

Research Article

An UAV-Assisted Edge Computing Resource Allocation Strategy for 5G Communication in IoT Environment

Hao Liu 

Zhengzhou Health Vocational College, Xingyang, Henan 450100, China

Correspondence should be addressed to Hao Liu; liuhao@zzhvc.edu.cn

Received 29 January 2022; Accepted 1 March 2022; Published 31 March 2022

Academic Editor: Shan Zhong

Copyright © 2022 Hao Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the computing capacity of existing mobile devices cannot fully meet the needs of users for communication quality, a computing resource allocation strategy for 5G communication in the Internet of Things (IoT) environment is proposed by applying UAV-assisted edge computing. First, a system model is constructed with the UAV deployed with mobile edge computing (MEC) servers to provide assisted computing services for multiple users on the ground. Based on the optimization of the UAV trajectory, communication scheduling, and the energy consumption model of the UAV, the problem of the total computational cost minimization is formulated. Then, the genetic algorithm is improved by introducing a penalty function to solve this problem, in which selection, crossover, and mutation operations are iterated to obtain the optimal allocation strategy for computational resources. Finally, a simulation platform is constructed to analyze the proposed method. The results show that the total cost and total time of the proposed strategy are better than other comparison strategies.

1. Introduction

With the increasing demands in applications and the rapid development of computing capacity, innovative applications are emerging quickly, making it difficult for the traditional 4G network architecture to serve exponential growth of data flows, diverse terminal devices, and various kinds of complex service scenarios [1]. The context of the Internet of Everything also places higher demands on wireless communications in terms of mobile performance, security, network delay, and communication reliability. At the same time, people have higher requirements for computational performance as artificial intelligence is developing quickly. However, the computing capacity of existing mobile devices cannot fully meet users' demands for communication quality [2, 3]. Therefore, mobile edge computing (MEC) technology is proposed to improve the system computing capacity so as to meet the users' demands for communication. In addition, 5G has become a key technology to achieve the best performance of MEC systems in the Internet of Things (IoT) environment, which can effectively solve the problems of delay, lack of storage, and poor computing

capacity in wireless communication networks by combining reasonable resource optimization strategies for computing, storage, and communication [4]. It can also give full play to the characteristics of MEC networks, such as low response delay in computation task offloading and good network scalability [5].

By introducing UAVs into the MEC system, the flexible deployment of UAVs allows users to jointly collaborate on computing and communication, which improves the performance of the MEC system [6, 7]. However, the relationship between the trajectory control strategy and the capacity and energy consumption of traditional communication systems is unclear due to the fast mobility of UAVs. It is necessary to combine the classical air-to-ground line-of-sight channel model with a comprehensive multiobjective strategy optimization to improve the effectiveness of the algorithm based on the mobility of UAV during the communication [8].

There have been some research studies on traditional computation task offloading and resource allocation optimization at home and abroad. In Ref. [9], an exact linear search-based algorithm is proposed for finding the optimal

solution by fixing the status update rate. As a design alternative, a low complexity concave-convex procedure (CCP) is also formulated for finding a near-optimal solution relying on the original problem's transformation into a form represented by the difference of two convex problems. In Ref. [10], the optimization problem of computational resource allocation was transformed into the optimal matching for weighted bipartite graphs, which was solved by adding fake containers, thus obtaining the optimal strategy for resource matching of tasks in edge servers. But further improvement should be considered on the quality of service. Reference [11] proposed a dynamic resource allocation algorithm in the cloud edge environment and obtained the optimal scheme of resource scheduling based on tabu search algorithm, but the multiobjective optimization effect is not good. Reference [12] designed a multiobjective resource allocation method, in which Pareto archiving evolution strategy is used to optimize the time cost of IoT application, load balance, and energy consumption of MEC server, so as to find the optimal resource allocation scheme, but there is still room for further optimization for calculating delay. In Ref. [13], an enhanced resource allocation strategy in MEC using the reinforcement learning-based MOACO algorithm was proposed, in which MEC is used to predict the factors affecting delay and evaluate the performance of adjacent user equipment. A cognitive agent model was established to evaluate the validity of resource allocation. Zhang et al. proposed a MEC resource offloading method based on the UAV architecture and alternatively solved the problem of computational resource energy consumption and communication delay combined with an iterative algorithm, which improves the efficiency of optimal resource management [14]. However, the performance of overall optimization for multiple UAVs still needs to be improved. An iterative optimization algorithm with a double-loop structure for the optimal configuration of multi-UAV-assisted MEC systems was proposed by Zhang et al. [15]. Based on partial computing offload mode, jointly optimize user association and CPU cycle frequency for resource scheduling. In Ref. [16], a strategy for formulating and solving computational offloading problems using a model-free reinforcement learning framework was studied. And select an optimal overhead-aware computational offloading action based on its state, enabling the optimization of computational offloading policies in wireless cellular networks in MEC. However, the communication delay is still large due to the inadequacy of communication policies.

Based on the aforementioned analysis, an UAV-assisted edge computing resource allocation strategy for 5G communications in the IoT environment is proposed to address problems of energy consumption and delay in existing MEC networks. The main contributions are summarized as follows:

- (1) In this article, a new mobile edge computing system is studied. In this system, the ground base station is used to serve the UAV for computing diversion, and some computing tasks are unloaded to the ground base station. Then, based on the energy consumption

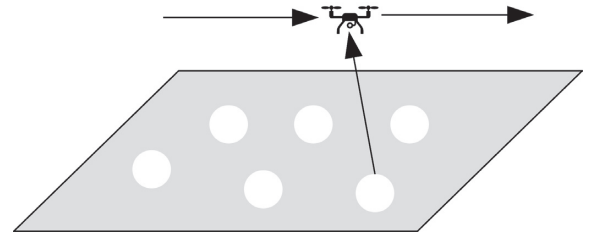


FIGURE 1: System model.

model of the rotor UAV, an optimization problem is formulated to minimize the total UAV energy consumption, At the same time, it meets the requirements of single target calculation and unloading of multiple ground base stations.

- (2) In order to reduce the total system cost, the proposed method applies an improved genetic algorithm to solve the problem, in which the penalty function and combinatorial coding are used to improve the global search capability so as to obtain the optimal resource allocation scheme while improving the allocation efficiency.

2. System Modeling

2.1. System Model. The scenario studied is the uplink communication in an UAV-assisted MEC system, where users on the ground perform computing tasks with the help of UAVs, as shown in Figure 1. In the system, there is a single-antenna rotary-wing UAV with a MEC server and N number of users on the ground, which can be denoted as $n \in \{1, 2, \dots, N\}$. Considering the three-dimensional Cartesian coordinate system, the position of the n th ground user can be denoted as $l_n = (x_n, y_n)$. In addition, as the UAV flies in a defined region and provides assisted computing service to the ground users, the initial and final positions of the UAV can be denoted as $U_I = (x_I, y_I)$ and $U_F = (x_F, y_F)$, respectively.

Assume that each ground user offloads a portion of the computational task to the UAV and the remainder is executed locally due to the lack of computing capacity. It is also assumed that the total execution time T can be divided equally into K timeslots, where $T = K\delta$; here, δ refers to the length of the timeslot which is small enough to ensure that the position of the UAV does not change within each timeslot and $k = \{1, 2, \dots, i \dots, K\}$ denotes that there are K number of timeslots. In the i th timeslot, the position of the UAV can be represented as $(x[i], y[i])$ and the maximum flight speed of the UAV is v_{\max} , and then, the trajectory of the UAV can be constrained as

$$\begin{aligned}
 U[0] &= U_I, \\
 U[K] &= U_F, \\
 \frac{\|U[i] - U[i-1]\|}{\delta} &\leq v_{\max},
 \end{aligned} \tag{1}$$

where the initial and final positions of the UAV during its flight are given and the flight speed of the UAV must not exceed the specified maximum flight speed. In addition, the energy consumption of the UAV during the flight in the i th timeslot can be calculated as

$$E^{\text{fly}}[i] = \alpha \|v[i]\|^2, \quad i \in k, \quad (2)$$

where $v[i] = [\|U[i+1] - U[i]\|/\delta]$ represents the flight speed of the UAV in the i th timeslot. $\alpha = 0.5J\delta$ and J is a constant related to the UAV payload.

2.2. Channel Model. Because the wireless channel between N ground users and UAV depends on the line-of-sight (LoS) channel. Therefore, the channel gain between the UAV and user n in the i th timeslot can be written as

$$h_n(i) = \frac{h_0}{d_n^2(i)} = \frac{h_0}{\sqrt{H^2 + \|U(i) - l_n\|}}, \quad (3)$$

where h_0 is the channel gain at a distance of 1 m, $d_n[i]$ is the distance between the user n and the UAV in the i th timeslot, and H denotes the fixed flight altitude of the UAV.

Additionally, assume that the transmitting power of the UAV is P_U and the offloading rate of the UAV data can be calculated as

$$\begin{aligned} R_n(i) &= B \log_2 \left(1 + \frac{P_U h_n(i)}{\sigma^2} \right) \\ &= B \log_2 \left(1 + \frac{\beta_0}{H^2 + \|U(i) - l_n\|} \right), \end{aligned} \quad (4)$$

where B represents the system bandwidth, σ^2 is the additive Gaussian white noise power, and β_0 represents the received signal to noise ratio (SNR) at a distance of 1 m.

2.3. Energy Consumption Model. From the above analysis, the energy consumption of the UAV consists of three main parts: the propulsion energy, computational energy, and communication energy. It is described in detail in the following section.

2.3.1. Propulsion Energy Consumption Model for UAV. The unit energy consumption (J/m) for the UAV during its flight at a speed of v can be calculated as

$$\begin{aligned} E_0(v) \triangleq \frac{P(v)}{v} &= P_0 \left(\frac{1}{v} + \frac{3v}{V_{tip}^2} \right) \\ &+ P_u \left(\sqrt{v^{-4} + \frac{1}{4V_0^4}} - \frac{1}{2V_0^4} \right)^{1/2} \\ &+ \frac{1}{2} \tau_0 \rho H A v^2, \end{aligned} \quad (5)$$

where V_{tip}^2 denotes the tip speed of the rotor blades and V_0 denotes the average rotor-induced hover speed. ρ is the air density. A denotes the area of rotor disc. τ_0 denotes the body drag ratio. P_0 and P_u denote the initial and process values of hovering power, respectively.

For any given UAV trajectory $\{U(i)\}$, the propulsion energy consumption can be expressed as

$$E_{fly}(T, \{U(i)\}) = \int_0^T P(\|v(i)\|) di, \quad (6)$$

where $v(i) \triangleq U(i)$ represents the speed of the UAV and $\|v(i)\|$ is the speed of the UAV at the moment i .

2.3.2. Communication Energy Consumption Model of UAV When Offloading Tasks. The time-division multiple access approach is considered here. Specifically, the UAV can establish contact with at most one ground user at any moment, and $\varphi_n(i) = 1$ indicates that the ground user n has established contact with the UAV. Otherwise, $\varphi_n(i) = 0$. $\{\varphi_n(i)\}$ is the user scheduling function, which should meet the constraints as

$$\sum_{n=1}^N \varphi_n(i) \leq 1, \quad \forall i \in [0, T]. \quad (7)$$

The data volume \bar{D}_n of the subtask offloaded from ground user n to the UAV is a function of the variable T , $U(i)$, and $\varphi_n(i)$, which can be expressed as

$$\begin{aligned} \bar{D}_n(T, \{U(i)\}, \{\varphi_n(i)\}) &= \int_0^T \varphi_n(i) R_n(i) di \\ &= B \int_0^T \varphi_n(i) \log_2 \left(1 + \frac{\beta_0}{H^2 + \|U(i) - l_n\|} \right) di. \end{aligned} \quad (8)$$

Hence, when a task packet with data size $(1 - \omega_n)D_n$ is offloaded to the ground base station at the moment i , we can get

$$\bar{D}_n(T, \{U(i)\}, \{\varphi_n(i)\}) \geq (1 - \omega_n)D_n, \quad \forall n \in N, \quad (9)$$

where D_n denotes users' total computation volume and ω_n denotes the bit allocation coefficient of the system.

Then, the communication energy consumption of the UAV when offloading tasks can then be obtained as

$$E_U = P_U \int_0^T \left(\sum_{n=1}^N \varphi_n(i) \right) di. \quad (10)$$

2.3.3. Computational Energy Consumption of UAV. The energy consumed in local execution is directly determined by the CPU workload F , which can be formulated as

$$F = D\zeta, \quad (11)$$

where D represents the total data volume of system tasks. ζ denotes the complexity of the application which obeys a Gamma distribution. In order to minimize CPU energy consumption, it is necessary to dynamically adjust the clock frequency in each CPU cycle to achieve the best working state. As a result, the computational energy consumption required to execute the computation task whose data size is $\bar{\omega}_n D_n$ within time T can be calculated as

$$E_{comp} = \frac{G(\bar{\omega}_n D_n)^3}{T^2}, \quad (12)$$

where the coefficient $G = 10^{-11}$ is a constant that takes the effective switching capacity and the probability of completion of the application execution into account.

To sum up, the total energy consumption of UAV is composed of propulsion energy consumption, calculated energy consumption, and communication energy consumption, which is calculated as follows:

$$E_{total} = E_{fly}(T, \{U(i)\}) + P_U \int_0^T \left(\sum_{n=1}^N \varphi_n(i) \right) + \sum_{n=1}^N \frac{G(\bar{\omega}_n D_n)^3}{T^2}. \quad (13)$$

2.4. Optimization Objectives. The optimization objective is to minimize the energy consumption of the UAV, which needs to take the user scheduling function $\varphi_n(i)$, the bit allocation coefficient of the system $\bar{\omega}_n$ and the UAV trajectory $\{U(i)\}$ into consideration. Then, the problem can be formulated as

$$\begin{aligned} & \min_{T, \{U(i)\}, \{\varphi_n(i)\}, \bar{\omega}_n} E_{total} \\ \text{s.t. } & \bar{D}_n(T, \{U(i)\}, \{\varphi_n(i)\}) \geq (1 - \bar{\omega}_n) D_n, \quad \forall n \in N, \\ & \|\dot{U}(i)\| \leq v_{max}, \quad \forall i \in [0, T], \\ & U(0) = U_I, \\ & U(T) = U_F, \\ & \varphi_n(i) \in \{0, 1\}, \quad \forall n \in N, i \in [0, T], \\ & 0 \leq \bar{\omega}_n \leq 1. \end{aligned} \quad (14)$$

It should be noted that the problem requires to jointly optimize the UAV trajectory $U(i)$, the communication scheduling $\varphi_n(i)$, and the bit allocation coefficient $\bar{\omega}_n$. And the trajectory $U(i)$ and the communication schedule $\varphi_n(i)$ are both continuous functions with respect to time i . Furthermore, the problem constraints are nonconvex and contain binary constraints, making it difficult to obtain optimal results.

3. Computational Allocation Strategy Based on an Improved Genetic Algorithm

Minimizing energy consumption is a nonlinear 0-1 programming problem, so the binary coded genetic algorithm is selected to solve the problem. Genetic algorithm is an efficient global search algorithm, which has certain advantages in solving complex problems such as nonlinear, multi-peaked, large space, and globalization. It is famous for simulating the law of "superiority and inferiority" in the evolution of organisms in nature. The main idea is to encode the solution of the objective optimization problem into chromosomes and then simulate the genetic evolution of the chromosomes to gradually approach the optimal solution.

3.1. Encoding and Initial Populations. When the genetic algorithm is applied to globally find the optimal offloading strategy of a task, each offloading strategy needs to be encoded into a corresponding chromosome to facilitate later genetic operations such as crossover and mutation [17]. In fact, encoding is the process of converting the feasible solutions of the actual problem from its solution space to the search space of the genetic algorithm. And choosing the appropriate encoding method is the fundamental task in solving the genetic algorithm.

There are a variety of encoding methods. Since in the energy consumption minimization model for task offloading, the scheduling decision variable φ_n is a binary vector, binary encoding is most suitable and conforms to the encoding principle of minimum symbol set. The population size is set to be Q and task offloading decisions are randomly generate, which should be encoded based on the binary chromosome encoding method. Then, the initial chromosome population $S(0)$ can be obtained.

Since an offloading decision variable φ_n contains n number of decisions and it is known that those subtasks that can only be executed locally do not require a decision, they always have 0 decision, that is, $\varphi_n = 0$. Suppose that a mobile application contains n number of subtasks, where ϑ_0 number of tasks must be executed locally and ϑ_1 number of tasks can be offloaded, and $\vartheta_0 + \vartheta_1 = n$. Therefore, it is sufficient to make offloading decisions for the ϑ_1 number of subtasks. Namely, a chromosome can be constructed by using ϑ_1 number of binary bits (genes) in the genetic algorithm, and after arriving at the optimal solution, the completed offloading decision φ_n can be formed by combining them with another ϑ_0 number of binary bits whose values are 0. An example of the combined encoding solution is shown in Figure 2.

As shown in Figure 2, the mobile application is divided into 10 subtasks, with 3 locally executed tasks and 7 offloading tasks. Only the decision variables of 7 offloading subtasks need to be encoded and solved. When the optimal solution is achieved, it is combined with the decision variables of the other 3 local subtasks to arrive at the final offloading decision.

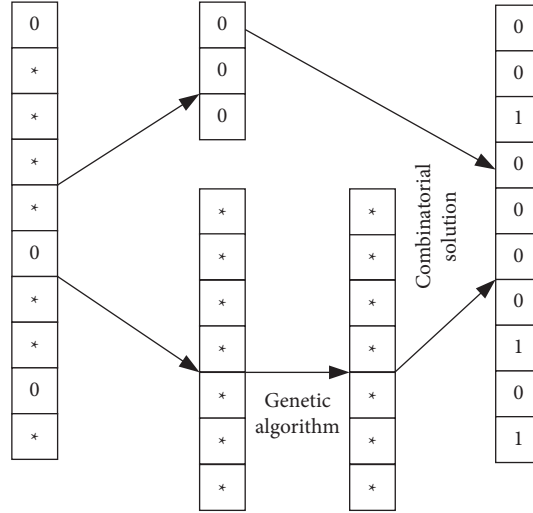


FIGURE 2: Combined encoding process.

3.2. Fitness Function. The optimization problem with constraints can be converted to an unconstrained problem by introducing a penalty function [18, 19]. The penalty function can combine the constraints with the objective function by introducing a multiplicative factor, and then, the optimization problem can be converted, while the optimal solution to the original optimization problem remains unchanged. The specific process can be written as

$$\begin{aligned} \phi = & E_{fly}(T, \{U(i)\}) + P_U \int_0^T \left(\sum_{n=1}^N \varphi_n(i) \right) + \sum_{n=1}^N \frac{G(\bar{\omega}_n D_n)^3}{T^2} \\ & + \sum_{i=1}^T \gamma_1 \times \bar{D}_n(T, \{U(i)\}, \{\varphi_n(i)\}) \\ & + \gamma_2 \times \|\dot{U}(i)\| + \gamma_3 \times \varphi_n(i), \end{aligned} \quad (15)$$

where γ_1 , γ_2 , and γ_3 are penalty factors, which are the weights of the inequality constraint functions in the optimization problem. The fitness function of the genetic algorithm can be combined with the original optimization problem based on the method of the penalty function. Since the proposed optimization problem is to minimize the total energy consumption of the task allocation under the constraints, the fitness function here will be designed to be inversely proportional to equation (15), which can be calculated as

$$\text{Fitness} = \frac{1}{\phi}. \quad (16)$$

3.3. Genetic Operations

3.3.1. Selection. The main function of selection is to simulate the operation of survival of the fittest in nature, select the chromosomes corresponding to some individuals with the best adaptation to produce the next generation, and eliminate the parents with poor adaptability [20, 21]. In this

process, the number of individuals in parent and offspring populations should be the same.

The most common strategy used in selection is the “proportional selection” strategy. For example, the probability of selecting the b th element with priority a in a j th population can be expressed as

$$P_j^{a,b} = \frac{\text{Fitness}_j^{a,b}(x_b)}{\sum_{b=1}^n \text{Fitness}_j^{a,b}(x_b)}, \quad x_b = \{0, 1\}. \quad (17)$$

Proportional selection is usually achieved using the Roulette Wheel Selection (RWS) algorithm, which yields probabilities $p_j^{a,b}$ that are always distributed in the interval $[0,1]$ and satisfy

$$\sum_{b=1}^n p_j^{a,b} = 1. \quad (18)$$

Thus, for a population with a number of priority tasks in each generation, assuming that the total number of a th priority tasks is W and the RWS algorithm first generates W random numbers in the interval $[0,1]$, $[0, 1]$ and divides the wheel into W parts according to the individual probabilities $p_j^{a,b}$. For each random number, if it is within $[\sum_{r=1}^{b-1} p_j^{a,r}, \sum_{r=1}^b p_j^{a,r}]$, the individual corresponded to the task will be selected to generate the next population with W number of individuals. Thus, the RWS algorithm can be regarded as a scheme that decides whether to select an individual based on the probability corresponding to its fitness value. Therefore, this selection method has no safeguard mechanism and may allow some well-adapted individuals to be lost in genetics. According to the proposed model, elitist strategy selection is used to ensure that the better-adapted individuals will not be lost in the next generation. The elitist strategy selection is an improvement of the basic genetic algorithm. In order to prevent the optimal solutions generated during evolution from being destroyed by crossover and mutation operations, the optimal one in each generation will be copied directly into the next generation [22, 23]. The genetic algorithm with the

introduction of elitist strategy tends to have faster convergence and better stability [24]. Since the execution time of tasks with higher priority affects the waiting time for the subsequent tasks with lower priority, the fitness value and the maximum time for the execution of tasks with the same priority should be considered simultaneously when performing elitist strategy selection.

3.3.2. *Crossover and Mutation.* Individuals are selected for the crossover and mutation operations based on the probabilities of crossover and mutation. The crossover operation for two 6×4 matrices is shown below, with the positions of the lines chosen at random:

$$\begin{array}{l} \text{Before crossover:} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\ \\ \text{After crossover:} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

3.4. *Genetic Update.* The new population is formed by combining the population obtained through the above operations with the initial population, and the energy consumption and the corresponding fitness value of each individual in the new population need to be calculated [25, 26]. The minimum value ϕ is taken as the result of this iteration, and the individual with the greatest fitness value is selected to update the population, that is, $i = i + 1$.

4. Experiments and Analysis

In the simulation experiments, numerical results will be provided to validate the proposed algorithm. The altitude of the UAV is set to $H = 100$ m, and the total communication bandwidth is $B = 3$ MHz. The SNR is $h_0 = 60$ dB, and the communication power consumption of the UAV is fixed at $P_U = 75$ W. The maximum flight speed is $v_{\max} = 60$ m/s, and the initial and final positions of the UAV are set to $U_I = [0, 0]^T$ and $U_F = [1200, 1200]^T$. The specific simulation parameter settings for the scenario when the number of users is $N = 50$ are shown in Table 1.

The parameters of the genetic algorithm are set as follows. The number of populations is 50. The maximum number of iterations is 500. The crossover probability is 0.7, and the initial mutation probability is 0.01. The penalty factors γ_1 , γ_2 , and γ_3 are set to be 107.

4.1. *UAV Trajectory.* The UAV trajectory based on the proposed method under different task execution time T is shown in Figure 3, where the red circle indicates the positions of the ground users.

TABLE 1: Simulation parameter setting.

Simulation parameters	Value
Flight altitude H	100 m
Channel bandwidth B	3 MHz
SNR received at reference distance 1 m h_0	60 dB
Communication power of UAV P_U	75 W
Maximum flight speed v_{\max}	60 m/s
Initial position U_I	$[0, 0]$
Final position U_F	$[1200, 1200]$
Number of users N	50

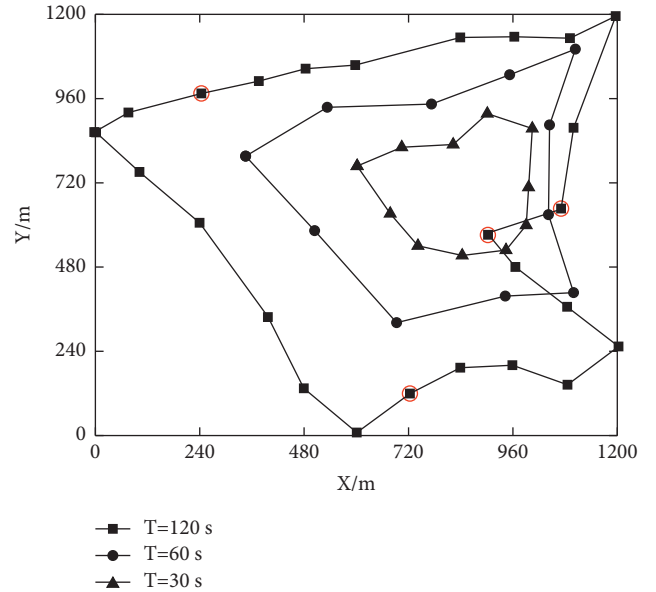


FIGURE 3: The map of UAV trajectory.

As can be seen in Figure 3, when $T = 30$ s, the UAV flies along a closed-loop trajectory and has a small flight range. The UAV is farther away from the user when offloading data and the communication quality is poor. As T increases, that is, $T = 60$ s and $T = 120$ s, the distance between the UAV and users decreases and the flight range expands, improving the communication quality. When T is large enough, the UAV continues to fly along a closed-loop trajectory and serves one user for a certain number of timeslots. Thus, it can be concluded that the larger T leads to the greater flight range of the UAV, making it closer to the users, and then brings better communication quality.

4.2. *Comparison with Other Algorithms.* To demonstrate the performance of the proposed method, it is analyzed in comparison with algorithms proposed in Ref. [10], Reference [12], and Ref. [16].

4.2.1. *Effect of the Number of Tasks on Total Cost.* When the weight of time in the total cost is 0.5, there are 10 levels of task priorities, and the effect of changes in the maximum number of tasks at each level on the total cost is shown in Figure 4. The horizontal coordinate indicates the maximum

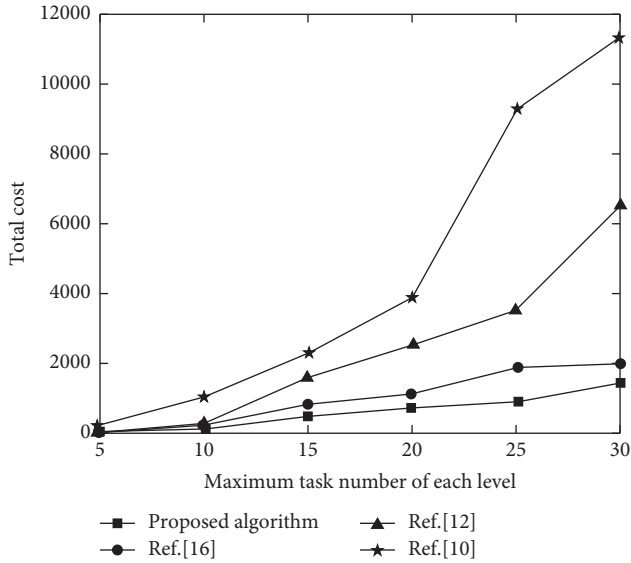


FIGURE 4: Effect of the number of tasks on total cost.

number of tasks at each level. In the experiment, for each level, a random number from 1 to the maximum is randomly generated and the tasks are numbered, and then, the dependencies on the tasks at the previous level are randomly generated according to the mutation of different tasks. The vertical coordinate shows the total cost corresponding to the allocation results obtained in different algorithms.

It can be seen in Figure 4 that the total cost grows gradually as the number of tasks at each level increases. Meanwhile, the lowest total cost is obtained in the proposed method as the UAV is used for computational task offloading and the improved genetic algorithm is introduced for problem-solving. When the task size is 30, the total cost reaches 1800. Both Ref. [10] and Ref. [12] use a traditional optimization algorithm, and the total cost grows most rapidly with the increase of tasks in each level. Reference [16] realizes the resource allocation based on a reinforcement learning network. However, it lacks a reasonable offloading target, so the total computational cost is higher than the proposed method. In addition, when there are more tasks per level, greater reduction in the total cost of executing all tasks can be achieved using the proposed method and the algorithm proposed in Ref. [16], while the proposed method has a more significant advantage.

4.2.2. Effect of the Number of Tasks on Total Time. When there are 10 levels in task priority, the effect of the change in the number of tasks on the total time to execute the same task is illustrated in Figure 5. The horizontal coordinate indicates the maximum number of tasks at each level, and the vertical coordinate indicates the total time to execute all tasks.

As depicted in Figure 5, the total computation time is the shortest due to the simplicity of the algorithm proposed in Ref. [10], which creates a weighted complete bipartite graph by adding false containers to achieve the optimal matching

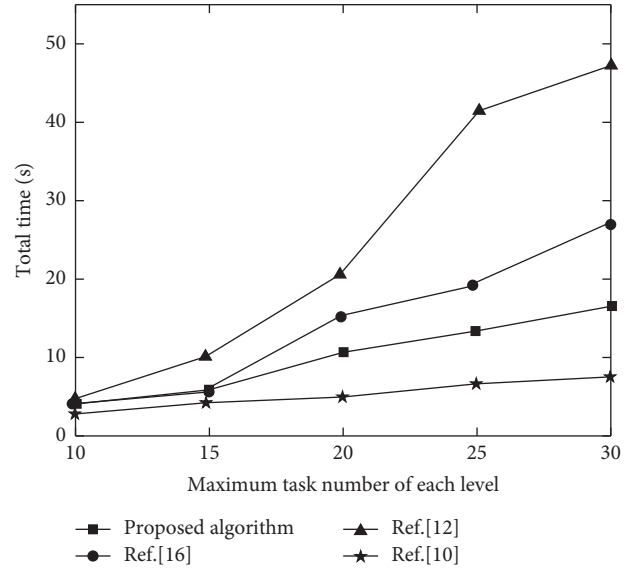


FIGURE 5: Effect of the number of tasks on total time.

of task resources in the edge server. Reference [12] uses a Pareto archiving evolutionary strategy to optimize time cost, load balancing of MEC servers, and energy consumption. The overall time consumed rises rapidly with the increase of tasks due to too many optimization targets and the slow search speed of the optimization algorithm. The reinforcement learning network in Ref. [16] is more complex, and its search time is longer compared to the proposed method, which uses an improved genetic algorithm. And the proposed method introduces the UAV to dynamically adjust the offloading strategies and reduce the data transmission time; thus, its total time cost is 17 s when the task size is 30, which is about 39% less than Ref. [16].

4.2.3. Effect of the Number of Mobile Terminals on Total Cost. The effect of the change in the number of mobile terminals on the total system cost is shown in Figure 6.

As illustrated in Figure 6, when the number of mobile terminals is less than 30, the computing resources are less and the MEC server resources are relatively sufficient. Therefore, most tasks are unloaded to the MEC server for execution, and the overhead obtained by collaborative scheduling of different algorithms is not different. With the increase of mobile terminals, resources in the MEC server gradually decrease as the collaborative scheduling uses the computational resources of the mobile terminals. The total cost of Ref. [10] and Ref. [12] grows rapidly to over 4×10^4 due to their poor performance in finding the optimal solution. The proposed method allocates resources based on an improved genetic algorithm and the UAV platform, and its system cost is lower than that of Ref. [16] which applies the reinforcement learning network. Specifically, the cost of the proposed method is only half of that of Ref. [16] when there are 100 mobile terminals in the system, demonstrating that the resource allocation strategy in the proposed method is ideal.

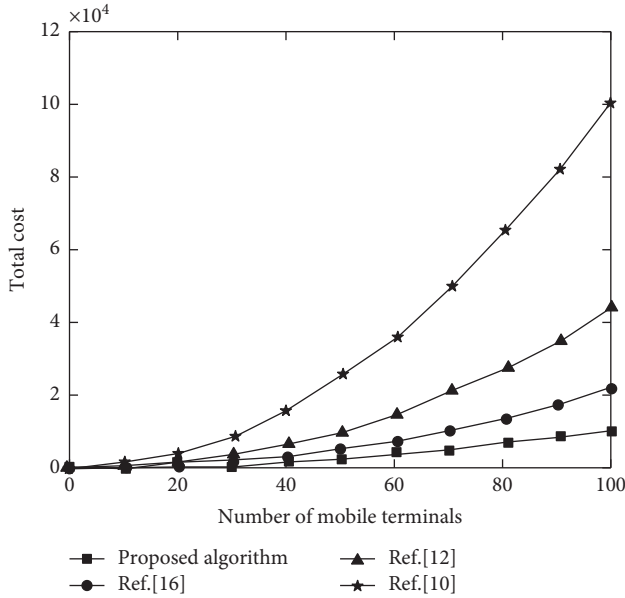


FIGURE 6: Effect of the number of mobile terminals on the total cost.

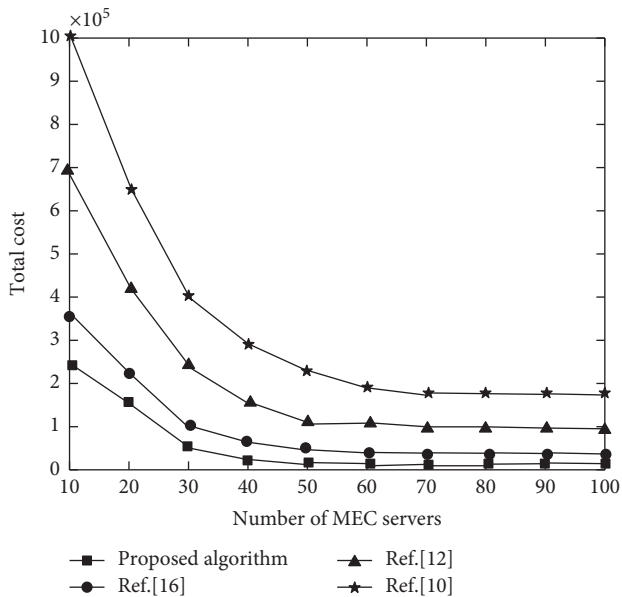


FIGURE 7: The effect of the number of MEC servers on the total cost.

4.2.4. Effect of the Number of MEC Servers on Total Cost. The effect of the change in the number of MEC servers on the total system cost is shown in Figure 7.

It is depicted in Figure 7 that the queuing time at the MEC server side decreases as the total computational cost gradually decreases. The proposed method has the smallest total computational cost, which is less than 0.2×10^5 when the number of MEC servers is 100. This is because the use of UAV allows dynamic adjust on the positions of MEC servers, causing less transmission loss. Furthermore, as the number of MEC servers continues to increase, all tasks are offloaded to the MEC server for execution when resources in the MEC server are sufficient. When the number of servers

exceeds 70, the overhead of the whole system is almost unchanged, which shows that only increasing the number of MEC servers cannot increase the total overhead of the system. With the increase of MEC servers, the queuing latency decreases and the cost when all tasks are offloaded to MEC servers decreases at first, but it remains constant when there are enough MEC servers.

5. Conclusion

As a key and fundamental technology in 5G, the MEC system needs to meet the requirements of low latency and low energy consumption. Applying the high flexibility of UAVs, an UAV-assisted edge computing resource allocation strategy for 5G communication in the IoT environment is proposed. A system model of an UAV deployed with a MEC server to provide assisted computing services for multiple users on the ground is constructed. And the minimization of total system cost, which is composed of channel model and energy consumption model, is solved by the improved genetic algorithm, obtaining the best resource allocation strategy. The simulation results show that

- (1) The system model introduces the UAV-assisted edge computing with high flexibility, which can effectively reduce the total system time to be less than 17 s when the number of tasks increases.
- (2) Improving the genetic algorithm by using a penalty function and combination coding can enhance its ability of global optimization so as to obtain a resource allocation scheme that minimizes the total system cost. The total cost of the proposed method is 1800 when the number of tasks per level is 30 and it is less than 0.2×10^5 when the number of MEC servers is 100.

The main scenario studied in this article is a single UAV providing assisted computing services to users, while multi-UAV scenarios are more important in practical applications. Therefore, in future research studies, the focus will be attached on the scenarios of multiple UAVs serving ground users.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in Internet of underwater things networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1775–1789, 2022.
- [2] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-

- sensitive mobile edge computing,” *Sustainable Computing: Informatics and Systems*, vol. 21, no. 3, pp. 154–164, 2019.
- [3] D. Zhu, C. Xiang, and S. Bing, “Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 99, pp. 304–313, 2018.
 - [4] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, “A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing,” *Future Generation Computer Systems*, vol. 108, no. 3, pp. 273–287, 2020.
 - [5] M. Mukherjee, S. Kumar, C. X. Mavromoustakis et al., “Latency-driven parallel task data offloading in fog computing networks for industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6050–6058, 2020.
 - [6] H. Wang, “Collaborative task offloading strategy of UAV cluster using improved genetic algorithm in mobile edge computing,” *Journal of Robotics*, vol. 2021, Article ID 3965689, 9 pages, 2021.
 - [7] Y. Zhang, M. Zhang, and C. Fan, “Computing resource allocation scheme of IOV using deep reinforcement learning in edge computing environment,” *EURASIP Journal on Applied Signal Processing*, vol. 2, no. 1, pp. 1–19, 2021.
 - [8] D. Wang, H. Qin, B. Song, X. Du, and M. Guizani, “Resource allocation in information-centric wireless networking with d2d-enabled MEC: a deep reinforcement learning approach,” *IEEE Access*, vol. 7, no. 10, pp. 114935–114944, 2019.
 - [9] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, “Age of information in energy harvesting aided massive multiple access networks,” *IEEE Journal on Selected Areas in Communications*, p. 1, 2022.
 - [10] Wei, Feng, and Zou, “Distance control strategy for deploying content replica servers in an edge cloud,” *The Journal of China Universities of Posts and Telecommunications*, vol. 26, no. 2, pp. 95–102, 2019.
 - [11] H. Tang, C. Li, J. Bai, J. Tang, and Y. Luo, “Dynamic resource allocation strategy for latency-critical and computation-intensive applications in cloud-edge environment,” *Computer Communications*, vol. 134, no. 6, pp. 70–82, 2019.
 - [12] Q. Liu, R. Mo, and X. Xu, “Multi-objective resource allocation in mobile edge computing using PAES for Internet of Things,” *Wireless Networks*, vol. 4, no. 3, pp. 1–13, 2020.
 - [13] 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, no. 4, pp. 355–364, 2020.
 - [14] X. Zhang, J. Zhang, J. Xiong, L. Zhou, and J. Wei, “Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating NOMA,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5613–5627, 2020.
 - [15] J. Zhang, L. Zhou, F. Zhou et al., “Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2114–2125, 2020.
 - [16] Y. Wei, Z. Wang, D. Guo, and F. Richard Yu, “Deep Q-learning based computation offloading strategy for mobile edge computing,” *Computers, Materials & Continua*, vol. 59, no. 1, pp. 89–104, 2019.
 - [17] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, “Energy-efficient computation offloading in vehicular edge cloud computing,” *IEEE Access*, vol. 8, no. 5, pp. 37632–37644, 2020.
 - [18] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, “UAV-aided mobile edge computing systems with one by one access scheme,” *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 664–678, 2019.
 - [19] M. Sheng, Y. Wang, X. Wang, and J. Li, “Energy-efficient multiuser partial computation offloading with collaboration of terminals, radio access network, and edge server,” *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1524–1537, 2019.
 - [20] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Risk-aware data offloading in multi-server multi-access edge computing environment,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405–1418, 2020.
 - [21] C. Gza, Z. Hao, and B. Yla, “5G network-oriented hierarchical distributed cloud computing system resource optimization scheduling and allocation,” *Computer Communications*, vol. 164, no. 5, pp. 88–99, 2020.
 - [22] S. F. Abedin, M. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, “Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network,” *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 489–502, 2019.
 - [23] R. Siddavaatam, I. Woungang, and A. Anpalagan, “Joint optimisation of radio and infrastructure resources for energy-efficient massive data storage in the mobile cloud over 5G HetNet,” *IET Wireless Sensor Systems*, vol. 9, no. 5, pp. 323–332, 2019.
 - [24] Y. Wei, F. R. Yu, M. Song, and Z. Han, “Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, 2019.
 - [25] S. Zhang, Y. Liang, J. Ge, M. Xiao, and J. Wu, “Provably efficient resource allocation for edge service entities using hermes,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1684–1697, 2020.
 - [26] C. Tang, C. Zhu, X. Wei, H. Wu, Q. Li, and J. J. P. C. Rodrigues, “Intelligent resource allocation for utility optimization in RSU-empowered vehicular network,” *IEEE Access*, vol. 8, no. 1, pp. 94453–94462, 2020.