International Journal of Innovative
Computing, Information and Control
Volume **18**, Number **5**, October **2022**

ICIC International ©2022 ISSN 1349-4198
pp. 1413–1429

ARTIFICIAL POTENTIAL FIELD-BASED RESOURCE ALLOCATION FOR MOBILE EDGE COMPUTING

Jianhua Liu, Zibo Wu, Jiaqing Shen, Jiajia Liu and Xiaoguang Tu

Institute of Electronic and Electrical Engineering
Civil Aviation Flight University of China
No. 46, 4th Section, Nanchang Road, Guanghan 618307, P. R. China
{ jianhuacafuc13; wuzibo; shengjiaqing; cafucljj; xguangtu }@cafuc.edu.cn

Received February 2022; revised June 2022

Abstract. *Mobile edge computing (MEC) is a promising technology for improving latency in mainly real-time applications, such as smart airports, smart maintenances, and AR/VR. And it is an effective technology that helps local servers or the cloud to release the network overload at a rush time. However, an alternative network environment and mobility-agnostic trajectory brings new challenges for minimizing communication latency. To solve these challenges, we present a resource allocation scheme based on an artificial potential field (RAAPF), which provides an adaptable resource allocation in the tradeoff between distance and node capacity conditions. Futhermore, the potential energy of RAAPF is transformed into a knapsack problem in the MEC to optimize the latency, and a heuristic algorithm is applied to finding the optimal solution. The simulation results show that the latency of our method is reduced by 25% compared to that of mobility-agnostic online resource allocation within a few node numbers; the latency gap between the two methods increases significantly as the number of nodes increases, and our scheme can always achieve a lower latency performance under different scenes.*
Keywords: Mobile edge computing, Resource allocation, Artificial potential field, Latency

1. **Introduction.** Mobile edge computing (MEC) is a paradigm for the decentralization of traditional cloud computing, which realizes "the last 100 meters" [1] data communication and results in a lower latency because of shorter transmission distance. A low time delay is a key index for real-time applications, such as HD live [2], AR/VR [3], target tracking [4], automatic drive [5], and novel coronavirus monitoring systems [6]. The 5th generation mobile networks (5G) combined with the Internet of Things (IoT) in MEC conduct large-scale communication and bring users a faster experience with restrained bandwidth [7].

MEC is proposed to share the pressure of the cloud and is designed to cope with tasks with feasible costs [8]. Instead of tasks moving to the remote cloud for mobile computing, MEC can be viewed as "cloud closer to the ground" [9]. The mobile edge server is located at the edge of the wireless network, closer to the user than the cloud, and can efficiently provide services to the surrounding users. MEC allows mobile terminals to migrate computing tasks to nearby mobile edge nodes, such as cellular base stations [10] and Wi-Fi access points [11]. Meanwhile, some data at the edge of the network can be processed and shared. Compared with cloud computing, MEC features low latency, location sensitivity, and improves quality of service (QoS) for streaming media and real-time applications [12, 13]. In addition, the development of artificial intelligence (AI) has become one of the most prevalent technologies for providing convenience and entertainment for users. MEC

DOI: 10.24507/ijicic.18.05.1413

is a practical technology that transforms some tasks from the local side to the edge node server to ensure a stable quality of experience (QoE) and reduces local CPU pressure [14]. Simultaneously, the distance between the user and the service node becomes closer by constructing the MEC frame. Furthermore, edge nodes around user terminals bring considerable signal coverage [15].

More available nodes indicate more choices of resource allocation for users. How to reduce service delay, including transmission delay and executing delay, is regarded as a mathematical programming problem under constraint conditions [16]. In [17], the authors proposed an IoT-Cloud model that improves the transmission delay and executing delay by optimizing the resource allocation strategy in MEC. The works in [18, 19] showed that using an unmanned aerial vehicle (UAV) in communication can reduce signal power fading because of the multipath effect and the end-to-end distance. Additionally, UAVs can be flexibly deployed in places where many land-based devices cannot reach; thus, MEC based on UAVs can be applied in some real-time scenes to satisfy a particular service. In [18], the tasks obeyed the queuing theory when selecting servers; however, the scheme was impractical because it only considered the task arrival rate but ignored the task size. The wireless connection is multi-cross within the signal coverage for an MEC communication system, and each edge node can share the workload on user terminal. The tasks are gathering and waiting for available feasible edge nodes that may be busy processing the previous tasks, and the targeted-direction tasks are transformed to the target nodes. In MEC framework, the users want to submit their tasks to edge nodes for rapid processing. The tasks will generate a task queue on an edge server. The server will allocate the tasks to select edge nodes. For example, edge node 1 is processing tasks which arrived in the task queue at the last time slot, and edge node 2 is selected to process the tasks at the current time slot. Other edge nodes are waiting as candidate nodes to be assigned tasks, please refer to Figure 1.

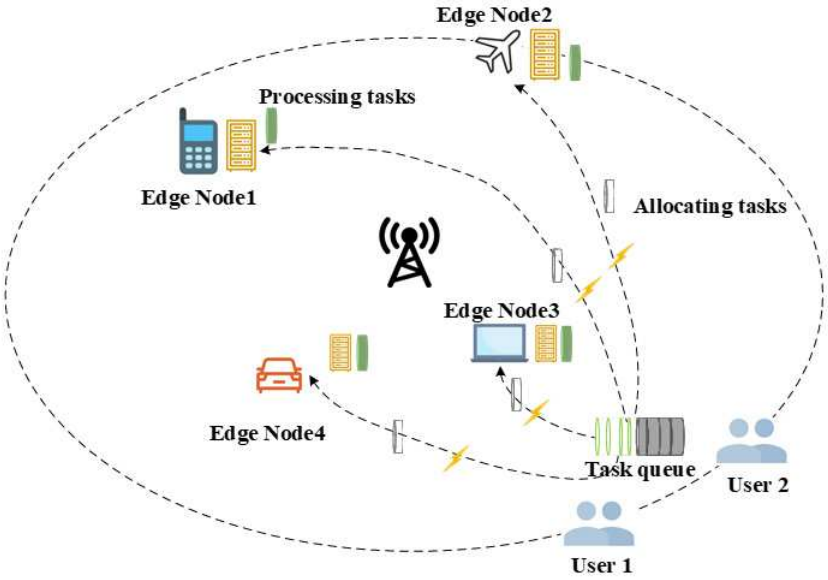


FIGURE 1. MEC framework

However, the actual communication environment is complex, and many problems need to be addressed, such as multipath reliability. Although more edge nodes bring a less sensitive delay [20], the service cost is expensive and uneconomical. Many recent works have paid more attention to the optimization algorithm for known resource allocation [21] rather than seeking an optimized task transmission solution, which deteriorates arithmetic

complexity and ignores the spare nodes in the whole communication environment. In this study, we use a heuristic algorithm (HA) to construct a resource allocation scheme which can assign reasonable edge node to tasks considering the distance, capacity, computing power, and task size.

The contributions of our work can be summarized in the following.

• We propose an artificial potential field in MEC and construct an RAAPF method. The computing power, remaining storage capacity, task size, and distance are built and restrained. Aiming at a min-latency scheme with various edge nodes using the RAAPF method, we find a solution when mathematical programming is established.

• The potential field is transformed into an NP-hard problem. To solve this problem, a knapsack model is established. The optimal solution is then determined by the simulated annealing algorithm (SA). The complexity of the SA is $O(N)$, where N is the total number of available edge nodes.

• Whether the model is in 4G or 5G, and whether the network status is free or busy, the RAAPF could carry out the best edge node selection in our experimental environment. Compared with the mobility-agnostic online algorithm (MOERA) [22] based on the regularization technique, the results show that RAAPF method provides users with an optimal scheme for task offloading, which achieves significant optimization for latency.

The rest of this paper is organized as follows. Section 2 introduces the related work and proposes an efficient framework for MEC networks and presents the similarities and differences between the artificial potential field and MEC field. We explain why the artificial potential field is adaptable to the MEC environment. In Section 3, the problem is formulated, and we propose an SA to find the optimal solution. Section 4 establishes the model under many restraint conditions. In terms of latency, for different user numbers, node numbers, CPU speeds, and task sizes, our model achieves better performance than other methods. We discuss some practical issues in Section 5. Finally, Section 6 concludes the paper and introduces future work.

2. **Related Work.**

2.1. **Edge computing.**

Computing offloading involves offloading decisions and resource allocation. The offloading decision mainly solves the problems of how to offload, how much to be offloaded, and what to be offloaded.

The offloading decision can be divided into dynamic offloading and static offloading [23]: static offloading means how to migrate the tasks which are determined before the offloading; dynamic offloading refers to how to migrate the tasks which are undetermined, and it needs to dynamically change the offloading strategy.

The user of edge computing divides the tasks that need to be processed, and trys to maintain the functional integrity of each part of the program after segmentation to facilitate resource allocation. A user terminal consists of a code parser, system parser, and decision engine [24]. Consequently, an offloading decision requires three steps [25]. First, the code parser determines which task can be assisted based on its application type and code/data partition. Second, the system parser is responsible for monitoring parameters such as available bandwidth, the size of data to be offloaded, or the energy consumed to execute the local application. Third, the decision engine determines whether to offload or not. In general, there are three means to calculate the offloading decision: local execution, full offloading, and partial offloading, respectively.

The optimization objectives of the offloading decisions can be divided into three types [26]: optimizing latency, optimizing energy consumption, and weighing latency and energy consumption. In detail, as for offloading decision to minimize latency: service latency,

which affects QoE, is a critical performance index. For real-time applications, the latency caused by task migration from the user terminal to the cloud is difficult to be accepted. All offloading decisions must meet the latency constraints in each mobile edge node. Therefore, to meet QoE requirements, an offloading decision to minimize latency for real-time applications is necessary. And for offloading decision to the tradeoff between latency and energy consumption: according to the specific requirement for some applications, researchers trade off the time delay and energy consumption to carry out the least cost loss. However, simply optimizing the service delay and ignoring the energy consumption of the mobile terminal will lead to a rapid decline in the battery power [27], which will lead to a reduction in QoE.

In addition, resource allocation focuses on how to allocate resources after the user terminal decides to offload [21]. Research on resource allocation mainly includes centralization and decentralization. For centralized resource allocation, the MEC server considers all the mobile information, including channel status information and computing requests, makes a resource allocation decision, and informs the mobile device of this decision. However, multi-users and multi-MEC servers are distributed in actual application environment, which has high computational complexity and extensive limitations [28]. Researchers present a series of distributed resource allocation algorithms to solve these problems [29, 30]. Kim et al. [31] proposed the method of logistics allocation, and its distributed resource allocation brings inspiration to the resource allocation scheme of edge computing.

With the rapid development of IoT, edge computing focuses on improving the QoE on various application scenes to overcome the long-latency challenge. Many previous works have proposed deploying servers in the edge network to obtain short-distance communication and low latency [32]. The resource allocation scheme in edge computing is significant. Machine learning is used in a sophisticated and huge communication network to obtain an adaptable value to meet user requirements [38]. Some heuristic algorithms, such as ant colony optimization (ACO), simulated annealing algorithm (SA), and particle swarm optimization (PSO), are applied to solving the optimal value in a complex mathematical model [34, 35]. Under many constraints, the min-latency problem is transformed into mathematical programming, for example, the problem is convex optimization using the Lagrange multiplier method, KKT, etc. [36, 37]. Building a neural network with a prepared training set, machine learning is still a prevalent method for pursuing the least loss function. However, the complexity is relevant to the neural network structure, and the convergence is difficult to guarantee [38]. Heuristic algorithm is a common solution for minimizing an objective function [39]. For the deployment of edge nodes, UAVs can set up an available wireless communication link anywhere without constraints to resolve channel fading due to the multipath effect caused by dense buildings [40]. However, because of the flexible but expensive costs of the deployment of UAVs, UAV communication in edge computing is used in a special area, whose energy consumption cannot be guaranteed.

In this paper, we propose a resource allocation method for optimizing the latency on full offloading. To find the most feasible edge node for each user at time slot t, the attractive potential energy is proposed, which is combined with the artificial potential field. To find the most suitable nodes for users, we consider the four indexes: node capacity, computational power, task size, and communication distance, respectively. The technique for order preference by similarity to an ideal solution (TOPSIS) method is used to score for each node to the user. We sort the result and transform the problem into a knapsack problem using SA. Combined with a rational scheme and heuristic algorithm, resource allocation based on an artificial potential field is proposed.

2.2. **Artificial potential field.** The traditional artificial potential field method assumes that the robot is moving in an abstract artificial force field. The artificial field consists of a repulsive potential field and an attractive potential field in the workspace. The goal position produces an attractive force that makes the mobile robot move toward it. In addition, obstacles generate a repulsive force, which is inversely proportional to the distance from the robot to the obstacles and is pointing away from obstacles. The robot moves from high to low potential fields along the negative of the total potential field. Consequently, the robot moving to the goal position can be considered from a high-value state to a low-value state. The artificial potential field method, including attraction and repulsion fields, is common for local path planning. The attraction force is generated by the goal point, which causes the item to move toward its direction. The repulsion force is generated by obstacles, which prevents the item from colliding with them. Setting the attraction and repulsion force in a special area, the item tends to move along the tendency. This tendency is the direction of gradient descent. The resultant force on the item at each point along the path is equal to the sum of the above two forces. The formula for the attraction field function is as follows [41]:

$$U_{att}(q) = \frac{1}{2}\xi d^2(q, q_{goal}) \tag{1}$$

The attractive force is given by the negative gradient of the attractive potential, where ξ is a positive scaling factor, $d(q, q_{goal})$ is the distance between robot q and goal q_{goal}.

The derivative of the attraction field with respect to distance is obtained, and the gravitational force formula is as follows:

$$F_{att}(q) = U'_{att}(q) = \xi(q, q_{goal}) \tag{2}$$

The repulsion field corresponds with distance and electricity; the formula is as follows:

$$U_{rep}(q) = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{\rho(q, q_{obs})} - \dfrac{1}{\rho_0}\right)^2, & \text{if } \rho(q, q_{obs}) \le \rho_0 \\ 0, & \text{if } \rho(q, q_{obs}) > \rho_0 \end{cases} \tag{3}$$

where η is a positive scaling factor, $\rho(q, q_{obs})$ denotes the shortest distance from robot q to obstacle, and ρ_0 is the largest impact distance of the obstacle.

Similarly, deriving repulsion field with respect to distance, the repulsion force is

$$F_{rep}(q) = U'_{rep}(q) = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{\rho(q, q_{obs})} - \dfrac{1}{\rho_0}\right)^2, & \text{if } \rho(q, q_{obs}) \le \rho_0 \\ 0, & \text{if } \rho(q, q_{obs}) > \rho_0 \end{cases} \tag{4}$$

When the attractive and repulsive forces are equal or almost equal, the potential force of the robot is zero. This causes the robot to be trapped in local minima or oscillations. However, the MEC potential field differs from the artificial potential field. The main difference is that there is no repulsive force in the MEC scenario. To find the optimal global solution, the RAAPF with an SA algorithm is proposed.

3. Problem Formulation.

3.1. **MEC field formulation.** For the MEC field, there are some commons and differences between the artificial potential field and MEC field, as shown in Table 1. Artificial field is suitable for path planning to search for the shortest path. Similarly, resource allocation means finding a proper node for a user to obtain the minimum potential energy including task size, communication distance, node residual capacity, and computational power. According to the artificial potential field Formulas (1)-(4), we know that the

TABLE 1. Comparison with artificial potential field and MEC field

	Artificial potential field	MEC field
Transmitting terminal	Negative charge	User terminal
Receiving terminal	Positive charge	Edge node
Workload	Quantity of electric charge	Task size
Transmission mode	Direct transmission	Direct transmission/ spreading the task
Impact factor	Charge amount, distance	Node capacity, computational power, task size, and communication distance
Interference	Interference between adjacent electronic fields	Signal interference between adjacent nodes

TABLE 2. List of main notations

Symbol	Meaning
I	The feature index matrix of edge node
C	The set of edge node capacity
f	The set of edge computing power
q	The set of task size
d	The set of communication distance
$\hat{q}$	The forward index of task size
$\hat{d}$	The forward index of communication distance
$\hat{I}$	The standardized feature index matrix of edge node
μ	Attactive potential energy
$x_{s,u,t}$	sth node could supply resource for uth user at t time slot
λ_u	Total task size for uth user

movement trend of an item is proportional to the nodes capacity and computing power. Because MEC prefers to deal with disassembly tasks, QoS deteriorates because of long distances, which results in an expensive cost. Obviously, the movement trend of tasks is inversely proportional to the task size and distance. The index item table is shown in Table 2.

It is assumed that there are four indexes for edge nodes in the MEC field, namely, node capacity, computational power, task size, and communication distance. In the MEC field, the larger node capacity and computing power are, the easier it is to attract tasks. The above two indexes items are called maximum indexes. The smaller size and communication distance are, the better the communication quality can be guaranteed. The above two items are called minimum indexes. The minimum index can be directly subtracted from the maximum value of the index to get the forward index. The formulas of task size and communication distance are as follows:

$$\hat{q}_i = \max\{q\} - q_i \tag{5}$$

$$\hat{d}_i = \max\{d_i\} - d_{i,j} \tag{6}$$

For n nodes in the MEC field, the matrix of four indexes is as follows:

$$I = \begin{pmatrix} I_{1,1} & I_{1,2} & I_{1,3} & I_{1,4} \\ I_{2,1} & I_{2,2} & I_{2,3} & I_{2,4} \\ \vdots & \vdots & \cdots & \vdots \\ I_{n,1} & I_{n,2} & I_{n,3} & I_{n,4} \end{pmatrix} = \begin{pmatrix} C_1 & f_1 & \hat{q}_1 & \hat{d}_1 \\ C_2 & f_2 & \hat{q}_2 & \hat{d}_2 \\ \vdots & \vdots & \cdots & \vdots \\ C_n & f_n & \hat{q}_n & \hat{d}_n \end{pmatrix} \tag{7}$$

For the ith edge node, the feature index vector is shown as follows:

$$I_i = \left(C_i, f_i, \hat{q}_i, \hat{d}_i\right) \tag{8}$$

Here, C_i denotes the present capacity of the ith node. f_i denotes the CPU computing power. $\hat{q}_i$ and $\hat{d}_i$ denote the ith task size and communication distance after forward processed, respectively. Then, in order to eliminate the influence of different dimensions of indexes, the forward matrix is standardized, and its standardization formula is as follows:

$$\hat{I}_{i,j} = \frac{I_{i,j} - \min\{I_{1,j}, I_{2,j}, \ldots, I_{n,j}\}}{\max\{I_{1,j}, I_{2,j}, \ldots, I_{n,j}\} - \min\{I_{1,j}, I_{2,j}, \ldots, I_{n,j}\}} \tag{9}$$

For the maximum solution $\hat{I}^*$, it can be expressed as follows:

$$\begin{aligned} \hat{I}^* &= \left(\hat{I}_1^*, \hat{I}_2^*, \ldots, \hat{I}_n^*\right) \\ &= \Big(\max\{c_1, c_2, \ldots, c_n\}, \max\{f_1, f_2, \ldots, f_n\}, \max\left\{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_n\right\}, \\ &\quad \max\left\{\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n\right\}\Big) \end{aligned} \tag{10}$$

For the minimum solution $\hat{I}^-$, it can be expressed as follows:

$$\begin{aligned} \hat{I}^- &= \left(\hat{I}_1^-, \hat{I}_2^-, \ldots, \hat{I}_n^-\right) \\ &= \Big(\min\{c_1, c_2, \ldots, c_n\}, \min\{f_1, f_2, \ldots, f_n\}, \min\left\{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_n\right\}, \\ &\quad \min\left\{\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n\right\}\Big) \end{aligned} \tag{11}$$

Finally, the potential energy of the ith node could be calculated as follows:

$$\mu_i = \frac{\sqrt{\sum_{j=1}^{4}\left(\widehat{I_j^*} - \widehat{I_{i,j}^*}\right)^2}}{\sqrt{\sum_{j=1}^{4}\left(\widehat{I_j^*} - \widehat{I_{i,j}^*}\right)^2} + \sqrt{\sum_{j=1}^{4}\left(\widehat{I_j^-} - \widehat{I_{i,j}^-}\right)^2}} \tag{12}$$

Here, μ_i is the attractive potential energy for the ith node. The communication quantity caused by multipath interference is ignored in the MEC field; only the attracted potential energy is considered. An edge node with a large capacity provides more cache to facilitate the next task invocation. The communication quantity corresponds directly to transmission distance. The signal is affected in the channel with long-distance, such as Gaussian noise and multipath fading, which reduces the signal-to-noise ratio and increases the bit error rate. In addition, because each edge node has limited computing resources, every node has more attractive potential energy for small tasks. The potential energy of each node to attract user requests in the MEC field is given by Formula (12). The simulation structure is shown in Figure 2.

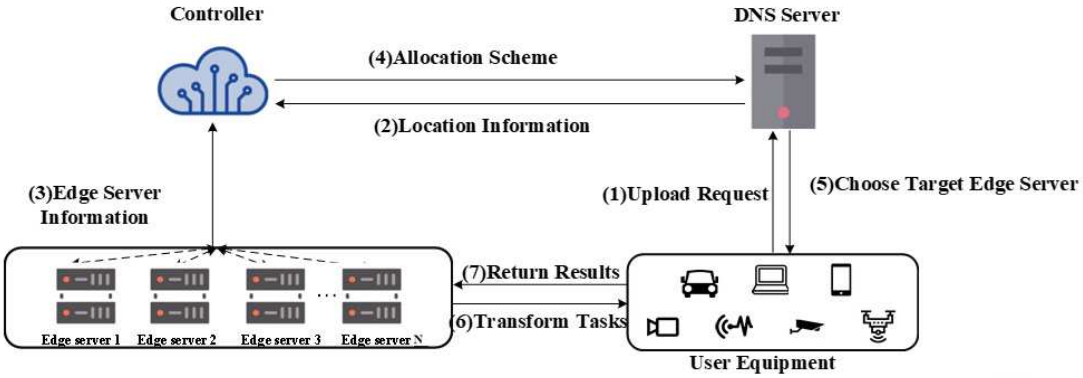


FIGURE 2. RAAPF structure

In Figure 2, a task is processed as follows:

1) User equipment (UE) uploads requests to the domain name system (DNS) server when an IoT user submits her tasks to the MEC service.

2) The requests are received by the DNS server, and the location information is sent to the controller.

3) Meanwhile, as a node set, each edge server provides information about capacity, computing power, work state, etc., to help the UE choose some feasible nodes.

4) According to the existing node information, the controller combines with the RAAPF method to select the most suitable node for the corresponding UE.

5) The DNS server chooses the target edge server for each requested UE and the communication channel between the UE and edge server.

6) UE sends tasks to the selected edge service. The edge service deals with the task of costing a certain executing time.

7) The result is received by the UE, and the users request is met with the RAAPF method.

RAAPF method combined with artificial potential field and simulated annealing measures the potential energy of each task to the node. Through standardization and forward procession, the TOPSIS method is used to calculate the maximum solution and minimum solution for many impact indicators. The problem of resource allocation is transformed into the knapsack problem. The MOERA method is a mobility-agnostic online algorithm. In order to reduce the delay of the whole communication scenario, all nodes are in a working state. Therefore, the user's task is broken down. For user tasks that cannot be decomposed, additional delays will be generated. Compared with MOERA, RAAPF method takes nodes as research objects. By scoring the four indexes of the user task, the potential energy attracted by the node to the task was calculated under the MEC field. RAAPF method can overcome the linearity deficiency of the MOERA method significantly and can find the global optimal solution and jump out of the local optimal solution. The RAAPF method is shown in Algorithm 1.

3.2. **Knapsack model setting.** The knapsack problem is a combinatorial optimization non-deterministic polynomial (NP) complete problem. As for the MEC, the potential energy could be regarded as the value of the item, whose impact factor is determined by Formula (12). The node capacity is regarded as the present weight of the knapsack. The value and weight of the items in the knapsack are dynamic, as shown in Formula (13).

$$\Delta C = \begin{cases} C_i, & \text{Choose the } i\text{th node} \\ C_i - C_{i+1}, & \text{Replace the } i\text{th node with the } (i+1)\text{th node} \\ C_{i+1} - C_i, & \text{Replace the } (i+1)\text{th node with the } i\text{th node} \end{cases} \tag{13}$$

Algorithm 1: RAAPF

01: Initialize velocities and positions of users and nodes randomly.
02: Calculate the distance $d_{i,j}$
03: For i, j do
04: For C_i, f_i, q_j, do
05: Calculate u_i for each user and node.
06: Use logarithmic function to reduce differences.
07: Find the maximum value of u corresponding to each node direction, and the rest of the u is called -1.
08: For each row, if u is not all -1, that means the optimal scheme needs to be selected for different number of users for the same node.
09: Solve the knapsack problem and select the user who could process at this time slot.
10: Calculate $T_{uploading}$, $T_{executing}$, and $T_{downloading.}$
11: End for
12: End for

Mathematical programming with constraints is shown as follows:

$$u = \max_{C_i, f_i, \hat{q}_i, \hat{d}_i} \sum_j \sum_i \mu_i x_{i,j} \tag{14}$$

$$\text{s.t.} \quad \sum_j q_j x_j \leq C_j \tag{14a}$$

$$x_{i,j} \in 0, 1 \tag{14b}$$

$$d_i \leq L_i \tag{14c}$$

Our object is to maximize u for each user connected to a selected node. We notice that the total potential energy u consists of each node potential energy and offloading factor. Constraint (14a) guarantees that the capacity constraint for each edge node is not violated. Constraint (14b) ensures that potential energy is generated only when the task is offloaded. $x_{i,j}$ is an offloading factor for each node-to-user service. $x_{i,j} = 0$ means the jth task is executed in local terminal, and $x_{i,j} = 1$ indicates the jth task is offloaded on the ith node. Constraint (14c) ensures communication distance is within the range of the edge node, and the communication distance is no more than L to ensure that the signal can cover the users. An SA algorithm is applied to solving the knapsack problem and finding the optimal offloading computing solution of the allocation resource. SA is a common optimization algorithm and an extension of the local search algorithm. However, it is different from the local search algorithm because it selects a state with a large target value in the neighborhood with a certain probability. SA has powerful global search performance because it uses many unique methods and technologies. The SA guides its search direction by changing the probability. The probability is used to guide the search process to move towards the region with a better solution. Although it seems to be a blind search method, it has a clear search direction.

The core idea of the SA is to randomly select points in the search area. Using the Metropolis criterion shown in Formula (15), we gradually converge the random moving point to the local optimal solution. The Metropolis criterion provides the opportunity to make the random moving point jump off the local optimal solution and continue to search for the global optimal solution.

$$P = \begin{cases} 0, & m + \Delta m \geq m \\ 1, & m + \Delta m \leq m \text{ and } \Delta f \geq 0 \\ \exp\left(-\dfrac{\Delta f}{t}\right), & \text{others} \end{cases} \tag{15}$$

Then, the uploading time and downloading time correspond to the distance, task size, and uploading/downloading speed. The executing time is generated by the CPU to analyze and process the tasks. The above three indexes are denoted in Formulas (16), (17), and (18), respectively.

$$T_{uploading} = \sum_s \sum_u \sum_t \frac{d_{u2ns,u,t} \frac{x_{s,u,t}}{\lambda_u}}{v_{uploading_t}} \tag{16}$$

$$T_{downloading} = \sum_s \sum_u \sum_t \frac{d_{n2us,u,t} \frac{x_{s,u,t}}{\lambda_u}}{v_{downloading_t}} \tag{17}$$

$$T_{executing} = \sum_s \sum_u \sum_t \frac{C_i q_j}{f_i} \tag{18}$$

Here, $\lambda_u = \sum_s \sum_t x_{s,u,t}$. Formula (19) denotes the total latency, which is the sum of the uploading time, executing time, and downloading time.

$$T = T_{uploading} + T_{executing} + T_{downloading} \tag{19}$$

3.3. **Time complexity.** It is convenient to check that the RAAPF algorithm can be finished in polynomial time since it only relies on solving a series of linear programming, from which we choose the simulated annealing algorithm due to its practical performance. As shown in Algorithm 1, the time complexity of simulated annealing algorithm is $O(N) * O(1) = O(N)$, where N is the total number of available edge nodes in each observation. The optimal total delay and reasonable resource allocation are achieved after obtaining the potential energy u for every iteration ($j \leq N$).

4. Evaluation.

4.1. **Experiment setups.** The simulation software is compiled by MATLAB2020a. The computer CPU is Intel(R) Core (TM) i5-9300H CPU@2.40GHz, and the RAM is 8.00GB. For each row of edge nodes in matrix μ, every value is obtained as the feasible solution to the knapsack problem. SA is applied to exploring the optimal solution for the migration of the task. Three methods, MOERA, RAAPF, and OPT, were compared with the latency under different parameter settings. MOERA decomposes the user requests into sub-tasks based on the regularization technique, which makes full use of all nodes in the communication environment [22]. Our proposed RAAPF method is inclined to the existing edge servers, and user requests information for optimizing SA latency. OPT is based on RAAPF, which eliminates the impact of RAAPF randomness and achieves the best performance.

4.2. **Experiment results.** Figure 3 shows the optimizing latency under different numbers of nodes with a free/busy status of 4G/5G. When many nodes in the model are occupied, RAAPF has few available option, similar to a greedy algorithm. As the number of nodes increases, RAAPF has a few available options to find a better solution. Therefore, latency has a tendency to decline in a short period. For the free status, this tendency is stable because each node has sufficient capacity to host and process tasks sent by users. However, for a busy status, the delay increases linearly after a short decrease. This is because there is not enough capacity on each node to deal with the current tasks, and the

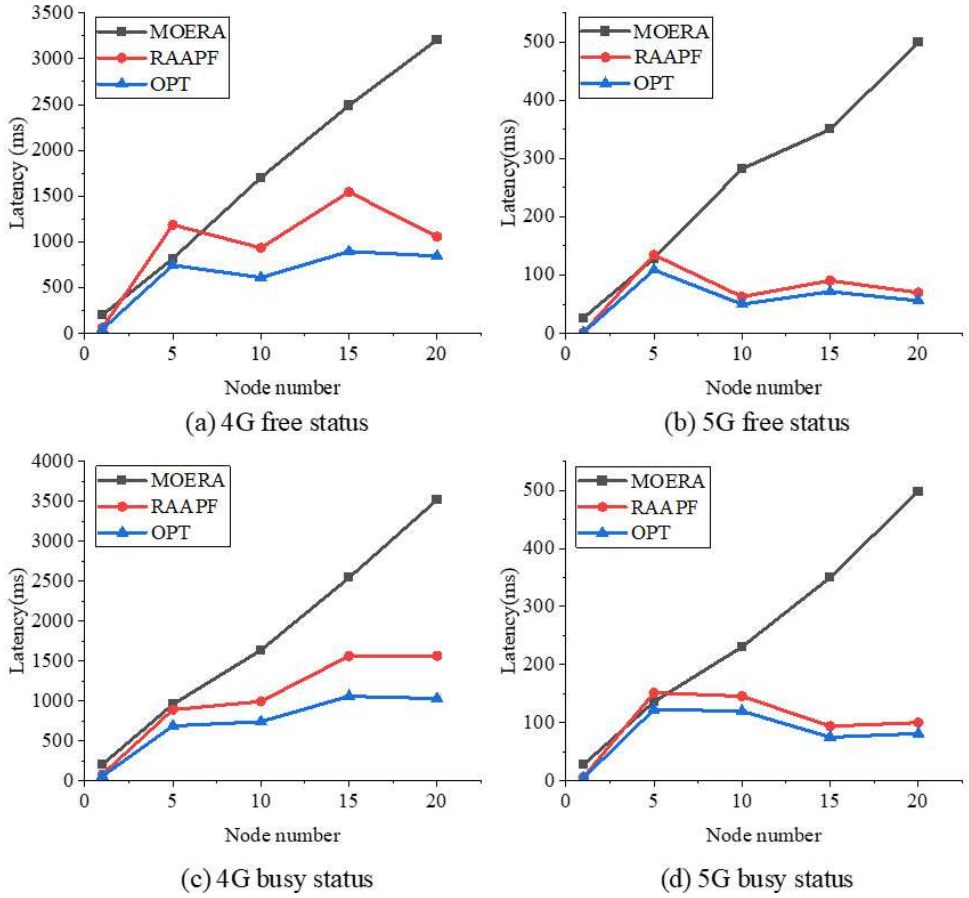


FIGURE 3. Latency comparison in different scenes

tradeoff between queuing the task and offloading task at the node farther away. The latency with the RAAPF method is reduced by 25% compared to that of mobility-agnostic online resource allocation within a few node numbers; the latency gap between the two methods increases significantly as the number of nodes increases. In detail, when the number of nodes is small, the total communication delay of RAAPF method and MOERA method is similar. When the number of nodes increases to 20, the delay gap widens significantly. In addition, latency result that the potential energy is proportional to the task size, and the delay is inversely proportional to the upstream and downstream speeds of 4G/5G. When RAAPF meets 100 users and ten edge nodes' services, the tradeoff between queuing the task at the node and offloading the task farther away is considered. Owing to the fast CPU executing speed, it is evident that 4G's upstream and downstream transmission speed is slower than those of 5G. Therefore, under the 4G network, the tasks prefer the nearest queuing solution. For 5G, waiting for the queue and migrating to other nodes lead less time delay.

There are ten nodes in the test range, and the CPU frequency of each node is set to be 1024MHz, the latency generated by the two methods with different numbers of users is shown in Figure 4. The figure reveals that RAAPF is more suitable for scenes with a large number of users. The MOERA method prefers linear segmentation of data processing and has a better performance for a small number of users. With the same number of nodes, increasing the growth of users will cause communication queuing and congestion. Because of the time gap caused by different waiting times and processing times, RAAPF has more potential than mean fragment filling in MOERA when the number of users is growing.

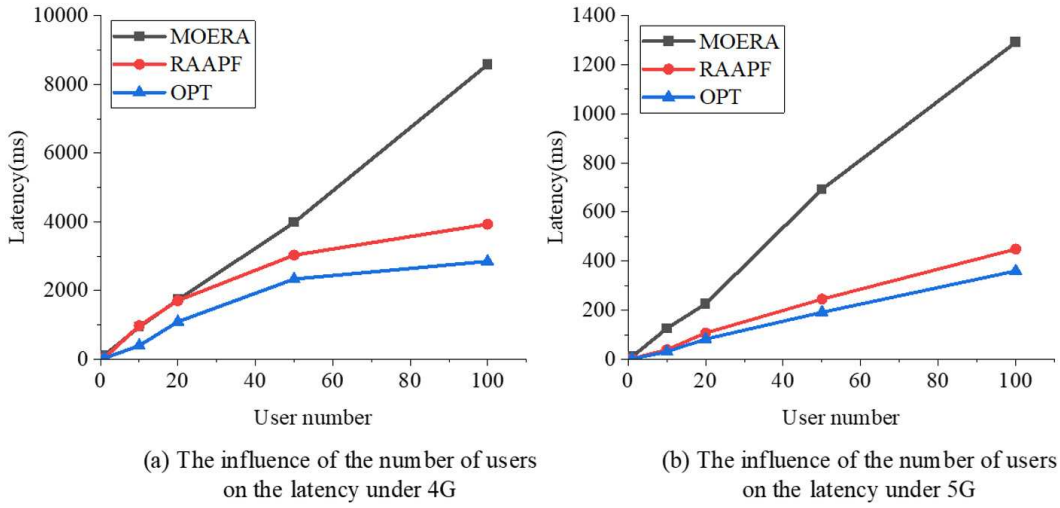


FIGURE 4. The influence between the number of users and latency under 4G/5G

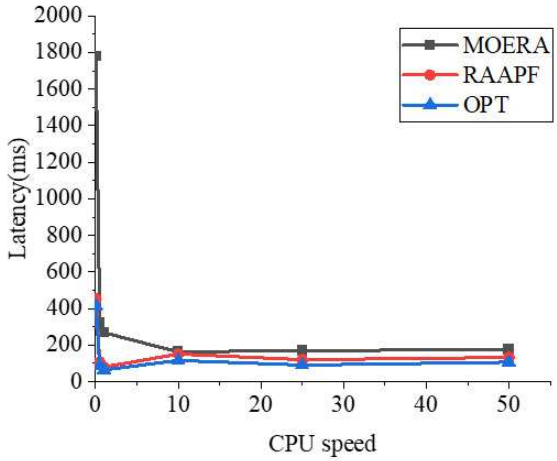


FIGURE 5. The influence between the number of CPU speed and latency under different methods

Under 4G, when the number of users is small, the delay of RAAPF method and MOERA method is very close. When the number of users increases to 100, the delay optimization of RAAPF scheme is 60.3% compared with MOERA scheme. RAAPF is already better than MOERA when the number of users is small under 5G. When the number of users increases to 100, the delay optimization of RAAPF method is 43.8% compared with MOERA method. We can get that the RAAPF method is suitable for scenarios with high network speed and heavy traffic.

The relationship between CPU speed and time delay is shown in Figure 5. The CPU speed primarily affects task executing time. With 20 users and ten nodes, different CPU speeds for each user require different times. In this figure, we can see that the lack of node computational power leads to task queuing and fails to meet the delay requirements. With continuous improvement in the CPU processing capacity, the delay decreases exponentially. This indicates that the computation capacity of the edge nodes plays a decisive role in the delay of task processing. As the CPU processing power increases, the delay tends to stabilize. This is because the available computing power can handle the tasks of users. However, the requests to the edge are generally small. Therefore, edge computing

does not need to deploy powerful computing nodes to meet users' requirements of high reliability and low delay.

Figure 6 shows the effect of different task sizes and the task distribution on time delay. In Figure 6, we can see that the task distribution has little impact on the delay. And with the request size increasing, the delay in the RAAPF method varies almost linearly. It reflected that the RAAPF is sensitive to the request sizes. It affects the weight in the knapsack problem, which may cause a direct impact on the outcome. And for MOERA method, because the total tasks are distributed to the whole edge nodes in communication scenarios, each edge node is processing tasks all the time. So the latency is convergent with a small task and strong node capacity.

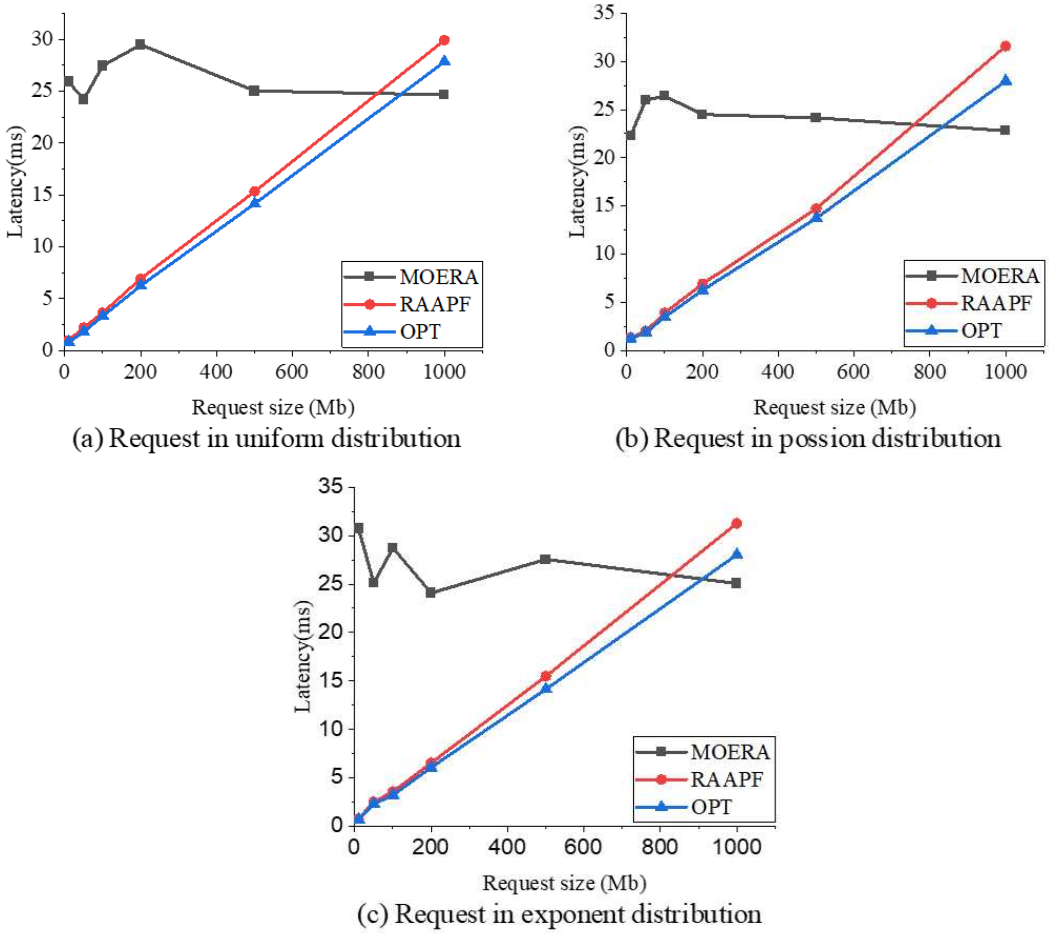


FIGURE 6. The influence between the request size and latency

5. **Discussions.** We now discuss some practical issues in implementing the proposed algorithm in real systems. What has been analyzed above, our proposed RAAPF method is suitable for scenarios with large numbers of users. RAAPF method can be applied to scenarios with large-scale nodes and users in smart airports, such as intelligent check-in and airport vehicle supervision. An edge cloud can be deployed in the scenario center and calculate the potential energy for node-to-user. According to Formula (12), the indexes, such as node capacity, computational power, communication distance, and task size, have been normalized and forward processed through the TOPSIS method. Simulation results show that the proposed RAAPF method can minimize communication delay. In addition, when the number of users increases, the minimum latency is required

compared to other resource allocation schemes. It can be seen that RAAPF method has better performance and can be well applied to airports, stations and other communication scenarios with heavy workload. Another issue is the workload distribution among the edge nodes. It is envisioned that modern mobile applications that are compatible with edge computing will be refactored based on the microservice architecture paradigm. With this microservice-based architecture, application requests can be partitioned and served by multiple instances of microservices on different edge nodes. Each instance of a microservice can be scaled separately by assigning an appropriate amount of resources to it.

6. **Conclusion.** In this paper, we proposed an RAAPF scheme that indicates how to choose and transform different nodes with the user moving randomly to achieve low latency. We formulated the relationship between min-latency and other performance indices as mathematical programming problems. Then, we solve these problems by considering two cases. 1) RAAPF provides an adaptable scheme for sensitive-latency applications in short-distance communication. 2) The QoS is guaranteed by the MEC, and the optimal latency is an alternative with an increasing number of nodes. For the first case, we constructed a formula to solve the best value of the artificial potential energy in our model. The scheme considers the effects of the significant and necessary parameters connected with well-communicative edge nodes. In the second case, the edge node scale is larger, and the signal coverage is wider. In a well-coverage signal environment, the complexity of multi-hop communication is decreasing, which results in a smooth communication network to improve the users QoS. The QoS of the 5G communication channel is more reliable than that of 4G. The numerical evaluation shows that the proposed algorithm is similar to MOERA in a busy state. Communicating with nodes as the number of nodes increases, the probability of the best choice we obtained is high; thus, we could have a low-latency service. Furthermore, the QoS and latency of 5G communication, on the short distance between the user and node, is superior to 4G.

However, the situation in the communication environment is complex and diverse. The RAAPF model needs a more reliable feature of information to further optimize the schedule of resource allocation so as to be more adaptive in order to be applied to extensive senses in the MEC field. We leave this for future exploration.

REFERENCES

[1] S. Math, L. Zhang and S. Kim, An intelligent real-time traffic control based on mobile edge computing for individual private environment, *Security and Communication Networks*, pp.1-11, DOI: 10.1155/2020/8881640, 2020.

[2] M. S. Elbamby, C. Perfecto and C. F. Liu, Wireless edge computing with latency and reliability guarantees, *Proc. of the IEEE*, vol.107, 2019.

[3] M. S. Elbamby, C. Perfecto and M. Bennis, Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord, *IEEE WCNC2018*, DOI: 10.1109/WCNC.2018.8377419, 2018.

[4] B. Yang, X. Cao and C. Yuen, Offloading optimization in edge computing for deep learning enabled target tracking by Internet-of-UAVs, *IEEE Internet of Things Journal*, 2020.

[5] X. Xu, Y. Xue, X. Li, L. Qi and S. Wan, A computation offloading method for edge computing with vehicle-to-everything, *IEEE Access*, vol.7, pp.131068-131077, DOI: 10.1109/ACCESS.2019.2940295, 2019.

[6] M. S. Rahman, N. C. Peeri, N. Shrestha, R. Zaki, U. Haque and S. H. A. Hamid, Defending against the Novel Coronavirus (COVID-19) outbreak: How can the Internet of Things (IoT) help to save the world?, *Health Policy and Technology*, vol.9, no.2, pp.136-138, 2020.

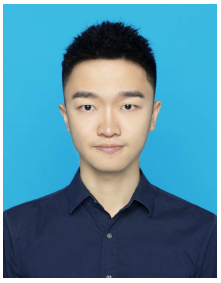
[7] P. Ligza, Static and dynamic parameters of hot-wire sensors in a wide range of filament diameters as a criterion for optimal sensor selection in measurement process, *Measurement*, p.151, 2019.

[8] B. Nasir, Z. Yan and A. Taherkordi, Mobile edge computing: A survey, *IEEE Internet of Things Journal*, 2017.
[9] S. P. Ahuja and N. Deval, From cloud computing to fog computing: Platforms for the Internet of Things (IoT), *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing*, pp.999-1010, DOI: 10.4018/978-1-7998-5339-8.ch047, 2021.
[10] K. Kim and C. S. Hong, Optimal Task-UAV-Edge matching for computation offloading in UAV assisted mobile edge computing, *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019.
[11] L. Li, M. Siew and T. Quek, Learning-based priority pricing for job offloading in mobile edge computing, *IEEE Internet of Things Journal*, 2019.
[12] Y. Siriwardhana, P. Porambage and M. Liyanage, A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications and technical aspects, *IEEE Communications Surveys and Tutorials*, DOI: 10.1109/COMST.2021.3061981, 2021.
[13] J. Ren, D. Zhang and S. He, A survey on end-edge-cloud orchestrated network computing paradigms, *ACM Computing Surveys (CSUR)*, vol.52, no.6, 2019.
[14] J. Wang, L. Zhao and J. Liu, Smart resource allocation for mobile edge computing: A deep reinforcement learning approach, *IEEE Trans. Emerging Topics in Computing*, 2019.
[15] J. Huang, S. Li and Y. Chen, Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing, *Peer-to-Peer Networking and Applications*, pp.1776-1787, 2020.
[16] V. D. Valerio and P. F. Lo, Optimal virtual machines allocation in mobile femto-cloud computing: An MDP approach, *Wireless Communications and Networking Conference Workshops*, pp.7-11, 2014.
[17] J. Liu, Y. Mao and J. Zhang, Delay-optimal computation task scheduling for mobile-edge computing systems, *IEEE International Symposium on Information Theory (ISIT)*, 2016.
[18] Z. Yang, C. Pan, K. Wang and M. Shikh-Bahaei, Energy efficient resource allocation in UAV-enabled mobile edge computing networks, *IEEE Trans. Wireless Communications*, vol.18, no.9, pp.4576-4589, DOI: 10.1109/TWC.2019.2927313, 2019.
[19] Z. Ning, P. Dong, M. Wen et al., 5G-enabled UAV-to-community offloading: Joint trajectory design and task scheduling, *IEEE Journal on Selected Areas in Communications*, vol.39, no.11, pp.3306-3320, DOI: 10.1109/JSAC.2021.3088663, 2021.
[20] A. Santoyo and C. Cervello, Edge nodes infrastructure placement parameters for 5G networks, *IEEE Conference on Standards for Communications and Networking*, 2018.
[21] P. Wang, C. Yao and Z. Zheng, Joint task assignment, transmission and computing resource allocation in multi-layer mobile edge computing systems, *Internet of Things Journal*, vol.6, no.2, 2018.
[22] L. Wang, J. Lei and J. Li, MOERA: Mobility-agnostic online resource allocation for edge computing, *IEEE Trans. Mobile Computing*, vol.18, no.8, pp.1843-1856, 2019.
[23] A. Khalili, S. Zarandi and M. Rasti, Joint resource allocation and offloading decision in mobile edge computing, *IEEE Communications Letters*, 2019.
[24] X. Ren, X. Lian, Q. Jia, T. Huang and Y. Li, Survey on computation offloading in mobile edge computing, *Journal on Communications*, vol.39, pp.138-155, 2018.
[25] K. Zhang, X. Gui, D. Ren, J. Li, J. Wu and D. Ren, Survey on computation offloading and content caching in mobile edge networks, *Journal of Software*, vol.30, pp.2491-2516, 2019.
[26] A. Khan, M. Othman, S. Madani and S. U. Khan, A survey of mobile cloud computing application models, *IEEE Communications Surveys and Tutorials*, vol.16, no.1, pp.393-413, 2014.
[27] M. Messous, H. Sedjelmaci and N. Houari, Computation offloading game for an UAV network in mobile edge computing, *International Conference on Communications*, 2017.
[28] B. Zhang, S. Guo, Y. Dong and D. Liu, Joint task offloading and data caching in mobile edge computing networks, *Computer Networks*, 2020.
[29] M. Liu, R. Yu and Y. Teng, Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing, *IEEE Trans. Wireless Communications*, vol.18, pp.695-708, 2019.
[30] F. Jia, H. Zhang and H. Ji, Distributed resource allocation and computation offloading scheme for cognitive mobile edge computing networks with NOMA, *IEEE/CIC International Conference on Communications in China (ICCC)*, pp.553-557, 2019.
[31] J.-M. Kim, G. Cho, C.-S. Ko and Y.-T. Park, Allocation of logistics sharing cost in refrigerated logistics warehouse, *ICIC Express Letters, Part B: Applications*, vol.12, no.10, pp.943-948, 2021.
[32] F. Zhang, J. Ge and C. Wong, Online learning offloading framework for heterogeneous mobile edge computing system, *Journal of Parallel and Distributed Computing*, 2019.

[33] S. Wang, M. Chen and X. Liu, A machine learning approach for task and resource allocation in mobile edge computing based networks, *IEEE Internet of Things Journal*, 2020.
[34] V. Shanmuganthan, M. Khari and N. Dey, Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT, *Computer Communications*, vol.151, pp.355-364, 2020.
[35] Y. Li and S. Wang, An energy-aware edge server placement algorithm in mobile edge computing, *2018 IEEE International Conference on Edge Computing (EDGE)*, pp.66-73, 2018.
[36] O. Munoz, A. Pascual and J. Vidal, Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading, *IEEE Trans. Vehicular Technology*, vol.64, no.10, pp.4738-4755, 2015.
[37] Y. Hao, M. Chen, L. Hu, M. S. Hossain and A. Ghoneim, Energy efficient task caching and offloading for mobile edge computing, *IEEE Access*, vol.6, pp.11365-11373, 2018.
[38] S. Wang, X. Zhang and Y. Zhang, A survey on mobile edge networks: Convergence of computing, caching and communications, *IEEE Access*, 2017.
[39] M. Lavorato, M. J. Rider and A. V. Garcia, A constructive heuristic algorithm for distribution system planning, *IEEE Trans. Power Systems*, vol.25, no.3, pp.1734-1742, 2010.
[40] S. Ahmed, M. Z. Chowdhury and Y. M. Jang, Energy-efficient UAV-to-user scheduling to maximize throughput in wireless networks, *IEEE Access*, vol.8, pp.21215-21225, DOI: 10.1109/ACCESS.2020.2969357, 2020.
[41] F. Uyanik, *A Study on Artificial Potential Fields*, Ph.D. Thesis, Middle East Technical University, Ankara, Turkey, 2011.

Author Biography

Jianhua Liu received the Ph.D. degree from Beihang University, Beijing, China, in 2013. He is currently an associate professor with the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China, Guanghan, China. His research interest includes information security, Internet of Things, and edge computing.



Zibo Wu received the B.S. degree in communication engineering from Tianjin University of Technology, China in 2019. He is currently pursuing the Master's degree with the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China. He is conducting researches in the fields of mobile edge computing and cloud computing in wireless communications.

Jiaqing Shen received the Master's degree from Northwestern Polytechnical University (NPU), Xi'an, China, in 2003. He is currently an associate professor with the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China, Guanghan, China. His research interest includes communication of IoT and aeronautical communication.

Jiajia Liu graduated from Sichuan University with a Master's degree in communication and information system in 2011. She is now an associate professor at Civil Aviation Flight University of China. Her research interests include edge computing and image processing.

Xiaoguang Tu is a Ph.D. candidate with the School of Communication and Information Engineering at University of Electronic Science and Technology of China (UESTC). His research interests include convex optimization, computer vision, and deep learning.