Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6722848, 13 pages
https://doi.org/10.1155/2022/6722848

Research Article

Intelligent Time Allocation for Wireless Power Transfer in Wireless-Powered Mobile Edge Computing

Xiaogang Dong,[1,2] **Zheng Wan**,[1] **Changshou Deng**,[2] **Wenying Wen**,[1] **and Yuxuan Luo**[1]

[1]*School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China*
[2]*School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, China*

Correspondence should be addressed to Zheng Wan; wanzheng97@163.com

Received 31 March 2022; Revised 30 June 2022; Accepted 11 July 2022; Published 24 August 2022

Academic Editor: Zhao Junhui

Wireless-powered mobile edge computing is a new network computing paradigm that combines with the advantages of wireless power transfer and mobile edge computing. When the harvest-then-offload protocol is adopted in this network, the time of wireless power transfer has a significant impact on system performance. If the time is too short, the user cannot harvest enough energy. If it is too long, the user will not have enough time to complete the task offloading. Both result in many of user tasks being discarded. To address this problem, DEWPT, a differential evolution-based optimization scheme for wireless power transfer time, is proposed in this paper. DEWPT is designed with a hybrid mutation operator and a perturbation-based binomial crossover operator. The hybrid mutation operator combines the benefits of two mutation operators with distinct characteristics, so that DEWPT not only has a strong exploration ability but also can quickly converge. Meanwhile, the perturbation-based binomial crossover operator improves DEWPT's ability to exploit local space. These two improvements effectively enhance DEWPT's optimization performance, which is beneficial to find the optimal time for wireless power transfer. Furthermore, to improve the optimization efficiency, micro-population is introduced into DEWPT. Finally, the computation completion ratio maximization model is used to validate the performance of DEWPT in the wireless-powered mobile edge computing network with multiple edge servers. Numerical results show that the computation offloading scheme integrating with DEWPT can achieve a higher computation completion rate than three benchmark schemes, and is competitive in complexity. This demonstrates that DEWPT is an effective time allocation scheme for wireless power transfer.

1. Introduction

In recent years, the rapid development of Internet of Things (IoT) technology has spawned numerous new intelligent applications [1], such as virtual and augmented reality [2], unmanned driving [3], intelligent video analysis [4], and intelligent industrial production lines [5]. These applications are often computation-intensive and time-sensitive and have a very high demand for the computation capacity of wireless devices (WDs). However, WDs usually only have low computation power due to volume and manufacturing cost constraints [6]. In addition, the lifetime of WDs is very finite because of the battery capacity limitation. Therefore, the stable and sustainable operation cannot be maintained. Under the goal of service first, the two performance limitations have become the bottleneck that must be broken through to improve the quality of IoT services.

Mobile edge computing (MEC) [7, 8] and wireless power transfer (WPT), as two promising technologies, have attracted widespread concern in many fields. In the MEC architecture, the computation-intensive tasks of WDs can be offloaded to MEC servers deployed at the network edge for execution, which can effectively expand the computing power of WDs. Meanwhile, WPT can continuously charge the battery of WDs. It provides a solution to prolong the lifetime of WDs. Combining the advantages of MEC and WPT results in a new computing paradigm, wireless-powered MEC (WP-MEC). WP-MEC has become a potential solution to solve the above bottlenecks of IoT. Due to this significant advantage of WP-MEC, the optimization design and application of WP-MEC have got a lot of attention in recent years. However, the majority of existing works only focus on the task offloading decision-making.

The allocation of WPT time is a crucial factor affecting the performance of WP-MEC when the harvest-then-offload protocol is employed. How to achieve the optimal allocation of WPT time is a problem worth of studying for the WP-MEC. The existing works usually adopt traditional optimization methods (such as bisection search) or mathematical programming methods (such as Lagrangian duality) to deal with the allocation of WPT time. However, these methods are only suitable for the simple network scenarios. In complex real network scenarios (such as the network with massive connections), they are either difficult to implement or achieve satisfactory optimization performance.

Based on the above shortcomings of the existing works, we focus on the allocation of WPT time for WP-MEC in this paper, aiming to achieve the optimal allocation of WPT time. To this end, a differential evolution-based optimization scheme for the WPT time allocation is proposed. The contributions of this paper are summarized as follows.

(i) A novel differential evolution algorithm is developed to optimize the time allocation for WPT in a WP-MEC with multiple edge servers, in which a hybrid mutation operator and a perturbation based binomial crossover operator are designed. The hybrid mutation operator combines the advantages of two mutation operators with distinct characteristics. The perturbation-based binomial crossover operator enhances search ability of the algorithm in the local space through a random perturbation

(ii) To improve the efficiency of the algorithm, a micro-population scheme is introduced into the algorithm. The scheme selects an appropriate population size based on trade-off between performance and efficiency. Therefore, it can guarantee the efficiency of finding a better WPT time without performance loss

(iii) A new task offloading scheme is constructed by integrating the algorithm into the computation completion rate maximization model. Extensive experiments are conducted to evaluate the performance of our proposed algorithm

The rest of this paper is organized as follows. The related works are discussed in Section 2. Section 3 presents the network model and problem formulation. The proposed scheme is described in Section 4, and the simulation results are given in Section 5. Finally, Section 6 concludes this work.

2. Related Works

Over the past few years, many existing works have studied the optimal design of WP-MEC from various perspectives. According to the optimization design objectives, these works can be divided into the following four categories.

2.1. Maximization of Computation Rate. Bi and Zhang [6] considered a multiuser WP-MEC system, where the user task offloading mode is binary mode. The authors designed a joint optimization scheme for WPT time, user computation mode, local CPU frequency, computation time, and offloading communication time, aiming to maximize computation rate. Huang et al. [9] extended the work of [6]. Deep reinforcement learning was introduced. It primarily addressed the issue of real-time selection of user computing mode in wireless channel time-varying scenarios. Zhou et al. [10] considered an unmanned aerial vehicle- (UAV-) enabled WP-MEC system. The computation rate maximization problem was investigated for both partial and binary computation offloading modes in this work. The two-stage algorithm and the three-stage alternative algorithm were designed for the two offloading modes, respectively. Simulation results showed that the two algorithms outperformed benchmark algorithms in terms of performance, converge rate, and computation complexity. However, it was assumed that users can simultaneously perform energy harvesting, local computing, and computation offloading.

For the WPT time, the bisection search algorithm was used in [6, 9] to find the optimal time of WPT while the work [10] directly assumes that the entire time frame is used for the WPT.

2.2. Maximization of Computation Efficiency. Zhou and Hu [11] studied the computation efficiency maximization problem of wireless-powered MEC networks under both partial and binary computation offloading modes. Different from other existing works, this work designed solution schemes for time division multiple access and nonorthogonal multiple access. Ji and Guo [12] considered a WP-MEC system including two users. It should be noted that this work only considered the offloading computation of user tasks. Based on the consideration of "doubly near-far" effect, the author devised a user cooperation scheme of task offloading to maximize the energy efficiency (the ratio of the user throughput to energy).

In these two works, the time allocation of WPT was optimized by the mathematical programming methods, such as Lagrangian method, Newton iteration method, and subgradient algorithm. Clearly, these methods are only suitable for the scenarios with a small user scale.

2.3. Minimization of Energy Consumption. The total energy consumption minimization problem of the wireless access point (WAP) was formulated in [13]. Subsequently, an optimal resource allocation scheme was developed for a practical scenario where latency-limited computation was required. Similar to [12], Hu et al. [14] also studied a two user WP-MEC system. However, the optimization objective was minimization of the WAP total transmission energy. The author first illustrated that the optimization is equivalent to a min-max problem. Next, a two-phase optimization method was devised to solve it. Wang et al. [15] investigated a multiple-user WP-MEC system. In their work, apart from the special scenarios where the channel state information and the task state information are completely known, the author further studied the optimization scheme under the more practical application scenarios and proposed a sliding-window based online resource allocation scheme by integrating with the sequential optimization.

Similarly, these works either used the mathematical programming methods or assumed that the entire time frame is used for WPT.

2.4. Maximization of Computation Completion Ratio. The concept of computation completion ratio (CCR) [16] was first introduced for the optimization design of WP-MEC. CCR is a vital metric, which can effectively indicate the computing performance of WP-MEC. Meanwhile, the WP-MEC with multiple edge devices was considered in their work. Under this network configuration, the author proposed the CCR maximization scheduling scheme, which is termed as CoCoRaM. CoCoRaM achieved a higher CCR through joint optimizing the WPT time allocation and computation scheduling.

In the CoCoRaM, the approximate optimal time allocation of WPT was obtained by constructing a set of candidate times. The number of candidate times in the set is greater than the number of users. That is, as the user scale increases, the number of candidate times will grow.

Although the optimization design of WP-MEC system has been extensively studied in the previously mentioned works, there are still some issues that require further investigation, for example, the optimal time allocation of WPT. Despite bisection search, methods based on mathematical derivation and approximation method have been used to find the optimal time allocation of WPT. However, these methods either require analytical knowledge of model and complex computations or can only be applied to specific network scenarios. Therefore, an effective optimization scheme for the time allocation of WPT still deserved further research.

Some main notations used in this paper are summarized in Table 1.

Table 1: Summary of some main notations.

Notation	Definition
N	Number of users
Q	Number of MEC servers
U	Set of users
S	Set of MEC servers
D_i	Task size of the ith user
T	Time frame duration
E^h	Harvested RF energy
E^o	Energy of task offloading
E^l	Energy of local computing
ϕ_i	CPU cycles of processing one bit data
CCR	Computation completion ratio
$f^{\max}$	The upper of user CPU power
$f(\bullet)$	Fitness function
x	Decision matrix
$\boldsymbol{t}^l$	Local computing time vector
$\boldsymbol{f}$	Local computing CPU frequency
t^0	Offloading time vector
P	Power of ET
pop	Population of DE
NP	Population size
F	Scale factor
CR	Crossover probability
$pool$	Mutation operators pool
Λ_1, Λ_2	Control parameters
CBs	Computation bits
$\mathcal{F}(x)$	Probability density function

3. Network Model and Problem Formulation

This section will describe the network model, the local computing model, and the offloading computing model, respectively. A CCR maximization problem is then formulated.

3.1. WP-MEC Network with Multiple Edge Servers. As illustrated in Figure 1(a), a wireless-powered mobile edge computing network is investigated in this paper. It consists of a radio frequency (RF) energy transmitter (ET) with a single-antenna, N mobile wireless users, and Q wireless access points (WAP). Each WAP integrated with an MEC server provides the computing service to users. $\mathcal{U} = \{u_1, \cdots, u_N\}$ represents the set of users, and the set of edge servers is denoted by $\mathcal{S} = \{s_1, \cdots, s_Q\}$. The ET broadcasts RF energy through WPT for all users. Each user can harvest the RF energy by a single-antenna energy receiver to charge its rechargeable battery. The network system employs the binary offloading rule. That is, by utilizing the harvested energy, users can complete their computing task at local or a chosen MEC server (i.e., offloading computing). Certainly, if the energy harvested by a user is not enough to complete its task locally or offload it to any MEC server, its task will be discarded. For one time frame with duration T, the harvest-then-offload protocol (HTOP) is used, and each user has a task of size D_i bits to accomplish. Therefore, local computing of users and RF energy harvesting from ET can be simultaneously executed while the energy harvesting and task offloading cannot be performed concurrently. The time division multiple access protocol is used to avoid the communication interference caused by multiple users offloading tasks to the same MEC server. Figure 1(b) gives an example of the time allocation between WPT and computing offloading of users. Due to the powerful computation power of the edge servers and the very small task results, the time of edge servers computing and sending results back can be safely ignored like in [6, 17, 18]. The amount of RF energy harvested by each user can be calculated by a linear energy harvesting as formula (1)

$$E_i^h = \mu P g_i \tau_0, \tag{1}$$

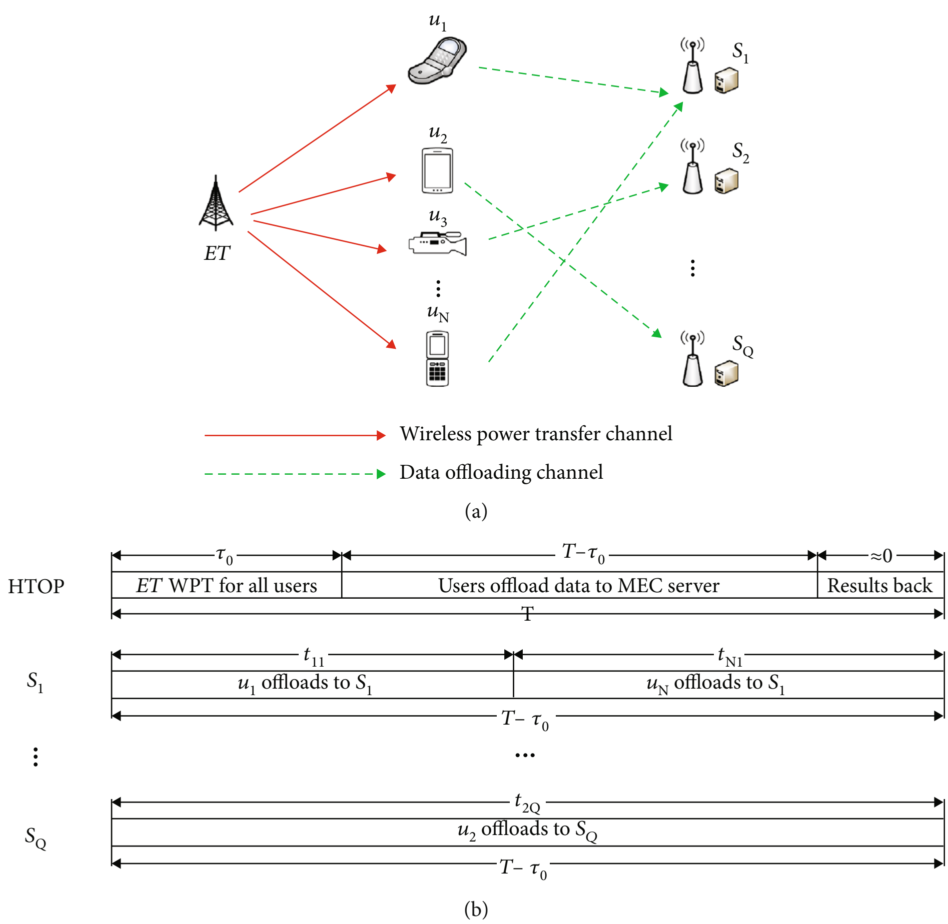

Figure 1: Network system. (a) WP-MEC network model. (b) Time allocation of HTOP.

where τ_0 is the time of WPT. The channel gain between ET and user u_i is denoted by g_i which remains constant in one time frame, and P is the energy transmission power of ET. μ denotes the energy harvesting coefficient.

3.1.1. Local computing model. If the user u_i performs its computation task locally, the relationship between CPU frequency and local computing time can be obtained by formula (2), and the energy consumption can be calculated by formula (3).

$$f_i t_i^l = D_i \phi_i, \quad (2)$$

$$E_i^l = k_i f_i^3 t_i^l, \quad (3)$$

where f_i denotes the CPU frequency, which can be adjusted between $(0, f^{\max}]$ by dynamic voltage and frequency scaling technique. t_i^l is the local computing time of u_i. ϕ_i represents the needing CPU cycles that u_i processes unit bit data. k_i is the computation energy efficiency coefficient of the processor's chip. The power consumption of the processor is modelled as $k_i f_i^3$ [6]. According to the constraint of energy, E_i^l cannot exceed E_i^h, i.e., $E_i^l \le E_i^h$.

3.1.2. Offloading computing model. When u_j offloads task to MEC server s_k for remote executing, the time of task uploading t_{jk}^o and the energy consumption E_{jk}^o can be obtained by formulas (4) and (5)

$$t_{jk}^o = \frac{D_j}{W \log_2\left(1 + \left(P_{jk} h_{jk} / \sigma^2\right)\right)}, \quad (4)$$

$$E_{jk}^o = P_{jk} t_{jk}^o, \quad (5)$$

where W is the transmission bandwidth. P_{jk} and h_{jk} are the transmission power and channel gain between u_j and s_k, respectively. σ^2 is the noise power. Likewise, E_{jk}^o must meet the energy constraint, i.e., $E_{jk}^o \le E_j^h$.

3.2. CCR Maximization Problem. As described in related work (Section 2), computation rate, computation efficiency, energy consumption, and computation complete ratio (CCR) are four common optimization objectives for the WP-MEC. Computation efficiency and energy consumption are only effective when all user tasks are guaranteed to be completed. Furthermore, the computation rate and the CCR are essentially equivalent when the user task information is a priori. But, as the ratio of processed computation data to the required computation data of all users, CCR can more directly reflect the optimization performance of the scheme. Thus, maximization CCR is selected as the optimization objective in this paper. According to the binary offloading rule, each user has $(1+Q)$ options for task execution, i.e., local and Q MEC servers. A variate x_{ik} ($k \in [0, Q]$) is used to indicate the choice of user task execution. If u_i performs its task at the kth option, $x_{ik}=1$. Otherwise, $x_{ik}=0$. Note that $k=0$ denotes local computing. So, the maximization problem of CCR is as formula (6)

$$
\begin{aligned}
&\underset{\tau_0,\mathbf{x},\mathbf{t^l},\mathbf{f},\mathbf{t^o}}{Max} \quad CCR = \frac{\sum_{i=1}^{N} x_{i0} D_i + \sum_{k=1}^{Q}\sum_{j=1}^{N} x_{jk} D_j}{\sum_{i=1}^{N} D_i},\\
&\text{s.t. :}\\
&\qquad C1 : \tau_0 \le T,\\
&\qquad C2 : f_1 \in [0, f^{\max}], \forall i \in [1, N],\\
&\qquad C3 : 0 \le t_i^l \le t, \forall i \in [1, N],\\
&\qquad C4 : 0 \le t_{jk}^o \le T - \tau_0, \forall j \in [1, N] \forall k \in [1, Q],\\
&\qquad C5 : \sum_{j=1}^{N} t_{jk} \le T - \tau_0,\\
&\qquad C6 : E_i^l \le E_i^h, E_{jk}^o \le E_j^h, i, j \in [1, N], k \in [1, Q],\\
&\qquad C7 : x_{ik} \in \{0, 1\}, \forall j \in [1, N], \forall k \in [0, Q],
\end{aligned} \tag{6}
$$

where τ_0 is a float variate. $\boldsymbol{x} = \{x_{ik} | i \in [1, N], k \in [0, \mathrm{Q}]\}$ is an integer matrix. $\boldsymbol{t}^l = \{t_i^l | i \in [1, N]\}$, $\boldsymbol{f} = \{f_i | i \in [1, N]\}$, and $\boldsymbol{t}^o = \{t_{jk}^o | j \in [1, N], k \in [1, Q]\}$ are three float vectors. Obviously, the maximization problem is a mixed integer nonlinear programming problem that is NP-hard and difficult to solve by the traditional optimization methods.

4. The Proposed Scheme of WPT Time

This section will first analyze the nested optimization structure of the optimization problem, i.e., formula (6). Then, a WPT time allocation algorithm based on the differential evolution algorithm is proposed. In this algorithm, a hybrid mutation operator and a perturbation-based binomial crossover operator are designed for achieving the optimal allocation of WPT time, and the micropopulation is introduced to improve the optimization efficiency.

4.1. Analysis of Optimization Problem. According to the feature of the WP-MEC system, all energy consumed by users comes from the RF energy harvested by them. In one time frame, WPT time τ_0 is the only factor that affects the amount of energy harvested by the user when other factors are fixed. Furthermore, the user offloading operation should be completed within $(T-\tau_0)$. Based on these, the following conclusions can be drawn.

(i) The WPT time τ_0 is a crucial decision factor. Too short τ_0, the users cannot harvest enough energy. Too long τ_0, the users have less time for task offloading

(ii) Given a τ_0, different decision matrices $\boldsymbol{x}$ will significantly affect the offloading performance. Thus, it is only makes sense to evaluate whether the τ_0 is optimal based on the optimal decision matrix $\boldsymbol{x}^*$ and resource allocation. Meanwhile, the resource allocations (i.e., $\mathbf{t^l}, \mathbf{f}, \mathbf{t^o}$) are closely related to the decision matrix $\boldsymbol{x}$

According to these analyses, optimization problem (6) is essentially a nested optimization problem that can be divided into two layers, i.e., the inner layer and outer layer (see Figure 2). The inner layer determines the optimal decision matrix $\boldsymbol{x}$ for a given WPT time, as well as the corresponding resource allocation. The outer layer optimizes the WPT time. The solution of the inner layer serves as the basis for evaluating the solution of the outer layer. The work of [16] modelled optimization problem of the inner layer as a generalized assignment problem (GAP) and designed the generalized assignment problem-based computation scheduling (GAP-CS) algorithm for finding the optimal decision matrix $\boldsymbol{x}^*$ and the corresponding resource allocation under the given WPT time.

In this work, the outer optimization problem is focused. To design an optimization scheme for WPT time, we must understand the relationship between WPT time and CCR which is the optimization objective of this work. Since the required computation data of all users is a priori, here, we simplify CCR to computation bits (CBs), the molecular part of formula (6), and observe their relationship through a sampling experiment. Assume that the CBs are a function of WPT time τ_0 as formula (7). The sampling experiment adopts an iterative search method to obtain the optimal decision matrix $\boldsymbol{x}^*$ and the corresponding resource allocation. The observation results are given in Figure 3.

$$
\mathrm{CBs} = f(\tau_0), \tau_0 \in [0, T]. \tag{7}
$$

From the results of Figure 3, it can be seen that the function curve obviously has multiple extreme points (see green oval mark). This shows that CBs are a nonmonotonic multimodal function of WPT time. Equivalently, CCR is a nonmonotonic multimodal function of WPT time. Therefore, the optimization of WPT is difficult to solve by the traditional search methods directly, such as bisection search.

4.2. The Allocation of WPT Time Based on DE. Differential evolution (DE), which is a heuristic evolution algorithm, was proposed by Store and Price [19]. A population pop with NP individuals is maintained, and each individual of pop denotes a solution to the optimization problem. The pop is initialled by a random way, and then, mutation (8), crossover (9), and selection operators are performed to update the pop generation by generation [20]. The widely used mutation and crossover operators [21] are as formulas

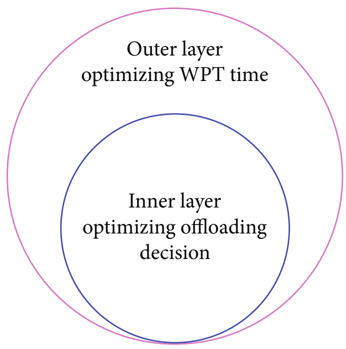

Figure 2: Nested optimization.

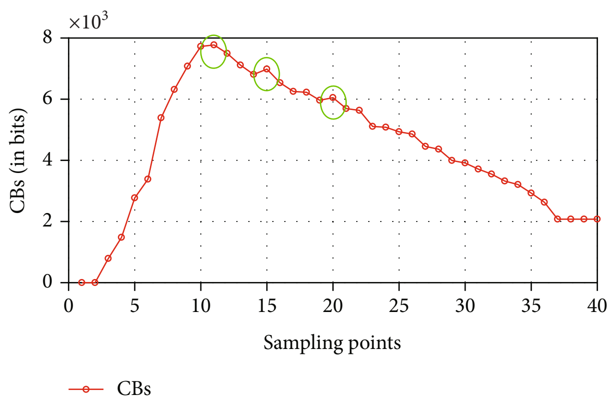

Figure 3: Results of sampling experiment.

(8) and (9)

$$V_i^g = X_{r1}^g + F \cdot \left(X_{r2}^g - X_{r3}^g\right), \tag{8}$$

$$U_{i,j}^g = \begin{cases} V_{i,j}^g, & \text{if } rand \leq CR \text{ or } j = J, \\ X_{i,j}^g, & \text{otherwise,} \end{cases} \tag{9}$$

where $i \in [1, NP]$ is the index of individual. $r1, r2, r3 (r1 \neq r2 \neq r3 \neq i)$ are generated in $[1, NP]$ by a random manner. g is the index of generation. V, U represent the mutation individual and trial individual, respectively. $j = 1, 2, \cdots, D$, and D is the number of gene in an individual. F and CR are scale factor and crossover probability, respectively. $J \in [1, D]$ is a uniformly distributed random number. $f(\cdot)$ is the fitness function.

Due to the powerful optimization performance, DE has been successfully applied in many of fields [22–24]. Inspired by these successful applications, a novel DE algorithm is designed for optimization the allocation of WPT time in this work. Next, the detailed designs of this algorithm will be given.

4.2.1. Hybrid mutation operator. In the DE algorithm community, there are many of mutation operators [21]. Some of them have outstanding global searching capability, which is conducive to find the global optimal solution, say, DE/rand/1, as formula (8). Others of them have excellent local searching capability and can speed up the convergence of the algorithm, say, DE/best/1, as formula (10)

$$V_i^g = X_{best}^g + F \cdot \left(X_{r1}^g - X_{r2}^g\right), \tag{10}$$

where best is the index of the optimal individual in the gth generation. The other parameters are same as formula (8).

For a given real-world optimization problem, it is not easy to choose the best one among different mutation operators. On the other hand, in order to improve quality of WP-MEC service, finding the optimal WPT time should not only meet the requirement of high precision but also enhance the real-time performance as much as possible.

Based on the above analysis, a new hybrid mutation operator is designed. The main idea of the hybrid mutation operator includes three key points.

(i) A pool of mutation operators *pool* is constructed, which consists of a mutation operator with outstanding global search performance (DE/rand/1) and a mutation operator with a fast convergence rate (DE/best/1)

(ii) A random number q is generated in each iteration according to a uniform distribution u. The probability density function $\mathscr{F}(x)$ of u is formula (11), where a is the lower bound, and b is the upper bound, respectively

$$\mathscr{F}(x) = \begin{cases} \dfrac{1}{b-a}, & a \leq x \leq b, \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

(iii) A fixed threshold θ is set in the algorithm's initial stage. In each generation, if q is greater than θ, the algorithm selects DE/rand/1 from *pool* to perform mutation. Otherwise, the DE/best/1 is selected from *pool* for mutation operation

The hybrid mutation operator can effectively improve the global search ability and accelerate the convergence rate of DE when solving the WPT time allocation problem. The detailed pseudocode of the hybrid mutation is given in Algorithm 1.

4.2.2. Perturbation-based binomial crossover. The crossover operator plays a crucial role in DE algorithm. Its major contribution is to improve the DE's exploitation performance. The binomial crossover operator, formula (9), has the characteristics of simple structure and easy implementation. However, there is still room for performance improvement, particularly when it comes to real-world applications. To further improve the exploitation performance of DE and enhance the efficiency of searching optimal WPT time in WP-MEC system, a perturbation-based binomial crossover

```
Input: pool, θ and the target individual
Output: mutation individual
generate a random number q by the uniform distribution u with the probability density function 𝓕(x)
if q < θ then
    select DE/rand/1 from pool, perform mutation to produce the mutation individual
else
    select DE/best/1 from pool, perform mutation to produce the mutation individual
return mutation individual
```

Algorithm 1: Hybrid mutation operator.

```
Input: the set of users 𝒰, the set of MEC servers S. The set of WP-MEC parameters para. The algorithm parameters: NP, F, CR,
maxIter, Λ1 = Λ2 = 0.1.
Output: the optimal CCR*
generate the initial population pop in a random way.
execute GAP-CS with para and each individual of pop to produce the decision
matrix x.
iter = 1
while iter ≤ maxIter do
    for i = 1 to NP
        select r1, r2, r3 from [1, NP]/i in random way
        execute the hybrid mutation operator to produce v_i via Algorithm 1
        execute the perturbation-based binomial crossover to produce u_i by formula (12)
        execute GAP-CS with u_i and para to produce decision matrix x and resource allocation t^l, f, t^o
          calculate CCR by formula (6)
          execute selection operator, and update the optimal CCR*
return the optimal CCR*
```

Algorithm 2: DEWPT.

as formula (12) is designed.

$$U_{i,j}^{g} = \begin{cases} V_{i,j}^{g} + \Lambda_1^{*}(-1)^{I_{\mathrm{rnd}}}, & \text{if } rand < CR, \\ X_{i,j}^{g} + \Lambda_2^{*}(-1)^{I_{\mathrm{rnd}}}, & \text{otherwise}, \end{cases} \tag{12}$$

where Λ_1 and Λ_2 are two fixed control parameters and the I_{rnd} is a random integer. The other notations are same as formula (9).

The main idea behind the perturbation-based binomial crossover is to improve the exploitation of the space around the target and mutant individuals through a random perturbation. In the actual implementation process, the perturbation range can be controlled by adjusting the sizes of Λ_1 and Λ_2. The values of Λ_1 and Λ_2 can be same or different according to the actual needs.

4.2.3. Micro-population for DE. The population size has a significant effect on the overall performance of DE algorithm [25]. When the population size is large, DE requires high computation cost and occupies a large amount of memory [26]. This makes DE difficult to apply to some application scenarios with small memory and high real-time requirements. Therefore, micro-population (no more than 10 individuals)-based DE algorithms have received high attention in recent years [27–29]. Furthermore, micro-population can meet the two goals of high efficiency and low computation cost of WP-MEC network optimization.

In view of the above analysis, we design a micro-population scheme for the DE algorithm to solve the WPT time allocation problem in this work. In order to obtain the optimal population size, we used several groups of population size less than 10 for pre-experiment (to save space, the pre-experiment process is omitted). Through verification and analysis, we found that DE has a higher probability of obtaining a better WPT time allocation when the population size is set to 6. Therefore, we set the population size of the proposed DE algorithm to 6.

Integrating the three aspects of design, we construct the solving framework for optimization problem (6), which is termed as DEWPT. In DEWPT, GAP-CS is used to solve the inner optimization problem. The pseudocode structure of DEWPT is described as Algorithm 2.

5. Simulation Results

In this section, we provide extensive simulations to evaluate the performance of our proposed scheme. The Reyleigh

Table 2: Parameter setting.

Parameters	Notations	Value
The duration of time frame	T	0.3 s
Transmission bandwidth	W	1.45 MHz
The ET power of broadcasting RF energy	P	3 w
The energy harvesting coefficient	μ	0.51
The upper of user CPU power	$f^{\max}$	0.5 GHz
The computation energy efficiency coefficient	k_i	10^{-28}
The CPU cycles of user processing one bit data (in cycles/bit)	ϕ_i	800
The transmission power of user offloading	p_{ik}	0.12 w
The scope of user task size (in kb)	D_i	[50,150]
The scale factor of DE	F	0.5
The crossover probability of DE	CR	0.5
Population size of DE	NP	6

fading channel model is used in all simulations. Moreover, Table 2 gives the configuration of other parameters. Noted that each plotted point in all figures represents the average results of 20 group of user tasks.

Furthermore, the following three schemes are used as benchmark schemes for comparison.

(i) CoCoRaM: CoCoRaM is an approximation algorithm proposed in [16], which optimizes the WPT time by constructing a candidate set of WPT times, and the optimal offloading decision is got by GAP-CS

(ii) LCO: the WPT time is set to the time frame T. All users either perform their tasks locally or discard them and cannot offload them

(iii) COO: all users cannot perform tasks locally and either offload them to a selected edge device or drop them. The optimal WPT time and the optimal offloading decision are got by CoCoRaM and GAP-CS, respectively

5.1. The Impact of WP-MEC Size. In this group of simulations, we will investigate and compare the impact of network size on the optimization performance of four schemes by varying the number of users and MEC servers.

First, we validate the impact of changes in the number of users. Therefore, in this simulation, the number of users is increased from 10 to 90 with the step size of 10, and the number of edge servers is fixed to 5. Figure 4 gives the results of this simulation.

Figure 4(a) shows the comparison of CCR achieved by DEWPT, CoCoRaM, COO, and LCO. From the experimental results, it can be found that the CCR obtained by each scheme decreases with the increase of the number of users. The CCR achieved by DEWPT is the highest among the four schemes compared. Figure 4(b) gives the gains of DEWPT to CoCoRaM. It shows that the CCR gains of DEWPT to CoCoRaM grow with increasing in the number of users. The maximal gain is achieved when the number of users is 70, which is near 3.5%. When the number of users exceeds 70, the CCR gain decreases, but still achieves a gain of at least 2%. The computation capacity of WP-MEC is limited because the number of MEC server is fixed. When the number of users reaches 70, the computation load of offloading users approaches the maximum computation capacity of WP-MEC, and so, the CCR can no longer grow. Therefore, the CCR decreases when the number of users exceeds 70. However, DEWPT outperforms its main competitor, CoCoRaM, in all cases. This shows that DEWPT can find the better WPT time because the same method is used for the inner layer optimization. In addition, Figures 4(c) and 4(d) show the CCR gains of DEWPT to CCO and LCO, respectively. Clearly, DEWPT achieved significant gains to CCO and LCO, with maximum gains approaching 35% and 28%, respectively.

Next, we validate the impact of changes in the number of MEC servers. Hence, in this simulation, the number of edge servers is increased from 4 to 9 with the step size of 1, and the number of users is fixed to 50. Figure 5 gives the results of this simulation.

Figure 5(a) shows that the CCR achieved by DEWPT, CoCoRaM, and COO keep growing with increasing in the number of edge servers. Among them, DEWPT outperforms CoCoRaM and COO. The CCR achieved by LCO does not change. This is expected, as LCO performance is not affected by changing the number of edge servers. Figures 5(b)–5(d) show the gains of DEWPT to the other three competitors, respectively. Clearly, DEWPT has significant positive gains for all three competitors. Likewise, DEWPT is superior to the major competitor CoCoRaM in all cases. It shows that DEWPT can find the better WPT time.

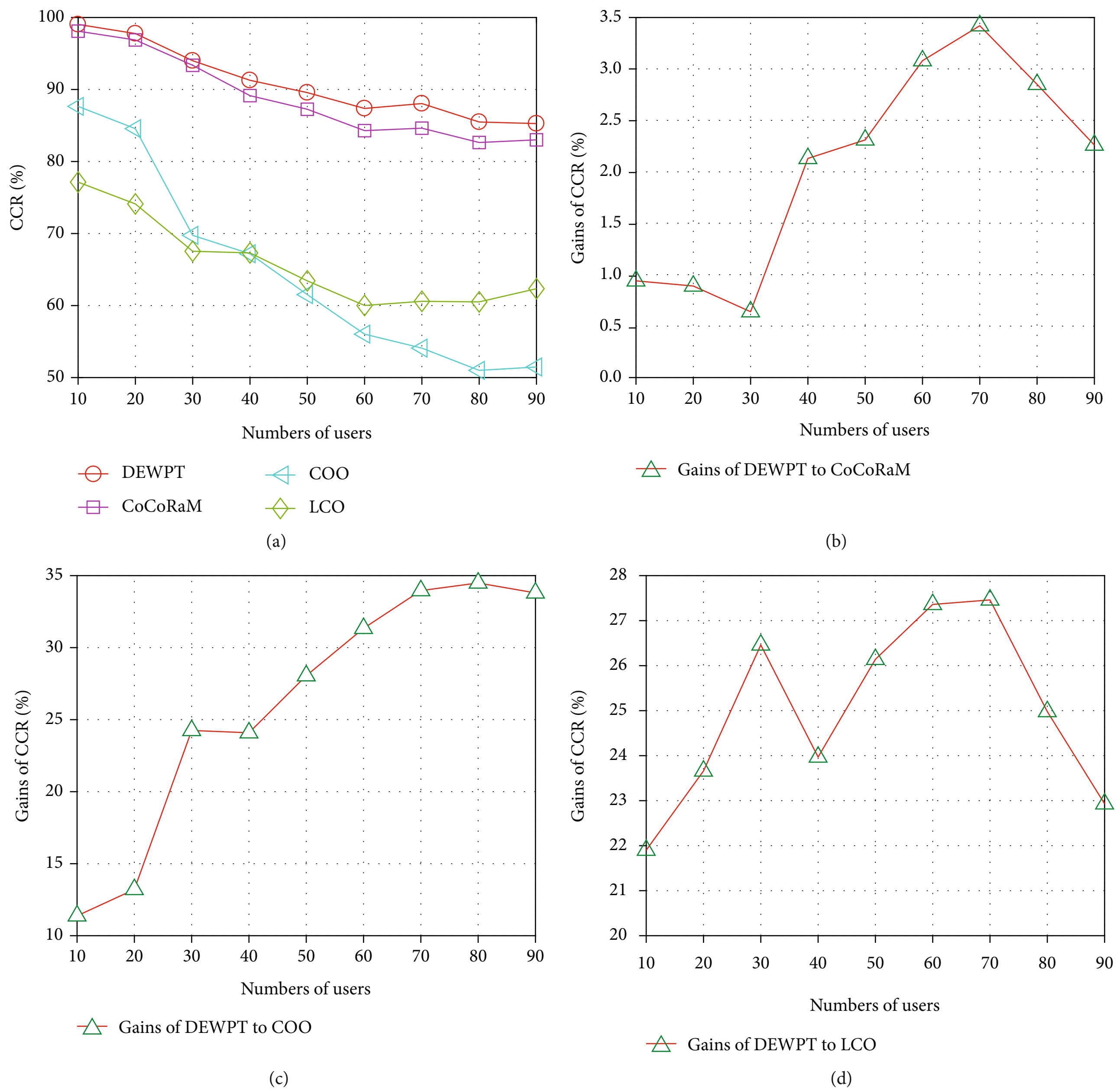

FIGURE 4: The impact of user scale size. (a) CCR comparison. (b) Gains of DEWPT to CoCoRaM. (c) Gains of DEWPT to COO. (d) Gains of DEWPT to LCO.

Based on these simulation results and analysis, it can be concluded that DEWPT has a better optimization performance than CoCoRaM, COO, and LCO.

5.2. Complexity Comparison. In this simulation, we will compare the computation complexity of DEWPT and its major competitor, CoCoRaM. For this purpose, we conduct the simulation under three user configurations (70 users, 80 users, and 90 users) and 5 edge servers. Figure 6 gives the results. The orange, purple, and green curves show the changes in CCR obtained by DEWPT with evolution iterations. The corresponding rectangle represents the CCR obtained by CoCoRaM.

As can be seen from Figure 6, the CCR achieved by CoCoRaM under three user configurations are 84.63%, 83.01%, and 82.64%, respectively. According to the analysis of [16], the size of candidate WPT times set of CoCoRaM is $N + \mathrm{Log}_{(1+\beta)}(T/\Delta t) + 3$. Therefore, the numbers of the candidate WPT time are higher than 70, 80, and 90 under three user configurations. That is, the numbers of WPT time evaluations are greater than 70, 80, and 90, respectively. However, under three user configurations, the CCR obtained by DEWPT exceeds that of CoCoRaM at the 4th, 5th, and 5th iteration, respectively. Because the population size of DEWPT is set to 6, the number of WPT time evaluations

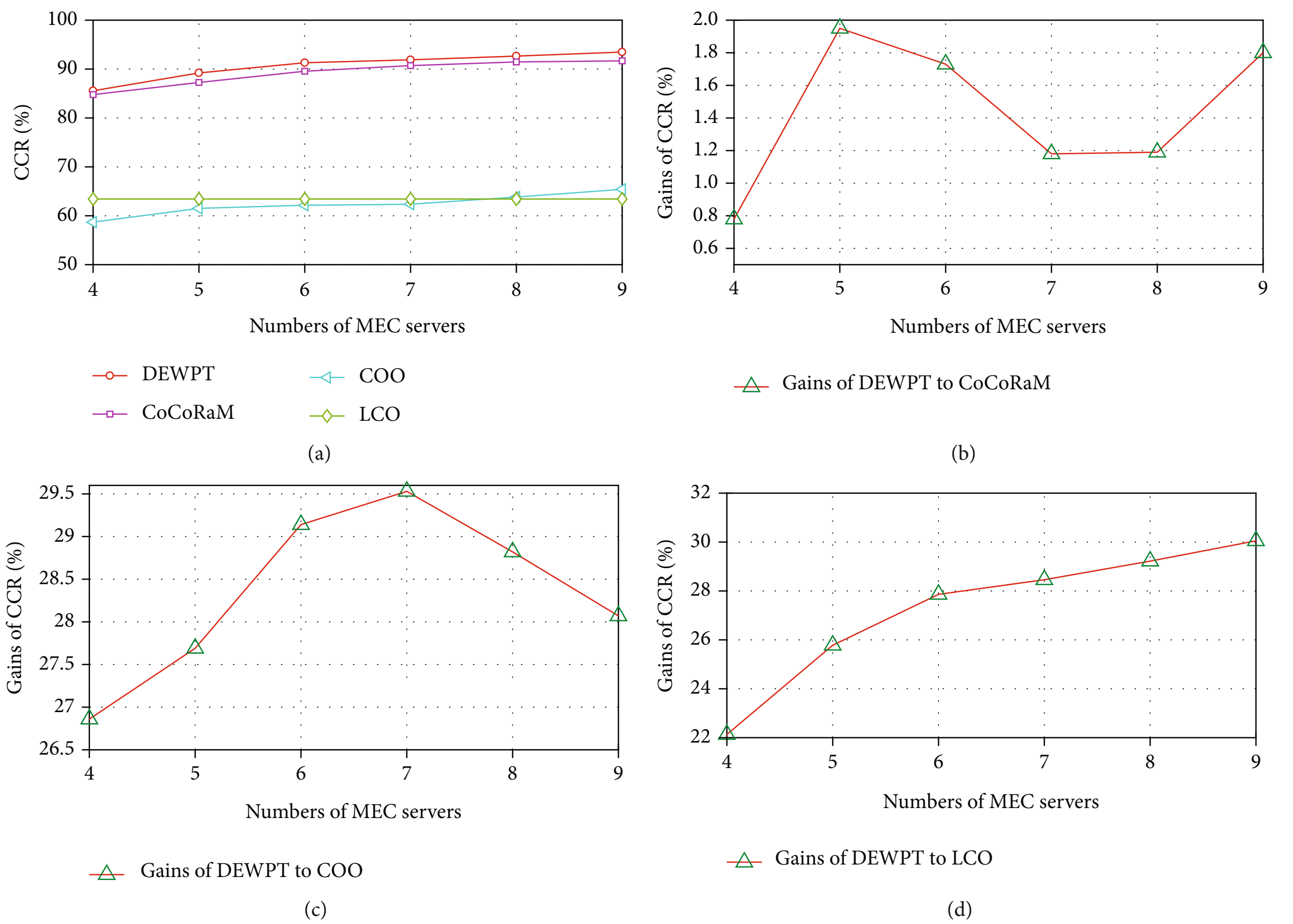

Figure 5: The impact of MEC server scale. (a) CCR comparison. (b) Gains of DEWPT to CoCoRaM. (c) Gains of DEWPT to COO. (d) Gains of DEWPT to LCO.

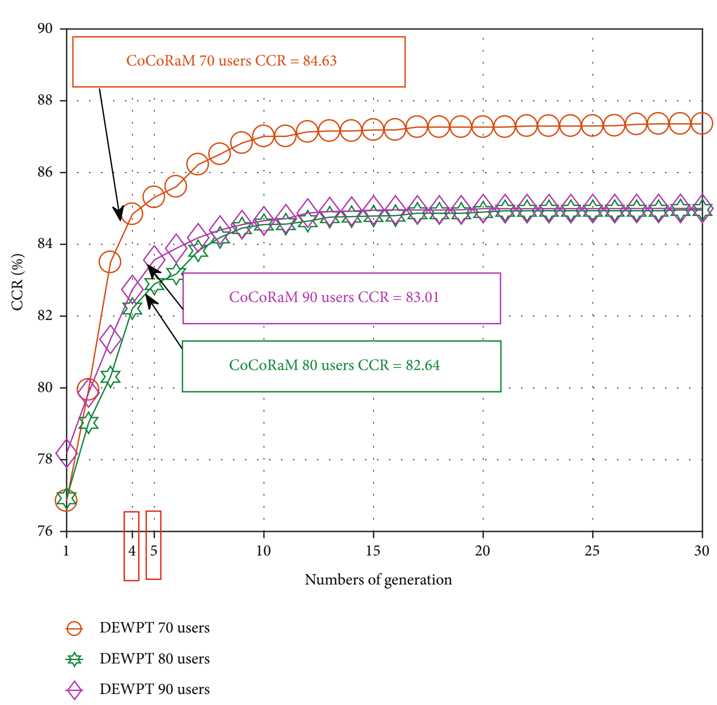

Figure 6: Complexity comparison.

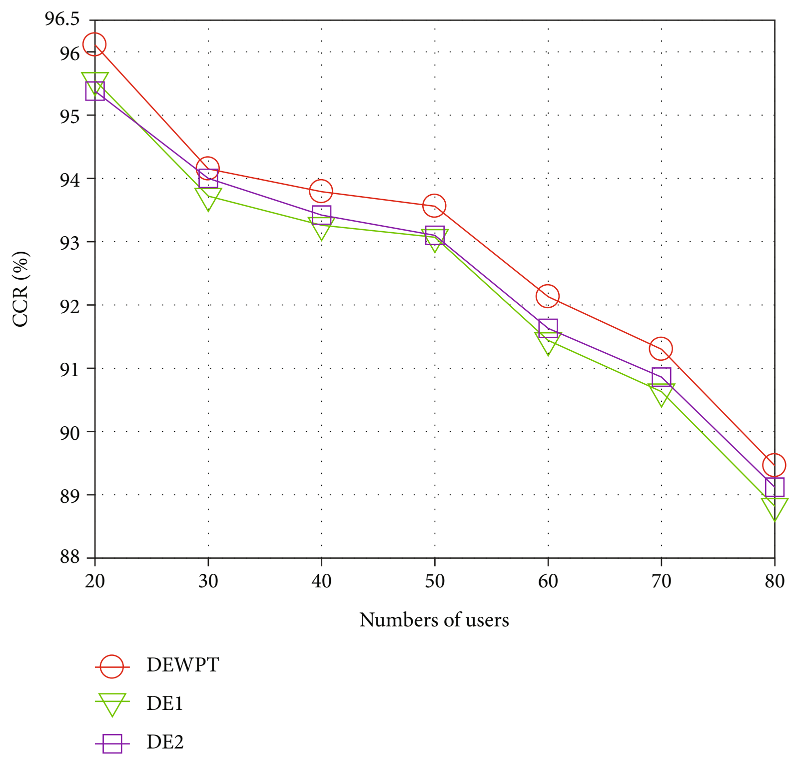

Figure 7: Comparison with the other DE algorithm.

is the product of the population size and the number of iterations, i.e., 24, 30 and 30, respectively. Obviously, DEWPT is much lower than CoCoRaM in terms of the number of WPT time evaluations, only about 1/3.

Thus, it can be concluded that DEWPT outperforms CoCoRaM in terms of the complexity.

5.3. Comparison with Other DE Algorithm. In this simulation, we will compare the DEWPT with the two DE algorithms which are termed as DE1 and DE2. The population size of three algorithms is all set to the same small population, i.e., $NP = 6$. In addition, the number of edge servers is set to 5, and the number of users increases from 20 to 80 with the step size of 10. DE1 only uses the mutation operator DE/rand/1 while DE2 only uses the mutation DE/best/1. At the same time, the traditional binomial crossover operator is used in them. Figure 7 gives the results of this simulation.

From Figure 7, it can be seen that DEWPT achieved a better CCR than DE1 and DE2 under different user numbers. The maximal gain is about 1% under different user numbers, because the optimal decision matrix can theoretically be obtained through the GAP-CS based on dynamic programming. Therefore, although the gain is small, it has shown that DEWPT can find a better WPT time than the other two DE algorithms. Hence, this proves that the hybrid mutation operator and the perturbation-based binomial crossover improve the optimization performance of DEWPT, and DEWPT is a better optimization algorithm for WPT time.

6. Conclusions

In this work, we studied the time allocation problem for WPT in WP-MEC network. A novel differential evolution-based optimization algorithm for WPT is developed, in which the hybrid mutation operator, perturbation-based binomial crossover operator, and the micro-population were designed to improve the algorithm's optimization performance. Then, the overall solving framework DEWPT for the maximization problem of computation completion ratio was constructed by integrating the algorithm. Finally, extensive numerical simulations showed that DEWPT can achieve a higher computation completion ratio than the other three benchmark schemes. This proved that the proposed algorithm can find the better WPT time, and it is an effective optimization scheme of WPT time for WP-MEC network.

A promising future research work is to design a priority-based user offloading order and jointly optimize WPT time allocation and offloading decision. Moreover, extending the work of this paper to more complex scenarios, such as intelligent reflecting surface-aided WP-MEC and nonorthogonal multiple access-assisted WP-MEC systems, is also a work worth studying.

Data Availability

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

Disclosure

Part of this paper has been published by the 2022 International Conference on Computer Communications and Networks (ICCCN 2022).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61961021 and 61961022), the Science and Technology Foundation of Jiangxi Province (nos. 20202BABL202019 and 20202BABL202036), and the Science and Technology Project of Jiangxi Education Department (no. GJJ180251).

References

[1] J. Zhao, L. Yang, M. Xia, and M. Motani, "Unified analysis of coordinated multipoint transmissions in mmWave cellular networks," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12166–12180, 2022.

[2] M. Ma, L. C. Jain, and P. Anderson, "Future trends of virtual, augmented reality, and games for health," in *Virtual, Augmented Reality and Serious Games for Healthcare 1*, pp. 1–6, Springer, 2014.

[3] X. Zhang, H. Gao, M. Guo, G. Li, Y. Liu, and D. Li, "A study on key technologies of unmanned driving," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 4–13, 2016.

[4] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1222–1233, 2013.

[5] N. D. Sarier, "Efficient biometric-based identity management on the blockchain for smart industrial applications," *Pervasive and Mobile Computing*, vol. 71, article 101322, 2021.

[6] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.

[7] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal, "Mobile-edge computing introductory technical white paper," *White paper, mobile-edge computing (MEC) industry initiative*, vol. 29, pp. 854–864, 2014.

[8] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–13, 2017.

[9] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2019.

[10] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[11] F. Zhou and R. Q. Hu, "Computation efficiency maximization in wireless-powered mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3170–3184, 2020.

[12] L. Ji and S. Guo, "Energy-efficient cooperative resource allocation in wireless powered mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4744–4754, 2019.

[13] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2018.

[14] X. Hu, K.-K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2375–2388, 2018.

[15] F. Wang, H. Xing, and J. Xu, "Real-time resource allocation for wireless powered multiuser mobile edge computing with energy and task causality," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 7140–7155, 2020.

[16] T. Zhu, J. Li, Z. Cai, Y. Li, and H. Gao, "Computation scheduling for wireless powered mobile edge computing networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 596–605, Toronto, ON, Canada, 2020.

[17] J. Zhao, X. Sun, Q. Li, and X. Ma, "Edge caching and computation management for real-time Internet of vehicles: an online and distributed approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2183–2197, 2021.

[18] S. Bi, L. Huang, and Y. J. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7519–7537, 2021.

[19] R. Storn and K. V. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[20] Y. Zhang, D. Gong, X. Gao, T. Tian, and X. Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67–85, 2020.

[21] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[22] Y. Wang, C. Chen, P. Huang, and K. Wang, "A new differential evolution algorithm for joint mining decision and resource allocation in a MEC-enabled wireless blockchain network," *Computers & Industrial Engineering*, vol. 155, article 107186, 2021.

[23] Y. Wang, H. Liu, H. Long, Z. Zhang, and S. Yang, "Differential evolution with a new encoding mechanism for optimizing wind farm layout," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1040–1054, 2018.

[24] S. Zhou, L. Xing, X. Zheng, N. du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[25] H. Salehinejad, S. Rahnamayan, and H. R. Tizhoosh, "Micro-differential evolution: diversity enhancement and a comparative study," *Applied Soft Computing*, vol. 52, pp. 812–833, 2017.

[26] H. Salehinejad, S. Rahnamayan, and H. R. Tizhoosh, "Exploration enhancement in ensemble micro-differential evolution,"

in *IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 63–70, Vancouver, BC, Canada, 2016.

[27] C. Brown, Y. Jin, M. Leach, and M. Hodgson, "JADE: adaptive differential evolution with a small population," *Soft Computing*, vol. 20, no. 10, pp. 4111–4120, 2016.

[28] Y. E. Yildiz and A. O. Topal, "Large scale continuous global optimization based on micro differential evolution with local directional search," *Information Sciences*, vol. 477, pp. 533–544, 2019.

[29] M. A. Sotelo-Figueroa, H. J. P. Soberanes, J. M. Carpio, H. J. Fraire Huacuja, L. C. Reyes, and J. A. Soria Alcaraz, "Evolving Bin Packing Heuristic Using Micro-Differential Evolution with Indirect Representation," in *Recent Advances on Hybrid Intelligent Systems, ser. Studies in Computational Intelligence*, vol. 451, pp. 349–359, Springer, 2013.