The Fisher Market Equilibrium Price Based Multi-Access Edge Computing Coalition Formation

Xiaolong Wang  
Lanzhou Jiaotong University  https://orcid.org/0000-0001-6137-7336

Jianwu Dang  (✉ dangjw25300@163.com)  
Lanzhou Jiaotong University

Shuxu Zhao  
Lanzhou Jiaotong University

Zhanping Zhang  
Lanzhou Jiaotong University

Yangping Wang  
Lanzhou Jiaotong University

Zhanjun Hao  
Northwest Normal University

Research Article

Keywords: Multi-access edge computing, Market equilibrium price, Computing resource block, Coalition formation

Posted Date: October 5th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-924666/v1

License: ☺️ This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
The Fisher Market Equilibrium Price based Multi-Access Edge Computing Coalition Formation

Xiaolong Wang¹, Jianwu Dang¹*, Shuxu Zhao¹, Zhanping Zhang¹, Yangping Wang¹ and Zhanjun Hao²

Abstract

The emergence of multi-access edge computing (MEC) aims at extending cloud computing capabilities to the edge of the radio access network. As the large-scale IoT services are rapidly growing, a single edge infrastructure provider (EIP) may not be sufficient to handle the data traffic generated by these services. The coalition method has been used in MEC for resource optimization, latency, energy consumption reduction, computation offloading, etc. However, the majority of research does not consider the price of computing resources corresponded to a container. Moreover, each SP does not choose EIP with the highest cost-performance to sign a medium/long-term computing resource purchase or lease contract. In this work, we consider a scenario with a collection of SPs with different budgets and several EIPs distributed in geographical locations. During the first phase, we get the market equilibrium price and select the optimal EIPs to make a deal by solving the Eisenberg-Gale convex program. In the second stage, using a mathematical model, we maximize EIP’s profits and form stable coalitions between EIPs by a distributed coalition formation algorithm. Numerical results demonstrate that the effectiveness of our method is significantly better than the existing model.

Keywords: Multi-access edge computing; Market equilibrium price; Computing resource block; Coalition formation

Introduction

Multi-access edge computing (MEC) has attracted much attention in academia and industry in enabling ultra-low latency requests, high-bandwidth, and nimble mobile
services [1, 2, 3]. In MEC, the computing resource is deployed on the edge side,\(^1\) closing to users. In contrast to the service centralization of mobile cloud computing,\(^2\) the computation requests from users can be offloaded to edge nodes (ENs) closing to\(^3\) users. As fusion of wireless communications and mobile computing, MEC is viewed\(^4\) as a key technology of next generation networks, e.g., 5G, Internet of Things (IoT),\(^5\) Internet of Me, Tactile Internet, Social Networks, etc. [4]. However, the power supply\(^6\) limitation of a single edge node raises the question of how an edge node provides\(^7\) high computing resources. As the large-scale IoT services, such as healthcare [5],\(^8\) smart cities [6], agriculture monitoring [7], and many others [8] are rapidly growing,\(^9\) a single EN structure may not be sufficient to handle these data traffic. In addition,\(^10\) cloud data centers (DCs) are often geographically distant from the end-user, which\(^11\) may cause high backhaul traffic. It may induce that the latency constraints of\(^12\) offloaded computation tasks cannot be satisfied due to the propagation delay from\(^13\) the ENs to the DCs. The above reasons drive physical resources and workload\(^14\) sharing between ENs in a peer-to-peer manner by forming coalitions [9].\(^15\)

A coalition is a collection consists of participants in a game to complete a specific task, generating with the arrival of new requests and breaking up with the completion of request processing [10, 11]. However, affected by the number of edge nodes, computing power, optimization targets, and constraint conditions, the coalition formation is more complicated in natural edge scenarios. Hence, it is one of the significant challenges in the field of edge computing [12, 13]. We currently consider a situation where service requests are hosted into containers executed through\(^16\) a virtualization platform running on ENs. The objective of each service is to single\(^17\) out the edge infrastructure providers (EIPs) with maximized cost-effectiveness and\(^18\) find its price in line with market laws under the premise of guaranteeing its QoS.\(^19\)

Furthermore, the EIPs’ goal is to maximize their profit by amortizing their costs,\(^20\) which requires that the aggregated request load submitted to containers must be\(^21\) intense enough to amortize its costs.\(^22\) The objective of services induces a novel market-based solution framework that\(^23\) aims to get the EIPs maximized cost-effectiveness and obtain a price in line with the market laws. The basic idea of the method used in this work is to assign different prices to a computing resource block (CRB) corresponding to a container of different EIPs. According to the market rules, highly in-demand CRBs are priced\(^24\)
high while under-demanded CRBs’ prices are low. Moreover, we assume that each SP has a specific budget for CRB procurement. And the budget is used to capture service priority. Given the CRBs prices, each SP buys CRBs from the EIPs with the maximized cost-effectiveness. When each SP spends the total budgets and CRBs of EIPs are entirely sell out, the resulting prices form a market equilibrium.

Aims at the second goal, if each EIP agrees to cooperate by sharing their requests and computing resources, they can increase their profits. In particular, an EIP can increase its net profit by turning off some ENs and transferring its workloads to other EIPs, or running workloads originated from other EIPs to improve its revenues. Here, depending on the market price and optimal EIPs for SPs, we use a suitable algorithm to decide what conditions the EIPs willing to cooperate. The main contributions of this work are summarized as follows:

- We consider a scenario with a collection of SPs with different budgets and several EIPs distributed in geographical locations. For the scenario, we introduce a system architecture of MEC in an area.
- By solving the Eisenberg-Gale convex program, each SP chooses EIP with the highest cost-performance to sign a medium/long-term computing resource purchase or lease contract and gets the market equilibrium price of the computing resource block.
- Depending on the results of the above phase, we use a mathematical model to maximize EIP’s profit and a distributed algorithm to form a stable coalition.
- We conduct the simulation experiments for the above question and demonstrate that the effectiveness of our method is significantly better than the existing model.

The rest of paper is organized as follows. We first discuss related work in Section, and introduce the system model in Section. In Section, we provide a formulation of the market equilibrium price generating and present the detailed description of the cooperative game. We discuss the simulation results that demonstrate the effectiveness of our approach in Section, followed by the conclusion in Section.

**Related work**

The coalition method has already been investigated in recent years to maximize the profit of EIP. Notably, the methods based on game theory have also been used in
the field of mobile/multi-access edge computing for resource optimization [14, 15], latency and energy consumption reduction [16, 17], computation offloading [18, 19], etc. GlebKoshevoy [20] et al. introduce Shapley value, nucleolus, kernel, and bar-gaining set in the cooperative game theory. Using time, utility, cost-performance stability principle or the reasonableness principle [22], it continuously removes the relatively unstable coalition structure in the space of strategy and finally retains the approximately optimal coalition structure. T. Ling [23] et al. proposed a distributed computation offloading algorithm to achieve the Nash equilibrium of the game, where the algorithm based on the cooperative game is essential to strive for a relative equilibrium state. X. Wang [24] et al. investigate the dynamic bargaining games with externalities and analyzes the effect of externality on the payoffs of players, which shows that externality affects the results of the bargaining game, and coalition structures affect the payoffs of players. Thus, they are suitable for multi-objective scenarios, especially for solving the optimization problem with economical cost and time cost as objectives. T. Li [25] et al. formulate the task scheduling problem as a distributed multi-device task scheduling game. It is proved that the task scheduling game is a potential game, which possesses a property of finite improvement and always owns a Nash equilibrium. It solved the cooperative work problem of the mobile edge computing based on AD-HOC, verified the conditions to avoid null solutions. Y. Zhang [26] et al. put forward a dynamic optimization model for flexible job-shop scheduling based on cooperative game theory, which provides a new real-time scheduling strategy and method. X. Cao [27] et al design the edge federation, an integrated edge computing model, to realize the transparent service provisioning across independent EIPs and the cloud. They characterize the service provisioning process under their edge federation as a linear programming optimization model and develop a service provisioning algorithm SEE. However, the majority of research does not consider the price of computing resources corresponded to a container. Even though considering the resource price, it is more in task offloading in MEC. Evidently, the two prices of resources are defined from different application perspectives. Furthermore, many approaches do not choose the EIPs, which maximize SPs’ profit, as game players before coalition formation. Different from the existing works, we consider the price of CRB and the EIPs with cost-performance.
System model

We introduce the system architecture of MEC in an area that belongs to the different regions in which EIPs and services are distributed. As shown in Figure, there are three layers: the user layer, the aggregation layer, and the MEC layer. First of all, various end-devices with different computation requirements, such as smartphones, connected vehicles, set-top-boxes, PCs, sensors, that have different computation requirements, generate amounts of requests that need to be processed at cloud DCs or MEC platforms in real-time. Next, the aggregation layer, which comprises of base stations, switches/routers, access points, is the communication bridge between users and MECs. In other words, requests from the user layer arrive at a Point of Aggregation (PoA) first and then transmit it to a MEC server for processing.

Indeed, universities, enterprises, hospitals, etc., can also offload their computation data to MEC through the aggregation layer. Furthermore, service providers like Netflix, YouTube, and Facebook would purchase many CRBs from EIPs, and then install their applications and services onto these EIPs’ nodes to serve their customers better.

We contemplate that EIPs are distributed in different geographic locations in an area with distinct configurations and limited computing capacities. Each region hosts a set of edge nodes to run the services which are in charge of processing the data generated by user devices located in that area. We assume that each service has a definite budget for CRBs procurement and offloads its requests/data to MEC as many as possible. The service budget may be related to its net profits from users. However, we do not focus on it to simplify the problem in this research. Through the maximum revenue generated by using the resource bundle of EIPs, we can evaluate the merit of an EIP to a service provider. An EIP may have different worth for different service providers because of the varied distance between an SP and an EIP. Intuitively, each SP would choose its favorite resource bundle with optimal cost performance.

There exists a platform between SPs and EIPs for the sake of collecting the information of EIPs (e.g., computing capacities) and SPs (e.g., budget, etc.) and computing an ME price of unit resource bundle, which maximize the SPs’ revenue and also fully allocates the EIPs’ resources. The ME price can act as an incentive mechanism to charge SPs and to reward MECs. On one side, if the amount of
EIPs’ resource bundles is fixed, the market-clearing price is increases with the SPs’ budget increase. On the other side, anchored the SPs’ budget, the price is inversely proportional to the number of EIPs’ resources. To format a rational coalition among EIPs, we focus on computing a market-clearing equilibrium price that SPs spent all budgets and EIPs sold out all resources.

Problem formulation

APs have the same objective as EIPs, which is to prompt their net profit as much as possible. Without loss of generality, we assume that it is independent among areas to allocate containers/CRBs for services. We only consider the single geographic district, and extend it to multiple zones is straightforward. In general, we deem the EIPs’ total computing capacity is steady, which is determined by predicting the whole region’s resource requirements. That is not the point on which we focus. Nevertheless, the requests from users are dynamic variance along with the users’ changing. Therefore, precisely predicting the demands originated from users is challenging. Moreover, to improve Qos and decrease EIPs’ expenditure as much as possible, the MEC coalition is indispensable. Namely, sharing workloads and resource bundles to serve SPs can reduce EIPs’ energy consumption costs and increase their revenues by hosting services on others’ EIP. It is noteworthy that each EIP prefers to cooperate with some EIP to gain more profit than others. The coalition formed is unstable without the above incentive mechanism. That is to say, a member in an alliance can always find a more profitable confederation that leaves the current league.

EIP Selection and CRB pricing model

We denote $\mathcal{M} = \{1, 2, \cdots, M\}$ and $\mathcal{S} = \{1, 2, \cdots, S\}$ as the set of EIPs and SPs, where $|\mathcal{M}| = M$, $|\mathcal{S}| = S$, respectively. Denote $i$ and $j$ as the SP and EIP index, respectively. Assuming that each EIP $j$ has $c_j$ homogeneous computing unit (e.g., VMs, servers, etc.). Let $a_{ij}$ is the portion of the resources on EIP $j$ allocated to SP $i$, $0 \leq a_{ij} \leq 1$. Then, $c_j a_{ij}$ is the number of computing units allocated to SP $i$ from EIP $j$. Denote $a_i = (a_{i,1}, a_{i,1}, \cdots, a_{i,M})$ as the vector of resources allocated to SP $i$ from all the EIPs. We assume that the computing resource is dividable to support hosting container (VM/Docker). Therefore, $c_j a_{ij}$ can be non-integer; and we define the budget of SP $i$ as $B_i$. 
The ultimate goal is choosing an EIP with the best cost performance for an SP.\(^1\)

Computing ME solutions include an equilibrium price vector \(p = (p_1, p_2, \ldots, p_M)\).\(^2\) \(p_j\) is the price of EIP \(j\), and a resource allocation matrix \(A\), the element in which\(^3\) at row \(i\)th and column \(j\)th is \(a_{ij}\). Note that \(p_j\) is the price of all the resources\(^4\) of EIP \(j\). Then one unit resource price equals \(\frac{p_j}{c_j}\), \(\forall j\). We define \(u_i(a_i, p)\) as the\(^5\) SP \(i\)'s utility function of the number of resources \(a_i\) received from EIPs under\(^6\) the resource price \(p\). Depend on the capacity constraints of EIPs, then we have\(^7\)

\[
\sum_{i=1}^{S} c_j a_{ij} \leq c_j, \forall j \in M \text{ and } \sum_{i=1}^{S} a_{ij} \leq 1, \forall j \in M.
\]

Each SP is a player in the\(^8\) Fisher market game, in which the player aims to maximize its utility subject to the\(^9\) SP budget constraint \(\sum_j a_{ij} p_j \leq B_i\).

**Definition 1**  A market outcome that maximizes the utility of each buyer subject to its budget constraint and clears the market is called a market equilibrium \([29]\).\(^{10}\)

In-state, \((p^*, A^*)\) is a market equilibrium if and only if satisfy the following conditions:

- **Condition 1**: For all \(i \in S\), \(a^*_i\) maximizes buyers \(i\)'s utility given prices \(p^*\) and budget \(B_i\), i.e., \(a^*_i = (a^*_i,1, \ldots, a^*_i,M) \in \arg \max u_i(a_i, p^*)\).

- **Condition 2**: Each item (e.g., computing resource block) \(j\) either is completely sold or has price 0, i.e., \((\sum_{i=1}^{S} a_{ij} - 1) p_j = 0, \forall j \in M\).

- **Condition 3**: All budgets get spent, i.e., \(\sum_{j=1}^{M} p_j a_{ij} = B_i, \forall i \in S\).

A market equilibrium is guaranteed to exist if each item is desired by at least one buyer and each buyer desires at least one item \([30]\). Here, condition one guarantees that the equilibrium allocation \(a^*_i\) maximizes the utility of SP \(i\) at the equilibrium prices \(p^*\) and SPs' budgets constraints \(B_i\). Condition two maximizes the EIPs' resource utilization, i.e., the resources of EIPs are entirely sold in the market. Condition three represents that each SP runs out of its budget to purchase resources from EIPs. Moreover, condition two and three is called the market clearing condition \([31]\).

**Service provider utility**

For the sake of representation simplicity, we use \(u_i(a_i)\) to denote the utility of SP \(i\), i.e., \(u_i(a_i, p) = u_i(a_i)\). The model considers linear functions for ease of investigation. We view the delay-sensitive applications and assume that the transmission
1. Bandwidth is large enough and the data size need transmission is relatively small. 2. Hence, we can neglect the transmission delay and only consider propagation latency 3. and processing delay [32, 33]. The sum total latency of a request from users con- 4. sists of three parts: the round-trip latency between a user and a PoA, such as base 5. station (BS) or access point (AP), the return journey delay between the PoA and 6. an EIP’s node which hosting the service application, and the processing latency of 7. request. A detailed depiction is shown in Figure. Note that we only investigate the 8. model from the aggregation layer to the MEC layer. For the sake of simplicity, we 9. assume that each service is located at one PoA in a region. If a service’s requests 10. from several PoAs, we can take sum over all the PoAs’ requests. When there is 11. more than one PoAs for a service, we take the sum over all PoAs requests processed 12. by the MEC. Let $T_{i}^{\text{max}}$ as the maximum latency tolerance of service $i$, then

\begin{equation}
    l^{n}_{ij} + l_{ij}^{p} \leq T_{i}^{\text{max}}, \quad \forall i, j.
\end{equation}

15. Denote $\lambda_{ij}^{\text{max}}$ as the maximum number of requests that EIP $j$’s nodes can process. Evidently, if $l^{n}_{ij} \geq T_{i}^{\text{max}}$, $\lambda_{ij}^{\text{max}} = 0$. We use M/M/1 queues to model the processing latency and assume the workload 20. is assigned among computing units evenly [32, 33, 34, 35]. The processing latency 21. can be represented as

\begin{equation}
    l_{ij}^{p} = \frac{1}{\mu_{ij} - \frac{\lambda_{ij}}{c_{ij}a_{ij}}}, \quad \forall i, j.
\end{equation}

25. where $\mu_{ij}$ is the processing rate of a computing unit of EIP $j$ to deal with service 28. $i$’s requests, $\lambda_{ij}$ is the arrival rate of request from service $i$ to EIP $j$. In addition, to 29. assure the stability of the queue, the inequality $\frac{\lambda_{ij}}{a_{ij}} < \mu_{ij}$ must hold. Otherwise, the 30. delay of the queue will be infinite. Combining formulas (1) and (2), we can derive

\begin{equation}
    \lambda_{ij} \leq c_{ij}a_{ij} \left( \mu_{ij} - \frac{1}{T_{i}^{\text{max}} - l^{n}_{ij}} \right), \quad \forall i, j.
\end{equation}
If \( l_{ij} < T_{i}^{\text{max}} \), the maximum number of requests from service \( i \) processed by the EIP \( j \) is

\[
\lambda_{ij}^{\text{max}} = \max\left\{ c_{j}a_{ij}\left( \mu_{ij} - \frac{1}{T_{i}^{\text{max}} - l_{ij}} \right), 0 \right\} \\
= c_{j}a_{ij} \cdot \max\left\{ \left( \mu_{ij} - \frac{1}{T_{i}^{\text{max}} - l_{ij}} \right), 0 \right\}, \forall i, j.
\]

Denote \( r_{i} \) as the profit of successfully serving a request from service \( i \) [34]. Hence, the revenue of service \( i \) \( u_{ij}(a_{ij}) = r_{i}g_{ij}c_{j}a_{ij} \). Let \( v_{ij} = r_{i}g_{ij}c_{j} \), and then we have

\[
u_{ij}(a_{ij}) = v_{ij}a_{ij}, \quad \forall i, j.
\]

Thus,

\[
u_{i}(a_{i}) = \sum_{j=1}^{M} u_{ij}(a_{ij}) = \sum_{j=1}^{M} v_{ij}a_{ij}, \quad \forall i.
\]

Obviously, \( r_{i}g_{ij} \) can be a valuation index which is the gain of service \( i \) from one unit resource of EIP \( j \). Accordingly, \( v_{ij} \) is a valuation index of all resources of EIP \( j \).

**Definition 2** A function \( u(\cdot) \) is homogeneous of degree \( d \), where \( d \) is a constant, if

\[
u(\alpha x) = \alpha^{d}u(x), \quad \forall \alpha > 0 \quad [36].
\]

It is simple to verify that \( u_{i}(a_{i}) \) is a linear function with degree \( d = 1 \).

**Solution**

Define the ratio \( \frac{v_{i}}{p_{j}} \) is the bang per buck [37] of EIP \( j \) to SP \( i \), which is the utility gained by SP \( i \) per unit amount of money spent on EIP \( j \). Thus, we can define \( \alpha(i) = \arg\max_{j} \left( \frac{v_{j}}{p_{j}} \right) \) as the maximum bang per buck (MBB) [37]. Only SP \( i \) buys resources from EIP of \( \alpha(i) \), will each SP spend entire budget and maximize its utility. Derive from **definition 2**, the buyers’ utility is linear. Hence, by solving the Eisenberg-Gale (EG) convex program, we can find the market equilibrium. The EG
convex program can be described as

\[ \max_{A, u} \sum_{i=1}^{S} B_i \ln u_i \]  

s.t. \( u_i = \sum_{j=1}^{M} v_{ij} a_{ij}, \quad \forall i \)  

\[ \sum_{i=1}^{S} a_{ij} \leq 1, \quad \forall j \]  

\[ a_{ij} \geq 0, \quad \forall i, j. \]  

By setting \( a_{ij} > 0 \) and its value is small enough such that the constraints (8)-(10) are stringent inequality. Thus, Slater's condition holds, and then Karush-Kuhn-Tucker (KKT) conditions are necessary and adequate for optimality [38]. Using \( \varphi_i, p_j, \) and \( \zeta_{ij} \) to represent the dual variables affiliated with restrictions (8), (9), and (10), severally. According to the Lagrangian, we have

\[ L(u, A, \varphi, p, \zeta) = \sum_{i} B_i \ln u_i + \sum_{j} p_j \left( 1 - \sum_{i} a_{ij} \right) + \]  

\[ \sum_{i} \varphi_i \left( \sum_{j} v_{ij} a_{ij} - u_i \right) + \sum_{i} \sum_{j} \zeta_{ij} a_{ij} \]  

The KKT conditions are

\[ \frac{\partial L}{\partial u_i} = 0 \Rightarrow \frac{B_i}{u_i} = \varphi_i, \quad \forall i \]  

\[ \frac{\partial L}{\partial a_{ij}} = 0 \Rightarrow p_j - B_i \frac{v_{ij}}{u_i} = \zeta_{ij}, \quad \forall i, j \]  

\[ u_i = \sum_{j} v_{ij} a_{ij}, \quad \forall i \]  

\[ p_j \left( 1 - \sum_{i} a_{ij} \right) = 0, \quad \forall j \]  

\[ \zeta_{ij} a_{ij} = 0, \quad \forall i, j \]  

\[ p_j \geq 0, \quad \forall j \]  

\[ \zeta_{ij} \geq 0, \quad \forall i, j. \]
From the KKT conditions, we can derive

\[ \frac{u_i}{B_i} \geq \frac{v_{ij} p_j}{B_j}, \quad \forall i, j \]  

(19)

if \( a_{ij} > 0 \) \( \Rightarrow \) \( \zeta_{ij} = 0 \) \( \Rightarrow \) \( \frac{u_i}{B_i} = \frac{v_{ij} p_j}{p_j} \), \( \forall i, j \)  

(20)

\( p_j > 0 \) \( \Rightarrow \) \( \sum_i a_{ij} = 1, \quad \forall j \)  

(21)

\[ \sum_i a_{ij} < 1 \Rightarrow \quad p_j = 0, \quad \forall j \]  

(22)

Note that variable \( p_j \) can be seen in the EG convex program as the pang per buck of EIP \( j \) to SP \( i \). The subsequent theorem reveals the connection between the EG convex program and the market equilibrium solution. Besides, some properties of the ME are also represented.

**Theorem 1**  
The optimal solution of the EG convex program is a market equilibrium. Particularly, the Lagrangian dual variables of the EIPs’ capacity restrictions (9) are the equilibrium prices. In addition, the allocation at the equilibrium is Pareto optimal and envy free, which also satisfies the sharing-incentive and proportionality properties [39].

At the equilibrium, the resource allocation maximizes the utility and exhausts each service’s budget. Moreover, each SP \( i \) only purchases resources from ENs in \( \alpha_i \). Additionally, SPs’ optimal utilities and equilibrium prices are exclusive.

**Optimal Coalition Allocation Utility**

An EIP \( j \) aims at increasing its net margin as much as possible. We assume that each service application is encapsulated in a container with a computing resource block, and each EIP’s edge nodes need a virtualization platform to run the container. \( EN (j) \) represents the set of edge nodes of EIP \( j \). Let \( w_z (\eta) \) be the expected power consumption of edge node \( z \).

\[ w_z (\eta) = \eta_z W_{z}^{max} + (1 - \eta_z) W_{z}^{min} \]  

(23)

where \( \eta \in [0, 1] \) is the utilization of resource (CPU) on EIP edge node \( z \), \( W_{z}^{max} \) and \( W_{z}^{min} \) represent the power consumption with idle and fully utilized CPU states,
respectively. The net margin rate of EIP $\alpha (i)$ denoted by $M_{\alpha(i)}$.\(^{1}\)

\[
M_{\alpha(i)} = R_{\alpha(i),i,n_i} - \left( P_{1,\alpha(i)}b_i(n_i < N_i) + \sum_{e \in EN(\alpha(i))} x_e w_e(\eta_e) E_{\alpha(i)} \right)
= R_{\alpha(i),i,n_i} - \left( P_{1,\alpha(i)}b_i(n_i < N_i) + \sum_{e \in EN(\alpha(i))} W_e(x_e, \eta_e) E_{\alpha(i)} \right)
\]

\(^{(24)}\)\(^{6}\)

where $W_e(x_e, \eta_e) = x_e W_e^{\min} + (W_e^{\max} - W_e^{\min}) \eta_e$; $x_e$ is a binary variable that depicts the edge node state; when $x_e = 1$, the state is switched on; conversely $x_e = 0$, the state is switched off. $n_i$ denotes the number of containers/CRBs allocated for service $i$, and $N_i$ is the minimum number of containers/CRBs that satisfies the delay requirement of service $i$, which can derive from the inequality (4) and the stability condition $\frac{\mu_i}{\lambda_i} < \mu_{ij}$ of the queue. $\eta_e$ represents the ratio of resources allocated to all the containers running on edge node $e$ to its total resource capacity. $E_j$ is the price (per unit of time) charged EIP $j$ for electricity. $b_i$ is binary variable. When $n_i < N_i$, $b_i = 1$, which represents that CRBs allocated for service $i$ cannot satisfy the Qos requirement, conversely, $b_i = 0$. When SP $i$’s QoS is not met, it will charge EIP $j$ for a monetary penalty $P_{i,j}$. $R_{\alpha(i),i}$ represents the revenue that EIP $\alpha (i)$ earns for running a container for processing its requests from SP $i$. In equation (24), the first term on the right means the sum of EIP $\alpha (i)$’s income to run $n_i$ containers, where the second term on the right denotes the overheads of EIP $\alpha (i)$.

The coalition formation algorithm (in Subsection ) needs to compute the coalition net margin $\Psi (\mathcal{C})$ to get the payoffs that EIPs gain in any coalition $\mathcal{C}$. By introducing the Mixed Integer Linear Program (MILP) model to solve a maximization problem, we can find the best allocation of the containers $\mathcal{V}$ onto the edge nodes $\mathcal{N}_e = \bigcup_{j \in \mathcal{C}} EN(j)$ to run the services $\mathcal{S}_e = \{ i : \alpha (i) \in \mathcal{C} \}$. Denote $m(i)$ as a mapping $\mathcal{V} \rightarrow \mathcal{S}_e$, which represents the service application run by container $i$. For the sake of simplicity, we define vector $\mathbf{R} = (R_{\alpha(i),i})_{i \in \mathcal{S}_e}$, $\mathbf{n} = (n_i)_{i \in \mathcal{S}_e}$, $\mathbf{b} = (b_i)_{i \in \mathcal{S}_e}$, $\mathbf{P} = (P_{1,\alpha(i)})_{i \in \mathcal{S}_e}$, $\mathbf{W} = (W_j(x_j, \eta_j))_{j \in \mathcal{N}_e}$, $\mathbf{E} = (E_{\alpha(i)})_{\alpha(i) \in \mathcal{C}, j \in EN(\alpha(i))}$ (here, $j$ is the
The MILP model can be described as

$$\text{maximize } \Psi(C) = R \cdot n - (b \cdot P + W \cdot E)$$

s.t. $\sum_{i \in N_c} y_{ij} \leq 1, \quad \forall j \in V \quad (25)$

$$\sum_{j \in V} y_{ij} \leq |V|x_i, \quad \forall i \in N_c \quad (26)$$

$$\eta_i = \sum_{j \in V} \frac{1}{L_i} y_{ij}, \quad \forall i \in N_c \quad (27)$$

$$\eta_i \leq x_i, \quad \forall i \in N_c \quad (28)$$

$$n_k = \sum_{i \in N_c} \sum_{j \in V, m(j) = k} y_{ij}, \quad \forall k \in S_c \quad (29)$$

$$n_k \leq N_k, \quad \forall k \in S \quad (30)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N_c \quad (31)$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in N_c, \; j \in V \quad (32)$$

$$\eta_i \in \mathbb{R}^+, \quad \forall i \in N_c \quad (33)$$

$$n_k \in \mathbb{N}, \quad \forall k \in S_c \quad (34)$$

In the above MILP model, the binary variable $x_i = 1$ if edge node $i$ is turned on, otherwise $x_i = 0$. The binary variable $y_{ij} = 1$ if container $j$ is allocated on edge node $i$, else $y_{ij} = 0$. The non-negative real variable $\eta_i$ denotes the ratio of resources allocated to all the containers running on edge node $i$ to its total resource capacity, and $L_i$ represents the maximum number of containers that can host on edge node $i$. The non-negative integer variable $n_k$ denotes the number of containers allocated for service $k$. The objective function $\Psi(C)$ of the maximization problem is the retained profits rate gained by coalition $C$ composed of EIPs. It is an extension of the net profit $M_{\alpha(i)}$ of EIP $\alpha(i)$ and has the following constraints.

Constraint condition (26) ensures that each container is hosted on one edge node at most. Limitation condition (27) assures that containers will not be allocated on edge nodes turned off. Equation (28) defines the value of variable $\eta_i$ as the ratio of the sum of containers distributed to the maximum number of containers that can host on edge node $i$. Moreover, inequation (29) guarantees that the ratio $\eta_i$ does not exceed one (edge node $i$ switched on) or zero (edge node $i$ turned off). Equation (30) defines the value of the variable $n_i$ as the number of containers allocated for
service $k$. Restriction condition (31) ensures that the number of containers assigned to each service cannot exceed the actual demand. Equations (32)-(35) define the domain of variables $x_i$, $y_{ij}$, $\eta_i$ and $n_k$, respectively.

Coalition preference function

We assume that each EIP is a rational participant in a cooperative game to make optimal strategic decisions to increase their profits. Thus, every EIP may form a coalition to pursue its maximum yields, in which each EIP (coalition member) must go along with sharing its resources with others. An alliance formation is a dynamic process, where an EIP moves from one coalition with a lower utility to another with a higher. We model the process as a coalition formation cooperative game. Foremost, an EIP must ensure that joining a league gained is no worse than working alone. In addition, to ensure that the benefits are not short-lived, the coalition, an EIP joined, needs to satisfy the following properties [40, 41].

- **Coalition stability property**: A coalition is stable if no participant can deviate from the current federation to reach a better one subjectively.
- **Fairness property**: Any member in the current coalition expects that the resulting profits divide fairly among all players.

Hence, the EIPs desire a method to decide whether to join a coalition.

By modeling the coalition formation process as a cooperative game with transferable utility [42], each EIP can cooperate with others to maximize its net profit by using a coalition formation algorithm. Here, we use a hedonic game, where any player’s gain is decided by the coalition members to which the player belongs, and the players’ preferences for their possible set of coalitions form the final alliances. In other terms, each participant does not consider other coalition players how grouped, and is only interested in the players which in the coalition he joined.

Given that $\mathcal{M}_s$ is the EIPs set selected, then a coalition $C \subseteq \mathcal{M}_s$ acts as an entity, representing an agreement that they must agree to share their resources and users among EIPs. At any time, we divide the participants set into a coalition partition $\Pi = \{C_1, C_2, \cdots, C_l\}$. Each element $C_k$ in partition $\Pi$ is disjoint for $k = 1, 2, \cdots, l$, i.e., $\bigcup_{k=1}^{l}C_k = \mathcal{M}_s$ and $C_i \cap C_j = \emptyset$, $i \neq j$. Denote $C_{\Pi} (i)$ as a coalition containing player $i$ in partition $\Pi$, $\forall i \in \mathcal{M}_s$. Define $\Psi (C | \Pi)$ as the utility value of a coalition $C$ in partition $\Pi$ of $\mathcal{M}_s$. For a hedonic game, the utility value of alliances
is independent of each other, i.e., \( \Psi (C \mid \Pi) = \Psi (C) \), which can be calculated by the formula in subsection. Let \( \phi_i (C, \Psi) \) be a fraction coalition utility value received from coalition \( C \) by \( i \), where \( i \in C \). That is to say, \( \phi_i (C, \Psi) \) is the payoff of \( i \) in coalition \( C \). Each player in cooperative games aims to get a stable coalition and gains rewards as high as possible. In parallel, it is critical to maintaining a stable coalition to distribute the federation value fairly among the members. Furthermore, every participant cannot improve its rewards by leaving the current coalition to join a new one. In a nutshell, the coalition value distribution is fair and can be reached using a profit allocation schedule with the following properties:

1. **Efficiency property**: \( \sum_{i \in C} \phi_i (C, \Psi) = \Psi (C) \).
2. **Symmetry property**: If \( \forall i, j \in M_s, \forall C \subseteq M_s \setminus \{i, j\} \) and \( \Psi (C \cup \{i\}) = \Psi (C \cup \{j\}) \), then \( \phi_i (C \cup \{i\}, \Psi) = \phi_j (C \cup \{j\}, \Psi) \).
3. **Dummy player**: If \( i \) is a dummy EIP, i.e., \( \Psi (C \cup \{i\}) - \Psi (C) = 0 \) for \( \forall C \subseteq M_s \setminus \{i\} \), then \( \phi_i (C, \Psi) = 0 \).
4. **Fairness**: If \( \forall i, j \in M_s, \forall C \subseteq M_s \setminus \{i, j\} \) and \( \Psi (C \cup \{i\}) = \Psi (C \cup \{j\}) \), then \( \phi_i (C \cup \{i, j\}, \Psi) - \phi_i (C \cup \{i\}, \Psi) = \phi_j (C \cup \{i, j\}, \Psi) - \phi_j (C \cup \{j\}, \Psi) \).

The efficiency property guarantees that the total coalition value is assigned. The symmetry property denotes that if two participants who make the same contribution to every subset composed by other players, they should obtain the same profit.

The dummy property represents that the members, nothing marginal contribution provided to any other coalitions, get zero profit. In addition, the fairness property means that any two participants who make the same contribution to the alliance will get the same payoff. Shapely value [44], a payoff allocation, satisfies the additivity, strong monotonicity and the above four properties. Here, we use the Shapely value defined in the literature [45], described as

\[
\phi_i (C, \Psi) = \sum_{B \subseteq C \setminus \{i\}} \frac{|B|! (|C| - |B| - 1)!}{|C|!} (\Psi (B \cup \{i\}) - \Psi (B))
\]

\[
= \sum_{B \subseteq C \setminus \{i\}} \left( \begin{array}{c} |C| \\ |B| \end{array} \right) (|C| - |B|)!^{-1} (\Psi (B \cup \{i\}) - \Psi (B))
\]

where \( B \) is all possible subsets that do not contain \( i \) of coalition \( C \), including the empty set.
Define $\succeq_i$ as a preference relation that EIP $i$ uses to compare all the possible coalition that it may join. The binary relationship needs to satisfy complete, reflexive, and transitive, etc., properties [41]. In other terms, $\forall C_1, C_2 \subseteq M_s$, where $i \in C_1$ and $i \in C_2$, $C_1 \succeq_i C_2$ represents that participant $i$ prefers being a member of $C_1$ than $C_2$, or at least $i$ prefers both equally. Hence, we use the following preference relation

$$C_1 \succeq_i C_2 \iff \Phi_i(C_1) \geq \Phi_i(C_2) \quad (37)$$

where $C_1$ and $C_2$ are any two coalitions that contain member $i$. $\Phi_i(\cdot)$ is the preference function, described as

$$\Phi_i(C, \Psi) = \begin{cases} 
\phi_i(C, \Psi) , & \text{if } C \notin h(i) \\
-\infty , & \text{otherwise}
\end{cases} \quad (38)$$

where $h(i)$ is a set that contains the coalitions having been evaluated [46, 47]. For member $i$, if $C_1$ is strictly better than $C_2$, we use notation $\succ_i$ to denote the preference relationship, i.e., $C_1 \succ_i C_2$.

Coalition Formation Algorithm

After deriving the price of the computing resource block, coalition utility, and preference function from subsection - , in this subsection, we use a distributed algorithm named DCFA (Distributed Coalition Formation Algorithm) based on the hedonic shift rule [48] to get a stable coalition. By introducing the suitable distributed state management algorithms [49, 50], coping with the distribution property of DCFA.

Describing simply the shift rule as, given a coalition partition $\Pi = \{C_1, C_2, \cdots, C_l\}$ on set $M_s$ and a preference $\succ_i$, if and only if $C_k \cup \{i\} \succ_i C_\Pi (i)$, player $i$ leaves its current coalition $C_\Pi (i)$ to join coalition $C_k \in \Pi \cup \emptyset$. The shift rule exploits the rational participants’ selfishness to move from low profit to higher, ignoring the influence of this behavior on other individuals in the same coalition. The objective of DCFA is to let each member find the possible alliances it may join and check whether it is preferred to join. If a member decides to leave the current coalition to join a new one, we put the current coalition into set $h(i)$ to avoid repeatedly visiting it. Given that each EIP can operate asynchronously and independently from others, we can implement DCFA using suitable distributed mechanisms, which...
are state retrieval and atomic state update. The previous one guarantees that any
member can obtain the current coalition partition; the second one updates the
coalition partition when any member decides. A similar issue instance can refer to
the literature [49, 50, 51].

Algorithm 1 DCFA

1: function DCF(eip_i, partition)

2: Initialize h(eip_i), c_current, c_best as an empty set, respectively.

3: while true do

4: p_current = partition

5: c_current = getCurrentCoalition(eip_i)

6: c_best = c_current

7: for c in p_current \ c_current do

8: if c not in h(eip_i) then

9: c_new = c ∪ {eip_i}

10: Φ(c_best) = getShapleyValue(c_best, eip_i)

11: Φ(c_new) = getShapleyValue(c_new, eip_i)

12: if Φ(c_new) > Φ(c_best) then

13: c_best = c_new

14: end if

15: end if

16: end for

17: if c_best != c_current then

18: add set c_current \ {eip_i} in h(eip_i)

19: p_best = (p_current \ {c_current, c_best \ {eip_i}}) ∪ (c_current \ {eip_i}, c_best)

20: partition = p_best

21: end if

22: if c_best == c_current then

23: break

24: end if

25: end while

26: end function

The function DCF accepts the variable eip_i and the global variable partition used to store the current coalition partition. The initial coalition partition is that every EIP is a coalition. In DCFA, A \ B represents that set A removes element B. The algorithm has two important properties, which are convergence and Nash-stability [41, 52].

Theorem 2 Starting from any initial coalition structure, the DCFA algorithm always converges to a final partition.

Theorem 3 Any obtained final partition with algorithm DCFA is Nash-stable.
Similar proof of theorem 2 and 3 can refer to the literature [53]. The property\(^1\) of Nash stability is a fundamental guarantee of convergence property. Moreover,\(^2\) introducing each member’s visited history set makes the number of alternatives\(^3\) decrease monotonically to end with a stable coalition configuration.

**Experimental Evaluation**

**Simulation Setup**

In this subsection, we consider a square region with \(10\text{km} \times 10\text{km}\) located in Anning District, Lanzhou (China), as shown in Figure. The locations of EIPs and SPs are generated randomly in the region. We randomly generated 100 EIPs and 1000 SPs locations in total, and assumed that each SP is located at one location. We randomly sampled with 8 EIPs, and 4 SPs from EIPs and SPs generated previously for the sake of clarity and analysis, i.e., \(M = 8, S = 4\). We assume that the delay between an SP and an EIP is proportional to their distance. Suppose that the price for an SP to serve a request successfully is 2 to 3 per 100000. The physical infrastructure of each EIP has four identical edge nodes, and everyone has 40 CRBs, whose maximum power consumption \(W_{\text{max}}^{k}\) and idle power consumption \(W_{\text{min}}^{k}\) are 200w and 100w, respectively. The electricity price \(E_i\) to every EIP \(i\) is 0.0001/Wh. The penalty \(P_{ij} = \beta R_{ji}\), where \(\beta\) is a factor. Here, let \(\beta = 10\). The value of \(\beta\) ensures that each EIP prefers to allocate its CRBs for the services it hosts. Budget of each SP ranges from 10 to 20, and we can normalize it to be 1 for generality. Parameters related to services are shown in Table 1.

**Experimental Results**

According to the parameters set before, we get the valuation index \(v_{ij}\) that is each service \(i\) for all resources of EIP \(j\), which is shown in Figure. Obviously, if we normalize \(v_{ij}\) corresponded to one CRB, it will have a generality. In this experiment, we assume that each EIP has an equal number of resources, such that the performance before and after normalization is equivalent. In Figure, each service corresponds to 8 EIPs evaluation indicators depicted by a bar graph. From Figure, we can easily find that the optimal EIPs corresponding to services \{1, 2, 3, 4\} are \{5, 2, 3, 1\}. The same conclusion can obtain from Figure to Figure. And from the distribution map of SPs and EIPs (Figure), we inferred that the result obtained above is reasonable. For instance, we only consider one influencing factor – distance,
SP 1 chooses the relatively farther EIP 5 instead of EIP 3. It is more reasonable to match SP 3 and EIP 3. Similarly, if SP 4 matches EIP 2, only can SP 2 match EIP 7. However, EIP 7 is farther from SP 2 than EIP 2, and EIP 1 satisfies SP enough. Overall, the matches are rational. Figure depicts the revenues of unit cost, i.e., bang per buck $\frac{v_i}{p_j}$, and Figure is the maximum bang per buck. Figure and are similar in shape, but they represent utilities on different EIPs and the top arrival rates supported by different EIPs, respectively.

By solving the EG program, we get the Fisher market equilibrium prices of all CRBs and unit CRB of EIPs under service providers’ budgets, which described in Figure. We set three budgets of SPs to analyze the impact of which on the market price; they are the benchmark budgets (10-20), double of the benchmark, and a half of the benchmark. The effects show in Figure.

From Figure, all the prices increase twice as the budget of each SP is double. By the same rule, all the prices reduce one-half, with each SP’s budget reduced by half. Hence, the value of the budget only affects the equilibrium prices by a scaling factor. When fixing the budget, the prices act as a primary means to allocate resources only.

In the coalition formation algorithm, we vary the request intensity (arrival rate) of services, and the corresponding required containers are changing with it. The detailed depiction of request intensity and corresponding containers show in Figure and Figure, where the red diamonds in these two figures represent the maximum request intensity and the corresponding required containers maximum supported by EIPs selected.

In Table 2, we present the coalitions formed, which correspond to different requests of services, where $E[\cdot]$ denotes the arithmetic mean operator. In the first place, depending on the columns Average Arrival Rate and Average $E[\phi_i]$, we have Arrival $Rate 3 > Arrival Rate 4 > Arrival Rate 2 > Arrival Rate 1$ and Average $E[\phi_i]$ $2 > Average E[\phi_i] 1 > Average E[\phi_i] 4 > Average E[\phi_i] 3$. As the requests increase, we find that the benefits of cooperation increase first and then decrease. For one thing, the increase in requests made them obtain more benefits, as more requests take up more resources. For another, as the requests increasing, the number of edge nodes switched off are decrease.
We depict the utilities of different methods (fixed one-one contract-FOOC, coalition) in Figure 2, capturing that the EIPs’ utilities obtained by cooperation are markedly better than FOOC. And the arithmetic means of utilities of the coalition method are also not less than FOOC. The experiments point out that cooperation brings more significant benefits, and as the requests increase beyond a certain amount, the advantages decrease gradually.

To analyze the effect of parameter $\beta$ on the coalition formed and the average utility of EIPs, we choose the arrival rate 2 as a case. The detailed results of the experiment are shown in Table 3 and Figure 3, where $\beta = \{1, 5, 10, 15, 20, 30\}$ and we can see that the average $\Psi \{i\}$ and $E[\phi_i]$ are decline linearly and monotonically with $\beta$ increasing, which also can obtain from equations (24) and (25). Moreover, the gradient of the average $\Psi \{i\}$ and $E[\phi_i]$ are approximately equal. We find that as $\beta$ increases, EIPs are more inclined to form a big coalition through further analysis. It may be induced by that each EIP prefers to join an enormous coalition to apportion more cost.

**Conclusion and future works**

In this work, we consider a scenario with a collection of SPs with different budgets and several EIPs distributed in geographical locations. We aim to develop a price strategy to select EIPs with the most cost-effectiveness for each SP and get a stable coalition among EIPs selected with maximum EIP profits for different workloads. For the scenario, we introduce the system architecture of MEC in an area. There are three layers: the user layer, the aggregation layer, and the MEC layer. Towards the first goal, we use the famous concept of General Equilibrium in Economics as an effective solution. It produces a Market Equilibrium with Pareto-efficient and fairness, etc., properties. We get the market equilibrium price and select the optimal EIPs for each SP by solving the Eisenberg-Gale convex program. Aim at the second purpose, depending on the price and EIPs selected, obtained by the first goal, we use a mathematical model to maximize EIP’s profit and an algorithm to form a stable coalition. Numerical results demonstrate the effectiveness of the method. In future work, we will consider the net profit of the service provider and analyze the EIPs profit and the coalition under its market equilibrium price.
Acknowledgements

We would like to thank Shuxu Zhao, Yangping Wang, the School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China, for providing the infrastructure and funding needed to carry out the proposed research work.

Funding

This work was partially supported by National Natural Science Foundation of China (No. 6206020135), Key Research and Development Program of Gansu Province (No. 20YF8GA123) and Innovation Star Project for Outstanding Postgraduates of Gansu Province (No. 2021CXZX-557).

Abbreviations

MEC: Multi-access edge computing; EIP: Edge infrastructure provider; SP: Service provider; CRB: Computing resource block; MBB: maximum bang-per-buck; EN: Edge nodes; IoT: Internet of things; DC: Cloud data center; EG: Eisenberg-Gale; BS: Base station; AP: Access point; PoA: Point of aggregation; ME: Market equilibrium; FOC: Fixed one-one contract; MCEP: Market clearing equilibrium price.

Competing interests

The authors declare that they have no competing interests.

Authors’ contributions

Xiaolong Wang: Formal analysis, Investigation, Methodology, Validation, Writing-original draft, Writing-review & editing. Jianwu Dang: Conceptualization, Project administration, Funding acquisition, Writing-review & editing. Shuxu Zhao: Funding acquisition, Methodology. Zhanping Zhang: Writing-review & editing. Yangping Wang: Project administration, Funding acquisition. Zhanjun Hao: Conceptualization.

Author details

1 School of Electronic and Information Engineering, Lanzhou Jiaotong University, LanZhou, 730070, China.
2 College of Computer Science & Engineering, Northwest Normal University, Lanzhou, 730070, China.

References

1. Hu, Y.C., Patel, M., Sabella, D., Sprecher, N., Young, V.: Mobile edge computing—a key technology towards 5G. ETSI white paper 11(11), 1–16 (2015).
2. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials 19(4), 2322–2358 (2017). doi:10.1109/COMST.2017.2745201
3. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: A survey. IEEE Internet of Things Journal 5(1), 450–465 (2017). doi:10.1109/JIOT.2017.2750180
4. Zhang, T.: Data offloading in mobile edge computing: A coalition and pricing based approach. IEEE Access 6, 2760–2767, 2017. doi:10.1109/ACCESS.2017.2785265
5. Hassanali eragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., Kantarci, B., Andreescu, S.: Health monitoring and management using internet-of-things (Iot) sensing with cloud-based processing: Opportunities and challenges. In: 2015 IEEE International Conference on Services Computing, pp. 285–292 (2015). doi:10.1109/SCC.2015.47. IEEE
6. Perera, C., Qin, Y., Estrella, J.C., Reiff-Marganiec, S., Vasilakos, A.V.: Fog computing for sustainable smart cities: A survey. ACM Computing Surveys (CSUR) 50(3), 1–43 (2017). doi:10.1145/3057266
7. Vasisht, D., Kapetanovic, Z., Won, J., Jin, X., Chandra, R., Sinha, S., Kapoor, A., Sudarshan, M., Stratman, S.: Farmbeats: An iot platform for data-driven agriculture. In: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17), pp. 515–529 (2017).
8. Byers, C.C.: Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled iot networks. IEEE Communications Magazine 55(8), 14–20 (2017). doi:10.1109/MCOM.2017.1600885
9. Hong, C.-H., Varghese, B.: Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. ACM Computing Surveys (CSUR) 52(5), 1–37 (2019). doi:10.1145/3326066
1. Pei, Z., Piao, S., Souidi, M.E.H., Qadir, M.Z., Li, G.: Coalition formation for multi-agent pursuit based on neural network. Journal of Intelligent & Robotic Systems 95(3), 887–899 (2019). doi:10.1007/s10846-018-0893-6

2. Hoefer, M., Vaz, D., Wagner, L.: Dynamics in matching and coalition formation games with structural constraints. Artificial Intelligence 262, 222–247 (2018). doi:10.1016/j.artint.2018.06.004

3. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: Vision and challenges. IEEE internet of things journal 3(5), 637–646 (2016). doi:10.1109/JIOT.2016.2579198

4. Cao, J., Zhang, Q., Shi, W.: Challenges and opportunities in edge computing. In: Edge Computing: A Primer, pp. 59–70. Springer, ??? (2018). doi:10.1007/978-3-030-02083-5_5

5. Barbarossa, S., Sardellitti, S., Di Lorenzo, P.: Joint allocation of computation and communication resources in multiuser mobile cloud computing. In: 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 26–30 (2013). doi:10.1109/SPAWC.2013.6612005. IEEE

6. Meng, S., Wang, Y., Miao, Z., Sun, K.: Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment. Peer-to-Peer Networking and Applications 11(3), 462–472 (2018). doi:10.1007/s12083-017-0544-x

7. Chen, M.-H., Dong, M., Liang, B.: Joint offloading decision and resource allocation for mobile cloud with computing access point. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3516–3520 (2016). doi:10.1109/ICASSP.2016.7472331. IEEE

8. Dinç, T.Q., Tang, J., La, Q.D., Quek, T.Q.: Offloading in mobile edge computing: Task allocation and computational frequency scaling. IEEE Transactions on Communications 65(8), 3571–3584 (2017). doi:10.1109/TCOMM.2017.2699660

9. Koshevoy, G., Suzuki, T., Talman, D.: Cooperative games with restricted formation of coalitions. Discrete Applied Mathematics 218, 1–13 (2017). doi:10.1016/j.dam.2016.09.003

10. Készty, L.A., Lauwers, L.: The coalition structure core is accessible. Games and Economic Behavior 48(1), 86–93 (2004). doi:10.1016/j.geb.2003.06.006

11. Tang, L., He, S.: Multi-user computation offloading in mobile edge computing: A behavioral perspective. IEEE Network 32(1), 48–53 (2018). doi:10.1109/MNET.2018.1700119

12. Wang, X., Liu, J.: A dynamic bargaining game with externalities. Journal of Systems Science and Complexity 31(6), 1591–1602 (2018). doi:10.1007/s11424-018-7358-6

13. Tianze, L., Muqing, W., Min, Z., Wenxing, L.: An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing. IEEE Access 5, 5609–5622 (2017). doi:10.1109/ACCESS.2017.2678102

14. Zhang, Y., Wang, J., Liu, S., Qian, C.: Game theory based real-time shop floor scheduling strategy and method for cloud manufacturing. International Journal of Intelligent Systems 32(4), 437–463 (2017). doi:10.1002/int.21868

15. Maxfield, R.R.: General equilibrium and the theory of directed graphs. Journal of Mathematical Economics
1. 27(1), 23–51 (1997). doi:10.1016/0304-4068(95)00763-6

23. Mas-Colell, A., Whinston, M.D., Green, J.R., et al.: Microeconomic Theory vol. 1. Oxford University Press New York, ?? (1995)

23. Liu, Z., Lin, M., Wierman, A., Low, S.H., Andrew, L.L.: Greening geographical load balancing. ACM SIGMETRICS Performance Evaluation Review 39(1), 193–204 (2011). doi:10.1145/2007116.2007139

33. Ardagna, D., Panico, B., Passacantando, M.: Generalized Nash equilibria for the service provisioning problem in cloud systems. IEEE Transactions on Services Computing 6(4), 429–442 (2012). doi:10.1109/TSC.2012.14

34. Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F.R., Han, Z.: Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching. IEEE Internet of Things Journal 4(5), 1204–1215 (2017). doi:10.1109/JIOT.2017.2688925

35. Tang, L., Chen, H.: Joint pricing and capacity planning in the iaas cloud market. IEEE Transactions on Cloud Computing 5(1), 57–70 (2014). doi:10.1109/TCC.2014.2372811

36. Roughgarden, T.: Algorithmic game theory. Commun. ACM 53(7), 78–86 (2010). doi:10.1145/1785414.1785439

40. Dreze, J.H., Greenberg, J.: Hedonic coalitions: Optimality and stability. Econometrica: Journal of the Econometric Society, 987–1003 (1980). doi:10.2307/1912943

41. Bogomolnaia, A., Jackson, M.O.: The stability of hedonic coalition structures. Games and Economic Behavior 38(2), 201–230 (2002). doi:10.1006/game.2001.0877

45. Aumann, R.J., Dreze, J.H.: Cooperative games with coalition structures. International Journal of Game Theory 3(4), 217–237 (1974). doi:10.1007/BF01766876

53. Guazzone, M., Anglano, C., Sereno, M.: A game-theoretic approach to coalition formation in green cloud federations. In: 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 618–625 (2014). doi:10.1109/CCGrid.2014.37. IEEE
Table 1 Parameters related to services.

| Service index | Service name  | Latency threshold (ms) | Number of containers | Service rate per container |
|---------------|--------------|------------------------|----------------------|---------------------------|
| 1             | Healthcare   | 5                      | 1                    | 500                       |
| 2             | Accelerated Video | 10                  | 1                    | 400                       |
| 3             | Virtual Reality | 20                   | 1                    | 300                       |
| 4             | Web Game     | 30                     | 1                    | 200                       |

Table 2 The Coalitions Formed with Different Arrival Rates

| Arrival index | EIP | Request Intensity λ | Average arrival rate | N_CRB | Ψ({i}) | E[φ_i] | Average E[φ_i] | Coalition |
|---------------|-----|----------------------|----------------------|-------|-------|--------|----------------|-----------|
| 19            | 5   | 32667.79             |                      | 66    | 8.7563| 12.05  |                |           |
| 20            | 1   | 1095.622             | 16114.82             | 5     | 1.4818| 5.699  | 8.96           | {1,2,3,5} |
| 21            | 3   | 22626.079            | 92                   | 8.1335| 11.15 |        |                |           |
| 22            | 1   | 8069.79              | 50                   | 4.5558| 6.93  |        |                |           |
| 23            | 5   | 41269.63             | 83                   | 8.7563| 11.67 |        |                |           |
| 24            | 2   | 23834.39             | 16789.925            | 100   | 8.9011| 12.00  | 9.29           | {5},{1,2,3}|
| 25            | 3   | 2857.55              | 12                   | 3.2534| 6.98  |        |                |           |
| 26            | 1   | 3558.13              | 22                   | 3.3476| 6.50  |        |                |           |
| 27            | 5   | 72456.64             | 145                  | 8.7563| 16.72 |        |                |           |
| 28            | 2   | 1351.85              | 18729.567            | 6     | 1.7802| 4.33   | 7.026          | {1,2,3,5} |
| 29            | 3   | 1087.94              | 5                    | 1.3539| 3.98  |        |                |           |
| 30            | 1   | 21.839               | 1                    | 0.1435| 3.07  |        |                |           |
| 31            | 5   | 16633.28             | 34                   | 9.9405| 10.39 |        |                |           |
| 32            | 2   | 50758.22             | 18610.117            | 212   | 8.9011| 13.09  | 8.58           | {5},{1,2,3}|
| 33            | 3   | 1557.73              | 7                    | 1.8995| 4.35  |        |                |           |
| 34            | 1   | 5491.239             | 34                   | 5.1799| 6.47  |        |                |           |
Table 3 The impact of $\beta$ on coalition formed

| $\beta$ | Average $\Psi(i)$ | Average $E[\phi_i]$ | Coalition         |
|-------|-------------------|---------------------|------------------|
| 1     | 7.4               | 10.42               | $\{1, 5\}, \{2, 3\}$ |
| 5     | 6.807             | 9.97                | $\{5\}, \{1, 2, 3\}$ |
| 10    | 6.065             | 9.29                | $\{5\}, \{1, 2, 3\}$ |
| 15    | 5.32              | 8.657               | $\{5\}, \{1, 2, 3\}$ |
| 20    | 4.58              | 8.03                | $\{1, 2, 3, 5\}$  |
| 30    | 3.095             | 6.94                | $\{1, 2, 3, 5\}$  |
Figure 1

System Architecture
Figure 2

Latency composition

Figure 3

Random distribution map of SPs and EIPs in Anning District, Lanzhou (China)
Figure 4

The evaluation indicator of Service to EIP
Figure 5

Revenue per unit cost (bang per buck)
Figure 6

Maximum revenue per unit cost (maximum bang per buck)
Figure 7

The utilities on different EIPs
Figure 8

The maximum arrival rate for different EIPs
Figure 9

Market equilibrium prices of CRB on different EIP
Figure 10

The impact of different budgets on the market equilibrium price
Figure 11

different arrival rates
Figure 12

the number of CRBs corresponding to arrival rates
Figure 13

utilities of different methods for different arrival rate
Figure 14

The impact of $\beta$ on average $\Psi\{i\}$ and $E[\phi_i]$.