

## Research Article

# Resource Optimization Technology Using Genetic Algorithm in UAV-Assisted Edge Computing Environment

Huijuan Sun  and Hongqi Xi 

*College of Computer and Information Technology, Henan Finance University, Zhengzhou, Henan 450046, China*

Correspondence should be addressed to Huijuan Sun; [sunhuijuan@hafu.edu.cn](mailto:sunhuijuan@hafu.edu.cn)

Received 16 February 2022; Revised 18 March 2022; Accepted 19 March 2022; Published 6 April 2022

Academic Editor: Shan Zhong

Copyright © 2022 Huijuan Sun and Hongqi Xi. 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 fixed edge computing systems can hardly meet the demand of mobile users for massive data processing, a computational resource allocation strategy using the genetic algorithm in UAV-assisted edge computing environment is proposed. First, a UAV-assisted mobile edge computing (MEC) system is designed to help users execute computation tasks through the UAV or relaying to the ground base station. Then, a communication model and a computation model are constructed to minimize the total system energy consumption by jointly optimizing the UAV offloading ratio, user scheduling variables, and UAV trajectory. Finally, the minimization of total system energy consumption is modeled as a nonconvex optimization problem and solved by introducing an improved genetic algorithm, so as to achieve a rational allocation of computational resources. Based on the experimental platform, the simulation of the proposed method is carried out. The results show that the total energy consumption is 650 J when the execution time is 110 s and the execution time is 17.5 s when the number of users is 50, which are both better than other comparison methods.

## 1. Introduction

In recent years, as cloud computing are gradually moving to the edge of the network, mobile edge computing (MEC) has emerged and developed rapidly, which mainly utilizes the spare computing capacity and storage of the edge network to execute computation-intensive and delay-sensitive tasks [1, 2]. Currently, scholars are actively trying to integrate MEC-related technologies with a wireless communication theory to better address long propagation delays, and the deployment of mobile base stations on unmanned aerial vehicle (UAV) is a possible solution to extend the coverage of wireless systems to the areas with limited available infrastructure for wireless access points [3].

UAV is a non-manned vehicle operated by remote control, with advantages of good maneuverability and small size, which is easy to carry and transport. Meanwhile, with great stability during low altitude flight, UAV can better acquire images obscured by clouds and other objects, which cannot be captured by the satellite light-sensing remote controller and other aerospace photography [4]. Due to its

high maneuverability and flexibility, UAV plays an increasingly prominent role in the communication system and is widely used in aerial surveillance, imaging, and traffic [5]. To maintain the safe operation of UAVs through real-time command/control and to enable new applications with artificial intelligence, it is becoming more important to enhance the communication and computing capabilities of UAVs [6, 7].

However, UAVs are limited by size, load capacity, and power consumption, which make the communication, computing capacity, and endurance time all suffer from a large impact. In order to settle this limitation, it is of great practical significance to design resource allocation strategies and optimal control methods for UAV trajectories to make the best use of limited energy of UAV, so as to better provide high-capacity, high-speed, and low-delay services for wireless communication edge computing systems [8].

At present, there have been some studies at home and abroad for computational resource allocation strategies in edge computing environment. According to [9], the mixed integer and nonconvex optimization problem was decoupled

into two sub-problems of task offloading decision and resource optimization, which were jointly optimized by using a designed iterative algorithm. The experimental results showed that the optimization performance is comparable to that of the exhaustive search. However, this algorithm takes a longer time for optimization and the efficiency needs to be improved. The authors of [10] proposed a dynamic multi-win game model based on incomplete information to solve the task offloading and edge resource allocation problems for multiple terminal users. Further optimization is needed for multiobjective task offloading and resource allocation in complex network situations. The authors of [11] proposed a computing resource allocation scheme for MEC scenarios based on DRL networks. The cost of task offloading still needs to be optimized. In order to investigate the relationship between the cost of computation tasks and time and energy consumption, a computation offloading strategy was proposed, which reduces the execution cost by collaboratively allocating computational resources between mobile devices and edge servers [12].

With the continuous development of machine learning and UAV communication technologies, deep learning and UAV assistance have been gradually applied in computational resource allocation. Authors in Ref. [13] proposed a multiobjective optimization algorithm based on the MOACO algorithm combined with reinforcement learning, which effectively improves the quality of service in edge computing by establishing an effective cognitive agent model to evaluate the resource allocation. However, with more iteration, the time efficiency needs to be improved. In the study [14], a multiagent reinforcement learning framework based on independent learners was proposed to achieve computation offloading and resource allocation for multiple private users in IoT edge computing networks. Authors in Ref. [15] introduced agents into task offloading and proposed a new framework for agent-enabled task offloading in UAV-aided MEC to help users, UAV, and edge clouds perform computation task offloading. UAV can improve wireless communication coverage to a certain extent, but the computation offloading strategy still needs further optimization.

Based on the above analysis, a computational resource allocation strategy using the genetic algorithm in the UAV-assisted edge computing environment is proposed to address the problem that the computing capacity of existing mobile devices (computers, cell phones) cannot fully meet the users' demand for communication quality. Based on the UAV-assisted edge computing system, a minimization model of total system energy consumption with jointly optimized UAV offloading ratio, user scheduling variables, and UAV trajectory is constructed. The minimization problem is solved using an improved genetic algorithm to achieve a reasonable allocation of computational resources.

## 2. System Model and Optimization Objective

**2.1. System Model.** As shown in Figure 1, (a) UAV-assisted MEC system consists of a cellular-connected UAV and multiple ground-requesting users.

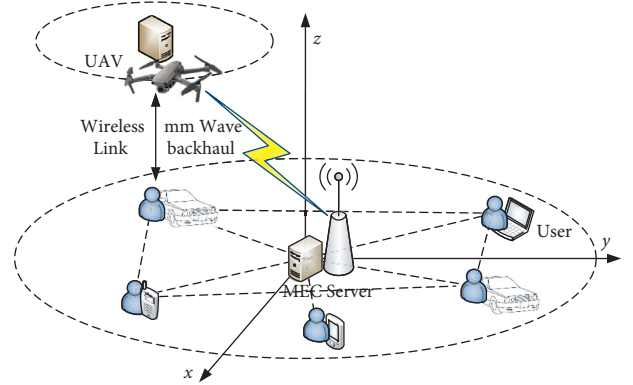


FIGURE 1: System model.

Both UAV and ground users have communication circuits and computing processors and are powered by batteries with a certain capacity. It is worth noting that different users have different computing capacities. The ground base station is deployed with multiple built-in MEC servers with high computing capacity, which can be regarded as an access point [16, 17]. In this scenario, the user has delay-sensitive computation tasks that require UAV to assist in computation. The UAV is deployed with MEC servers to help users to execute the task on itself or relay to the ground base station for execution.

**2.2. Communication Model.** In the system model, users are fixed on the ground and the position can be denoted as  $q$ ,  $q = [x, y]$ , where  $x$  and  $y$  are the horizontal coordinates of the user. UAV keeps flying at the same altitude  $H$  ( $H > 0$ ) and its position at  $i$  th timeslot can be expressed as  $q_u = [x_u[i], y_u[i]]$ ,  $0 \leq i \leq I$ .

It is assumed that the wireless channel between the UAV and users is a Line of Sight (LOS) channel. Therefore, the channel power gain between the UAV and users can be expressed as follows:

$$h[i] = \delta_0 d^{-2}[i] = \frac{\delta_0}{H^2 + \|q_u[i] - q\|^2}, \quad i \in I, \quad (1)$$

where  $\delta_0$  is the channel power gain within unit distance ( $d_0 = 1m$ ) and  $d^{-2}[n]$  is the horizontal distance between the UAV and the user at  $i$  th timeslot,  $i = 1, 2, \dots, I$ .  $\|\cdot\|$  is the Euclidean norm.

### 2.3. Computation Model

**2.3.1. Local Execution.** When users execute the task locally, the execution time can be calculated as follows:

$$T_n^{loc}[i] = \frac{(1 - \eta_n[i] \vartheta_n[i]) C_n D_n[i]}{f_n}, \quad (2)$$

where  $\eta_n[i]$  is the scheduling variable of user  $n$ .  $\vartheta_n[i]$  is the offloading ratio of user  $n$  at  $i$  th time slot.  $D_n[i]$  denotes the bit size of the computation task of user  $n$  at  $i$  th timeslot.  $C_n$  denotes the CPU frequency required when user  $n$  executes a

task of 1 bit.  $f_n$  denotes the local computing capacity of user  $n$ .

Then, the energy consumption for local execution can be written as follows:

$$E_n^{\text{loc}}[i] = \gamma(1 - \eta_n[i]\vartheta_n[i])D_n[i]C_n f_n^2, \quad (3)$$

where  $\gamma$  is a coefficient that depends on the chip architecture.

**2.3.2. UAV-Assisted Computation.** For UAV-assisted computation, it can be divided into two parts: the transmission process and the assisted computation. Thus, the UAV transmission time delay can be calculated as follows:

$$T_n^{\text{tr}}[i] = \frac{\eta_n[i]\vartheta_n[i]D_n[i]}{v_n[i]}, \quad (4)$$

where  $v_n[i]$  is the transmission rate between user  $n$  and the UAV at  $i$  th time slot. Then, the transmission energy consumption can be written as follows:

$$E_n^{\text{tr}}[i] = P_0 \frac{\eta_n[i]\vartheta_n[i]D_n[i]}{v_n[i]}, \quad (5)$$

where  $P_0$  is the rated transmission power of the UAV.

For the UAV-assisted computation, in each timeslot, the UAV serves multiple ground users, the execution time of UAV serving user  $n$  at  $i$ th timeslot for the assisted computation can be calculated as follows:

$$T_n^{\text{UAV}}[i] = \frac{\eta_n[i]\vartheta_n[i]D_n[i]C_n}{f_n^{\text{UAV}}}, \quad (6)$$

where  $f_n^{\text{UAV}}$  is the computing capacity of the UAV when serving user  $n$ . Hence, the energy consumption for the UAV-assisted computation can be formulated as follows:

$$E_n^{\text{UAV}}[i] = \gamma' \eta_n[i]\vartheta_n[i]D_n[i]C_n f_n^{\text{UAV}2}, \quad (7)$$

where  $\gamma'$  is a coefficient related to the chip architecture of UAV.

**2.4. Optimization Objective.** According to the aforementioned UAV-assisted computation model, the corresponding optimization problem is designed. It is assumed that user local execution and UAV-assisted computation are performed simultaneously, and the UAV needs to perform the computation and return the results at the same time slot. Because the size of the returned results is much smaller than that of the UAV-assisted computation task, the energy consumption for returning results can be neglected [18, 19]. The objective is to minimize the total system energy consumption by jointly optimizing the UAV offloading ratio, the user scheduling variables and the UAV trajectory. Thus, the optimization problem can be formulated as follows:

$$\Gamma = \min_{\{\eta_n, \vartheta_n, q\}} \sum_{i=1}^I E^{\text{fly}}[i] + \sum_{i=1}^I \sum_{n=1}^N (E_n^{\text{UAV}}[i] + E_n^{\text{tr}}[i] + E_n^{\text{loc}}[i]), \quad (8)$$

$$\begin{aligned} \text{s.t. } & q[1] = q[I], \\ & \frac{\|q[i+1] - q[i]\|}{\tau} \leq V_{\text{max}}, \quad i \in I, \end{aligned} \quad (9)$$

$$0 \leq \vartheta_n[i] \leq 1, \quad \forall n, i, \quad (10)$$

$$\sum_{n=1}^N \eta_n[i] \leq 1, \quad \forall i \quad (11)$$

$$\eta_n[i] \in \{0, 1\}, \quad \forall n, i, \quad (12)$$

$$v_n[i] \geq \eta_n[i] \cdot v, \quad \forall n, i, \quad (13)$$

$$\sum_{i=1}^I \sum_{n=1}^N T_n^{\text{loc}}[i] \eta_n[i] \leq T, \quad \forall n, i, \quad (14)$$

$$\sum_{i=1}^I \sum_{n=1}^N (T_n^{\text{tr}}[i] + T_n^{\text{UAV}}[i]) \eta_n[i] \leq T, \quad \forall n, i, \quad (14)$$

where  $E^{\text{fly}}[i]$  is the energy consumption of the UAV during its flight.  $V_{\text{max}}$  is the maximum flight speed of the UAV.  $\tau$  is the length of each timeslot.  $v$  is the flight speed of the UAV.

Equation (9) shows the constraint of the UAV trajectory. Equation (10) defines the constraint of the offloading ratio. Equation (11) represents the user scheduling variables.

Equation (12) ensures the communication quality during offloading. Equation (13) represents the constraint of user local execution delay. Equation (14) represents the constraint of user transmission delay and UAV-assisted computation delay.

### 3. Computational Resource Allocation Strategy Using Improved Genetic Algorithm

Genetic algorithm is a relatively mature algorithm that searches for the global optimal solution by simulating the process of biological evolution in nature. Its parallelism is suitable for distributed edge computing. Moreover, the genetic algorithm has good global search capability, so that an appropriate solution set for resource allocation can be obtained by using the genetic algorithm with multiobjective optimization.

Therefore, this paper introduces an improved genetic algorithm, which can not only improve the search efficiency but also obtain the further optimization objective by establishing a dynamic adjustment model of crossover rate and mutation rate, while reflecting the advantages of distributed execution of tasks. In the following sections, the implementation of resource scheduling of the proposed model is addressed based on the genetic algorithm.

**3.1. Basic Genetic Algorithm.** The basic genetic algorithm was proposed by J. H. Holland of the University of Michigan, USA, which is the basis for other genetic algorithms. Moreover, other improved genetic algorithms are developed by adding new mechanisms. The flow chart of basic genetic algorithm is shown in Figure 2.

The basic genetic algorithm generates optimal individuals by simulating the selection, crossover, and mutation operations in biological genetic and evolutionary processes and finds the optimal solution through an adaptive search process [20, 21].

#### 3.2. Improved Genetic Algorithm

**3.2.1. Encoding and Population Initialization.** The proposed method adopts binary encoding where a chromosome corresponds to a solution of the optimization problem. The defined chromosomes are encoded using real numbers. Therefore, it is necessary to convert the genes in the chromosome to be represented in binary. Moreover, the genes in the chromosome are all real numbers, which can be converted into a four-bit binary number and then be replaced in the corresponding genes. Moreover, this is the encoding method used in the improved genetic algorithm [22, 23]. For example, a chromosome sequence is (3, 5, 8) when encoded in real numbers, then the chromosome sequence is converted into (0011, 0101, 1000) after binary encoding.

However, the objective function value is calculated mainly based on real numbers, which requires reconvertng the binary-coded chromosome sequence into a real-coded chromosome sequence [24]. Moreover, after crossover and

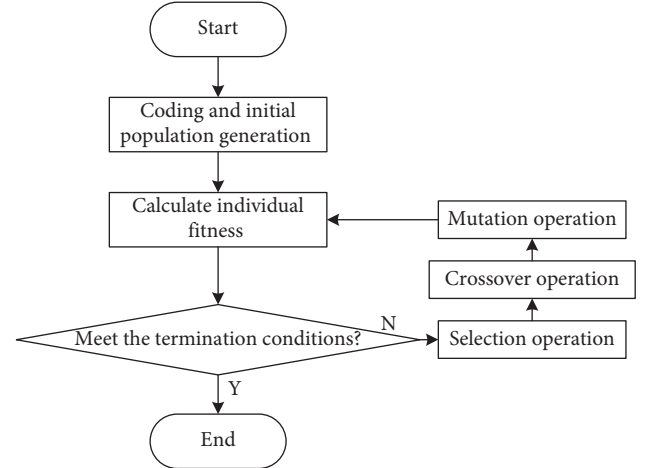


FIGURE 2: Flow chart of basic genetic algorithm.

mutation operations, the value of genes converted to real numbers may exceed the range of edge node numbers, so a remainder approach is used [25, 26]. For example, the total number of edge nodes is defined to be 7 and thus the binary code is {1101}, which will be converted to a decimal number of 13. The remainder of 13 divided by 7 is 6, then the corresponding edge node number of this binary code is 6.

Set the population size as  $S$ , the task sequence as  $S_T$ , the length as  $a$ , and the number of edge nodes in the random network topology graph as  $m$ . Each chromosome is composed of a random combination of binary numbers whose length is  $4*a$ , and every four binary numbers will be converted into a decimal number, which represents the number of edge nodes.

**3.2.2. Fitness Function.** The fitness function is the key to evaluate the direction of population evolution in task scheduling of terminal users in edge computing. In order to ensure that the fitness function can remain meaningful when the value of optimization objective  $\Gamma$  is equal to 0, it is defined as follows:

$$f(\Gamma) = e^{-\Gamma}. \quad (15)$$

**3.2.3. Genetic Operations.** The selection operation is performed by a roulette wheel. Assuming that the size of the population is  $S$  and the fitness value of an individual  $s$  is  $f_s$ , and  $p_s$  is the probability of an individual  $s$  being selected, the probability can be calculated as follows:

$$p_s = \frac{f_s}{\sum_{j=1}^S f_j}. \quad (16)$$

The crossover operator is used to generate new individuals by swapping the positions of chromosome genes of the parent individuals. The traditional single-point crossover method is used, in which a crossover point is set between any two adjacent genes in each chromosome sequence  $S_F$ , then a crossover point is arbitrarily selected in the first  $w^{-1}$  genes,

and all genes after that crossover point are replaced with each other [27].

In the mutation operation, each chromosome sequence  $S_F$  uses a basic position mutation, where one gene is arbitrarily selected in  $w$  number of genes and replaced by a random number between 1 to  $w$ . Thus, a new chromosome is generated and the diversity of the population can be ensured.

**3.3. Steps of Task Scheduling.** The steps of the improved genetic algorithm are as follows.

- (1) Population initialization. The user submits a terminal request and randomly sets the initial population  $g(0)$  based on the request. Set the iteration counter  $t = 0$ , and  $g(t)$  indicates the  $t$ th generation of population.
- (2) Calculate the fitness value. The fitness value of an individual is calculated according to the fitness function in the improved genetic algorithm.
- (3) Judge the termination condition. When the number of evolutionary generations reaches the specified number of iterations  $\Theta$ , the result should be given and an edge computation task scheduling strategy is found; otherwise, go to step (4).
- (4) Selection operation. The selection operation of the populations is performed using the roulette wheel.
- (5) Crossover operation. According to the crossover method in the designed improved genetic algorithm, two individuals are randomly selected among the individuals produced by the selection operation and are performed based on the crossover rate, so as to generate new individuals.
- (6) Mutation operation. According to the mutation method in the improved genetic algorithm, new individuals are generated by crossover operation and the selected individuals among the new individuals are operated according to the mutation rate to generate new individuals.
- (7) Compare the fitness value. If the fitness value of the mutated individuals is smaller than that of parent individuals, they will be replaced by the parent generation. If it is greater than or equal to the fitness value of the parent generation, the parent generation will be replaced. Then, a new generation of individuals  $g(i + 1)$  is generated.
- (8) Update the iteration counter. Set the iteration counter as  $t = t + 1$ , then go to step (2).

## 4. Experiment and Analysis

The UAV trajectory and the performance of the proposed improved genetic algorithm are demonstrated by numerical simulations. The system simulation parameters are set as shown in Table 1.

The parameters of the improved genetic algorithm are set as follows. The number of iterations is 00, the crossover rate

is 0.75, the mutation rate is 0.01, and the initial population size is 25.

**4.1. UAV Trajectory.** The UAV trajectory optimized by the improved genetic algorithm during different task execution time  $T$  is shown in Figure 3, where the red circles indicate the positions of ground users.

From Figure 3, it can be seen that when  $T = 30$  s, the UAV flies along the closed-loop trajectory and the flight range is small, the UAV is far from users when offloading data, and the communication quality is poor. With the increase of  $T$ , and when  $T = 60$  s and  $T = 120$  s, the distance between the UAV and users decreases and the flight range expands, thus the communication quality improves. When  $T$  is large enough, the UAV continues to fly along the closed-loop trajectory and serves one user in several timeslots. In addition, it can be found that the UAV has a larger flight range and is closer to users when  $T$  is larger, and the communication quality is better.

**4.2. Convergence Performance of Algorithms under Different Computation Tasks.** In order to demonstrate the convergence of the improved genetic algorithm, it is compared with the traditional genetic algorithm, and the results are illustrated in Figure 4.

It is depicted from Figure 4 that the proposed improved genetic algorithm executes more tasks within the same number of iterations, and when the number of iterations exceeds 70, the amount of computation tasks tends to converge to 3450 M bit. Because it is optimized in encoding and genetic operations, the amount of computation increases. However, the process is slightly more complicated, resulting in longer processing time, and the number of iterations that tend to be stable is about 10 times higher than that of the original genetic algorithm. On the whole, the proposed algorithm can achieve effective convergence performance.

**4.3. Relationship between Total System Energy Consumption with Task Execution Time.** To demonstrate the performance of the total energy consumption of the proposed method, it is compared with those of the [9, 13, 15], and the results are shown in Figure 5.

As can be seen in Figure 5, with the increase of task execution time, the total energy consumption of the system also increases. Meanwhile, the proposed UAV-assisted computational resource allocation method requires lower energy consumption and has better performance than the other three methods. Moreover, the total energy consumption required by the proposed method is always the least while the task execution time is increasing. The total energy consumption is 650 J when the execution time is 110 s. As the proposed method executes the computation task with the assistance of UAV and optimizes the UAV trajectory using the improved genetic algorithm, it can dynamically and timely adjust the computation tasks so as to reduce the energy consumption. Reference [9] introduces

TABLE 1: System simulation parameters.

System parameters	Value
Number of users $N$	10
Flight altitude $H$	150 m
Channel power gain $\delta_0$	-75 dB
CPU frequency $C_n$	1200 cycles/bit
$\gamma = \gamma'$	$10^{-27}$
UAV computing capacity $f_n^{UAV}$	1400 Hz
User local computing capacity $f_n$	[200, 500] Hz
User fixed transmission capacity $P_0$	50 dB-m
Maximum flight speed of UAV $V_{max}$	60 m/s

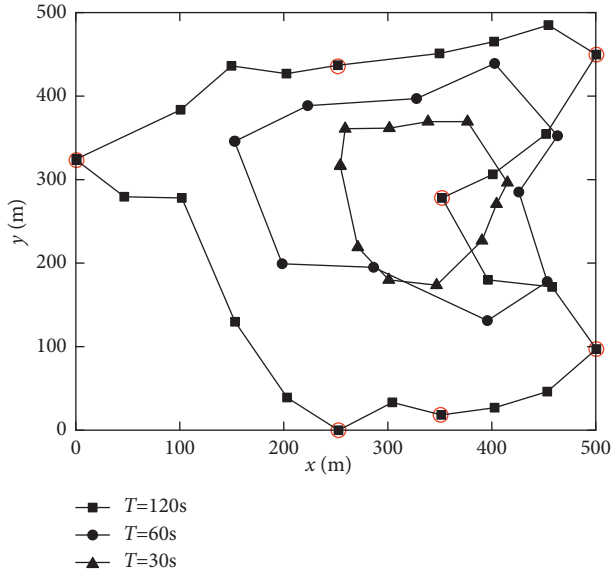


FIGURE 3: UAV trajectory.

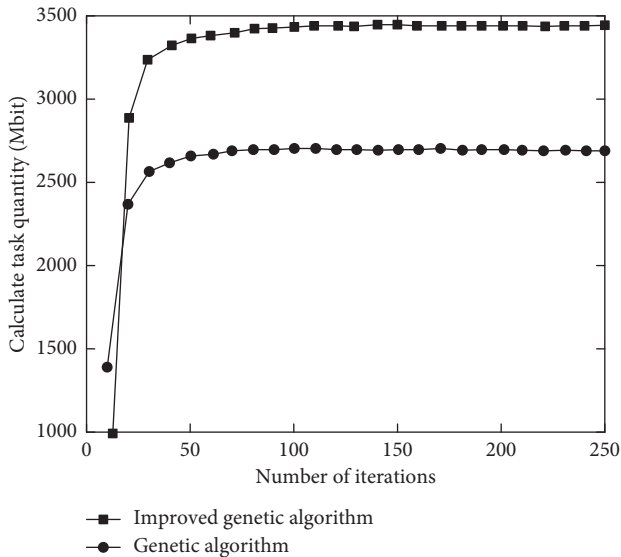


FIGURE 4: Convergence performance of genetic algorithms before and after improvement.

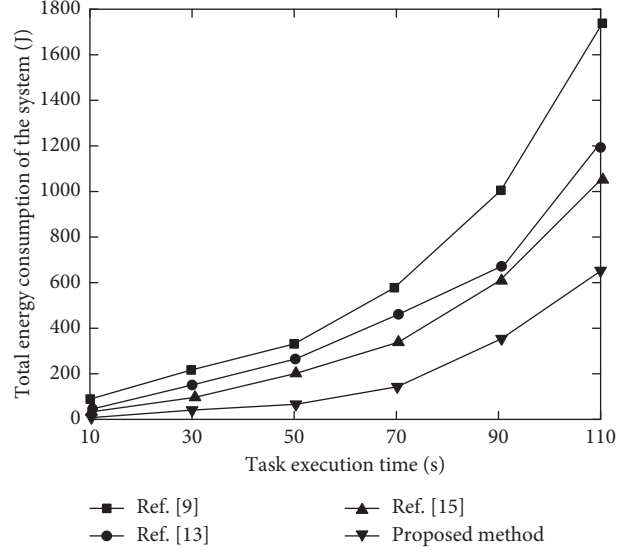


FIGURE 5: Total energy consumption of the system under different methods.

mixed integer and nonconvex optimization to obtain task offloading decision and resource optimization. However, the optimization method is traditional, the obtained resource allocation strategy is not reasonable enough, and the total energy consumption of the system is higher than 1600 J. Reference [13] combines MOACO algorithm and reinforcement learning for multiobjective resource allocation, which can effectively improve the quality of users' computing services, but the computational process is complex and the energy consumption is high. Similarly, [15] introduces UAV agents to assist in the offloading of computation tasks. However, it lacks an effective optimization algorithm to schedule its trajectory, so the energy consumption during UAV flight is high and its total system energy consumption is 62.5% higher than the proposed method.

**4.4. Performance Comparison with Other Methods.** In addition, to further validate the advantage of the proposed method in term of execution time, it is compared with [9, 13, 15], and the results are depicted in Figure 6.

It can be seen from Figure 6 that the task execution time increases continuously with more users, but the proposed method has the slowest increase and an obvious advantage in terms of execution time. The execution time is 17.5 s when the number of users is 50. The proposed method uses an improved genetic algorithm to optimize the UAV trajectory, it reduces energy consumption by lessening the blind flight time of the UAV and it offloads the tasks nearby to further reduce the transmission time. Compared with the proposed method, [15] lacks an effective algorithm to optimize the UAV trajectory, resulting in a longer flight time and a higher overall execution time, which is similar to that of [13]. Thus, the maximum execution time is more than 25 s. The optimization method in [9] is simple and easy to implement, and the execution time is shorter when the number of users is small. However, when the number of users increases, it is

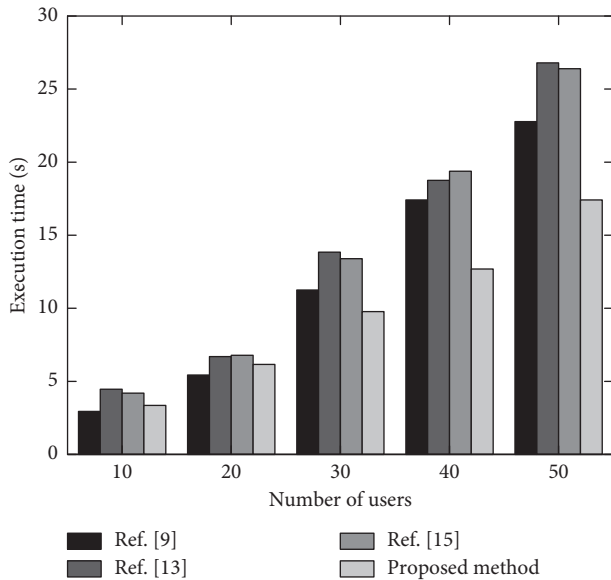


FIGURE 6: Execution time of different methods.

unable to handle the huge amount of data, resulting in the significant rise of the execution time.

## 5. Conclusion

With the explosive growth of IoT devices, cloud computing can no longer meet the demand of some time-sensitive computation tasks. For this reason, a computational resource allocation strategy using the genetic algorithm in the UAV-assisted edge computing environment is proposed. With the advantages of high mobility, low cost, and easy carrying, the UAV is introduced in the edge computing environment to assist in task offloading. The flexible scheduling of the UAV enables efficient allocation of user tasks and reduces execution time. In order to optimize the flight trajectory of the UAV, the problem of minimizing the total energy consumption of the system is solved by applying an improved genetic algorithm. Then, the optimal flight scheme can be obtained, that is, the optimal resource allocation strategy, which greatly reduces the system energy consumption and improves the transmission efficiency. The results based on the UAV simulation platform show as follows:

- (1) By real number encoding, selection, crossover, and mutation operations, the performance of the genetic algorithm in finding the optimal solution is effectively improved, and the data size of computational tasks can reach 3450 M bit when the number of iterations exceeds 70.
- (2) The introduction of the UAV allows dynamic response to users' computational demands and improves the efficiency of the computation offloading. When the number of users is 50, the execution time is 17.5 s, and at the same time, when the execution time is 110 s, the total energy consumption is 650 J, which effectively reduces the system energy consumption.

The UAV-assisted MEC system only considers the scenarios where UAV is connected with users or the ground. When the UAV only has limited computing capacity, the case that UAV collects users' data and offloads tasks to the ground base station for assisted computing is also worth studying.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## References

- [1] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3415–3426, 2020.
- [2] H. C. Ke, H. Wang, H. W. Zhao, and W. J. Sun, "Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing," *Wireless Networks*, vol. 27, no. 5, pp. 3357–3373, 2021.
- [3] J. Feng, L. Liu, Q. Pei, F. Hou, T. Yang, and J. Wu, "Service characteristics-oriented joint optimization of radio and computing resource allocation in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9407–9421, 2021.
- [4] L. Deng, P. Yang, W. Liu, L. Wang, S. Wang, and X. Zhang, "NAAM-MOEA/D-Based multitarget firepower resource allocation optimization in edge computing," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5579857, 14 pages, 2021.
- [5] Z. Song, Y. Liu, and X. Sun, "Joint task offloading and resource allocation for NOMA-enabled multi-access mobile edge computing," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1548–1564, 2021.
- [6] B. Baek, J. Lee, Y. Peng, and S. Park, "Three dynamic pricing schemes for resource allocation of edge computing for IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4292–4303, 2020.
- [7] Z. Wang, S. Zheng, Q. Ge, and K. Li, "Online offloading scheduling and resource allocation algorithms for vehicular edge computing system," *IEEE Access*, vol. 8, no. 9, pp. 52428–52442, 2020.
- [8] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-Aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 99, pp. 3341–3356, 2020.
- [9] Y. Fu, X. Yang, P. Yang et al., "Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2, no. 1, pp. 1–16, 2021.
- [10] B. Wang and M. Li, "Resource allocation scheduling algorithm based on incomplete information dynamic game for edge computing," *International Journal of Web Services Research*, vol. 18, no. 2, pp. 1–24, 2021.
- [11] Y. Zhang, M. Zhang, C. Fan, F. Li, and B. Li, "Computing resource allocation scheme of IOV using deep reinforcement

- learning in edge computing environment,” *EURASIP Journal on Applied Signal Processing*, vol. 21, no. 1, pp. 1–19, 2021.
- [12] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, “Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm,” *Sensors*, vol. 21, no. 3, pp. 779–787, 2021.
- [13] V. Shanmuganthan, M. Khari, N. Dey, R. G. Crespo, and Y. H. Robinson, “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. Liu, J. Yu, Z. Feng, and Y. Gao, “Multi-agent reinforcement learning for resource allocation in IoT networks with edge computing,” *China Communications*, vol. 17, no. 9, pp. 220–236, 2020.
- [15] W. A. Rui, C. A. Yong, B. An, T. A. Alamoudi, and R. Nour, “Agent-enabled task offloading in UAV-aided mobile edge computing,” *Computer Communications*, vol. 149, no. 2, pp. 324–331, 2020.
- [16] S. Li, S. Lin, L. Cai, W. Li, and G. Zhu, “Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3384–3398, 2020.
- [17] J. Yan, S. Bi, and Y. Zhang, “Offloading and resource allocation with general task graph in mobile edge computing: a deep reinforcement learning approach,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5404–5419, 2020.
- [18] Z. Yang, C. Pan, J. Hou, and M. Shikh-Bahaei, “Efficient resource allocation for mobile-edge computing networks with NOMA: completion time and energy minimization,” *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7771–7784, 2019.
- [19] J. Zhao, Q. Li, Y. Gong, and Z. Ke, “Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [20] L. I. Shulei, D. Zhai, D. U. Pengfei, and T. Han, “Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks,” *Science China Information Sciences*, vol. 62, no. 2, pp. 1–3, 2019.
- [21] L. Yang, G. Xu, J. Ge, P. Liu, and X. Fu, “Energy-efficient resource allocation for application including dependent tasks in mobile edge computing,” *KSII Transactions on Internet and Information Systems*, vol. 14, no. 6, pp. 2422–2443, 2020.
- [22] Y. Guo, F. Liu, N. Xiao, and Z. Chen, “Task-based resource allocation bid in edge computing micro datacenter,” *Computers, Materials and Continua*, vol. 58, no. 2, pp. 777–792, 2019.
- [23] H. T. Chien, Y. D. Lin, C. L. Lai, and W. Chien-Ting, “End-to-End slicing with optimized communication and computing resource allocation in multi-tenant 5G systems,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2079–2091, 2020.
- [24] 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.
- [25] M. Dai, Z. Su, Q. Xu, and N. Zhang, “Vehicle assisted computing offloading for unmanned aerial vehicles in smart city,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1932–1944, 2021.
- [26] K. Wang, Z. Xiong, C. Lin, Z. Pan, and H. Shin, “Joint time delay and energy optimization with intelligent overclocking in edge computing,” *Science China (Information Sciences)*, vol. 63, no. 04, pp. 154–169, 2020.
- [27] M. Zeng and V. Fodor, “Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access,” *Ad Hoc Networks*, vol. 98, no. 3, pp. 1–13, Article ID 102060, 2020.