A Novel Genetic and Intelligent Scheme for Service Trading in IoT Fog Networks

https://doi.org/10.3991/ijim.v16i10.30479

Ayoub Alsarhan[1](✉), Bashar Alkhawaldeh[2],
Atalla Fahed Al-Serhan[3], Muhsen Alkhalidy[2]
[1]Department of Information Technology, Hashemite University, Zarqa, Jordan
[2]Computer Science Department, Hashemite University, Zarqa, Jordan
[3]Department of Business Administration, Al-Bayt University, Al-Mafraq, Jordan
ayoubm@hu.edu.jo

Abstract—The evolution of the current centric cloud to distributed clouds such as fog presents a suitable path to counteract the intolerable processing delays for time-critical applications. It is anticipated that more fog nodes (FN) will be connected to the IoT paradigm to improve the quality of service and meet the requirements of emerging IoT applications. Typically, the owner manages these FN nodes opening up promising doors towards new business opportunities. Thus, this paper considers fog computing driven network that consists of a set of FNs, distributed on the network edge to serve cloud clients. Cloud service provider (CSP), in turn, can offer new services, define a profile for each service, and set generate revenue. However, new schemes should be developed to make this dynamic business model economically feasible. In this context, we propose a new intelligent scheme for service trading, in which a new genetic algorithm is developed for selecting a set of optimal clients that maximize CSP's profit using game theory for setting the service price. Game theory captures the conflict between cloud clients and CSP, where clients and CSP try to maximize their respective utilities. While CSP attempts to maximize profit, each client tries to negotiate for a lower service price. Simulation results stress that the CSP can maximize profit by utilizing computational resources efficiently and selecting service requests with the highest possible bid.

Keywords—fog computing, Internet of Things (IoT), game theory, genetic algorithm

1 Introduction

The Internet of Things (IoT) has been thriving in terms of the number of connected devices/things as well as the applications. The industry and academia anticipate that all of electronics devices can be soon connected to the internet, revolutionizing our life style. [1–5]. As a result, huge computing resources are needed to process the large data generated by all types of IoT devices [1–4]. The past years have witnessed the significant innovation of cloud computing technologies and rapid deployment of practical cloud computing systems. With the tremendous growth in IoT and the dramatic increase in demand for innovative services, new problems have emerged in cloud computing. Indeed, the widespread demand for data and the emergency of new services are inevitably leading to the so-called Resource Crisis. This crisis leads to the failure to meet the requirements of all cloud users in terms of availability, performance, robustness, and so on. Hence, the evolution of the current centralized model of cloud computing to new paradigms (heterogeneous, federated, distributed clouds) presents a suitable path to counteract this crisis by proposing advanced technologies and, most importantly, by adopting new infrastructures to reduce end-user delays, increase the offered services, improve and differentiate their quality. Recently, fog computing has emerged as new distributed cloud computing paradigm where edge devices are used to carry out significant amount of processing, storage, communication locally and routed over the internet backbone [5]. In this paradigm, all computing resources are distributed close to end devices over the IoT network [28–30].

In the smart city applications, extra devices are required sometimes for delivering effective clients service and quick solutions. For example, nearby FNs can be used to maintain proper operation of the smart city services provided in the event of malfunctions or system failures. Each client must pay for CSP to access computing resources [5]. CSP aims to maximize profit by efficiently manage the computing resources. Clients can access computing resources in cloud and able to access to fog's resources if they are close to clients to reduce latency. In the cloud market, in any geographic region, there are multiple CSPs along where clients able to choose the CSP that provides the best service at the lowest price. In order to maximize the profit, the key concern of the CSP is to attract more clients and get large sufficient computing resources to meet the clients' demands. However, CSP cannot serve infinite number of clients since the number of available computing resources are finite. Clients select CSP according to their requirements. It is worth indicating that it is important to examine the economic issues that have a profound impact on the quality of service (QoS) and the cost of service paid by cloud clients [4–5]. CSP has to specify the optimal service price to attract customers as much as possible in line with the demand for service and the expected revenue, which is the basis for the CSP's strategies. The price of service has a significant impact on customer's demand that leads to a cyclical reliance on a traditional supply-demand scenario.

In our work, genetic trading algorithm (GTA) is used to select a set of clients that will be served. We deviate from the notion of static prices for service and allow multiple CSP's to dynamically set service prices based on the service demand in fog computing market. Such a market mechanism is more realistic because there is competition between CSPs to set prices and there is no centralized authority to control the service price. We prove the existence of a price (Nash) equilibrium among competing CSPs, where no user can increase the benefit by changing the price. In summary, the contributions of this paper are summarized as follows:

1) This study, to the best of our knowledge, is the first to explore the emerging market of trading resources in edge computing where the required computation is performed on distributed edge devices close to clients.

2) A game theory based algorithm is proposed to captures non-cooperative interactions among CSP and clients. It takes into account the time-varying availability of computational resources and clients' requirements.
3) A novel, lightweight, and practical scheme is developed for profit-aware dynamic provisioning of fog services with no assumptions and minimal information about service demand. The trading problem is formulated as an optimization problem. For this problem, GA is proposed to select requests that maximize CSP's profit.
4) A model for the rational reaction of clients is proposed. Each client selects his bid without any cooperation or information exchange with other clients. The main concern of client is getting service with the lowest price.

This paper is organized as follows. We describe the system model in Section 2 and show related works in Section 3. We formulate the problems in Section 4 and then we describe our scheme. We evaluate the performance of the proposed scheme in Section 5. Finally, the is concluded and future research directions are given.

2 System model

Fog computing market refers to current state of system, e.g., service demand, service price, and complete description of available resources, etc. In this market, each CSP has some computing resources. The main concern of CSP is hiring these resources to generate extra revenue. Fog computing market is made up of large and small companies with different service demand. The market consist of *N* CSPs that compete for a total clients *M*. Each client submits a request to CSP that represents the client strategy. Each CSP owns *K* computing resources. The request for i^{th} client is represented as follows:

$$S_i = \{d_i, b_i\} \tag{1}$$

where d_i denotes the amount of computing resources requested and b_i denotes the corresponding bid that the i^{th} client is willing to pay. Each client has knowledge of his own bid, but does not have knowledge of the other bids. In our work, fog market parameters change over time in line with the system requirements, such as workload, service demand, and the hiring cost computing resources.

We assume that the client's arrival request follows the Poisson distribution and each request has an arrival rate λ. Service time μ is assumed to be exponentially distributed for each request. Requests are sent via a wireless communication network. The cost of renting a computing resource is *C*. At the beginning of each auction phase, the price decision for each resource is made by CSP. CSP faces the challenge of the winner-determination problem in such a way so that the CSP maximizes profit by choosing a bundle of bidders provided that the total fog resources do not exceed *K*. For clear exposition, the primary notations used throughout the problem description are summarized in Table 1.

Table 1. List of relevant notation

Notations	Description
N	Number of CSPs in the system
K	Number of computational resources
λ	Request arrival rate
μ	service time for each request
M	Number of clients in the system
C	cost of renting computational resource
x_i	number of allocated resources for ith user
S_i	request for ith client
W	number of selected requests for service
d_i	Size of request
b_i	bid for ith client
T	Time horizon
$F(b_i)$	joint cumulative distribution function
N_j	profit from jth request is
m	number of different fog resources
A	allocation vector
x_i	total number of allocated resources for ith user
$\varphi\ (d_i, b_i)$	conditional virtual function for ith request
v_{-i}	truth values for other clients
$O(A)$	objective function for genetic trading algorithm
$O_{-i}(A)$	objective function after accepting ith client's request
$O_{-N_i}(A)$	objective function after rejecting ith client's request
e_i	efficiency of ith resource
v	resource sustainability
$D(A)$	demand function for ith resource
E_i	expected payoff for ith client
b_i^*	optimal bid for ith client
S	remaining capacity
$G(b_i, S)$	genetic trading allocation rule for serving ith client
$q(R, S)$	payment rule for a client i
U_i	the utility for ith client
$L(S_i)$	action space for ith request
B_i	maximum budget for ith client

3 Related work

Fog computing is promising technology that enables delivering functionality to customers by set of low-power fog nodes most expediently. In [6], authors proposed new conceptual framework for fog's computing resource management. The problem was formalized as an optimization problem. The objective function for this problem was defined as providing delay-sensitive utilization of computational resources in fog market. Authors modeled resource management in fog market as a non-cooperative game in [7]. The fog market consists of Infrastructure and Service Provider (ISP), end Service Users (SUs), and Edge Resource Owners (EROs). In this market, edge resources were leased to SUs by ISP. For computational resource trading, a two-stage dynamic game was used. Equilibrium analysis was presented to stress economic benefits obtained by ISP, SUs, and EROs under whatever conditions. In [8], authors presented a new scheme for solving service pricing in fog computing market. Dynamic pricing scheme was presented to enable blockchain-based monetization and automated payment in cryptocurrency for services trading in fog market. Ethereum blockchain was used to govern the interactions between devices and fog nodes. Authors investigated service selection in a fog market where customer choses service provider from two providers (CSPs). M/M/∞ queue and M/M/1 queue were used to model CSPs and clients, respectively. Stackelberg game was adopted to analyze the interaction of the two CSPs and clients where both set the prices first, and then the clients decide to select CSP based on performances and prices. The problem of maximizing the profit of the CSPs was considered in [10]. CSP receives computing requests from clients and serve these requests by leveraging computing service of participating cloudlets. However, maximizing the operating profit for CSP is a challenging problem due to uncertain service demand and complexity in computing resource allocations. Authors proposed the Lyapunov optimization technique combined with the technique of weight perturbation to tackle this problem. The proposed control algorithm makes online decisions to admit requests and to allocate computing resources. Authors developed new distributed market framework for pricing the offloading service in [11]. Furthermore, they conducted a detailed analysis of the incentives for offloading CSP and conflicts arising from the interactions of different participators. Stackelberg game was used to model the interactions between the offloading service providers and the offloading service consumers in the considered market framework.

In [12] authors studied service pricing in cloud market. The problem was formulated as maximization problem where many objectives were taken into account. These objectives include: the energy consumption, delay, and price of cloud services. On the CSP side, authors proposed new pricing strategy by formulating a profit maximization problem. The problem of resource allocation in fog market was studied in [13]. Game theory was used to formulate the competition between clients for service. Each client aims to maximize his own utility, which reflects his satisfaction towards the service in terms of delay. Authors proved the existence of a pure Nash equilibrium.

In [14], authors addressed the tasks offloading from the base station to vehicular fog nodes. They proposed new mechanism for task assignment. The main objective of the proposed mechanism was minimizing the network delay from a contract-matching integration perspective. Pricing-based stable matching algorithm was used for service pricing. In [15], authors proposed new strategy to offload tasks from clients to Fog Nodes. The problem was formulated as a matching game. The main concern of the proposed strategy was minimizing the service time by taking into account both computational and communications costs. In [16], several methods were used for multiple task scheduling in fog computing market. These methods include: Local Regression (LR), Inter Quartile Range (IQR), Local Regression Robust (LRR), Non-Power Aware (NPA), Median Absolute Deviation (MAD), Dynamic Voltage and Frequency Scheduling (DVFS) and The Static Threshold (THR) methods.

Authors in [17] proposed new scheme for spatial task scheduling and resource optimization. The aim of the proposed method was minimizing the total cost of CSP by cost-effectively scheduling all new requests while meeting request' delay-bound constraints. In [18], a new dynamic programming algorithm was introduced for mobile-to-cloud transmission scheduling of the offloaded task's data. The proposed algorithm considers multiple service class where each class has different computing power and different service charge. The problem was formulated as an optimization problem where the main concern of the proposed solution is minimizing the cost of service. Novel crowdsourcing-based QoS supported mobile cloud service framework was introduced in [19]. The proposed framework meets clients' satisfaction by sensing their context information and providing appropriate services to each of the clients. The framework adapts cloud service dynamically based on client's activity context, social context, service context, and device context. In [20], authors introduced new framework for fog service provisioning. The main objective of the proposed framework is meeting low latency and QoS constraints while minimizing the service cost. Authors in [21] presented new multi-objective framework to find suitable fog nodes for serving clients' requests. The main objective of the proposed framework is achieving a trade-off between the cost of resources and average service delay. The problem was formulated as a mixed-integer linear programming and it solved using the weighted goal programming. In [22], authors studied the task scheduling algorithm using a hybrid approach. The proposed scheme combines two of the most widely used biologically-inspired heuristic algorithms, the genetic algorithms and the bacterial foraging algorithms in the computing cloud. The key objective of the scheduling algorithm was minimizing the makespan and reducing the energy consumption. The aim of this paper is to provide an effective trading algorithm that makes optimal use of the resources to maximize CSP's profit. Unlike other approaches, the resource allocation problem is formulated as maximization problem for each user to make its computation offloading decision. In order to maximize the objective function, the CSP selects winners using a dynamic genetic algorithm. Clients in the market bid to acquire some computational resources. Game theory is used to model the conflicting objectives between players in the game. This market is more realistic since multiple clients deliver multiple tasks to CPS. It is assumed that no centralized authority exists in this market to decide the service price.

4 Fog computing resources trading using GTA

In this section, we introduce the computing resources trading model in fog computing market. In this market, CSP is greedy and seeks for the maximum profit. Clients compete among themselves to get service from CSP in lowest price. The resources for the CSP are computing resources that have been statically allocated. Clients, on the other hand, get service depending on the benefits they wish to obtain and the prices they pay for CSP.

4.1 CSP profit function

The problem of trading computing resources can be modeled as the classical knapsack problem, where the objective is to fill a sack of finite capacity with several computing resources such that the total profit of the selected requests in the sack is maximized. The requests are served if the total demand for service is less than market capacity (i.e. number of resources). GTA is applied if the demand is exceeded the number of computing resources. The CSP has a "knapsack" of given capacity $K \in$ R that he wants to fill with clients' requests in a profit maximizing way in at most $T < \infty$ period. Each new request is characterized by the bid (i.e., weight of request), and the number of required resources (i.e., size of request). We use a non-cooperative game to model the price competition among players of the game.

Definition 1. A non-cooperative game has two types of players: CSPs and clients. CSP chooses his strategy first (i.e. service price) and then clients make decision sequentially. The strategy of each of the players is the service price. We assume CSP knows the bids of clients at each decision epoch. CPS sets the service price based on demand and clients' bids. While CSP attempts to maximize his own profit, clients try to pay less.

The request information is private for each client but its distribution is assumed to be known which is the joint cumulative distribution function $F(b_i)$ with continuously differentiable density $f(b_i) > 0$, However, service demand is unknown and it is independent over time. CSP serves requests based on the bid and the size of request (i.e. number of required fog resources). The generated profit from j^{th} request is computed as follows:

$$N_j = \sum_{i=1}^{m} (d_i b_i - C_i) \tag{2}$$

where C_i is the cost of renting fog i^{th} resource and m is the number of different fog resources.

Definition 2. A general fog computing market is one where the service provider (CSP) must pay a service cost C_i for the allocation i^{th} resource.

Each client has a single private value for getting some computing resource and quasi linear utility. The utility for each client is the truth value for a computing resource which should be less than the service price. The outcome of the GTA is the allocation vector which can be represented as follows:

$$A = \{(x_1, N_1)\ (x_2, N_2), \ldots, (x_n, N_n)\} \tag{3}$$

where x_i is the total number of allocated resources for i^{th} user. The conditional virtual function for i^{th} request is computed as follows:

$$\varphi(d_i, b_i) = b_i - \ (1 - \frac{F(b_i \mid d_i)}{f(b_i \mid d_i)}) \tag{4}$$

Definition 3. A general feasibility fog market is one where there is feasibility constraint for serving clients. For GTA, the feasible solution is a set of clients whom cumulative payments are more than the cost of service and serving this set does not violate any constraint in the market. The main concern of GTA is finding the allocation vector that maximize the profit. The problem of allocating fog resources can be formulated as follows:

$$\max_{A} \sum_{j=1}^{W} N_j \tag{5}$$

subject to:

$$\sum\nolimits_{j=1}^{W} \sum\nolimits_{i=1}^{m} d_i \leq K$$

Lemma 1. In a fog market, for i^{th} client and all truth values for other clients v_{-i}, the objective function for GTA for i^{th} client is a step function.

Proof: CSP either serve i^{th} client or reject his request using GTA. We write the objective function for both decisions (i.e. serving or rejecting the request). The objective function for GTA after accepting the i^{th} client can written be as follows:

$$O(A) = N_i + \sum\nolimits_{j=1}^{W-1} N_j \tag{6}$$

The objective function can be written as follows:

$$O(A) = N_i + O_{-i}(A) \tag{7}$$

where $O_{-i}(A)$ is the objective function for finding the optimal allocation vector that maximize CSP's profit after accepting i^{th} client's request. Clearly, $O_{-i}(A)$ is not function of N_i. However, if the i^{th} request is rejected, the objective function can be written as follows:

$$O(A) = O_{-N_i}(A) \tag{8}$$

Notice that $O_{-N_i}(A)$ is not function of N_i. GTA selects i^{th} client based on the generated profit. Therefore, GTA admits i^{th} request when:

$$N_i + O_{-i}(A) \geq O_{-N_i}(A) \tag{9}$$

Solving for N_i, the i^{th} request is served whenever:

$$N_i \geq O_{-N_i}(A) - O_{-i}(A) \tag{11}$$

Clearly, the admission policy of CSP for any request i is a step function which can be written as follows:

$$o(i) = O_{-N_i}(A) - O_{-i}(A) \tag{12}$$

The step function describes the impact of serving i^{th} client on other clients. Serving i^{th} client reduces the offered resources for other clients. Hence, the main concern of GTA is profit maximization where a client is served if and only if the total profit is maximized and this action causes the externality impact on other clients (i.e. reducing number of fog resources).

Theorem 1. The profit maximization problem in fog computing market using GTA is dominant strategy incentive compatible.

GTA maximizes the profit for a CSP for all possible strategies (i.e. bids) of clients. The algorithm makes no assumption about the information available to clients about others. GTA solicits clients in a truth-manner, and it does not have knowledge of the service demand in future.

Definition 4. In fog market, CSPs are correlated since their actions affect the profit of each other.

4.2 Client strategies

The service demand increases, if the computing resources available to the CSP creates high utility for clients. The utility of service reflects the actual value that the client is willing to pay to the quality of service. The following quadratic function is used to quantify the utility gained by a client [27]:

$$U(A) = \sum_{i=1}^{n} d_i e_i - \frac{1}{2}\left(\sum_{i=1}^{n} d_i^2 + 2v\sum_{i\neq j} d_i d_j\right) - \sum_{i=1}^{n} b_i d_i \tag{13}$$

where e_i is the efficiency of i^{th} resource, and v is the resource sustainability ($0.0 \le v \le 1$). The sustainability resource reflects the ability of using other resources. For example, when $v = 0$ a client cannot switch to use another computing resource. However, a client can freely switch to any resource for $v = 1$. The demand function for i^{th} resource can be extracted by computing the first derivative utility function as follows:

$$D(A) = \frac{(e_i - b_i)(vN) - v\sum_{i\neq j}(e_j - b_i)}{(1-v)(v(N-1)+1)} \tag{14}$$

Generally, if the utility derived from the i^{th} resource is still greater than the bid b_i, the client will pay for the service. The optimal bid for i^{th} client b_i^* should be the minimum price the CSP should accept. Obliviously, CSP declines lower bids. Thus, the bid for i^{th} client is the measure of contention among the fog market clients and shows the marginal increase in utility by purchasing the service.

Lemma 2. Dominant strategy of the i^{th} client is the bid b_i^* that increase the likelihood of winning the auction.

Proof. The probability that the i^{th} client offers the highest bid is computed as follows:

$$p(b_i^*) = F(b_i)^{n-1} \tag{15}$$

The request for i^{th} client will be admitted as the offer is equal or greater than the minimum acceptable bid. Assume t_i is the truth value for a service. The expected payoff for i^{th} client is computed as follows:

$$E_i = \sum_{j=1}^{m} (d_i t_i - d_i b_i) \tag{16}$$

The optimal bid for i^{th} client is computed as follows:

$$b_i^* = \sum_{\forall j \in W} N_j - \sum_{i\neq j, \forall j \in W} N_j \tag{17}$$

We assume that i^{th} client does not submit his optimal bid b_i^*. Accordingly, the client has two options for selecting the bid:

1. The bid is less than the optimal price b_i^* and the request will be rejected since CSP selects only the highest bids. Thus, the client's payoff will be 0.

2. The bid is more than the optimal price b_i^* and the request will be granted but with less payoff for the client. So, i^{th} client will not be interested to increase the offer.

Clearly, if the i^{th} client's request is admitted, the maximum payoff is E_i and bidding any other price will decrease the payoff. Thus, the dominant strategy of the i^{th} client is the bid b_i^*. At any period t, the GTA is called deterministic and Markovian if it uses nonrandom rules for resource allocating that depends only on the bids, and on the still available capacity denoted by S.

For each request i, the GTA rule can be described as follows:

$$G(b_i, S) \to \{1, 0\} \tag{18}$$

where 1 (0) the request is served (rejected). Indeed, the request is served when the quantity of fog computing resources are adequate to serve the request and the customer is willing to pay for this. However, CSP will not allocate more resources than the requested amount since this action will not increase the profit of CSP. The payment rule for a client i can be expressed as follows:

$$q(R,S) \to Sb_i^* \tag{19}$$

Lemma 3. GTA is a deterministic, Markovian allocation policy if and only if for every request, the following two conditions are met:

(i) For all (b_i, S), $b_i' \geq b_i$, $G(b_i, S) = 1$, $G(b_i', S) = 1$.
(ii) The payment function Sb_i^* is nondecreasing in S.

When the two conditions are met, the payment function can be expressed as follows:

$$q(R,S) = \begin{cases} Sb_i^*, & if\ G(b_i^*, S) = 1 \\ 0, & if\ G(b_i^*, S) = 0 \end{cases} \tag{20}$$

Proof of Lemma 3. There are two cases for i^{th} request:

(1) $G(b_i, S) = 1$, where $b_i < t_i$. Then, the reported profit N_i. However, if the client offers b_i which greater than b_i and CSP will admit the request since it will generate more profit. However, the client's payoff will decrease according to Eq. (16). Hence, b_i' is not profitable bid. Consequently, equation (17) can be rewritten as follows:

$$b_i^* = inf\left\{G\left(b_i, S\right) = 1\right\} \tag{21}$$

(2) $G(b_i, S) = 0$, where $b_i \geq t_i$. Assume the client submits b_i'. If $(b_i', S) = 0$, then the utility for client is 0 and the offer will be rejected. Assume now that $b_i' < b_i$. By the form of the profit function for a CSP, the bid b_i' will be rejected since it reduces the CPS's profit. Hence, the client should offer the optimal bid for guaranteeing the approval of his request.

Assume first, by contradiction, that condition (1) in the lemma is not satisfied. Then there exist (b_i', S) and (b_i, S) such that $b_i' > b_i$, $G(b_i, S) = 1$, and $G(b_i', S) = 0$. Then, the following inequality is obtained:

$$b_i' S - C_i \geq b_i S - C_i \tag{22}$$

Selecting b_i that generates less profit than b_i' contradicts the implementability of GTA. Therefore, condition (1) must be held. Now, assume condition (2) is not satisfied. Then there exist b_i, and b_i' such that $b_i' > b_i$ but $b_i S > b_i' S$. In order to maximize his utility, the i^{th} client offers b_i to get the utility U_i:

$$U_i = t_i S - b_i S \tag{23}$$

Therefore, condition (2) holds for this case.

Theorem 2. Game theory converges to a pure Nash equilibrium with the highest possible payoff, in the pricing game, when the GTA is applied.

Proof. Let the current state of bids be $b = (b_1, b_2, b_3, \ldots, b_m)$ and GTA is applied to select the winners. Without loss of generality assume that CSPs indexes imply the players , i.e., only CSPs 1 to $N - 1$ are correlated, meaning that their decisions affects the profit of each other. Then, if i^{th} CSP selects b_i^* using GTA at first round of auction that is not affected by other CSPs in the fog market. Clearly, all CSPs in the market try to maximize their profit by selecting the optimal bid. Hence, the total profit of all CSPs will be the maximum one. The total profit of all CSPs in fog market after b_i^* is determined by i^{th} CSP can be expressed as follows:

$$b_i^* + \sum_{i \neq j, \forall j \in W} b_j \geq \sum_{\forall j \in W} b_j \tag{24}$$

Clearly, if prices for other CSPs are optimal, then the price b_i^* is the optimal one for the i^{th} CSP since it will maximize the total expected profit. Hence, CSPs will not change their prices in the future. The same is true for clients in fog market.

4.3 Computational resource trading problem using genetic trading algorithm

Genetic trading algorithm tries to maximize the reported profit by selecting a subset of requests to be served so that the total amount of requested resources is less than or equal to a given limit and the total profit is as large as possible. For each request, CSP either accept or reject the request. The action space for i^{th} request is given by:

$$L(S_i) = \{a_i : a_i \in \{0, 1\}\} \tag{25}$$

where $a_i = 0$ denotes request rejection, $a_i = 1$indicates that the CSP accepts serving i^{th} request. Every request has a size weight d_i and a value b_i. The goal is to maximize the profit of serving optimal requests. Mathematically, the problem can be formulated as follows:

$$\max \sum\nolimits_{j \in w} \sum\nolimits_{i=1}^{m} (d_i \; b_i - C_i) \tag{26}$$

$$\text{s.t.} \; \sum\nolimits_{j \in w} \sum\nolimits_{i=1}^{m} d_i \leq K$$

$$\sum_{i=1}^{m} b_i \leq B_i$$

B_i is the maximum amount that the i^{th} request can pay for CSP (i.e., CSP's budget). Genetic algorithm has many applications such as capital budgeting, inventory control routing and project scheduling [26]. John Holland [24] introduced genetic Algorithm (GA) in the 1960s. GA is heuristic search algorithm inspired by the process of natural selection and genetics developed. In order to use the GA for determining the optimal subset of requests that contribute to maximize CSP profit, we first need to represent them in a binary string. In GA, the requests belong to $\{1; 0\}$, and the binary representation is sufficient. In evaluation step of the GA the quality of each candidate solution in the population is determined. In our work, the value of the objective function is defined in Eq. (5). In the selection phase of GA, the selection operator performs the following tasks:

- Extracting good solutions in the current population.
- Generating multiple copies of the good solutions.
- Eliminating bad solutions from the current population.

In order to generate new population, the selection operator chooses a chromosome from the current generation's population. Several algorithms have been proposed for selecting a good solutions [24–25]. In our work, Tournament technique [24] is adopted for selecting good solution. The crossover operator creates the new solutions by combining the genes of one individual with those of another. New individuals inherit characteristics of parents. We use the two- and three-points crossover [25]. In mutation phase, the worst genes in term of profit are selected randomly. Then, they are changed oppositely to create new population. The mutation does not perform when the corresponding solutions generate a good profit.

5 Performance evaluation

In this section, the performance of the proposed scheme (GTA) is evaluated and compared with greedy auction scheme [23]. The requests in greedy scheme are served based on bids and on the free number of resources. The client with the highest bid wins the auction. The GTA and greedy schemes are implemented in the C language. In this section, the performance of the proposed scheme is evaluated through simulation. The uniform distribution is used to generate the parameters for each request (i.e. number of required resources, and bid). Each simulation run consists of 100000 requests. The results are averaged over enough independent runs so that the confidence level is 95% and the relative errors do not exceed 5%. We examine the performance under different parameter settings.

5.1 Simulation results

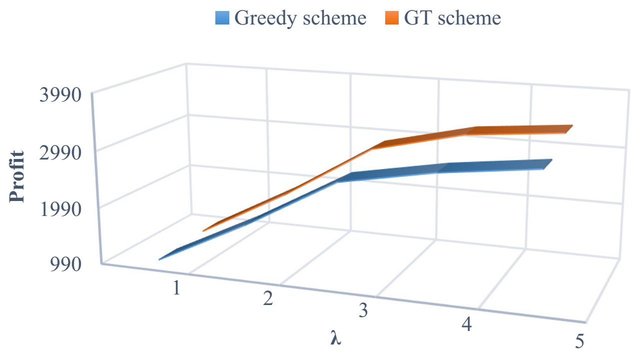

Fig. 1. CSP's profit under different system loads

Figure 1 shows the effect of varying load on the reported profit for GTA and the greedy scheme. The figure shows that the proposed scheme always achieves more profit than greedy scheme. GTA scheme outperforms greedy scheme because it spans all the possible combinations of fitting requests and finds the one with the highest total profit. The greedy scheme selects the highest bid at any time and continue to serve requests until no more available resource. GTA scheme spreads the net widely in the search space (i.e. a request queue). It tries all possible subsets of requests and select the subset

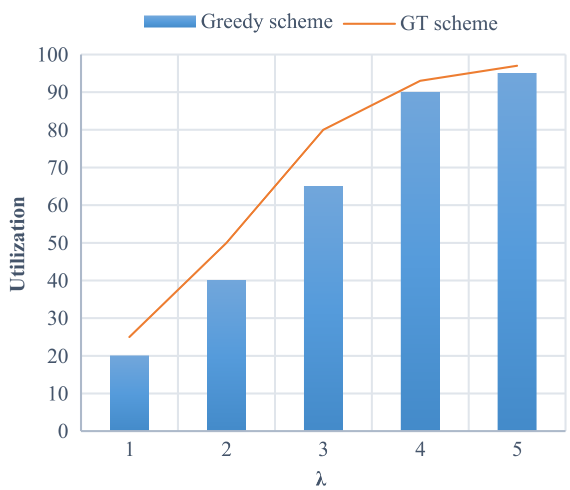

Fig. 2. Resource's utilization under different system loads

with highest total profit. Note that, the profit is low in the low values of load (i.e. request arrival rate) and subsequently increases with high demand for service. For lower service demand, the competition for service is low and bidders are dubious. Therefore, they offer low bids. With increase in competition, potential bidders emerged as expected and raised the reported profit. We observe that the proposed auction generates 5–10% more profit compared to the greedy scheme. Higher profit requires high competition among clients (i.e. increment in the number of requests). Figure 1 shows that the profit increases for both schemes as the number of served requests increases (load) but after certain number of load the profit reaches the steady state since the available resources are fully utilized. We further present the results of resource utilization with different system loads in Figure 2. Our scheme performs better than greedy scheme. Our scheme utilizes the whole resources because it considers both the number of resources and the price for each resource. It determines the number of each resource to include in a collection so that the total number is less than or equal to a given number of resources. This improves the reported profit significantly. Greedy scheme always favors requests with high request size (i.e. amount of resources required) and high bid, thus discouraging low potential bidders.

Figure 3 shows the average wining price under different system loads. We observe that the greedy scheme always prioritizes requests with higher bids with ignoring

Fig. 3. Average wining price under different system loads

selecting optimal subset of requests that generate maximum profit. Although the average of winning price for greedy scheme is higher than GTA, it fails to yield optimum benefit.

Next, we evaluate the scalability of our scheme for a large fog market. The profit comparison of the two schemes is shown in Figure 4 that shows the impact of resources quantity on the profit of GTA and greedy schemes. It is clear that the profit increases as the number of resources quantity increases. Since GTA utilizes the free resources

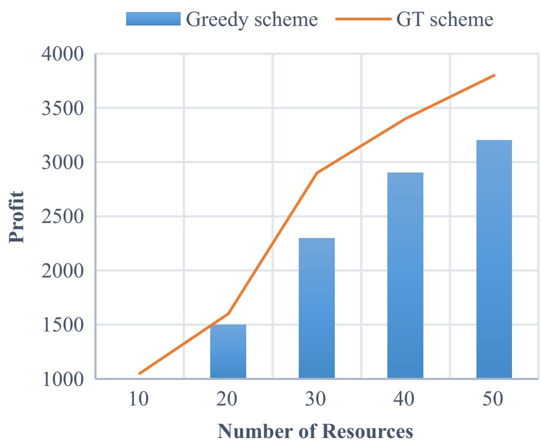

Fig. 4. CSP's profit under different market sizes

efficiently, it generates more profit than greedy scheme. Clearly, the effective/admitted rate of GTA is strictly higher than that of greedy scheme, since GTA takes into account the sizes of requests when allocating resources and prices. The likelihood of finding a set of requests with highest total profit increases as the number of resources increases. GTA trades its resources for a greater number of clients that leads to a higher profit as we see in Figure 4.

We illustrate greedy and GTA schemes' performance as we increase the service demand in Figure 5. The figure shows the impact of service demand on the acceptance rate of client's requests. Clearly, the acceptance rate of GTA is strictly higher than that of greedy, since GTA utilizes the whole resources for serving clients. Clearly, there are more requests served by GTA, which lead to a higher profit and higher utilization of resources. For high service demand, the acceptance rate decreases significantly for both schemes because the number of resources insufficient to accommodate large number of requests.

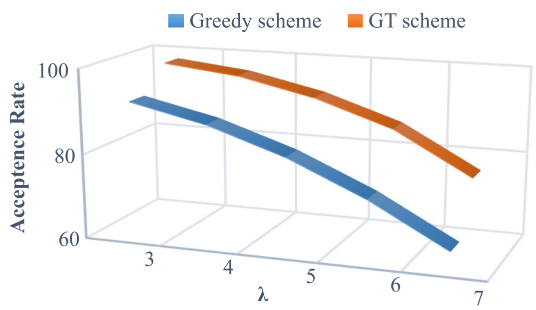

Fig. 5. Request's acceptance under different system loads

We plot the lowest winning price of i^{th} CSP competitors against profit for i^{th} CSP in Figure 6. The results illustrate the impact of CSPs' prices on the profit of i^{th} CSP. Clearly, the profit of GTA is strictly higher than that of greedy, since GTA considers the competition with other CSPs and utilizes the whole resources for serving clients. A higher price0 rival increases significantly the profit of i^{th} CSP for both schemes because clients wish to pay less for service for maximizing their utility.

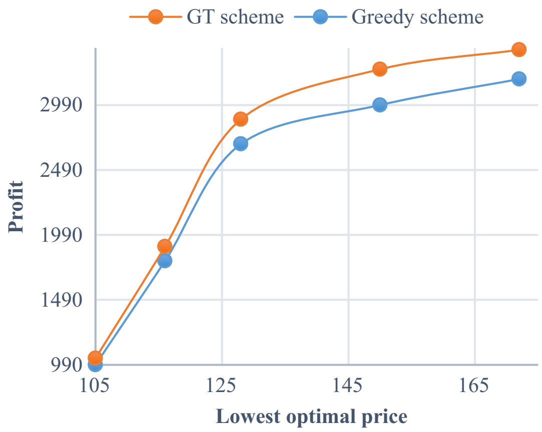

Fig. 6. CPS's profit under different prices for other CSPs

6 Conclusion and future work

An increasing number of applications with low latency requirements motivate new paradigm of computing where computing resources and storage on the edges of a network are rented to execute tasks and to store data for customers. In this paper, we studied the fog-computing market where client lease CSP's computing resources and storage for transaction processing. For this market, CSP is compensated for the computing resources and storage contributions. In particular, an innovative concept has been proposed to lease free computational resources. In order to maximize CPS profit, GA has been proposed to select a set of clients' requests.

The trading problem is modeled and inspired by the game theory, in which the clients attempt to be served in lowest price, while the CSP aims to maximize the profit. The proposed scheme considers the desire of clients to pay less cost. Our game theoretic analysis demonstrates that CSP can derive the optimal service price in any scenario. The results demonstrate our scheme ability to maximize CPS' profit in a variety of scenarios. Future extensions of the present work may involve carrying similar analysis on real market. Further investigations into different fitness functions that include QoS constraints such as latency.

7 References

[1] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang et al., "Fog Computing: Survey of Trends Architectures Requirements and Research Directions", *IEEE Access*, vol. 6, pp. 47980–48009, 2018. https://doi.org/10.1109/ACCESS.2018.2866491

[2] R. Mahmud, R. Kotagiri, R. Buyya, Fog Computing: A Taxonomy, Survey and Future Directions. In: Di Martino B., Li KC., Yang L., Esposito A. (eds) Internet of Everything. Internet of Things (Technology, Communications and Computing). Springer, Singapore, 2018. https://doi.org/10.1007/978-3-319-94890-4

[3] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee", *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2018. https://doi.org/10.1109/TCOMM.2017.2787700

[4] J. Moura, and D. Hutchison, "Game Theory for Multi-Access Edge Computing: Survey, Use Cases, and Future Trends," *IEEE Communications Surveys & Tutorials*, vol. 21, no.1, pp. 260–288, 2018. https://doi.org/10.1109/COMST.2018.2863030

[5] A. Alsarhan, "Game Theoretic Approach for Virtual Machines in IoT Fog Networks," 2018 15th International Multi-Conference on Systems, Signals & Devices (SSD), Hammamet, 2018, pp. 710–715. https://doi.org/10.1109/SSD.2018.8570585

[6] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource Provisioning for IoT Services in the Fog," *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Macau, 2016, pp. 32–39. https://doi.org/10.1109/SOCA.2016.10

[7] D. Kim, H. Lee, H. Song, N. Choi, and Y. Yi, "On the Economics of Fog Computing: Inter-Play among Infrastructure and Service Providers, Users, and Edge Resource Owners," *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, 2018, pp. 1–6. https://doi.org/10.1109/ICC.2018.8422773

[8] M. Debe, K. Salah, M. H. Ur Rehman and D. Svetinovic, "Monetization of Services Provided by Public Fog Nodes Using Blockchain and Smart Contracts," in *IEEE Access*, vol. 8, pp. 20118–20128, 2020. https://doi.org/10.1109/ACCESS.2020.2968573

[9] X. Li, C. Zhang, B. Gu, K. Yamori and Y. Tanaka, "Optimal Pricing and Service Selection in the Mobile Cloud Architectures," in *IEEE Access*, vol. 7, pp. 43564–43572, 2019. https://doi.org/10.1109/ACCESS.2019.2908223

[10] W. Fang, X. Yao, X. Zhao, J. Yin, and N. Xiong, "A Stochastic Control Approach to Maximize Proton Service Provisioning for Mobile Cloudlet Platforms," IEEE Trans. Syst., Man, Cybern., Syst., vol. 48, no. 4, pp. 522–534, 2018. https://doi.org/10.1109/TSMC.2016.2606400

[11] K. Wang, F. C. M. Lau, L. Chen, and R. Schober, "Pricing Mobile Data Offloading: A Distributed Market Framework," IEEE Trans. Wireless Commun., vol. 15, no. 2, pp. 913–927, 2016. https://doi.org/10.1109/TWC.2015.2480394

[12] H. Shah-Mansouri, V. W. S. Wong, and R. Schober, "Joint Optimal Pricing and Task Scheduling in Mobile Cloud Computing Systems," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5218–5232, 2017. https://doi.org/10.1109/TWC.2017.2707084

[13] Hamed Shah-Mansouri and Vincent W. S. Wong, "Hierarchical Fog-Cloud Computing for IoT Systems: A Computation Offloading Game", *Internet of Things Journal IEEE*, vol. 5, no. 4, pp. 3246–3257, 2018. https://doi.org/10.1109/JIOT.2018.2838022

[14] Zhenyu Zhou, Pengju Liu, Junhao Feng, Yan Zhang, Shahid Mumtaz, and Jonathan Rodriguez, "Computation Resource Allocation and Task Assignment Optimization in Vehicular Fog Computing: A Contract-Matching Approach", *Vehicular Technology IEEE Transactions on*, vol. 68, no. 4, pp. 3113–3125, 2019. https://doi.org/10.1109/TVT.2019.2894851

[15] F. Chiti, R. Fantacci, and B. Picano, "A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems," *in IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5089–5096, Dec. 2018. https://doi.org/10.1109/JIOT.2018.2871251
[16] Singh, Simar Preet, et al," "Dynamic Task Scheduling using Balanced VM Allocation Policy for Fog Computing Platforms," *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 433–456, 2019. https://doi.org/10.12694/scpe.v20i2.1538
[17] Haitao Yuan, Jing Bi, and MengChu Zhou, "Spatial Task Scheduling for Cost Minimization in Distributed Green Cloud Data Centers", *Automation Science and Engineering IEEE Transactions on*, vol. 16, no. 2, pp. 729–740, 2019. https://doi.org/10.1109/TASE.2018.2857206
[18] Sung-Tae Hong and Hyoil Kim, "QoE-Aware Computation Offloading to Capture Energy-Latency-Pricing Tradeoff in Mobile Clouds", *Mobile Computing IEEE Transactions on*, vol. 18, no. 9, pp. 2174–2189, 2019. https://doi.org/10.1109/TMC.2018.2871460
[19] D. Yao, C. Yu, L. Yang, H. Jin, "Using Crowdsourcing to Provide QoS for Mobile Cloud Computing", *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 344–356, 2019. https://doi.org/10.1109/TCC.2015.2513390
[20] Ashkan Yousefpour, Ashish Patil, Genya Ishigaki, Inwoong Kim, Xi Wang, Hakki C. Cankaya, Qiong Zhang, Weisheng Xie, and Jason P. Jue, "FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework", *Internet of Things Journal IEEE*, vol. 6, no. 3, pp. 5080–5096, 2019. https://doi.org/10.1109/JIOT.2019.2896311
[21] F. M. Dalvand and K. Zamanifar, "Multi-Objective Service Provisioning in Fog: A Trade-Off Between Delay and Cost Using Goal Programming," *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, Yazd, Iran, 2019, pp. 2050–2056. https://doi.org/10.1109/IranianCEE.2019.8786694
[22] W. Vickrey, "Couterspeculation, Auctions, and Competitive Sealed Tenders", J. Finance, vol. 16, no. 1, pp. 8–37, 1961. https://doi.org/10.1111/j.1540-6261.1961.tb02789.x
[23] David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Pearson, 1989.
[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Massachusetts, 1989.
[25] WM. Spears. The Role of Mutation and Recombinaison in Evolutionary Algorithms. Ph.D. Thesis, George Mason University, Fairfax, VA, 1998.
[26] Aytug, H., Khouja, M., and Vergara, F.E., "Use of Genetic Algorithms to Solve Production and Operations Management Problems: A Review," *International Journal of Production Research*, vol. 41, no. 17, pp. 3955–4009, 2003. https://doi.org/10.1080/00207540310001626319
[27] N. Singh and X. Vives, "Price and Quantity Competition in a Differentiated Duopoly," *RAND J. Economics*, vol. 15, no. 4, pp. 546–554, 1984. https://doi.org/10.2307/2555525
[28] Ayoub Alsarhan, Islam Almalkawi, and Yousef Kilani "A New Covid-19 Tracing Approach using Machine Learning and Drones Enabled Wireless Network," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 22, pp. 111–126, 2021. https://doi.org/10.3991/ijim.v15i22.22623
[29] Fady Esmat, Fathel Samann, Adnan Mohsin Abdulazeez, Shavan Askar, "Fog Computing Based on Machine Learning: A Review," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 12, pp. 21–46, 2021. https://doi.org/10.3991/ijim.v15i12.21313
[30] Mais Haj Qasem, Alaa Abu-Srhan, Hutaf Natoureah, Esra Alzaghoul, "Fog Computing Framework for Smart City Design", *International Journal of Interactive Mobile Technologies*, vol. 14, no. 1, pp. 109–125, 2020. https://doi.org/10.3991/ijim.v14i01.9762

8 Authors

Ayoub Alsarhan received the B.E. degree in computer science from the Yarmouk University, Jordan, in 1997, the M.Sc. degree in computer science from Al-Bayt University, Jordan, in 2001, and the Ph.D. degree in electrical and computer engineering from Concordia University, Canada, in 2011. He is currently Professor with the Department of Information Technology, Hashemite University, Zarqa, Jordan. His research interests include cognitive networks, parallel processing, cloud computing, machine learning, and real-time multimedia communication over the Internet.

Bashar Igried received his Ph.D. in Computer Science from the Swansea University, UKA in 2017, M.Sc. in Computer Science from the University of Middle East University in 2011, and B.Sc. in Computer Science from the Hashemite University, in 2009. He is currently an Assistant Professor at the Computer Science Department of the Hashemite University, Zarqa, Jordan. His research interests include software engineering, cloud computing, service level agreements, requirement engineering, cloud computing database, information retrieval, big data, and database systems.

Atalla Alserhan received his Ph.D. in 2016, M.Sc. in Computer Science from the Yarmouk University in 2001, and B.Sc. from the Mutah University in 1990. He is currently an Assistant Professor at the Department of Information Technology of the Hashemite University, Zarqa, Jordan. Her research interests include Marketing, Total Qualty Management, Strategic Planning, Innovation, Creativity.

Muhsen Alkhalidy received his M.Sc. in Computer Science from the Al-Albayt University in 2008, and B.Sc. in Computer Science from the Mutah University in 1990. He is currently Lecturer at the Computer Science Department of the Hashemite University, Zarqa, Jordan. His research interests include wireless network, security, and Industrial Data analysis.

Article submitted 2022-02-28. Resubmitted 2022-04-08. Final acceptance 2022-04-08. Final version published as submitted by the authors.