



Game-Based Multi-MD with QoS Computation Offloading for Mobile Edge Computing of Limited Computation Capacity

Junyan Hu, Chubo Liu, Kenli Li, Keqin Li

► To cite this version:

Junyan Hu, Chubo Liu, Kenli Li, Keqin Li. Game-Based Multi-MD with QoS Computation Offloading for Mobile Edge Computing of Limited Computation Capacity. 16th IFIP International Conference on Network and Parallel Computing (NPC), Aug 2019, Hohhot, China. pp.16-27, 10.1007/978-3-030-30709-7_2 . hal-03770553

HAL Id: hal-03770553

<https://inria.hal.science/hal-03770553>

Submitted on 6 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Game-based Multi-MD with QoS Computation Offloading for Mobile Edge Computing of limited computation capacity

Junyan Hu^{1,2}, Chubo Liu^{1,2,4}, Kenli Li^{1,2}, and Keqin Li^{1,2,3,4}

¹ College of Computer Science and Electronic Engineering, Hunan University, Hunan, China, 410082.

junyanhu@hnu.edu.cn lkl@hnu.edu.cn liuchubo@hnu.edu.cn

² National supercomputing center in Changsha, Hunan, China, 410082

³ Department of Computer Science, State University of New York, New Paltz, New York 12561, USA. lik@newpaltz.edu

⁴ Authors for correspondence

Abstract. Mobile edge computing (MEC) is becoming a promising paradigm of providing cloud computing capabilities to the edge network, which can serve mobile devices (MDs) with computation-intensive and delay-sensitive tasks. Facing with high requirements of many MDs, it's essential for MEC with limited computation capacity to serve more MDs with QoS. For each mobile device, it is also desirable to have a low energy consumption with an expected deadline. To solve above problems, we propose a Game-based Computation Offloading (GCO) algorithm, which includes the task offloading profile and the transmission power controlling with the method of non-cooperative game. Our mechanism maximizes the number of served MDs with deadline, as well as minimizing the energy consumption of each MD whose task is executed on MEC. Specifically, Given the allocation of transmission power, a Greedy-Pruning algorithm is proposed to determine the number of tasks executed on MEC. Besides, each MD adopts his/her transmission power controlling strategy to compete the computation resource of MEC or minimize the energy consumption. A game model for illustrating the problem of task offloading is formulated to find a proper transmission power for each task and is proved the existence of Nash equilibrium solution. Experiments are simulated to evaluate the proposed algorithm in terms of effectiveness evaluation.

Keywords: Mobile edge computing · Nash equilibrium · non-cooperative game theory · task offloading · power controlling.

1 Introduction

Nowadays, Mobile Devices (MDs) are indispensable part in our daily life [1, 2]. With the popularity of smart MDs, many new computation-intensive and delay-sensitive applications have higher demands on quality of service (QoS) [3]. However, the limited resources of MDs e.g., battery, computation capacity,

cannot meet their own needs. Therefore, how to meet the high QoS requirements of multiple MDs with low energy consumption is an urgent problem to be solved.

Mobile edge computing (MEC) provides high-bandwidth, high-computing resources for nearby MDs to meet the high QoS demands for computation-intensive and latency-sensitive applications via edge network [4, 5]. For a multi-device MEC system with multiple parallel computation tasks requiring computing resources, MEC can be viewed as a small cloud with limited resources (processing speed, CPU cycle). Facing with resource requests from numerous devices, MEC should propose a resource allocation strategy that maximizes the number of served MDs with QoS requirements. For each MD, it has an expected value of delay, and on this basis, it is desirable to have a minimum energy consumption. The transmission rate and the received computation resource of each MD are affected by other MDs. Thus, if there are many devices that offload their tasks, the QoS experience of each MD will be deduced. In order to compete the resource for CPU cycle, a suitable transmission power controlling strategy mechanism for each MD should be proposed.

The remainder of the paper is organized as follows. In Sect. 2, we introduce the related work. Sect. 3 describes the system model and presents the problem that needs to be solved. In Sect. 4, we consider the problem as a non-cooperative game and propose Algorithm GCO to compute the Nash equilibrium solution. In Sect. 5, extensive experiments results indicate the feasibility of our algorithms. We conclude the works of this paper in Sect. 6.

2 Related Work

Task offloading for user requirements in MEC has been studied by many scholars and most of studies are analyzed from computational offloading, latency, storage, and energy efficiency. [6–8] are considered from optimizing the energy consumption of users. In [6], Chen *et al.* computed the energy harvesting for MEC by using Lyapunov Optimization method. Besides, some works and models considered from guaranteeing the deadline or minimizing average delay [9–12]. Fan *et al.* proposed an application aware workload scheduling mechanism for IoT based on MEC to minimize the average delay of application resource requests in [10]. [11] solved the problem of minimizing delay by using the method of one-dimensional search. And then in [13] and [14], Zhang and Chen *et al.* considered the proportional overhead on power consumption and latency. In addition, [15–17] optimized the transmission to achieve the offloading balance in the MEC by controlling the transmission power. In [15], Rodrigues *et al.* proposed a workload balance strategy for cloudlets to minimize the cost by using Transmission Power Control (TPC). In [16], Mao *et al.* minimized the weighted sum of the execution delay and energy consumption by optimizing the transmission. Different from above all, our work considers not only from the view of serving the maximum number of MDs with deadline constraint, but also from the perspective of each MD's minimum energy consumption.

Game theory plays an increasingly important method in MEC [18–21]. In [18], Chen *et al.* analyzed the multi-task offloading problem for MEC under the condition of multi-channel from the view of game theory. In [21], by using the theory

of Minority Games, Ranadheera *et al.* proposed a novel distributed server activation mechanism for computation offloading which guaranteed energy-efficient activation of servers as well as satisfaction of users quality-of-experience (QoE) requirements in terms of latency. Heuristically, our work introduces an adaptive transmission power mechanism in the competing process for limited-computation resources provided by MEC. We formulate a non-cooperative game-based mechanism for MEC's offloading decision making and MDs' power control.

3 System Model

We denote $\mathcal{N} = \{1, 2, \dots, N\}$ as the set of N MDs, each of which has computation-intensive and time-sensitive task to be completed. Let τ_n be the task of n , and the requirement of MD τ_n can be denoted as a tuple (c_n, d_n, T_n) , where c_n denotes the total number of required CPU cycles, d_n denotes the size of the input task data, and T_n denotes the expected time required to complete task τ_n . The task can be computed either locally on the mobile device or remotely executed on MEC via computation offloading. Therefore, we denote the decision profile $X = \{x_1, x_2, \dots, x_N\}$ as the set of indicator function for N MDs, where $x_n \in \{0, 1\}$. If the task of MD n is computed on MEC \mathcal{S} , $x_n = 1$, otherwise, $x_n = 0$. Besides, we denote J as the set of mobile devices, where $J = \{n | x_n = 1\}$. Here we consider the computational capacity of MEC \mathcal{S} , denoted as \mathcal{C} , is limited. If a MD prepares to offload his task to MEC \mathcal{S} , the energy and time consumption of communication and computation are considered.

3.1 Communication Model

If MD n offloads task τ_n to remotely edge execution, the input data should be transmitted to MEC servers of \mathcal{S} . Given the decision profile X and J , the communication rate of MD n ($n \in J$) via the wireless channel can be denoted as

$$r_n(X, \mathcal{P}) = B \log_2 \left(1 + \frac{p_n G_n}{\eta_0 + \sum_{i \in J \setminus \{n\}} p_i G_i} \right). \quad (1)$$

Here B is the channel bandwidth, and for simplicity, we only consider one channel. $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ is the transmission power profile of all MDs and each p_n can be chosen from the interval $[p_n, \bar{p}_n]$. Further, G_n , related to the environment and the distance, denotes the channel gain between MD n and MEC \mathcal{S} and η_0 is the background noise power. In (1), let $I_n = \sum_{i \in J \setminus \{n\}} p_i G_i$ be the sum of interference from other MDs who belong to set J . Note that the transmission rate can be affected not only the transmission power of itself but also the MDs which offload tasks to MEC \mathcal{S} .

3.2 Computation Model

If task τ_n of mobile device n is offloaded to MEC \mathcal{S} to execute, i.e., $x_n = 1$, the completion time will contain communication time and computation time. We define the completion time as

$$t_{n,off} = \frac{d_n}{r_n(X, \mathcal{P})} + \frac{c_n}{f_n} = \frac{d_n}{B \log_2 \left(1 + \frac{p_n G_n}{\eta_0 + \sum_{i \in J \setminus \{n\}} p_i G_i} \right)} + \frac{c_n}{f_n}, \quad (2)$$

where f_n is the computation capability (i.e., CPU cycles per second) assigned to MD n by the MEC \mathcal{S} . Therefore, the energy consumption can be denoted as

$$E_{n,off}(X, \mathcal{P}) = \frac{p_n d_n}{B \log_2(1 + \frac{p_n G_n}{\eta_0 + \sum_{i \in J \setminus \{n\}} p_i G_i})}. \quad (3)$$

3.3 MEC's resource allocation strategy

From the perspective of MEC \mathcal{S} with limited resource, serving as many MDs as possible is its primary goal. We consider distributed resource allocation for MDs, and model it as $\max_X |J|$ with the constraints $t_n \leq T_n, n \in J$ and $\sum_{n \in J} f_n \leq \mathcal{C}$, where $|\cdot|$ is the number of elements in set \cdot .

Theorem 1. *The issue $\max_X |J|$ that maximize the number of tasks with QoS executed on MEC is NP-hard.*

Algorithm 1 Greedy-Pruning algorithm

Require: $\mathcal{N}, \mathcal{P}, G, B, \mathcal{C}$.

Ensure: $J, f_n(J)$.

```

1:  $J \leftarrow \mathcal{N}, J_1 \leftarrow \{\emptyset\}, J_2 \leftarrow \{\emptyset\};$ 
2: Calculate each  $f'_n(J)$  ( $n \in J$ ) based on Eq. (4);
3: while ( $\sum_{k \in J} f'_k(J) > \mathcal{C}$ ) do
4:    $J_1 \leftarrow J_1 \cup \{\arg \min_i \sum_{j \in J \setminus \{i\}} f'_j(J \setminus \{i\})\};$ 
5:   while ( $J_1 \neq J_2$ ) do
6:      $J_2 \leftarrow J_1;$ 
7:     for ( $l \in J_2$ ) do
8:        $J_1 \leftarrow (J_1 \setminus \{l\}) \cup$ 
          $\{\arg \min_i \sum_{j \in \mathcal{C}_{\mathcal{N}}^{J_1 \setminus \{l\}} \setminus \{i\}} f'_j(\mathcal{C}_{\mathcal{N}}^{J_1 \setminus \{l\}} \setminus \{l\})\};$ 
9:    $J \leftarrow \mathcal{N} \setminus J_1;$ 
10: return  $J, f_n(J)$ .
```

In order to solve the problem $\max_X |J|$, we propose a Greedy-Pruning algorithm (Algorithm 1). Let $f'_n(J)$ be the critical point of computation capability that MD n needs.

$$f_n(J) \geq \frac{c_n}{T_n - \frac{d_n}{\gamma_n(X, \mathcal{P})}} = f'_n(J). \quad (4)$$

Assuming $\sum_{n \in \mathcal{N}} f'_n \leq \mathcal{C}$, then there is $J = \mathcal{N}$. Otherwise, MEC \mathcal{S} needs to filter out some MDs to maximize the number of beneficial MDs with QoS. In algorithm 1, J is the set of MDs to be selected, and J_1 is the set of MDs to be filtered out. In the outer *while* loop of the line 3-9, once $\sum_{k \in J} f'_k(J) > \mathcal{C}$, an appropriate MD will be added to J_1 to check whether the condition $\sum_{k \in \tilde{J}} f'_k(\tilde{J}) \leq \mathcal{C}$ is satisfied, where \tilde{J} is the updated J . In each round of preparation to remove a MD to

J_1 , we use $\min_{j \in \tilde{J}} f'_j(\tilde{J})$ as the objective function. But removing MD i in J that minimizes $\sum_{j \in J \setminus \{i\}} f'_j(J \setminus \{i\})$ directly does not guarantee that updated J is globally optimal. If there is always

$$l = \arg \min_i \sum_{j \in (J \cup \{l\}) \setminus \{i\}} f'_j((J \cup \{l\}) \setminus \{i\}). \quad (5)$$

for any MD $(l \in \mathcal{N} \setminus J)$, J is optimal.

3.4 Power Control Strategy of Mobile Device

In this section, we explore that how to minimize each MD's energy consumption within the expected delay range. As can be seen from Eq. (2), $t_{n,off}$ decreases as p_n increases. Given \mathcal{F} and the expected time T_n required to complete the task τ_n , $t_{n,off} \leq T_n$ can be introduced as follows

$$p_n \geq (2^{\frac{d_n}{(T_n - \frac{\tau_n}{f_n})B}} - 1) \left(\frac{\eta_0 + I_n}{G_n} \right) = p'_n. \quad (6)$$

We denote p'_n as the critical power of MD n . If $p'_n > \bar{p}_n$, MD n will not choose to execute his task τ_n on MEC. We assume $p'_n \leq \bar{p}_n$ and consider the energy consumption of MD n in the interval $[\max\{p'_n, p_n\}, \bar{p}_n]$.

In each round, MD n , who does not execute his task τ_n on the MEC, can increase p_n to provide his own competitiveness. This leads to two outcomes: removing one of the other MDs in J or adding to the set J directly.

Removing one of the other MDs in J : Increasing p_n to p_n^1 and satisfying the conditions

$$\begin{aligned} \arg \min_k \sum_{j \in J_3} f'_j(J_3 \setminus \{k\}) &\neq n, (J_3 = J \cup \{n\}), \\ \min_k \sum_{j \in J_3} f'_j(J_3 \setminus \{k\}) &\leq \mathcal{C}. \end{aligned} \quad (7)$$

Adding to the set J directly: Increasing p_n to p_n^2 and satisfying the condition

$$\sum_{j \in J \cup \{n\}} f'_j(J \cup \{n\}) = \mathcal{C}. \quad (8)$$

Considering the energy consumption and $p_n \leq \bar{p}_n$, we define $\tilde{p}_n = \min\{p_n^1, p_n^2, \bar{p}_n\}$, where \tilde{p}_n is the updated p_n in next round. We denote $\tilde{\mathcal{P}} = (\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_N)$. We propose a Binary search algorithm (Calculate- $\tilde{\mathcal{P}}(\cdot)$) to update p_n .

4 Game Formulation And Analyses

4.1 Game Formulation

Let $\mathcal{P}_{-\mathbf{n}} = (p_1, \dots, p_{n-1}, p_{n+1}, \dots, p_N)$ be the transmission power profile of all MDs except MD n . Let \mathcal{P}_n be the set of power and decision making strategies for MD n , i.e., $p_n \in \mathcal{P}_n$. Given other MDs' transmission power $\mathcal{P}_{-\mathbf{n}}$, MD n would like to select a proper decision p_n to compete the computation resource of MEC \mathcal{S} and minimize his own energy consumption, under the condition of satisfying QoS. The objective function of MD n can be written as follows $\min E_n(X, \mathcal{P})$.

Algorithm 2 Calculate $\tilde{\mathcal{P}}(\cdot)$

Require: $\mathcal{N}, \mathcal{P}, G, B, \mathcal{C}, J, \varepsilon$.
Ensure: $\tilde{\mathcal{P}}$.

- 1: **for** $n \in J$ **do**
- 2: $\bar{p}_n = p_n$;
- 3: **for** $n \in \mathcal{N} \setminus J$ **do**
- 4: $l_1 p_n \leftarrow p_n, r_1 p_n \leftarrow \bar{p}_n$;
- 5: $l_2 p_n \leftarrow p_n, r_2 p_n \leftarrow \bar{p}_n$;
- 6: **while** $(|r_1 p_n - l_1 p_n| > \varepsilon)$ **do**
- 7: $mid_1 \leftarrow \frac{l_1 p_n + r_1 p_n}{2}$;
- 8: **if** Conditions in Eq. (7) are satisfied **then**
- 9: $r_1 p_n \leftarrow mid_1$;
- 10: **else**
- 11: $l_1 p_n \leftarrow mid_1$;
- 12: $p_n^1 \leftarrow r_1 p_n$;
- 13: **while** $(|r_2 p_n - l_2 p_n| > \varepsilon)$ **do**
- 14: $mid_2 \leftarrow \frac{l_2 p_n + r_2 p_n}{2}$;
- 15: **if** Condition in Eq. (8) is satisfied **then**
- 16: $r_2 p_n \leftarrow mid_2$;
- 17: **else**
- 18: $l_2 p_n \leftarrow mid_2$;
- 19: $p_n^2 \leftarrow r_2 p_n$;
- 20: $\bar{p}_n = \min\{p_n^1, p_n^2, \bar{p}_n\}$;
- 21: **return** $\tilde{\mathcal{P}}$.

The strategy set of MEC \mathcal{S} is \mathcal{X} and his objective function is maximizing the beneficial number of MDs $|J|$. Then, the multi-device computation offloading game can be represented as G , where $G = \{(\mathcal{P}_n)_{n \in \mathcal{N}}, \mathcal{X}; (E_n)_{n \in \mathcal{N}}, |J|\}$.

For all MDs, $\mathcal{P}^* = \{p_1^*, \dots, p_N^*\}$ is the optimal countermeasure strategy. That is to say, for MD n and any $p_n \in \mathcal{P}_n$, there is $E_n(p_n, \mathcal{P}_{-\mathbf{n}}^*) \geq E_n(p_n^*, \mathcal{P}_{-\mathbf{n}}^*)$. For MEC \mathcal{S} and any $X = (x_1, x_2, \dots, x_N)$, $|J(X^*)| \geq |J(X)|$.

4.2 Nash Equilibrium Existence Analysis

Theorem 1 Given $\mathcal{N}, G, B, \mathcal{C}$, and $p_n \geq \max\{p_n', \underline{p}_n\}$, non-cooperative game strategies for N MDs and MEC \mathcal{S} $\mathcal{M} = (\mathcal{N}, \{\mathcal{P}_n\}_{n \in \mathcal{N}}, \{E_{n,off}\}; \mathcal{S}, \mathcal{X}, |J|)$ have a Nash equilibrium $\langle \mathcal{P}^*, X^* \rangle$, ($p^* \in \mathcal{P}_n, X^* \in \mathcal{X}$).

Proof. We easily know that $\frac{\partial E_{n,off}(X, \mathcal{P})}{\partial p_n} > 0$ ($p_n > 0$). Based on Eqs. (1) we can obtain that

$$\frac{\partial^2 r_n(X, \mathcal{P})}{\partial p_n^2} = -\frac{BG_n^2}{\ln 2(\eta_0 + I_n + p_n G_n)^2}. \quad (9)$$

$E_{n,off}(X, \mathcal{P})$ is taken the second derivative with respect to p_n , and it yields that

$$\frac{\partial^2 E_{n,off}}{\partial p_n^2} = \frac{d_n B^2 G_n}{(\eta_0 + I_n) \mu r_n^3 \ln 2} \left[\left(-1 - \frac{1}{\mu}\right) \log_2 \mu + \frac{2}{\ln 2} \left(1 - \frac{1}{\mu}\right) \right], \quad (10)$$

where $\mu = \frac{\eta_0 + I_n + p_n G_n}{\eta_0 + I_n}$ and $\mu > 1$.

Let function $g(x) = (-1 - \frac{1}{x}) \log_2 x + \frac{2}{\ln 2}(1 - \frac{1}{x})$. We analyse function $g(x)$, and its derivative for x is

$$g'(x) = \frac{-x + \ln 2 \log_2 x + 1}{x^2 \ln 2}. \quad (11)$$

Let function $s(x) = -x + \ln 2 \log_2 x$. When $x \geq 1$, $s(x)$ is monotonically decreasing, and $s(x) \leq s(1) = 0$. Therefore, when $x \geq 1$, $g'(x) < 0$, $g(x)$ is monotonically decreasing, and $g(x) \leq g(1) = 0$. Because $\mu > 1$, the second derivative of $E_{n,off}(X, \mathcal{P})$ with respect to p_n is always less than 0, i.e., $\frac{\partial^2 E_{n,off}}{\partial p_n^2} \leq 0$ ($p_n \geq \max\{p'_n, \underline{p}_n\}$). Based on $\frac{\partial E_{n,off}(X, \mathcal{P})}{\partial p_n} > 0$ and the power variable of each MD is a closed interval, $E_{n,off}(X, \mathcal{P})$ takes the minimal value when $p_n = \max\{p'_n, \underline{p}_n\}$. Thus, $p_n^* = \max\{p'_n, \underline{p}_n\}$, and for any $p_n \geq p_n^*$, there always is $E_n(p_n, \mathcal{P}_{-n}^*) \geq E_n(p_n^*, \mathcal{P}_{-n}^*)$.

For MEC \mathcal{S} , J^* is the first set in Algorithm Greedy-pruning that satisfies the following conditions: (1) $\sum_{k \in J^*} f'_k(J^*) \leq \mathcal{C}$; (2) for any MD $l \in \mathcal{N} \setminus J^*$, there is always

$$l = \arg \min_i \sum_{j \in (J^* \cup \{l\}) \setminus \{i\}} f'_j((J^* \cup \{l\}) \setminus \{i\}).$$

Then, the maximum number of beneficial MDs with QoS will no longer decrease. Therefore, for any offloading scheduling profile $X \in \mathcal{X}$ satisfying the conditions $t_n \leq T_n, n \in J$ and $\sum_{n \in J} f_n \leq \mathcal{C}$, there always will be $|J(X)| = \sum_{n \in \mathcal{N}} x_n \leq |J(X^*)| = \sum_{n \in \mathcal{N}} x_n^*$.

4.3 Nash Equilibrium Solution Computation

We propose a Game-based Computation Offloading (GCO) algorithm 3 to find the equilibrium solution.

Algorithm 3 Game-based Computation Offloading (GCO)

Require: $\mathcal{N}, \underline{\mathcal{P}}, \bar{\mathcal{P}}, G, B, \mathcal{C}, \varepsilon, \delta$.

Ensure: \mathcal{P}, J, X .

```

1:  $\mathcal{N}(0) \leftarrow \mathcal{N}$ ;
2:  $s \leftarrow 1$ ;
3:  $p_n(0) \leftarrow \underline{\mathcal{P}}(\mathcal{N}(s-1))$ ;
4:  $\langle J(0), f_n(J(0)) \rangle \leftarrow \text{GP}(\mathcal{N}(0), p_n(0), G, B, \mathcal{C})$ ;
5:  $t \leftarrow 0$ ;
6: while  $|P(t+1) - P(t)| < \delta$  do
7:    $p_n(t+1) \leftarrow \text{Calculate\_}\tilde{\mathcal{P}}(\mathcal{N}(t), p_n(t), G, B, \mathcal{C}, J(t), \varepsilon)$ ;
8:    $\langle J(t+1), f_n(J(t+1)) \rangle \leftarrow \text{GP}(\mathcal{N}(s), p_n(t+1), G, B, \mathcal{C})$ ;
9:    $t \leftarrow t+1$ ;
10:  $J \leftarrow J(t)$ ;
11:  $\mathcal{N}(s) \leftarrow \mathcal{N}(s-1)$ ;
12: while  $\mathcal{N}(s) \neq \mathcal{N}(s-1)$  do
13:    $s \leftarrow s+1$ ;
14:   loop steps 3 to 11;
15: return  $\mathcal{P}, J, X$ .
```

5 Simulations

5.1 Simulation Settings

We evaluate the system performance of the proposed GCO based on the interaction of MEC \mathcal{S} and multiple mobile devices in this section. We consider 50 MDs in this system. The size of the input task data d_n of each MD n is randomly selected from the interval $(0, 2]$ MB and the total number of required CPU cycles $c_n = d_n \cdot w_n$, where w_n is the workload requirements of task τ_n ($w_n \in [100, 500]$ cycles/bit). Similarly, the expected Time T_n of MD n also follows a uniform distribution with $(0, 3]$ s. The minimum transmission power p_n is 100mWatts, and the maximum value is randomly selected from the interval $[1000, 3000]$ mWatts. We consider MEC \mathcal{S} has a coverage range of 50m. The computational capacity C of MEC \mathcal{S} is 1GHz. The bandwidth $B = 10$ MHz and the background noise power $\eta_0 = -100$ dBm. Based on the wireless interference model for urban cellular radio environment, the channel gain $G_n = dis_n^\alpha$, where dis_n is the distance between MD n and the MEC \mathcal{S} and $\alpha = -4$ is the path loss factor.

5.2 Convergence of Algorithm GCO

Figs. 1 and 2 illustrate the convergence process of transmission power for each MD by executing our proposed GCO algorithm. With the number of iterations increasing, the transmission power of each MD is increasing and then the curve reaches to a stable value. During the process of computing, some MDs will withdraw the resource competition if the transmission power is higher than their accepted maximum value, i.e., $p_i > \bar{p}_i$. Fig. 2 is the transmission power curve of MDs who cancel to compete the computation resource of MEC \mathcal{S} . Moreover, we can know that the transmission power can be obtained after 6 iterations, which shows high efficiency of our proposed algorithm.

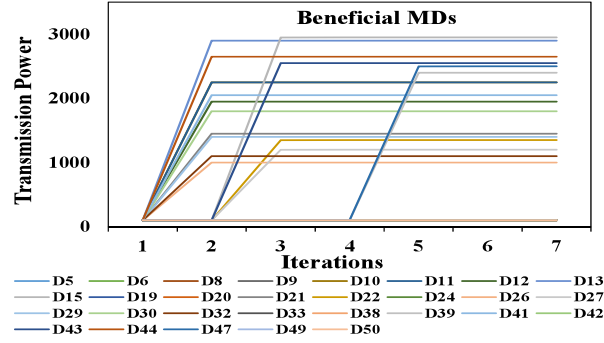


Fig. 1. Change of transmission power and beneficial MDs in the iterative process.

Figs. 3 and 4 is a curve of the number of MDs who obtain computing resources provided by MEC \mathcal{S} and a bar graph of the average energy consumption during the iterative process, respectively. At the beginning, each MD's transmission power is set as the initial value, i.e., the minimum value. The number of MDs with QoS served by the MEC \mathcal{S} with limited computing resources is 23 and the average energy consumption is about 70. Each MD increases its transmission power to compete for the computing resources of MEC \mathcal{S} , which causes the average energy

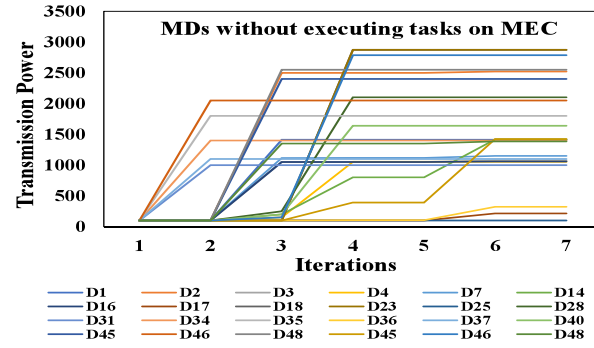


Fig. 2. Change of transmission power and non-beneficial MDs in the iterative process.

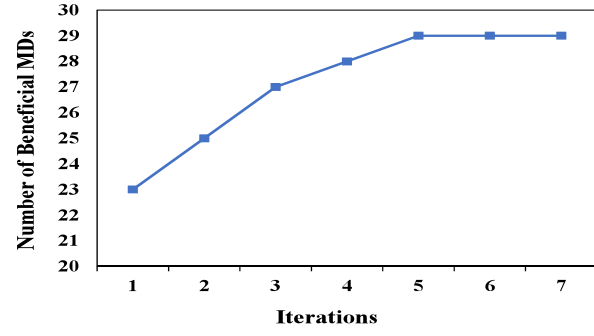


Fig. 3. The process change of number of beneficial MDs.

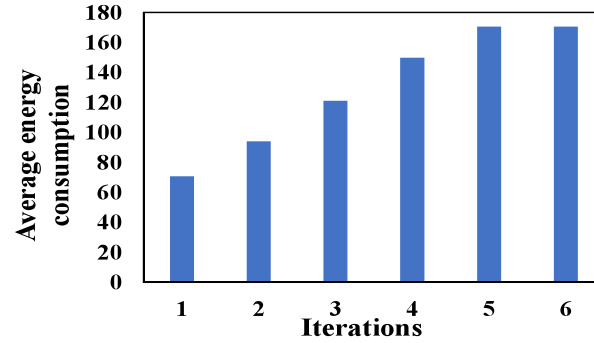


Fig. 4. The process change of average energy consumption.

consumption to rise during the iteration, as shown in Fig. 4. In Fig. 3, after several rounds of mutual negotiation between MDs, the number of MDs who can use the computing resources provided by MEC \mathcal{S} gradually increases and maintains stable at the value 29 as the number of iterations increases.

5.3 Performance Evaluation

The performance of GCO algorithm is evaluated from two respects: the number of iterations and the execution time. The variable is the number of MDs N ,

which increases by 10 from 10 to 50. For each N , we repeat the experiment many times. The experimental results are shown in Figs. 5 and 6.

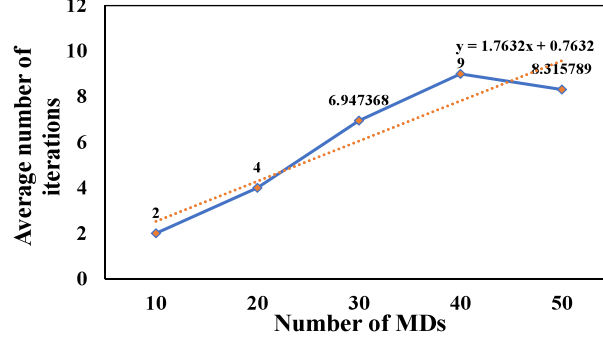


Fig. 5. Average iterative times of different scales of MDs.

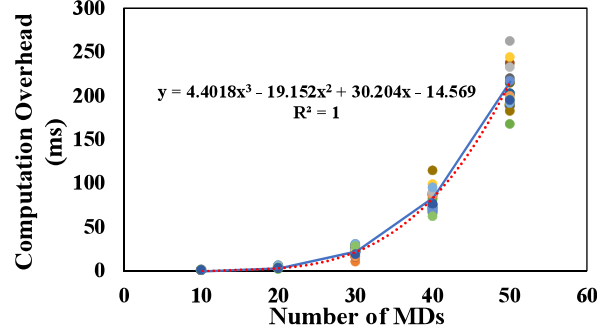


Fig. 6. Computation Overhead of different scales of MDs.

Fig. 5 and Fig. 6 show the number curve of iterations and iterative time curve of Algorithm GCO as the number of MDs increases, respectively. The blue line is an average curve in each figure. In Fig. 5, the general trend of the curve increases linearly and slowly. Besides, even if the number of MDs is 50, the average number of iterations is very small. In Fig. 6, as the scale of MDs increases, the computation overhead curve increases in a polynomial. The red dashed line is the trend line of the computation overhead curve, which is a second order polynomial. The fitting degree of the trend line and the time curve is 0.9908. When the number of MDs reaches 50, the average overhead is 225ms, which is rapid and shows the high efficiency of our proposed algorithm.

6 Conclusions

Our study focuses on the task offloading problem of one MEC and multiple MDs with delay deadlines. From the perspective of non-cooperative game theoretical method, the number of served MDs with delay deadline and the energy consumption of all tasks executed on MEC \mathcal{S} are alternately optimized. We prove the

existence of Nash equilibrium solution and propose GCO algorithm to solve it. Besides, the convergence of the algorithm is also analyzed. Extensive simulated experiments results validate and show the feasibility of our proposed method.

7 Acknowledgments

The research was partially funded by the National Key R&D Program of China (Grant No. 2018YFB1003401), the Program of National Natural Science Foundation of China (Grant No.61751204).

References

1. N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, Feb 2018.
2. P. Porambage, J. Okwuibe, M. Liyanage, T. Taleb, and M. Ylianttila, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–1, 2018.
3. Z. Ning, X. Wang, and J. Huang, "Mobile edge computing-enabled 5g vehicular networks: Toward the integration of communication and computing," *IEEE Vehicular Technology Magazine*, vol. PP, no. 99, pp. 1–1, 2018.
4. W. Kai, Y. Hao, Q. Wei, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. PP, no. 99, pp. 1–6, 2018.
5. H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiberwireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, May 2018.
6. W. Chen, W. Dong, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE TSC*, vol. PP, no. 99, pp. 1–1, 1939.
7. L. Yang, H. Zhang, L. Ming, J. Guo, and J. Hong, "Mobile edge computing empowered energy efficient task offloading in 5g," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2018.
8. W. Feng, X. Jie, W. Xin, S. Cui, W. Feng, X. Jie, W. Xin, S. Cui, W. Feng, and X. Jie, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, March 2017.
9. C. Min and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, March 2018.
10. F. Qiang and N. Ansari, "Application aware workload allocation for edge computing based iot," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.
11. J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *IEEE International Symposium on Information Theory*, April 2016.
12. S. Xiang and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1481–1484, 2017.
13. Z. Jiao, X. Hu, Z. Ning, C. H. Ngai, Z. Li, J. Wei, J. Cheng, and B. Hu, "Energy-latency trade-off for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.
14. X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, pp. 1–1, Oct. 2015.
15. T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE TC*, vol. PP, no. 99, pp. 1–1, 2018.
16. Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Wireless Communications and Networking Conference*, 2017.

17. X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 774–777, 2017.
18. C. Xu, J. Lei, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
19. X. Hu, K. K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 1–1, April 2018.
20. K. Li, "A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing," *IEEE Transactions on Sustainable Computing*, vol. PP, no. 99, pp. 1–1, 2018.
21. S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wireless Communications Letters*, 2018.