps: bandwidth reservation step and VM placement step

Table 2 Description and direction of CNCE messages in bandwidth reservation step

Message	Message description	Direction
VJR	Virtual machine joining request	From any virtual machine $vm_i$ to broker
PJR	Provider joining request	From cloud service provider (CSP) to the broker (the entity CSP, holds information of physical hosts)
WPM	Walrasian Pareto message	From broker to CSP and all VMs

Table 3 Contents of CNCE messages in bandwidth reservation step

Message	Message content	Comments
VJR	$\langle \{d_{vm_i}^{s_j}   s_j \in S\}, \langle b_{vm_i}, \langle d_{max}^{s_j}, \langle d_{min}^{s_j} \rangle \rangle \rangle$	Contains service demand, budget, and downloading capacity of any virtual machine $vm_i$
PJR	$\langle \sum_{l=1}^{N_H} g_{h_l}^{s_j}, 1 \leq j \leq N_S \rangle$	Contains the aggregation of services production functions concerning all physical hosts
WPM	$\langle d_{y_m}^{s_j}(\mathbf{P}^*), 1 \leq j \leq N_S \rangle, \langle g_{h_l}^{s_j}(\mathbf{P}^*), 1 \leq j \leq N_S \rangle, \langle \mathbf{P}^w \rangle$	Contains the calculated demands and new price vector for Walrasian general equilibrium



**Table 4** Mathematical symbols and notations used in VM placement step

Symbol	Symbol description
$x_{ij}$	A binary variable which equals 1 if any virtual machine $vm_j$ is placed on the physical host $h_i$ and is zero otherwise
$y_i$	A binary variable which equals 1 if the physical host $h_i$ is selected, and is zero otherwise
$cpu_j$	CPU usage of virtual machine $vm_j$
$cpu_i^{max}$	Maximum affordable CPU by the host $h_i$
$\eta_i$	CPU utilization concerning the physical host $h_i$ .
$pow_i^{busy}$	Consumed power of physical host $h_i$ in active mode
$pow_i^{idle}$	Consumed power of physical host $h_i$ in passive mode
$pow_i$	Consumed power of physical host $h_i$
$mem_j$	Memory usage of virtual machine $vm_j$
$mem_i^{max}$	Maximum affordable memory by the physical host $h_i$
$bw_i^{max}$	Maximum affordable capacity (bandwidth rate) by the physical host $h_i$

message to the broker. A VJR message contains parameters like service demand, budget, and downloading capacity of virtual machine  $vm_j$ . On the other hand, CSP node sends a PJR message to the broker. As is the case in real-world cloud data-centers, the CSP node stores all required parameters of physical hosts. A PJR message contains parameters like the aggregation of production functions subject to all physical hosts. From this point forward, the broker identifies the characteristics of VMs and hosts and indeed can compute the Walrasian general equilibrium optimum price for demands of VMs and hosts. The broker sends the calculated demands to all nodes using WPM message and also modifies the price vector for upcoming requests using the excess demand, as described before in Eq. (8). The CSP, in turn, will send the equilibrium price and other required information back to physical hosts. At this point, the bandwidth reservation step is finished. From this point forward, the VM placement routine will be started.

### 3.2 VM placement problem

We now proceed to describe the VM placement step and its protocol. As discussed before, the bandwidth reservation step uses the producer–consumer theory and also leverages the concept of Walrasian equilibrium. This process determines the optimal demands of bandwidth rates for all VMs. Now, in second step, we aim to place VMs on appropriate hosts in such a way that the consumed power in the DC is minimized. In order to solve the VM placement problem, we use the approach of [28] with some modifications. The description of mathematical symbols and notations which are used in this section is given in Table 4.

Let  $\eta_i$  denote the CPU utilization concerning the physical host  $h_i$ . Similar to [28], we can compute  $\eta_i$  as following:

$$\eta_i = \frac{\sum_{j=1}^{N_{VM}} x_{ij} \cdot cpu_j}{cpu_i^{max}}, \quad x_{ij} \in \{0, 1\}, \quad 1 \leq i \leq N_H, \quad (34)$$

where  $x_{ij}$  denotes a binary variable which equals 1 if virtual machine  $vm_j$  is placed on the physical host  $h_i$  and is zero otherwise. Simply, Eq. (34) aggregates the total CPU usage of all VMs running on physical host  $h_i$ , and then divides the total CPU usage by maximum affordable CPU in host  $h_i$ .

Previous studies, for example [28], have shown that the consumed power of a real-world server scales linearly with CPU utilization. So, we define the consumed power of physical host  $h_i$  with the following equation:

$$pow_i = \left( pow_i^{\text{busy}} - pow_i^{\text{idle}} \right) \cdot \eta_i + pow_i^{\text{idle}} \quad (35)$$

where  $pow_i^{\text{busy}}$  and  $pow_i^{\text{idle}}$  denote consumed power of physical host  $h_i$  in active mode and passive mode, respectively.

Now, we outline the power-aware VM placement problem:

$$\text{Min } f_2 = \sum_{i=1}^{N_H} y_i \cdot pow_i \quad (36)$$

$$\text{s.t. } \sum_{j=1}^{N_{VM}} x_{ij} = 1, \quad 1 \leq i \leq N_H \quad (37)$$

$$\sum_{j=1}^{N_{VM}} x_{ij} \cdot cpu_j \leq y_i \cdot cpu_i^{\text{max}}, \quad 1 \leq i \leq N_H \quad (38)$$

$$\sum_{j=1}^{N_{VM}} x_{ij} \cdot mem_j \leq y_i \cdot mem_i^{\text{max}}, \quad 1 \leq i \leq N_H \quad (39)$$

$$\sum_{j=1}^{N_{VM}} x_{ij} \cdot \sum_{k=1}^{N_S} d_{vm_j}^{sk} \leq y_i \cdot bw_i^{\text{max}}, \quad 1 \leq i \leq N_H \quad (40)$$

$$x_{ij}, y_i \in \{0, 1\}, \quad 1 \leq i \leq N_H, \quad 1 \leq j \leq N_{VM} \quad (41)$$

The problem defined in Eq. (36) seeks to minimize the aggregation consumed power of all selected physical hosts. As stated before,  $x_{ij}$  denotes a binary variable which equals 1 if any virtual machine  $vm_j$  is placed on the physical host  $h_i$  and is zero otherwise. Similarly,  $y_i$  denotes a binary variable which equals 1 if the physical host  $h_i$  is selected, and is zero otherwise. The constraint in Eq. (37) guarantees that each VM is merely placed on exactly one physical host. The constraint in Eq. (38) refers to the CPU usage limitation. The left-hand side expression in Eq. (38) calculates the aggregation of CPU usages subject to all virtual machines which are placed on the same physical host  $h_i$ . The right-hand side expression represents the

maximum CPU which is affordable by the designated host  $h_i$ . Constraints in Eqs. (39) and (40) are similar to Eq. (38), but subject to memory usage and demanded bandwidth, respectively. The term  $\sum_{k=1}^{N_s} d_{vm_j}^{s_k}$  in Eq. (40) calculates the aggregation of required bandwidth which is demanded by virtual machine  $vm_j$  concerning all services. In order to remember the explanations about  $d_{vm_j}^{s_k}$ , refer to Sect. 3.1 and Eqs. (2)–(5).

It has been proven previously that the VM placement problem is NP-hard [44]. Metaheuristic approaches are used extensively to reduce the time complexity of NP-hard problems. In the previous researches like [30], the advantages of cuckoo search optimization (CSO) algorithm, as an efficient metaheuristic approach have been investigated. Authors in [27] showed that the performance of CSO is better than PSO and GA approaches [27]. Therefore, we have used CSO algorithm to solve the VM placement step. As explained in detail in Sect. 3.1, the major contribution of this research is in designing microeconomics-based bandwidth reservation mechanism. However, our minor contribution is modifications which we have applied to CSO for the VM placement step. Due to space limitations, we omit detailed descriptions of VM placement with CSO. Interested readers can refer to [30] to see the full procedure. Here, we only highlight the modifications which we have made to VM placement with CSO. In the CSO algorithm, determination of initial radius of cuckoos (VMs) for oviparity in nests (hosts) is carried out in random space. The randomness of initial radius of cuckoos' nests may force them not to place their eggs in a uniform search space. Therefore, it is likely for a group of cuckoos to locate in the same radius, which in turn, results in slow convergence of the CSO algorithm. We have made some modifications in initial random selection of cuckoos' nests which results in more fairness in distribution of the nests. For this purpose, at first, we determine the number of cuckoos at bootstrap. Then, we divide the search space by the number of initial cuckoos in order to determine the radius for cuckoo oviparity. Using Eq. (42), the difference between upper bound and lower bound is obtained and is divided by the number of cuckoos.

$$\text{Dist} = (\text{UB}_{\text{old}} - \text{LB}_{\text{old}}) / \text{Num Cuckoos} \tag{42}$$

The search space should be changed using the oviparity distance obtained from Eq. (42). Note that the lower bound threshold should not be changed but the upper bound threshold will be equal to lower bound plus the oviparity radius. So, we have:

$$\text{LB}_{\text{New}} = \text{LB}_{\text{old}} \tag{43}$$

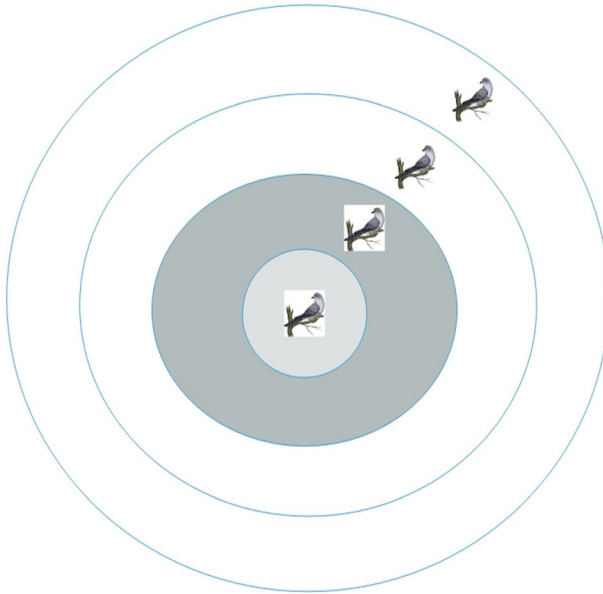
$$\text{UB}_{\text{New}} = \text{LB}_{\text{New}} + \text{Dist} \tag{44}$$

As shown in Fig. 3, for new coming cuckoos, the calculated distance will be added to the upper bound and the lower bound thresholds:

$$\text{LB}_{\text{New}} = \text{LB}_{\text{New}} + \text{Dist} \tag{45}$$

$$\text{UB}_{\text{New}} = \text{UB}_{\text{New}} + \text{Dist} \tag{46}$$

Based on the CSO algorithm, VMs are fairly distributed in search space and after oviparity, some of eggs are destroyed. One of the important sections of the CSO



**Fig. 3** Illustration of the proposed idea for fair distribution of cuckoo eggs across search space before oviparity

algorithm is levy flight of cuckoos for immigration after eggs maturation. Authors in [44] proposed an approach which combines GA with simulated annealing (SA) in order to minimize the energy consumption of the cloud datacenter. This algorithm falls into the category of metaheuristic approaches. Typically, metaheuristic optimization approaches help to avoid trapping in local optima. The main concept of SA has been inspired from annealing process of metals in metallurgy [44]. Figure 4 demonstrates the modifications which we have made in CSO by SA algorithm in order to place VMs optimally.

Some examples of placements in levy flight are demonstrated in Figs. 5 and 6 based on SA algorithm. In each figure, the upper array is resulted from previous steps of CSO algorithm (the number of eggs, oviparity radius and nest selection). In this array, the index of each cell represents the identification of associated VM. Also, the content of each cell shows the identification of physical host.

- (a) *VM migration based on transition operator* as is shown in Fig. 5, by applying *transition operator* in the modified CSO algorithm, at first, one of VMs located on physical machine “1” is migrated to physical machine “2,” and then one of VMs located on physical machine “3” is migrated to physical machine “1.”
- (b) *VM migration based on swap operator*: as it is shown in Fig. 6, by applying *swap operator* in the modified CSO algorithm, physical machines “3” and “5” swap VMs located on themselves with each other. This placement comes up when these two physical machines do not have enough resources (e.g., capacity) to

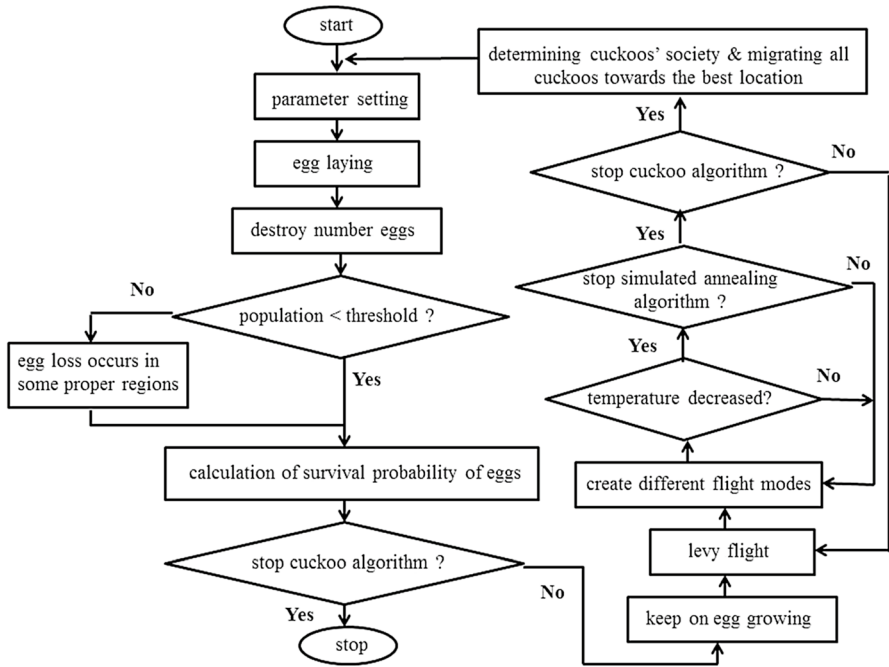


Fig. 4 Flowchart of modified cuckoo search optimization (CSO&SA) algorithm combined with simulated annealing (SA) for VM placement step

Fig. 5 Random placement and migration using “transition” operator with modified CSO algorithm

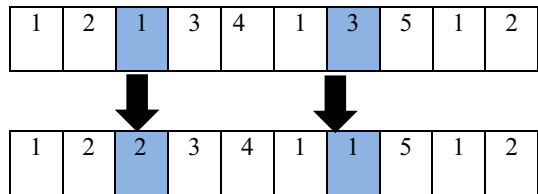
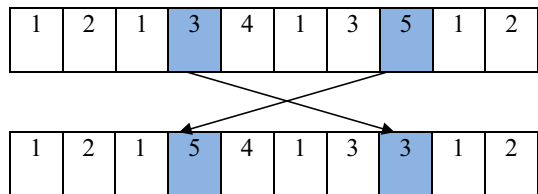


Fig. 6 Random placement and migration using “swap” operator with modified CSO algorithm



admit new virtual machines. The only benefit of swap operation in this state is reducing the consumed power of physical hosts.

We now proceed to explain our modifications to CSO algorithm: if the value of fitness function in the new physical host is better than that of the previous location, the correspondent array will be fed into the levy flight step of CSO algorithm. In this

**Table 5** Description and direction of CNCE messages in VM placement step

Message	Message description	Message direction
HAD	Host address	From cloud service provider to virtual machine
GSV	Give service virtual machine	From virtual machine to physical host

**Table 6** Contents of CNCE messages in VM placement step

Message	Message content	Comments
HAD	$\langle H_i \rangle$	Contains host address
GSV	$\langle \{ \text{Demand}_{vm_i}^{s_j}   s_j \in S \} \rangle$	Contains the amount of demanded service

way, the likelihood of attaining global optimum placements will be raised and CSO algorithm won't be trapped in local optima.

We explained the reservation step of the mechanism, before, in Sect. 3.1 (upper half of Fig. 2). Now, in the lower half of Fig. 2, we represent primitives and transferred messages between network entities in placement step of CNCE. Note that in the second step of the mechanism, each VM can find its appropriate place on physical machines using modified CSO placement algorithm. From now on, we refer to the modified algorithm as CSO&SA. The description of messages transferred between entities in placement step of CNCE is shown in Table 5. Also, the contents of the messages are shown in Table 6. As shown in Fig. 6, in the placement step, the cloud service provider sends HAD messages to VMs. These messages contain the addresses of appropriate physical hosts subject to the optimization goals. Now, each VM refers to the corresponding physical host by GSV message. This message contains the amount of demanded service. The Walrasian equilibrium allocation (resource reservation) and VM placement operations in CNCE are shown in Algorithm 1.

**Algorithm 1** Walrasian equilibrium allocation (resource reservation) and VM placement operations in CNCE

<b>VM:</b>	
1: Send <b>VJR</b> message to broker	/* VJR contains virtual machine joining request */
2: On Receiving <b>HAD</b> message from CSP	/* HAD contains host address $\langle H_1 \rangle$ */
2.1: Send <b>GSV</b> message to host	/* GSV contains the amount of demanded service */
<b>broker:</b>	
1: On receiving <b>PJR</b> message from CSP	/* WPM contains Walrasian Pareto allocation message */
1.1: calculate WEA using Eqs. (6)-(7)	
1.1: send <b>WPM</b> message to CSP	
2: On receiving <b>VJR</b> message from VM	/* WPM contains Walrasian Pareto allocation message */
2.1: calculate WEA using Eqs. (2)-(5)	
2.2: send <b>WPM</b> message to VM	
<b>CSP:</b>	
1: Send <b>PJR</b> message to broker	/* PJR contains provider joining request */
2: On Receiving <b>WPM</b> message from broker	
2.1: Send <b>HAD</b> message to VMs	/* HAD contains host address $\langle H_1 \rangle$ */
<b>host:</b>	
1: On Receiving <b>GSV</b> message from VM	
1.1: find best placement using Eqs. (36)-(41)	/* calculating best placement with CSO&SA algorithm */
1.2: allocate the resource to VMs based on calculated CSO&SA algorithm	

## 4 Performance evaluation

In this section, the performance evaluation of the proposed microeconomics-based system is explained using the well-known CloudSim tool (version 3.03) running on a 64-bit Intel® Core™ i5-8269U Processor with 6 MB Cache, 4 Cores, 4.20 GHz CPU frequency, and 8 GB RAM. The CloudSim was first developed with the aim of modeling and simulating cloud computing systems in Melbourne University [45]. As stated before in Sect. 3, the proposed CNCE system comprises two steps: in the first step the bandwidth reservation is carried out and in the second step, the placement of VMs on physical hosts is carried out. In order to simulate the first and second steps, we use the simulation settings which were introduced before in

[8, 27], respectively. The number of physical hosts in our simulation is fixed at 10. The number of VMs varies according to four scenarios as 20, 30, 45, and 65 virtual machines. The CPU processing rate of each VM varies using a uniform distribution in the interval (400, 700) MIPS. The required memory is chosen as 512 Mega Bytes. The physical hosts of cloud are considered heterogeneous in nature and uniformly distributed in the interval (9000, 10,000) MIPS. Also, the provided bandwidth of physical hosts is considered uniformly distributed in the interval (4000, 5000) bps. Since the CloudSim is a Java-based simulator, we imported Java Optimization Modeler (JOM) library [43] into the CloudSim to solve the above-mentioned optimization problem.

One of the significant metrics in microeconomics is the social welfare of the users (or equivalently the welfare of VMs). In fact, the concept of social welfare reflects the level of quality of service (QOS) in CNCE system. Equation (47) represents the aggregate social welfare (ASW) of VMs:

$$ASW = f_1 = \sum_{i=1}^{N_{VM}} u_{vm_i}, \quad (47)$$

Clearly, the social welfare of each VM depends on its utility value which in turn, depends on its reserved bandwidth. By dividing ASW to the total number of VMs, we can obtain the mean social welfare (MSW) of VMs in CNCE as the following:

$$MSW = \frac{ASW}{N_{VM}} \quad (48)$$

As stated before in Sect. 2, our proposal falls into the category of non-strategic pricing models in the sense that the action of a particular VM in a given time does not affect the actions of other VMs. Clearly, it is better to compare the performance of the proposed method with the research activities carried out in non-strategic scope. For this purpose, we compare the performance of the system with that of CloudSim simulator. This simulator has not any predetermined mechanism for pricing the cloud resources. In Fig. 7, the value of ASW of VMs in CNCE has been compared with that of the non-priced approach of CloudSim using Eq. (47). According to the figure, it is obvious that the value of ASW of VMs increases as the number of VMs increases.

In Fig. 8, the value of MSW of VMs in CNCE has been compared with that of the non-priced approach using Eq. (48). As is evident in Fig. 8, the amount of social welfare is remarkable using CNCE method. The interesting thing in the figure is that the value of MSW in CNCE decreases slightly after a pick with 30 VMs! From the viewpoint of producer–consumer theory, the reason behind this phenomenon is quite clear: Since resources on physical hosts (actually, the producers in CNCE) are limited, when the number of VMs (the consumers in CNCE) exceeds 30, the economic system slightly enters into the saturation state where the resources are no longer sufficient to serve VMs with initial levels of qualities. This, in turn, results in decreasing the demanded bandwidth rates of VMs and decreasing their utility level subject to Eq. (1). Also, regard to Eq. (8) it is clear that as the number of VMs increases, the



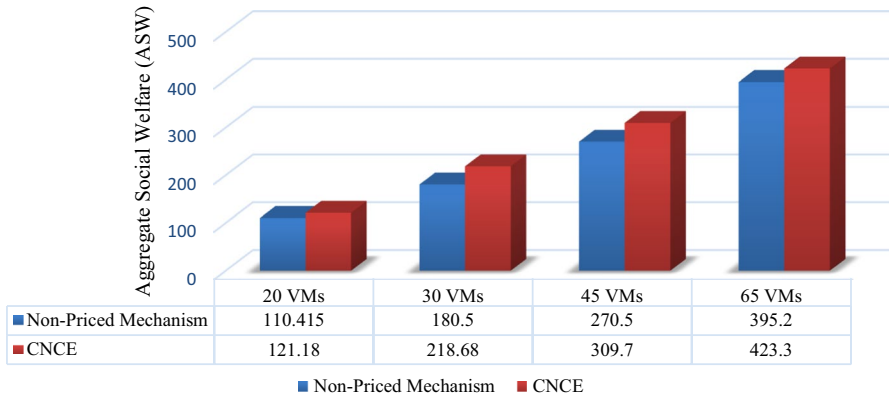
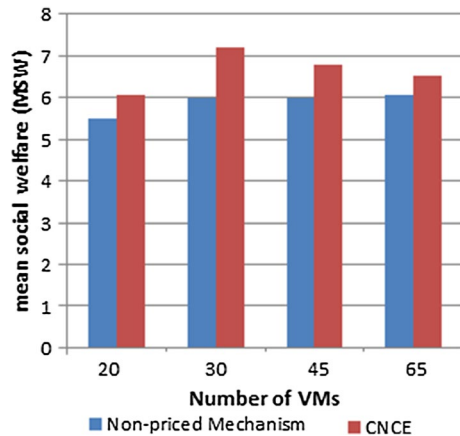


Fig. 7 Aggregate social welfares of VMs in CNCE in comparison with that of non-priced case

Fig. 8 Average social welfares of VMs in CNCE in comparison with that of non-priced case



service prices will experience a gradual rise too. In order to make a better sense of what CNCE mechanism does, in Fig. 9 we have depicted the service price against VMs’ social welfare in the form of radar diagram. As is evident in the figure, when settings of 45 and 65 virtual machines are used, the price of services drastically rises (red color) while, changes in VMs’ social welfare are not noticeable (green color). In other words, CNCE mechanism is scalable in the sense that the system performance does not degrade drastically with the increase in traffic load.

In statistics, *Mann–Whitney U* is a rank-order test. This test is used for assessing the distribution of two independent experimental approaches. It merges two independent dataset and ranks data elements in ascending order, irrespective of their original group. In this manner, the *Mann–Whitney U* test could determine the location and range of each data element within the merged dataset [46]. We have used the well-known SPSS, statistical software tool (version 22), in order to analyze the simulation results [47]. Figure 10 shows the mean rank for two approaches. Note

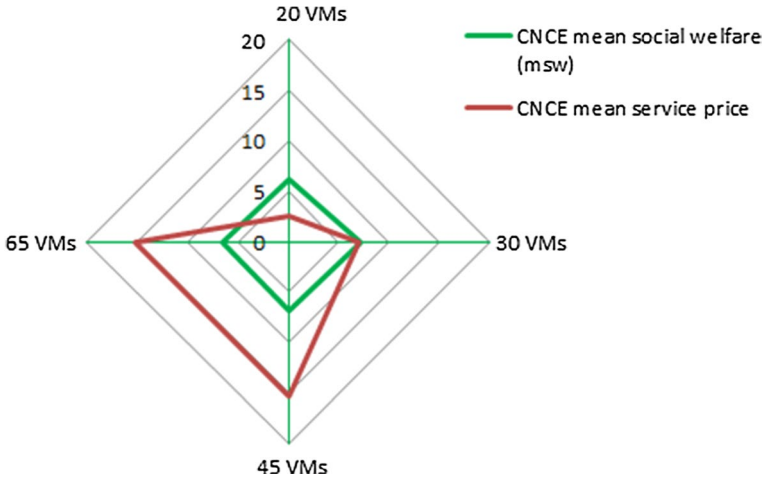


Fig. 9 Scalability of CNCE: system performance does not degrade drastically with the increase in traffic load

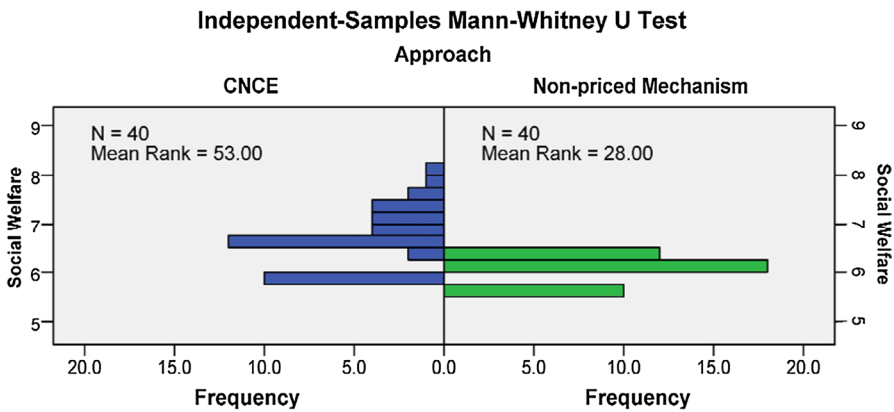
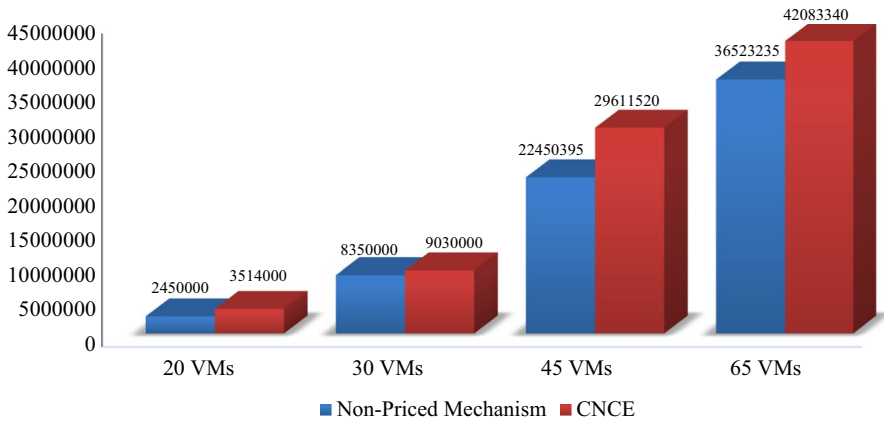


Fig. 10 Mann-Whitney *U* test: CNCE gets more shares of social welfare compared to non-priced case

that for each approach, we have run the experiments 40 times on CloudSim simulator. As is evident in Fig. 10, CNCE has higher mean rank compared to non-priced case. It means that CNCE has a positive and remarkable effect on social welfare of users and succeeds to get more shares.

Figure 11 shows the aggregate revenue of hosts in CNCE in comparison with that of non-priced approach for different number of VMs. Recall from Eq. (6) that physical hosts have the role of producers in CNCE. Clearly, CNCE attains considerably large amounts of revenue compared to that of non-priced case. This is in accordance with Eqs. (6)–(7), where the providers in CNCE try to regulate their production level in such a way that leads to maximization in their revenue.



**Fig. 11** Aggregate revenue of hosts in CNCE in comparison with those of the non-priced case for different number of VMs

Now, let us evaluate the fairness (or distributive justice) in CNCE. Literature survey in microeconomics shows that there exist numerous definitions subject to fairness. Here, we use a simple definition of fairness. For this purpose, at first we define the variance of social welfare concerning virtual machines in CNCE as following:

$$\delta^2 = \frac{1}{N_{VM}} \sum_{i=1}^{N_{VM}} (u_{vm_i} - MSW)^2, \tag{49}$$

Clearly, the less is  $\delta^2$  value, the more justice there exists in the society. In other words, fairness in CNCE indicates the amount of variations of VMs’ social welfare from the average social welfare. Regarding the dependence of this parameter to the rate of social welfare, it can be compared with other mechanisms like Ramsey pricing [25]. Figure 12 shows  $\delta^2$  value obtained from CNCE in comparison with those of the non-priced and Ramsey mechanisms for different number of VMs. As it can be seen from the figure, the proposed method has the least variance in all settings. It is worth mentioning that in Ramsey pricing, there are different prices and hence, the social welfare of VMs have more fluctuations.

Results which we have presented until now belong to the first step of the proposed system. Now, we proceed to discuss the simulation results concerning the second step, namely VM placement. We defined the power-aware VM placement problem, before, in Eqs. (36)–(41). The consumed power of each physical host is considered to be 1000 W and 400 W in active and passive modes, respectively. Figure 13 shows the comparison of average power consumption for different number of VMs and different placement strategies. It is worth mentioning that the developers of Cloud-Sim implemented the instrument using a number of VM placement approaches, the most famous of which are first-fit decreasing (FFD) and best-fit decreasing (BFD) schemes. FFD starts with the most active host and tries to pack every VM in it before going to the next host. If no suitable space is found for the VM, the following

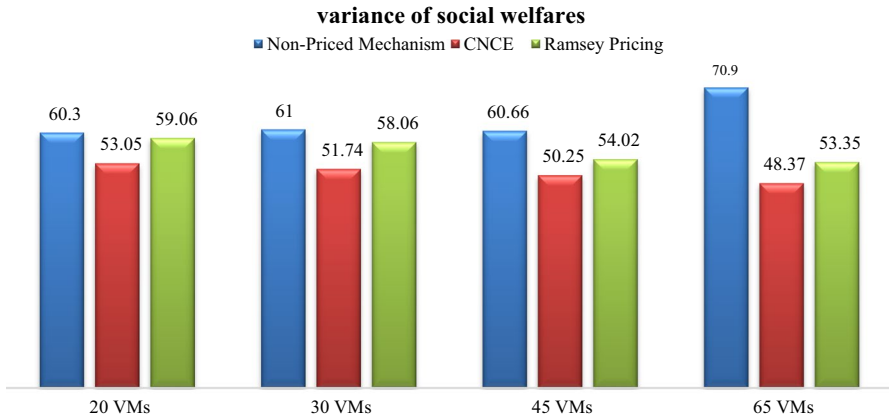


Fig. 12 Variance of social welfares (fairness) in CNCE in comparison with those of the non-priced and Ramsey mechanisms for different number of VMs

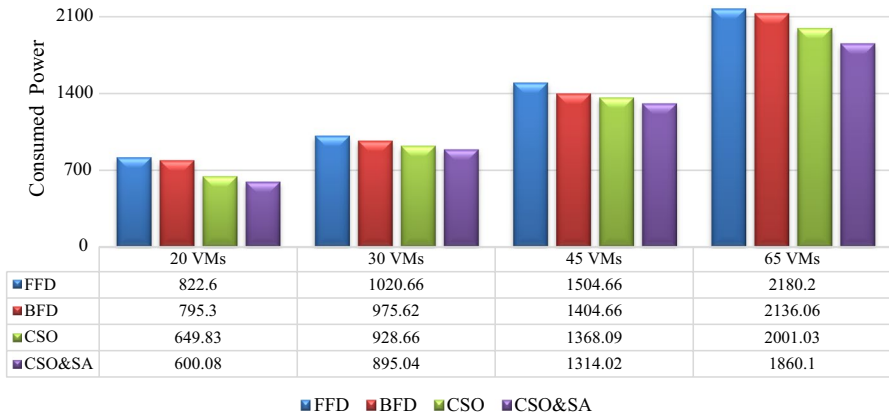


Fig. 13 Comparison of mean power consumption for different number of VMs and different placement strategies

host is selected to be put in the new space. Like FFD, BFD arranges VMs in non-increasing order. Subsequently, it picks up a bin that entails minimum empty space once the VM is packed [48]. We have compared our proposed placement approach (CSO&SA) with original CSO as well as FFD and BFD approaches. Note that since metaheuristic approaches find extensive Pareto efficient solution points in different runs, we have sketched the figure by averaging over 40 runs of the algorithms. The simulation results show that our proposed placement algorithm has less consumed power in comparison with other approaches for all settings of VMs. As discussed before in Sect. 3-2, the reason lies in good distribution of cuckoos as well as creating new opportunities due to applying the SA scheme.

In statistics, analysis of variances (ANOVA) test is used to determine the significant difference between the averages of multiple (more than two) experimental

**Table 7** The ANOVA table for power consumption

	Sum of squares	df	Mean square	F	Sig.
Between groups	108653.400	3	36217.800	.106	.955
Within groups	4101046.358	12	341753.863		
Total	4209699.758	15			

**Table 8** LSD pairwise comparisons subject to consumed power for different approaches: FFD, BFD, CSO, and CSO&SA (Dependent variable: Power consumption, LSD)

(I) Approach	(II) Approach	Mean difference ( $I - J$ )	SE	Sig.	95% Confidence interval	
					Lower bound	Upper bound
FFD	BFD	54.11000	413.37263	.898	-846.5516	954.7716
	CSO	144.83500	413.37263	.732	-755.8266	1045.4966
	CSO & SA	214.42000	413.37263	.613	-686.2416	1115.0816
BFD	FFD	-54.11000	413.37263	.898	-954.7716	846.5516
	CSO	90.72500	413.37263	.830	-809.9366	991.3866
	CSO & SA	160.31000	413.37263	.705	-740.3516	1060.9716
CSO	FFD	-144.83500	413.37263	.732	-1045.4966	755.8266
	BFD	-90.72500	413.37263	.830	-991.3866	809.9366
	CSO & SA	69.58500	413.37263	.869	-831.0766	970.2466
CSO & SA	FFD	-214.42000	413.37263	.613	-1115.0816	686.2416
	BFD	-160.31000	413.37263	.705	-1060.9716	740.3516
	CSO	-69.58500	413.37263	.869	-970.2466	831.0766

approaches. The SPSS uses default value of 95% confidence interval. So, if significance value (abbreviated “Sig.”) is larger than .05 we accept null (basic) hypothesis  $H_0$ . The null hypothesis indicates that the averages of all four approaches (FFD, BFD, CSO, and CSO&SA) are almost equal. Table 7 shows the descriptive statistics of the ANOVA analysis subject to power consumption.

Since Sig. value is greater than .05, the hypothesis  $H_0$  is rejected. It means that there exists at least one approach which its average power consumption differs from that of other approaches! Now, we refer to Table 8, namely Post Hoc analysis. The Post Hoc analysis in Table 8 represents pairwise comparisons among four approaches. Here, we have used Fisher’s least significant difference (LSD) method which is one of Post Hoc test methods [49]. In the first row of Table 8, the FFD approach is denoted by  $I$  and other approaches are denoted by  $J$ . As is evident from the first row of Table 8, the value of  $I - J$  ( $I$  minus  $J$ ) is positive for all comparisons. So, it is concluded that the FFD approach consumes more power compared to other approaches.

Figure 14 shows the average power consumption for all approaches. As is evident from the figure, the proposed approach has the least value of consumed power among all.

Figure 15 shows the Pareto front diagram subject to two objective functions,  $f_1$  and  $f_2$  for different number of VMs and different placement strategies. Recall from

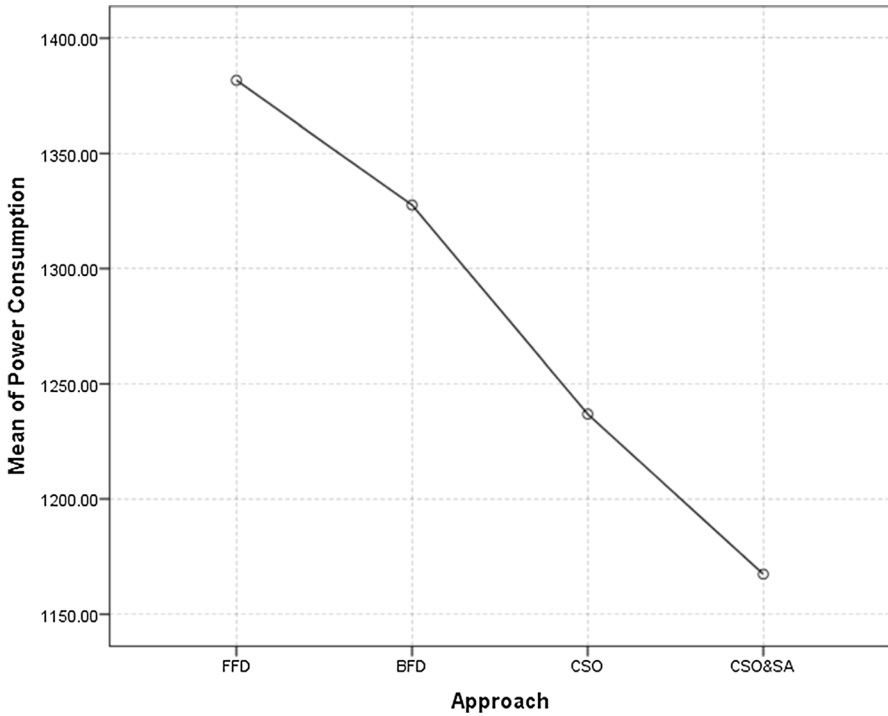


Fig. 14 Average power consumption for different approaches: FFD, BFD, CSO, and CSO&SA

Eqs. (2) and (36) that  $f_1$  and  $f_2$  represent the aggregate social welfare and the consumed power, respectively. As is evident from Fig. 15, our proposed method succeeds to find more Pareto optima points in comparison with other approaches. Also, note that the consumed power decreases with decreasing the number of VMs. The reason behind this phenomenon is that when the number of VMs decreases, the system probably needs to switch on fewer numbers of physical hosts.

Figure 16 shows the comparison of number of required migrations for different number of VMs. As it can be seen from the figure, the proposed placement approach requires fewer numbers of migrations in order to converge to optimum points.

Finally, we present our last evaluation subject to the proposed algorithm. Figure 17 shows the comparison of time complexity for different number of VMs. As it can be seen from the figure, the proposed placement approach has less time complexity in order to converge to optimum points. This type of complexity is based on the time required by each mechanism such as creating initial population and updating solution. The most evolutionary algorithms (EAs) have, at each iteration, a complexity of  $O(d \cdot s + c \cdot s)$ , where  $d$  is the dimension of the problem,  $s$  is the population size, and  $c$  is the cost of the objective function. The time complexity of the most objective functions is  $O(d)$  or even higher. So, usually is the second term which plays the significant role to determine the total time complexity. Furthermore, EAs usually perform  $e$  iterations, where  $e$  is the maximum number of evaluations performed by the objective function. Thus, the time complexity becomes

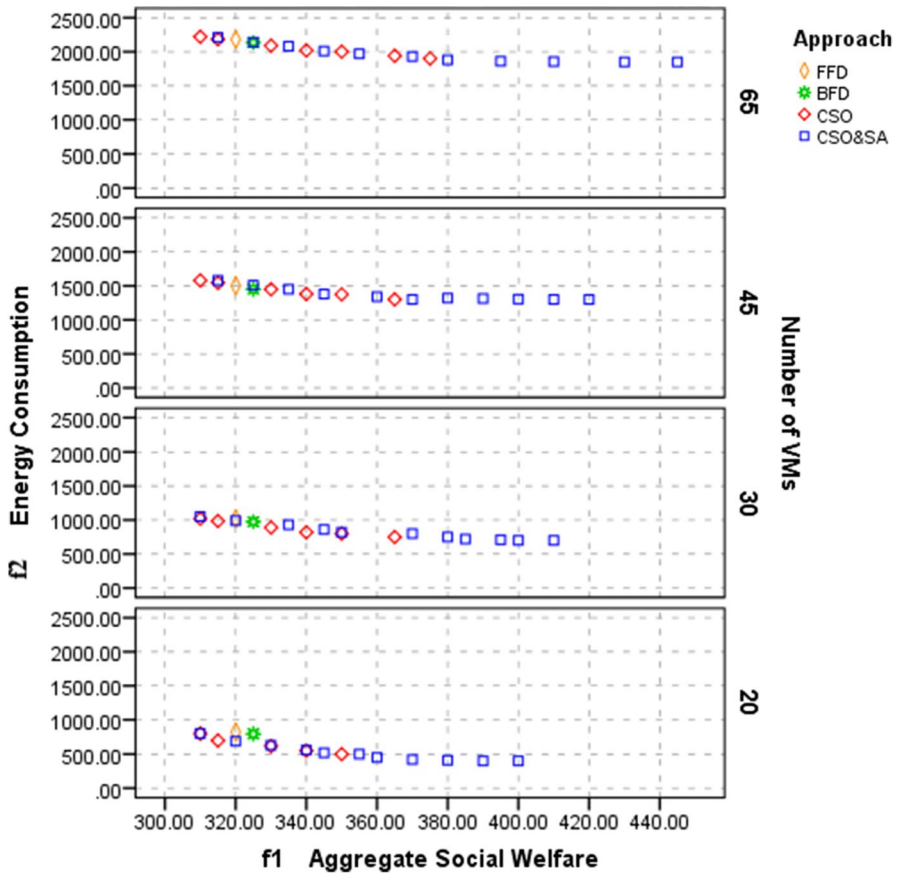


Fig. 15 Pareto front for different number of VMs and different placement strategies

$O(d \cdot e + c \cdot e)$ . Once again, the second term tends to determine the total time complexity. That’s why in EAs, the quality of an algorithm is frequently measured by the amount of evaluations it performs. As discussed previously, we made some modifications in CSO algorithm which reduces the parameters  $c$  and  $e$ . That is why our proposed approach attains lower time complexity in Fig. 17.

### 5 Concluding remarks and future trends

In this paper, we targeted one of the most practical concepts of cloud computing, namely the resource management. This process, itself comprises “resource reservation” and “VM placement” operations in which potential players are VMs and physical hosts. We made a comprehensive review of previous research activities as well as outlining their goals, limitations and criteria. We formulated the resource reservation problem using the theory of producer–consumer of microeconomics.

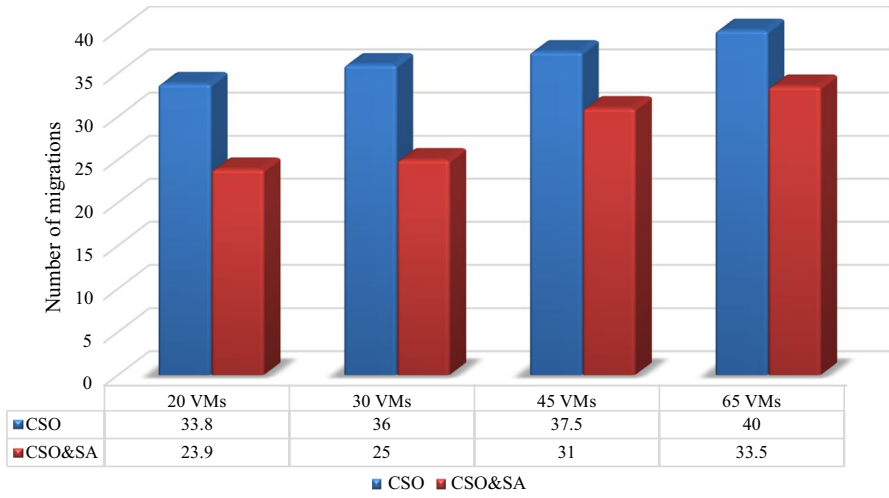


Fig. 16 Comparison of number of required migrations for different number of VMs

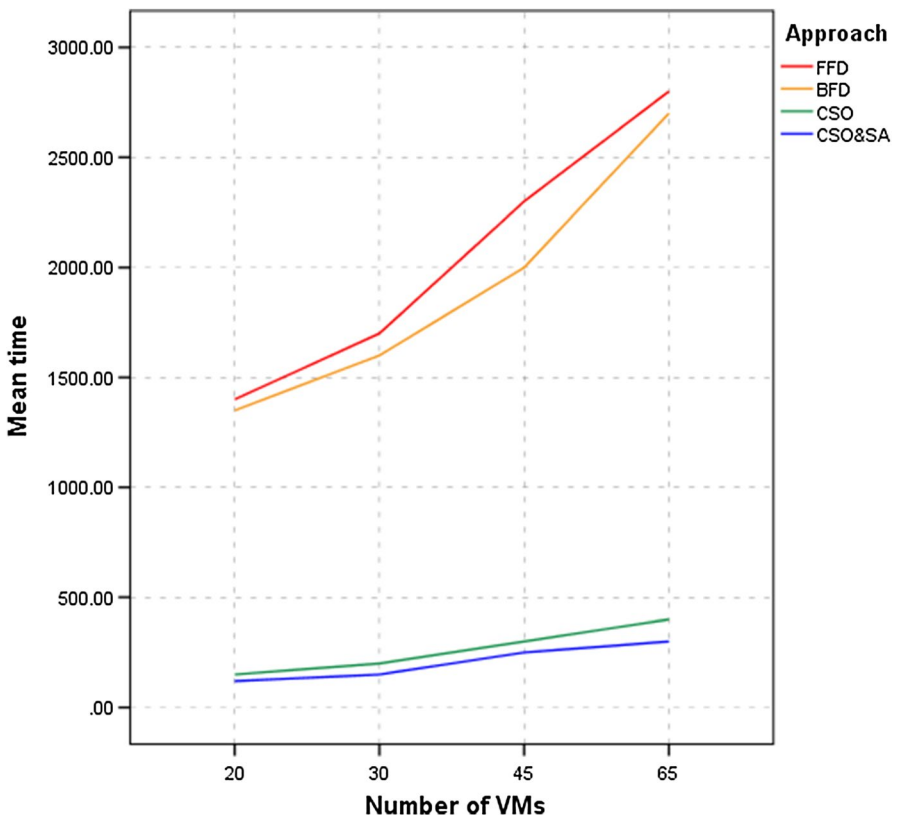


Fig. 17 Comparison of time complexity for different number of VMs



The designed mechanism at first identifies the required bandwidth of VMs subject to Walrasian equilibrium concept, and then tries to find a near-optimal placement for each VM over physical hosts using metaheuristic approach. We proved that the aggregation of users' social welfare is Pareto efficient. The proposed economic system results in maximizing the users' welfare and minimizing the consumed power in physical hosts. The proposed economic system falls into the category of non-strategic economic mechanisms in the sense that the actions of VMs associated with the users do not affect by the actions of other VMs.

In microeconomics, there exist other interesting and challenging non-strategic pricing mechanisms. One of the possible lines of research is to model the interactions between VMs, brokers, and physical hosts by the theory of "exchange economy." Another important field of research in the future is attention to the type of VMs. This will lead to more realistic modeling of cloud entities.

## References

1. Singh S, Jeong Y-S, Park JH (2016) A survey on cloud computing security: issues, threats, and solutions. *J Netw Comput Appl* 75:200–222
2. Kaur A, Kalra M (2016) Energy optimized VM placement in cloud environment. In: *Confluence 2016, 6th International Conference on Cloud System and Big Data Engineering*, pp 141–145
3. Elhabbash A, Samreen F, Hadley J, Elkhatib Y (2019) Cloud brokerage: a systematic survey. *ACM Comput Surv* 51(6):119:1–119:28
4. Alazzam H, Alhenawi E, Al-Sayyed R (2019) A hybrid job scheduling algorithm based on Tabu and Harmony search algorithms. *J Supercomput*. <https://doi.org/10.1007/s11227-019-02936-0>
5. Ferretti M, Santangelo L, Musci MJ (2019) Correction to: optimized cloud-based scheduling for protein secondary structure analysis. *Supercomputing* 75:3521. <https://doi.org/10.1007/s11227-019-02931-5>
6. Singh S, Chana I (2016) A survey on resource scheduling in cloud computing: issues and challenges. *J Grid Comput* 14:217–264
7. Nguyen NC, Wang P, Niyato D, Wen Y, Han Z (2017) Resource management in cloud networking using economic analysis and pricing models: a survey. *IEEE Commun Surv Tutor* 19(2):954–1001
8. Divakaran DM, Gurusamy M, Sellamuthu M (2014) Bandwidth allocation with differential pricing for flexible demands in data center networks. *Comput Netw* 73:84–97
9. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2018) Exploiting task elasticity and price heterogeneity for maximizing cloud computing profits. *IEEE Trans Emerg Top Comput* 6(1):85–96
10. Du B, Wu C, Huang Z (2019) Learning resource allocation and pricing for cloud profit maximization. In: *Published in AAAI*
11. Cong P, Li L, Zhou J, Cao K, Wei T, Chen M, Hu S (2018) Profit-driven dynamic cloud pricing for multiserver systems considering user perceived value. *IEEE Trans Parallel Distrib Syst*. <https://doi.org/10.1109/TPDS.2018.2843343>
12. Baranwal G, Malaviya M, Raza Z, Vidyarthi DP (2018) A negotiation based dynamic pricing heuristic in cloud computing. *Int J Grid Util Comput*. <https://doi.org/10.1504/IJGUC.2018.090230>
13. Wu C, Toosi AN, Buyya R, Ramamohanarao K (2018) Hedonic pricing of cloud computing services. *IEEE Trans Cloud Comput Cloud Comput*. <https://doi.org/10.1109/TCC.2018.2858266>
14. Lee I (2019) Pricing schemes and profit-maximizing pricing for cloud services. *J Revenue Pricing Manag* 18:112. <https://doi.org/10.1057/s41272-018-00179-x>
15. Lee I (2018) Developing pricing strategies for cloud service providers in a competitive market. In: *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, Zurich, Switzerland. <https://doi.org/10.1109/ucc-companion.2018.8653578>
16. Rahman S, Sultana A, Islam A, Whaiduzzaman M (2018) Group based resource management and pricing model in cloud computing. *Int J Comput Sci Inf Technol (IJCSIT)*. <https://doi.org/10.5121/ijcsit.2018.10403>

17. Nan G, Zhang Z, Li M (2019) Optimal pricing for cloud service providers in a competitive setting. *Int J Prod Res.* <https://doi.org/10.1080/00207543.2019.1566655>
18. Shi W, Wu C, Li Z (2018) A shapley-value mechanism for bandwidth on demand between datacenters. *IEEE Trans Cloud Comput* 6(1):19–32
19. Wei W, Fan X, Song H, Fan X, Yang J (2018) Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing. *IEEE Trans Serv Comput* 11(1):78–89
20. Mohammadi A, Rezvani MH (2017) Optimization of virtual machines placement based on micro-economics theory. In: *KBEL'17, Cloud Network, Proceedings of 4th IEEE International Conference on Knowledge-Based Engineering and Innovation*, pp 299–303, Tehran, Iran
21. Kaur A, Gupta P, Singh M, Nayyar A (2019) Data placement in era of cloud computing: a survey, taxonomy and open research issues. *Scalable Comput Pract Exp* 20:377–398
22. Lin B, Zhu F, Zhang J, Chen J, Chen X, Xiong N, Lloret J (2019) A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. *IEEE Trans Ind Inform.* <https://doi.org/10.1109/tii.2019.2905659>
23. Oh K, Chandra A, Weissman J (2017) TripS: automated multi-tiered data placement in a geo-distributed cloud environment. In: *Proceedings of the 10th ACM International Systems and Storage Conference, SYSTOR 2017, Haifa, Israel, May 22–24*, pp 12:1–12:11
24. Ren X, London P, Ziani J, Wierman A (2018) Datum: managing data purchasing and data placement in a geo-distributed data market. *IEEE/ACM Trans Netw* 26:893–905
25. Wanis B, Samaan N, Karmouch A (2016) Efficient modeling and demand allocation for differentiated cloud virtual-network as-a service offerings. *IEEE Trans Cloud Comput* 4(4):376–391
26. Moulik S, Misra S, Gaurav A (2017) Cost-effective mapping between wireless body area networks and cloud service providers based on multi-stage bargaining. *IEEE Trans Mob Comput* 16(6):1573–1586
27. Chowdhury MR, Mahmud MR, Rahman RM (2015) Implementation and performance analysis of various VM placement strategies in CloudSim. *J Cloud Comput* 4:20
28. Jamali S, Malektaji S, Analoui M (2017) An imperialist competitive algorithm for virtual machine placement in cloud computing. *J Exp Theor Artif Intell* 29:575–596
29. Tavakoli-Someh S, Rezvani MH (2019) Multi-objective virtual network function placement using NSGA-II meta-heuristic approach. *J Supercomput.* <https://doi.org/10.1007/s11227-019-02849-y>
30. Sait SM, Bala A, El-Maleh AH (2016) Cuckoo search based resource optimization of datacenters. *Appl Intell* 44(3):489–506
31. Scarpiniti M, Baccarelli E, Naranjo PGV, Uncini A (2018) Energy performance of heuristics and meta-heuristics for real-time joint resource scaling and consolidation in virtualized networked data centers. *J Supercomput* 74(5):2161–2198
32. Vinueza Naranjo PG, Baccarelli E, Scarpiniti M (2018) Design and energy-efficient resource management of virtualized networked fog architectures for the real-time support of IOT applications. *J Supercomput* 74(6):2470–2507
33. Bermejo B, Juiz C, Guerrero C (2019) Virtualization and consolidation: a systematic review of the past 10 years of research on energy and performance. *J Supercomput* 75(2):808–836. <https://doi.org/10.1007/s11227-018-2613-1>
34. Uzaman SK, Khan AR, Shuja J, Maqsood T, Rehman F, Mustafa S (2019) A systems overview of commercial data centers: initial energy and cost analysis. *Int J Inf Technol Web Eng* 14(1):42–65. <https://doi.org/10.4018/jjitwe.2019010103>
35. Feng S, Xiong Z, Dusit N, Wang P, Wang S (2018) Joint pricing and security investment for cloud-insurance: a security interdependency perspective. In: *IEEE Wireless Communications and Networking Conference, Barcelona, Spain, Apr 2018*. IEEE
36. Negi P, Mishra A, Gupta BB (2013) Enhanced CBF packet filtering method to detect DDoS attack in cloud computing environment. *Int J Comput Sci Issues* 10(1):142–146
37. Plageras AP, Psannis KE, Stergiou C, Wang H, Gupta BB (2017) Efficient IoT-based sensor BIG data collection-processing and analysis in smart buildings. *Future Gener Comput Syst* 82:349–357. <https://doi.org/10.1016/j.future.2017.09.082>
38. Stergiou C, Psannis KE, Kim B-G, Gupta B (2016) Secure integration of IoT and cloud computing. *Future Gener Comput Syst.* <https://doi.org/10.1016/j.future.2016.11.031>
39. Gupta BB (2018) *Computer and cyber security: principles, algorithm, applications, and perspectives*. CRC Press, Taylor & Francis, Boca Raton, p 666

40. Gupta BB, Agrawal DP, Yamaguchi S (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Hershey
41. Lee JK, Moon SY, Park JH (2017) CloudRPS: a cloud analysis based enhanced ransomware prevention system. *J Supercomput* 73:3065. <https://doi.org/10.1007/s11227-016-1825-5>
42. Jehle GA, Reny PJ (2001) Advanced microeconomic theory. Addison Wesley Longman, Boston
43. JOM (Java Optimization Modeler). <http://www.net2plan.com/jom/>. Accessed 5 May 2018
44. Liu D, Sui X, Li L (2016) An energy-efficient virtual machine placement algorithm in cloud data center. In: ICNC-FSKD 2016, 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Changsha, China, August, pp 719–723
45. Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw Pract Exp* 41(1):23–50
46. Kowalski J, Tu, XM. Modern Applied U Statistics. Wiley: New York. 2007; 1-378
47. SPSS (1968) Statistical package for social science. <https://www.ibm.com/analytics/spss-statistics-software>. Accessed 5 May 2018
48. Zhao H, Wang J, Liu F, Wang Q, Zhang W, Zheng Q (2018) Power-aware and performance-guaranteed virtual machine placement in the cloud. *IEEE Trans Parallel Distrib Syst* 29(6):1385–1400
49. Fisher GG (2002) Work/personal life balance: a construct development study. Doctoral Dissertation, ProQuest Information & Learning

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Alireza Mohammadi<sup>1</sup> · Mohammad Hossein Rezvani<sup>1</sup> 

✉ Mohammad Hossein Rezvani  
mhossein.rezvani@gmail.com; rezvani@qiau.ac.ir

Alireza Mohammadi  
alireza.mohammadi@qiau.ac.ir

<sup>1</sup> Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran