IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 July 2024, accepted 8 August 2024, date of publication 16 August 2024, date of current version 12 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444607

RESEARCH ARTICLE

A Multi-Objective Evolutionary Approach: Task Offloading and Resource Allocation Using Enhanced Decomposition-Based Algorithm in Mobile Edge Computing

CHUNYANG YU[1], YIBO YONG[1,2], YANG LIU[1], JIAN CHENG[3,4,5], AND QIANG TONG[1]

[1]Software College, Northeastern University, Shenyang 110819, China
[2]Foshan Graduate School of Innovation, Northeastern University, Foshan 528300, China
[3]Research Institute of Mine Artificial Intelligence in Chinese Institute of Coal Science, Beijing 100013, China
[4]State Key Laboratory of Intelligent Coal Mining and Strata Control, Beijing 100013, China
[5]Tiandi Science and Technology Company Ltd., Beijing 100013, China

Corresponding authors: Jian Cheng (jiancheng@tsinghua.org.cn) and Yibo Yong (yongyibo@stumail.neu.edu.cn)

This work was supported in part by the Key Project of Science and Technology Innovation and Entrepreneurship of Tian Di Science and Technology Company Ltd. (No. 2022-TD-ZD004, 2022-2-TD-ZD001), National Natural Science Foundation of China und Grant 62472079, the Key Technologies R&D Program of Liaoning Province (2023JH1/10400082), and Guangdong Basic and Applied Basic Research Foundation (2024A1515012016).

ABSTRACT With the rapid growth of the mobile computing techniques, a wide variety of mobile edge computing (MEC) applications have emerged recently, aiming to provide computationally intensive and delay-sensitive network services. Through MEC, various complex tasks of mobile devices can be offloaded to the edge of network system for execution by edge servers, which greatly reduces the local computing burden. However, how to effectively allocate computational and communication resources in edge-cloud remains a challenging task, especially when multiple mobile users and edge servers are involved. In this paper, we propose a decomposition-based multi-objective optimization algorithm based on estimation-of-distribution models (MOEA/D-EoD) to deal with the task offloading and resource allocation problem in MEC. Especially, considering the features of multi-user and multi-server cloud-edge-end collaboration wireless MEC system, we construct a joint optimization model of task offloading and resource allocation, where limited communication and computational resource constraints are considered. To deal with the optimization model, we design an efficient decomposition-based algorithm, which incorporates two novel estimation-of-distribution models to deal with discrete and continuous decision variables of the problem. Experimental results obtained from benchmark test suites DTLZ and ZDT demonstrate that the proposed method exhibits significantly superior performance compared to other comparative algorithms. The proposed model and algorithm are simulated and tested on different test instances, and experimental results show the effectiveness and efficiency of our proposed method.

INDEX TERMS Mobile edge computing, task offloading, resource allocation, multi-objective optimization, estimation-of-distribution model.

I. INTRODUCTION

The swift advancement of mobile computing technologies, such as 5G networks [1], [2], a wide variety of mobile devices, including smartphones, mobile robots and wearable devices [2], have emerged explosively, which also leads to a variety of computationally expensive intelligent applications [3], [4], such as natural language processing [5], virtual reality [6], big data analytic [7], face recognition [8], and ultra-high-definition video [9]. Such applications usually

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

involve stringent latency and computation requirements, but their mobile devices often grapple with resource constraints, characterized by limited processing power and modest battery capacity. Thus, it is rather difficult to deal with these computationally expensive and latency-sensitive tasks. Moreover, the explosive demand for massive data computing poses a big challenge to traditional cloud computing paradigm. The concentricity of cloud resources would incur many serious issues, such as high communication latency, network instability and low bandwidth, which is contrary to the essential requirements of emerging applications, and then greatly hinders the development of these applications. A natural way is to tackle the complex tasks where data is generated, i.e., sinking the capabilities of cloud computing to the network edge side where massive data is generated.

Accordingly, MEC [10] is developed based on the above basis, which offloads the computation-intensive and latency-sensitive tasks from mobile users burdened by resource constraints to nearby edge servers with rich resources. As a complement to cloud computing, MEC focuses on the intelligence at the network edge, which is able to play a key role in small-scale, real-time intelligent analytic. The core concept of MEC is to place a server at the near end of the base station connected to the backhaul network, which enables the efficient processing of data generated at the network edge, obviating the necessity for its transmission to the central cloud infrastructure, which substantially diminishes transmission latency and alleviates the demands on the backhaul network [11]. In fact, the MEC server is owned by the network operator, and mobile users need to pay a fee for processing tasks with the help of edge servers [12].

Through the offloading of latency-sensitive and compute-intensive tasks from local devices to nearby edge servers, MEC is able to reduce the latency and energy consumption and increase the service quality of service. However in fact, the process of the task offloading still relies on reliable wireless communication between mobile devices and edge servers, which generates additional energy consumption and latency, and then results in undesirable network congestion and even paralysis during data transmission. Moreover, compared to cloud servers, edge servers suffers from the limitation of computing resources in dealing with various compute-intensive tasks at the same time [13], which also limits the development of MEC. Therefore, the efficient offloading of computing tasks and the efficient allocation of execution resources have become the key to ensure the effectiveness and efficiency of the operations in the MEC system [14].

Currently, joint decision-making of task offloading and resource allocation within multi-user MEC system with consideration of interference have not been well studied [15], [16], where multiple optimization targets need to be considered simultaneously [17], [18], [19], [20], [21], [22]. In response to the imperative of efficiently allocating the limited computing and communication resources in the MEC system, a joint task offloading and resource allocation model for multi-user and multi-server MEC networks is proposed, which can improve network performance and bring better service experience to mobile users. The primary goal of the model is to minimize energy consumption, latency and cost of MEC. More specifically, the proposed model consists of the following parts: (i) Where should tasks of mobile devices be performed? That is, how to determine whether the decision mobile users' tasks are placed locally or on the edge cloud, and on which edge servers; (ii) If the task decides to offload, how is the uplink transmission power during the offloading process determined? How are the wireless channels required for transmission allocated? (iii) If the task decides to offload, how can the edge cloud rationally allocate compute resources?

Existing research has not adequately investigated the task offloading and resource allocation problem in multi-user MEC systems. To enhance network performance and provide a superior service experience for mobile users, a joint optimization model that simultaneously considers energy consumption, delay and cost is required. To deal with the joint optimization model of task offloading and resource allocation, we design an efficient decomposition-based algorithm based on dual estimation-of-distribution (EoD) models, termed as MOEA/D-EoD. The proposed model and algorithm aim to address the optimization problem under limited communication and computing resource constraints, while handling both discrete and continuous decision variables to find representative solutions on the Pareto front of the problem. Then, to deal with such mixed-variable joint optimization model, an EoD model combining histogram estimation and kernel density estimation algorithms is proposed for sampling points on continuous variables to deal with continuous decision variables, while an EoD model is designed to sample points on the discrete one, aiming to deal with the discrete decision variables. These two models are incorporated into a decomposition-based framework to solve each subproblem of the joint optimization model. In this way, representative solutions on the Pareto front of the problem can be obtained. Through simulations and testing of different test instances, the experimental results demonstrate significant improvements in both effectiveness and efficiency of the proposed method.

Our contributions are as follows.

- A joint optimization model of task offloading and resource allocation for multi-user and multi-server MEC system, including task placement decisions, transmission power control, communication resource allocation, and computational resource allocation, is proposed.
- An effective scheme is proposed to minimize the 3-objective optimization problem, which include the energy consumption, latency and price cost, while ensuring excellent resource utilization and flexibility of the MEC system. More specifically, we design an efficient MOEA/D-EoD algorithm.

- Comparative experiments conducted between our proposed algorithm and other existing algorithms have consistently revealed that our task offloading and resource allocation strategy leads to a substantial enhancement in performance.

II. RELATED WORK

To address the limitations about constrained computing power and battery capacity in mobile devices, mobile cloud computing (MCC) systems are first proposed and developed [23]. Accordingly, MCC can alleviate the two challenging issues encountered by resource-poor mobile devices, i.e., two major mobile computing deficiencies of resource-poor mobile devices and battery limitations. Mobile devices can transfer compute-intensive tasks over the network to run in remote cloud data centers to relieve the pressure locally [24]. Ravi et al [25] introduces a multi-criteria decision offloading method and a fuzzy switching strategy to diminish the local energy consumption of mobile devices and to improve the service availability. Ravi et al. [26] focus on the additional energy consumption generated during communication by proposing a new framework to diminish the communication overhead. Sanaei et al. [27] proposed a service-based arbitration multilayer infrastructure to minimize offloading latency. Chen et al. [28] simultaneously considered energy consumption and maximum delay to model the problem as a mixed-integer problem and put forth an efficient scheme to solve it. However, cloud data centers are frequently situated at a considerable distance from mobile devices, and the network conditions can have a significant impact on computing tasks. The latency that accompanies the data transfer process may result in latency-sensitive tasks not being completed successfully, or the transfer energy consumption may be too high for the mobile device's battery storage to support.

To address the shortcomings of MCC, many researchers have started to study MEC systems with servers located near mobile devices. MEC servers typically represent compact data centers deployed by cloud operators and telecom operators. Multi-user MEC systems are mainly classified as centralized and distributed. For centralized MEC systems, Chang et al. [29] devised a threshold-based optimal resource allocation strategy aimed at minimizing the local energy consumption to extend the device battery life. Reference [30] provides optimal resource allocation schemes to minimize system latency for three models: local, edge cloud and partial compression offloading, respectively. Liu et al. [31] jointly considered the influence of transmission power control on energy consumption and latency in the MEC system while implementing a task offloading strategy and proposed an efficient semi-distributed algorithm for solving it. For distributed MEC systems, an iterative algorithm is developed in [32] to minimize the overall user energy consumption based on the continuous convex approximation technique. Both [33] and [34] concentrate on the impact of energy usage and delay in MEC systems to design offloading strategies.

Efficient and fair resource allocation to meet multi-objective demands is a crucial challenge in cloud computing, vehicular cloud computing and edge computing. Feng et al. [35] formulated the resource allocation problem in cloud computing as a multi-objective optimization problem considering overall task execution time, resource reservation, and quality of service (QoS) for each task. They proposed a Pareto-dominated particle swarm optimization algorithm to search for the multi-objective optimal solution. In 2022, Wei et al. [36] addressed the resource allocation problem in vehicular cloud computing by proposing an improved NSGA-II algorithm, effectively optimizing resource allocation schemes. In the same year, Apinaya Prethi [37] tackled the resource management and task scheduling problem in the edge layer by proposing a multi-objective Krill Herd optimization algorithm. Their approach achieved optimized resource allocation and task scheduling during VM migration, enhancing the lifecycle of fog-edge networks. In 2021, Xue et al. [38] proposed a joint optimization strategy for task offloading and resource allocation to maximize system processing capacity. They decomposed the problem into three subproblems: resource allocation, task allocation and sub-channel allocation. In 2024, Umer et al. [39] proposed a multi-objective task-aware offloading and scheduling framework (MTOSF) for the IoT logistics domain. The framework prioritizes delay-sensitive tasks and compute-intensive tasks, employing a priority-based offloader for classification.

Multi-objective optimization algorithm has emerged as a prominent research area in recent years [40], [41], [42], [43], [44], aiming to simultaneously satisfy multiple performance metrics such as latency, throughput and energy consumption. To tackle the intricate resource allocation challenges, researchers have employed a diverse range of advanced multi-objective optimization algorithms, seeking optimal or near-optimal resource allocation schemes while considering multiple objectives. Existing approaches for task offloading and resource allocation with discrete and continuous mixed variables exhibit limitations. To address this challenge, we propose an EoD-based MOEA/D algorithm that effectively tackles the optimization problem involving discrete and continuous mixed variables. In summary, task offloading and resource allocation domain continues to face numerous challenges and opportunities. As technology advances and application demands grow, research in this field will continue to deepen, providing more efficient and reliable solutions for practical cloud computing applications.

III. MODELS

As illustrated in Fig. 1, the typical MEC system comprises a single macro base station (MBS), M wireless small base stations (SBSs) and N mobile users, where each user u_i has a pending computational task T_i that is computational intensive or delay-sensitive. The set of users can be designated as $U = \{u_1, u_2, \cdots, u_N\}$. The computational task T_i is composed of three variables, denoted by $T_i = (s_i, c_i, t_i), i \in N$,

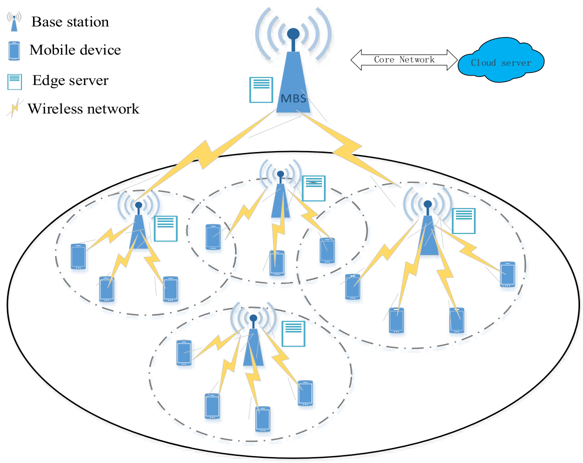

FIGURE 1. Depiction of the multi-user MEC system in a multi-channel wireless environment.

where s_i represents the offloaded data size of this task T_i, c_i and t_i represent the cumulative count of the required CPU cycles and the maximum permissible delay in accomplishing the task, respectively. Each BS including MBS and SBSs is associated with an edge server that can be used to deal with the computational tasks offloaded by the mobile users. Let $BS = \{bs_0, bs_1, bs_2, \cdots, bs_N\}$ denote the set of BSs, where bs_0 represents the MBS. In addition, MBS covers all the SBSs and it is connected to the cloud server via the core network. Since edge servers are closer to mobile devices, they can well tackle the delay-sensitive tasks of mobile users at the same time greatly decrease the total delay. Cloud servers have abundant computing power and storage space and can serve more mobile devices than edge servers, and thus they can better handle computational intensive and delay-tolerant tasks and highly reduce total energy consumption.

The communication state, computation state and the price cost are the three key factors to affect the offloading and resource allocation decisions. Thus, these three models are presented in detail separately. The important parameters are listed in Table 1.

A. COMMUNICATION MODEL

During the wireless transmission in the MEC system, the same base station can access multiple mobile devices, which may lead to transmission interference and decrease the reliability of the system [45]. To mitigate mutual interference during the transmission process within the same base station, the wireless channel bandwidth B used for transmission can be equally divided in to a set of α sub-channels, with each sub-channel's size denoted as $W = B/\alpha$. In order to maintain the orthogonality of wireless transmission across mobile users within the same BS, an orthogonal sub-channel is assigned to each mobile user.

Based on the above mentioned, each SBS can serve up to α mobile users simultaneously. Each accessed mobile device uses a separate wireless sub-channel. In the MEC system,

TABLE 1. Parameters.

Parameter	Definition
N	Number of users of mobile devices
M	Number of base stations
$U=\{u_1,u_2,\cdots,u_N\}$	Set of mobile users
$BS=\{bs_0,bs_1,bs_2,\cdots,bs_M\}$	Set of BSs where bs_0 denotes the MBS
T_n	The pending computational task of user u_n
s_n	Data size for the offloading of task T_n
c_n	Cumulative count of CPU cycles required of task T_n
t_n	Maximum permissible delay of task T_n
B	Transmission bandwidth of mobile users
α	Number of divided sub-channels
W	Size of divided sub-channels
x_n	Offloaded decision variable of user u_n
$X=\{x_n,n\in U\}$	Offloading decision vector
$\Lambda=\{\eta_n^m,n\in U,m\in BS\}$	Wireless channels allocation profile
η_n^m	User u_n chooses offloaded target base station bs_m
$p_{n,m}$	Transmission power of user u_n offloading tasks to the base station bs_m
$h_{n,m}$	Channel gain between the user u_n and the target base station bs_m
$F^L=\{f_1^L,f_2^L,\cdots,f_N^L\}$	The CPU frequency of mobile users
κ	The hardware architecture of the mobile device
λ_n^m	The proportion of computing resources allocated to user u_n by bs_m
$F^C=\{f_0^C,f_1^C,f_2^C,\cdots,f_M^C\}$	The computing power provided by the edge cloud to mobile users
d_m	The per price of edge server carried by base station bs_m
$r(x_n)$	The actual cost price of user u_n
$e(x_n)$	The actual total energy consumption of user u_n
$t(x_n)$	The actual total time delay of mobile user u_n

mobile users can decide to handle their tasks locally or offload onto a nearby edge server (or cloud server) for execution via the MBS. Let $X = \{x_n, n \in U\}$ denote the offloading decision vector and $x_n = \{0, 1, 2\}$ represent the offloaded decision variable of user u_n. If the mobile user u_n chooses to offload to edge server through nearby SBS or MBS, that is to say $x_n = 1$, the sub-channel used for transmission will be assigned to u_n via the base station bs_m, where the mobile user selects the destination BS for offloading. For simplicity, the target base station bs_m selected by the mobile user u_n can also be denoted as η_n^m. Thus, we have the wireless channels allocation profile $\Lambda = \{\eta_n^m, n \in U, m \in BS\}$ which satisfies $\sum_{n=1}^{N} \eta_n^m \leq \alpha, \forall m = 0, 1, \ldots, M$. Here, $x_n = 2$ means that the mobile user chooses to offload to cloud server for execution via the MBS. $x_n = 0$ means that the mobile user chooses to process the task locally.

1) LOCAL COMPUTING

If $x_n = 0$, it means that the mobile user u_n decides to perform the computational task locally, and the communication resources are not allocated to this mobile user

u_n. Accordingly, both transmission delay and transmission energy consumption of this mobile user u_n are 0.

2) MOBILE EDGE COMPUTING

For mobile users that offload tasks to MBS, which satisfy $\eta_n^0 = 1$, the noise power only involves the background noise power, rather than the inter-cell interference power. For mobile users that offload tasks to SBSs, although those users connected to the same SBS through different orthogonal sub-channels can effectively reduce the mutual interference within the base station, they are still affected by the interference among different base stations. Let $p_{n,m}$ denote the uplink transmission power of user u_n that offloads tasks to the edge server. Then, the user uplink data rate $r_{n,m}^{ul}$ is defined as follows

$$r_{n,m}^{ul} = \begin{cases} \frac{B}{\alpha} \times \log_2\left(1 + \frac{p_{n,m} \times h_{n,m}}{\sigma^2}\right) \\ m = 0 \\ \frac{B}{\alpha} \times \log_2\left(1 + \frac{p_{n,m} \times h_{n,m}}{\sigma^2 + \sum\limits_{j \in BS\setminus\{m\}} \sum_{i \in U} x_i \times p_{i,j} \times h_{i,j}}\right) \\ m = 1, 2, \ldots, M \end{cases} \tag{1}$$

where σ^2 denotes the background noise power, $\sum_{j \in BS\setminus\{m\}} \sum_{i \in U} x_i p_{i,j} h_{i,j}$ denotes the inter-cell interference power, $h_{n,m}$ represents the channel gain between the user u_n and the destination base station bs_m, which considers the effects of path loss, antenna gain and shadowing.

Subsequently, the transmission delay of mobile user u_n during the offloading process is formulated as

$$t_n^t = \frac{s_n}{r_{n,m}^{ul}} \tag{2}$$

Similarly, we can further gain the access to the transmission energy consumption of user u_n by

$$\varepsilon_n^t = \frac{p_{n,m} \times s_n}{r_{n,m}^{ul}} \tag{3}$$

3) CLOUD COMPUTING

The transmission delay of mobile user u_n, who offloads tasks to the cloud server, is the aggregate of the delay associated with offloading to MBS and the delay of transmission in the core network. It is formulated as

$$t_{n,Cloud}^t = \frac{s_n}{r_{n,0}^{ul}} + \frac{s_n}{r_{n,core}} \tag{4}$$

where $r_{n,core}$ is the user uplink data rate in the core network.

In our work, the transmission energy consumption of the core network is ignored. Then the transmission energy consumption $\varepsilon_{n,Cloud}^t$ of the user u_n offloading tasks to the cloud server can be defined as

$$\varepsilon_{n,Cloud}^t = \frac{p_{n,0} \times s_n}{r_{n,0}^{ul}} \tag{5}$$

In real scenarios, the distance from the mobile users to cloud server is usually very large, that is, $r_{n,core} \ll r_{n,m}^{ul}$, so $t_{n,Cloud}^t \gg t_n^t$.

Since the computation result size of the user task is typically considerably smaller than the amount of data at the time of offloading, we ignore the transmission delay of the returned processing result. It is important to highlight that our algorithm can still be used for offloading and resource allocation when the returned latency cannot be omitted.

B. COMPUTATION MODEL

In the MEC system, the mobile users have the discretion to determine whether to carry out their tasks locally or offload them to the edge server, which is influenced by local hardware capabilities, the prevailing network conditions, and the cost of renting an edge server. This section delves into a detailed examination of the execution latency and energy consumption entailed in task execution.

1) LOCAL COMPUTING

When $x_n = 0$, it implies that the mobile user u_n decides to perform the computational task locally, and accordingly the system must take into account both the time delay of completing the task and the energy consumed for the local processing. We assume that the local processing power of mobile devices, denoted by $F^L = \{f_1^L, f_2^L, \cdots, f_N^L\}$, is differentiated. Let $f_n^L \in F^L$ denote the computation capability of user u_n. Then, the local execution time for computation task T_n is defined as follows

$$t_n^{exe,L} = \frac{c_n}{f_n^L} \tag{6}$$

In MEC system, the CPU clock frequency is about linearly related to the voltage supply under low voltage limits [23], [27]. Thus, the energy consumption associated with a CPU cycle can be symbolized as $\kappa(f_n^L)^2$, where κ reflects a coefficient linked to the hardware architecture. As a result, The following formula may be used to calculate the energy used locally while computation job T_n is being executed

$$\varepsilon_n^{exe,L} = \kappa(f_n^L)^2 \times c_n \tag{7}$$

2) MOBILE EDGE COMPUTING

In MEC system, there is variability in the computing power of different edge servers. Let $F^C = \{f_0^C, f_1^C, f_2^C, \cdots, f_M^C\}$ denote the computing power provided by the edge cloud to mobile users for processing tasks. The edge cloud execution delay and energy consumption can be calculated by, respectively:

$$t_n^{exe,C} = \frac{c_n}{\lambda_n^m \times f_m^C} \tag{8}$$

and

$$\varepsilon_n^C = \frac{p_{n,m}^C \times c_n}{\lambda_n^m \times f_m^C} \tag{9}$$

where λ_n^m denotes the proportion of computing resources allocated to mobile device u_n via the offload destination base station bs_m, which satisfies $\sum_{n=1}^{N} \lambda_n^m \leq 1, \forall m = 0, 1, \ldots, M$, and $p_{n,m}^C$ represents the energy consumption per cycle of the edge server of SBS bs_m to process task of mobile users u_n.

3) CLOUD COMPUTING

When x_n=2, the cloud execution delay $t_n^{exe,Cloud}$ can be calculated by

$$t_n^{exe,Cloud} = \frac{c_n}{f_{Cloud}^C} \tag{10}$$

where f_{Cloud}^C is the computing power provided by the cloud to mobile users for processing tasks, which usually satisfies $f_{Cloud}^C \gg f_m^C \gg f_n^L$.

In the MEC system, our primary focus centers on assessing the energy depletion of mobile devices and edge servers. At this juncture, we do not calculate the energy consumption entailed in processing tasks by the cloud server; this aspect will be explored in our future research endeavors.

C. COST MODEL

Edge servers and cloud servers are not free to mobile devices, while users will need to pay a fee to the edge cloud provider or cloud provider when they decide to offload. We make the assumption that the cost per unit of the edge cloud server owned by the base station bs_m is d_m and the per price of cloud server is $2 \times d_m$. This actual cost $r(x_n)$ is related to the computing resources allocated by the edge server bs_m or cloud server to that mobile user u_n, as shown below:

$$r(x_n) = \begin{cases} 0 & x_n = 0 \\ \lambda_n^m \times d_m & x_n = 1 \\ 2 \times d_m & x_n = 2 \end{cases} \tag{11}$$

IV. PROBLEM FORMULATION

We present a joint optimization scheme of MEC that consists of task offloading, transmission power control, and resource allocation, where the resource allocation involves both communication resource allocation and computation resource allocation. We commence by formulating a multi-objective optimization model, accompanied by a set of constraints. Following this, we develop a multi-objective optimization algorithm designed to resolve this
problem.

A. JOINT OPTIMIZATION MODEL

In real MEC system, it is necessary to consider a set of constraints, such as the limited communication and computing resources at the edge side, and the limited battery capacity and computing power at the user side in the MEC system. Accordingly, we in this work consider the following three issues:

(1) How to Offload Tasks. Our task involves ascertaining whether the tasks of mobile users should undergo local execution or be offloaded to a nearby edge server or the cloud server. A discrete decision problem might be employed to describe this situation. In addition, it is also needed to identify the destination base station for the offloaded task.

(2) How to Control Transmission Power. When a mobile user opts to offload its tasks, we are faced with the task of establishing how the mobile device determines the suitable transmission power during offloading.

(3) How to Allocate Resources. For the mobile user deciding to offload its tasks, we need to establish the procedures by which the destination base station allocates communication resources and the corresponding edge cloud server allocates computation resources.

The actual total energy consumption $e(x_n)$ of mobile user u_n consists of energy consumption of transmission and computation, which can be expressed as

$$e(x_n) = \begin{cases} \varepsilon_n^{exe,L} & x_n = 0 \\ \varepsilon_n^t + \varepsilon_n^C & x_n = 1 \\ \varepsilon_{n,Cloud}^t & x_n = 2 \end{cases} \tag{12}$$

Similarly, the actual total delay $t(x_n)$ of mobile user u_n includes transmission delay of mobile user u_n and computing delay, which can be defined as

$$t(x_n) = \begin{cases} t_n^{exe,L} & x_n = 0 \\ t_n^t + t_n^{exe,C} & x_n = 1 \\ t_{n,Cloud}^t + t_n^{exe,Cloud} & x_n = 2 \end{cases} \tag{13}$$

The optimization problem in MEC system is multi-objective optimization problem, characterized by multiple objectives. These objectives encompass minimizing mobile device energy consumption, task delay, and the total cost while adhering to defined constraints, which is defined as follows

$$\begin{aligned}
&\textit{minimize} \quad \mathrm{E} = \sum_{i=1}^{N} e(x_i) && \text{(14a)}\\
&\textit{minimize} \quad \mathrm{T} = \sum_{i=1}^{N} t(x_i) && \text{(14b)}\\
&\textit{minimize} \quad R = \sum_{i=1}^{N} r(x_i) && \text{(14c)}\\
&\textit{subject to} \quad x_i \in \{0, 1, 2\},\ \forall i = 1, \ldots, N. && \text{(14d)}\\
&\qquad 0 \leq p_{i,j} \leq p_i^{\max},\ \forall i = 1, \ldots, N,\\
&\qquad \forall j = 1, \ldots, M. && \text{(14e)}\\
&\qquad p_{i,j} = 0,\ \forall x_i = 0. && \text{(14f)}\\
&\qquad D_i \leq t_i,\ \forall i = 1, \ldots, N. && \text{(14g)}\\
&\qquad \sum_{i=1}^{N} \lambda_i^j \leq 1,\ \forall j = 1, \ldots, M. && \text{(14h)}
\end{aligned}$$

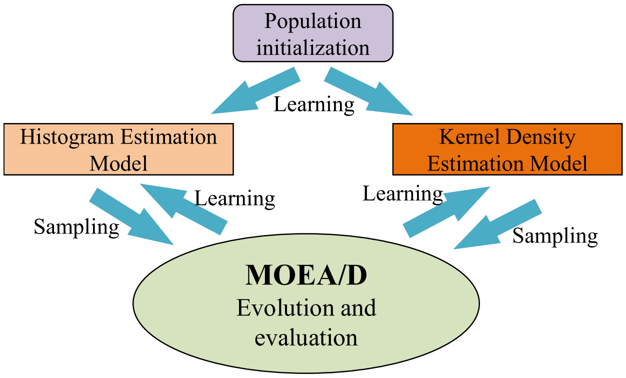

FIGURE 2. Framework of MOEA/D-EoD.

$$\sum_{i=1}^{N} \eta_i^j \leq \alpha, \ \forall j = 1, \ldots, M. \tag{14i}$$

$$\sum_{i=1}^{N} \eta_i^0 + \sum_{i=1}^{N} x_i \leq \alpha, \ \forall (\eta_i^0 = 1 \vee x_i = 2). \tag{14j}$$

The first objective function (Eq. 14a) aims to minimize the total energy consumption of all the mobile devices. The second objective function (Eq. 14b) aims to minimize the total delay of all the tasks, including the transmission delay and computing delay. Similar to the above, the third objective function (Eq. 14c) is to minimize the total cost of all mobile users.

In the above objective functions, there are a set of constraints. The constraint Eq. 14d shows that each task can either be executed locally or offloaded onto any of the edge servers or cloud server. The two constraints Eq. 14e and Eq. 14f offer the upper and lower limits of the transmitted power during the task offloading, where $p_i^{\max}$ denotes the maximum acceptable transmission power for mobile user u_n. No transmission power is required when the task is decided to be executed locally. The constraint Eq. 14g ensures that the completion time delay for computation task T_n cannot exceed its given upper limit t_n. The constraint Eq. 14h indicates that any edge server cannot allocate more computing resources to the associated mobile devices than its computation capacity. The constraint Eq. 14i implies that each SBS can only serve the maximum of α mobile devices. Then, the constraint Eq. 14j means that the MBS and cloud server can serve the maximum of α mobile devices together.

Mathematically, the above optimization problem is a mixed integer nonlinear program, which is difficult to solve due to its exponential complexity. To deal with this complex problem, we convert it into a hybrid coding problem, and use the multi-objective evolutionary optimization strategy [14] to solve this joint optimization problem. With the continuous development and improvement of multi-objective evolutionary optimization algorithms [10], these algorithms are also applied as a means of policy solving in mobile edge computing.

$x_n \in \{0\} \cup SC$	$\eta_n^m \in \{0\} \cup BS$	$\lambda_n^m \in [0,1]$	$p_{n,m} \in \left[0, p_n^{\max}\right]$

FIGURE 3. Gene string by hybrid coding.

B. ALGORITHM DESIGN

The problem studied is a multi-objective mixed-variable optimization problem, and to address it, we develop an efficient decomposition-based algorithm based on dual estimation-of-distribution (EoD) models, termed as MOEA/D-EoD, which uses MOEA/D as the basic algorithmic framework and combines two novel EoD learning models as reproduction operators for generating offspring. Fig. 2 presents the framework of MOEA/D-EoD. This algorithm employs a hybrid Individual generation strategy, utilizing both histogram estimation and kernel density estimation models to learn discrete and continuous variable knowledge, respectively. Subsequently, it samples new individuals based on the acquired knowledge and inputs them into MOEA/D for further evolution and evaluation.

1) HYBRID CODING IMPLEMENTATION

The genes in MOEA/D-EoD represent the task offloading strategy, the destination base station, the proportion of computational resources allocated to the destination base station, and the task transmission power, respectively. A complete individual comprises four genes, collectively representing a solution. Since the four decision variables are of diverse types, we use mixed coding. First of all, due to the division of each wireless channel into α sub-channels of equal size, we denote the set of sub-channels of the target base station bs_m used for task transmission as $SC_m = \{1, \cdots, \alpha\}$, and the set of sub-channels of all base stations is denoted by $SC = \{SC_1, SC_2 \cdots, SC_M\}$. In order to simplify the coding, we concurrently take into account the selection of sub-channels during the task offloading process. Because the mobile user's offloading decision is a binary variable, for encoding purposes, we consolidate the offloading decision and channel selection into a integer representation, denoted by $x_n = \{0\} \cup SC$, $x_n = 0$ implies that the task is executed locally. Next, we discuss the second coding segment. η_n^m represents the destination base station bs_m selected by the mobile user u_n when deciding to unload, and is the same as the first coding segment, which is also an integer variable in the range of $\{0, 1, \cdots, M\}$. Thirdly the computational resource λ_n^m allocated to the destination base station is a continuous variable taking values in [0, 1]. Finally, the corresponding transmission power variable $p_{n,m}$ is a floating point variable in $[0, p_n^{\max}]$. The gene string for the hybrid encoding comprises various types of variables, and its encoding is illustrated in Fig. 3.

2) FITNESS FUNCTIONS

The fitness functions are the criteria employed to assess the quality of individuals within a population and are expressed through formulas (14a), (14b), and (14c). The constraints are delineated by (14e) through (14j).

3) OVERALL FRAMEWORK

The primary framework of MOEA/D-EoD is illustrated in Algorithm 1. 1) Initialization: The initial step involves the initialization of N_{pop} reference vectors (line 1 of Algorithm 1), which are uniformly distributed. Here, N_{pop} denotes the size of the population. Subsequently, the neighborhood of each subproblem is established by considering the T neighbors associated with each reference vector (line 2 of Algorithm 1). Following this, the indices of both continuous and discrete decision variables are determined, guided by the problem's characteristics (line 3 of Algorithm 1). Based on these indices, an initial population P of size N_{pop} is generated through random initialization (line 4 of Algorithm 1). Subsequently, all individuals within population P undergo evaluation via constraint functions. In cases where the constraints are not met, appropriate remedies are applied for constraint satisfaction. Following the constraint-related adjustments, the ideal point z^* is computed (line 5 of Algorithm 1). 2) Reproduction: Following this step, the process of reproduction ensues. Initially, for each subproblem, we identify all individuals in the corresponding neighborhood, referred to as *archive*. Subsequently, adhering to the predefined number of bins, we proceed as follows: for continuous variables, we partition the bins through clustering and then create the histogram model denoted as m_c, along with estimating the probability density function for the data within each bin using kernel density estimation. For discrete variables, we employ an incremental learning approach to construct the histogram-based probabilistic model, denoted as m_d (Algorithm 1, lines 9-11). These models are influenced by the individuals in the *archive* and the indices of both continuous and discrete variables. Through sampling from these probability models, a novel individual, denoted as x_{new}, is generated (line 12 of Algorithm 1). Notably, in the case of continuous variables of x_{new}, a bin k is determined by a random number r. Subsequently, continuous variables are generated in the interval $[x_{j,k}, x_{j,k+1}]$ based on the estimated probability density function. For a discrete variable, an available value d from $\{0, 1, \cdots, x_i^u\}$ is chosen by a probability that is produced randomly. Subsequently, x_{new} is subjected to constraint-based evaluation, and if any constraints are breached, remedial measures are implemented (line 13 of Algorithm 1). This process concludes with the subsequent updating of the ideal point z^* (line 14 of Algorithm 1). 3) Environmental selection: Utilizing the scalarization function g^* (such as Chebyshev, PBI, etc.), the individuals within the present neighborhood, along with the newly generated individual x_{new}, are transformed into scalar values. Subsequently, a comparison is made between x_{new} and other individuals within the present neighborhood, and retain those individuals who perform better (line 15-19 of Algorithm 1).

Algorithm 1 MOEA/D-EoD

Input:
N_{pop} : Population size
N_{gen} : Number of iterations
T: Size of neighborhood
K: Size of clusters
Output:
P: Final population
1. Initializing N_{pop} weight vectors $(w_1, w_2, \cdots, w_{N_{pop}})$;
2. Determine the set of neighbors B for each weight vector;
3. According to the coding strategy, the index of continuous variables s_c and the index of discrete variables s_d are obtained;
4. P = Population-Initialization (N_{pop}, s_c, s_d);
5. $z^* = (z_1^*, z_2^*, \cdots, z_{N_{obj}}^*), z_i^* = \min(f_i(x_1), f_i(x_2), \cdots, f_i(x_{N_{pop}}))$;
6. **for** each $G = 1$ to N_{gen} do
7. Normalize objective functions;
8. **for** each $i = 1$ to N_{pop} do
9. Select the neighborhoods of the i-th individual, named *archive*;
10. (m_c, d_c) = Continuous-Variable-Model $(archive, s_c, K)$;
11. m_d = Dicrete-Variable-Model $(archive, s_d, G)$;
12. x_{new} = Offspring-Generation $(m_c, d_c, m_d, s_c, s_d)$;
13. If all constraints are not met, x_{new} is repaired;
14. Update z^*;
15. **for** each $j \in B(i)$ do
16. **if** $g^*(x_{new}|w_j, z^*) < g^*(x_j|w_j, z^*)$
17. $x_j = x_{new}$;
18. **end**
19. **end**
20. **end**
21. **end**
22. **return** P

4) EOD MODELS

Our target is on optimizing a multi-objective mixed-variable problem. Notably, EoD algorithm's applicability remains unaffected by the nature of decision variables. To accommodate diverse variable types, this paper introduces two novel EoD operators. These operators serve the purpose of generating new individuals by conducting sampling for continuous and discrete variables individually.

(1) Continuous variable model (CVM): The fundamental concept behind CVM involves estimating the distribution across the entire continuous variable space through the utilization of marginal histograms derived from samples within that continuous space. Initially, for every continuous variable x_i, a division into K bins is executed. Each bin is characterized by an interval, denoted as $[x_{i,j}, x_{i,j+1}]$, where $x_{i,j}$ and $x_{i,j+1}$ represent the lower and upper bounds of the interval. First

determine the first bin and the last bin according to the following formula:

$$x_{i,1} = \frac{1}{2} \times (x_i^l + x_i^{\min}) \tag{15}$$

$$x_{i,K+2} = \frac{1}{2} \times (x_i^u + x_i^{\max}) \tag{16}$$

where $x_i^{\min}$ and $x_i^{\max}$ are the smallest value and the largest value of the continuous variable x_i, respectively, and x_i^l and x_i^u are the lower boundary and upper boundary of the continuous variable x_i. Regarding the intermediate range $[x_{i,2}, x_{i,K+1}]$, which is partitioned into K clusters $\{Cluster_1, Cluster_2, \cdots, Cluster_K\}$ through K-Means clustering, the j-th bin of the continuous variable x_i is represented as follows:

$$\begin{aligned} x_{i,j} = \frac{1}{2}(&\max(Cluster_{j-1}) \\ &+ \min(Cluster_j)), j = 2, \cdots, K+1 \end{aligned} \tag{17}$$

Subsequently, the count of points contained within each bin is tallied. Given the potential existence of empty bins, and to ensure comprehensive coverage of the entire search space, such bins lacking points are assigned an exceedingly minute count denoted as ϵ. The count of individuals within the j-th bin of the continuous variable x_i, denoted as $Count_{i,j}$, is then defined as follows:

$$Count_{i,j} = \begin{cases} Count_{i,j}, & Count_{i,j} > 1 \\ 1, & Count_{i,j} \leq 1 \end{cases} \tag{18}$$

Continuous variables are characterized through a model reliant on the count of individuals within each bin. The height of the j-th bin pertaining to the continuous variable x_i can be denoted as follows:

$$m_{i,j}^c = \frac{Count_{i,j}}{\sum_{k=1}^{K+1} Count_{i,k}} \tag{19}$$

When constructing the histogram model for each interval $[x_{i,j}, x_{i,j+1}]$, and assuming that the set of points falling within this interval is denoted as $\{x_{i,j}^1, x_{i,j}^2, \cdots, x_{i,j}^{N_{in}}\}$, we apply the kernel density estimation method to estimate the probability density function for the data within that interval. In cases where no points are present within the interval, we assume that the data points within the interval follow a uniform distribution. The resulting distribution function within an interval is defined as follows:

$$d_{i,j}^c(x) = \begin{cases} \frac{1}{N_{in} \times h} \sum_{k=1}^{N_{in}} Kernel(\frac{x - x_{i,j}^k}{h}), & N_{in} > 0 \\ \text{Uniform}(x_{i,j}, x_{i,j+1}), & N_{in} = 0 \end{cases} \tag{20}$$

where N_{in} represents the solutions' amount within the current interval, h is a parameter governing the smoothness of the kernel function, and $Kernel(\cdot)$ denotes the selected kernel function. In this study, we opt for the Gaussian kernel function $GKernel(\cdot)$, which is defined as follows:

$$GKernel(x) = \exp(-\frac{x^2}{2 \times h^2}) \tag{21}$$

where h is defined as:

$$h = \left(\frac{4 \times \text{std}^5(x_{i,j}^1, x_{i,j}^2, \cdots, x_{i,j}^n)}{3 \times n} \right)^{\frac{1}{5}} \tag{22}$$

where std$(\cdot)$ is the function for computing the standard deviation. The pseudocode of the CVM is shown in Algorithm 2.

Algorithm 2 Continuous-Variable-Model

Input:
P: Population
s_c : Indices of continuous variables
K: Number of clusters
Output:
m_c : Set of continuous probability models
d_c : Set of probability density functions
for each $i \in s_c$ do
  Determine $x_{i,1}, x_{i,K+2}$ using Eq. (15) and Eq. (16);
  $\{Cluster_1, Cluster_2, \cdots, Cluster_K\}$ = KMeans (P);
  Employing Eq. (17) to calculate the boundary $[x_{i,j}, x_{i,j+1}]$;
  for each $k \in \{2, \cdots, K+1\}$ do
    Employing Eq. (18) to get the number of individuals in k-th bin;
    Employing Eq. (19) to update the probability model m_c;
    Employing Eq. (20) to estimate the probability density function d_c of each interval;
  end
end
return m_c, d_c

(2) Discrete variable model (DVM): To facilitate sampling within the discrete space, this paper employs an incremental learning approach for constructing models pertaining to discrete variables. This approach integrates both historical and current information. First, define the height of the j-th bin of the discrete variable x_i:

$$m_{i,j}^d = \frac{WCount_{i,j}}{\sum_{k=1}^{x_i^u} WCount_{i,k}}, j \in \{0, 1, \cdots, x_i^u\} \tag{23}$$

where $WCount_{i,j}$ represents the weighted count of solutions whose discrete variable x_i is equal to j. Subsequently, the $N_{archive}$ individuals in *archive* are arranged in ascending order through a sorting technique, such as non-dominated sorting. The individual positioned at rank r will then contribute to an

Algorithm 3 Discrete-Variable-Model

Input:
P: Population
s_d : Indices of discrete variables
G: Number of current generations
Output:
m_d : Set of discrete probability models
1. **for** each $i \in s_d$ do
2. **for** each $j \in \{0, 1, \cdots, x_i^u\}$ do
3. Determine the individuals whose x_i equals to j;
4. Employing Eq. (22) to calculate the weighted count;
5. Employing Eq. (23) to update the probability model;
6. **end**
7. **end**
8. return m_d;

increment in the height of the j-th bin to which it is affiliated:

$$\Delta m_{r,j}^d = \frac{N_{archive} - r + 1}{\sum_{k=1}^{N_{archive}} \frac{N_{archive}-k+1}{N_{archive}}} \tag{24}$$

Then $WCount_{i,j}$ and $\Delta m_{r,j}^d$ are related as follows:

$$WCount_{i,j} = \sum_{r=1}^{N_{archive}} \Delta m_{r,j}^d \times \delta_{i,r,j} \tag{25}$$

when $x_r = j$, $\delta_{i,k,j} = 1$; otherwise, $\delta_{i,k,j} = \epsilon$, which is a small value. Following this, the probability model undergoes an incremental update as follows:

$$m_{i,j}^d (G) = (1 - \frac{G}{N_{gen}}) \times m_{i,j}^d(G-1) + \frac{G}{N_{gen}} \times \frac{WCount_{i,j}}{\sum_{k=1}^{x_i^u} WCount_{i,k}} \tag{26}$$

where G and N_{gen} are the current and maximum number of generations, respectively. The pseudocode of the DVM is shown in Algorithm 3.

C. COMPUTATIONAL COMPLEXITY

The computational complexity of population evolution based on the MOEA/D algorithm is $O(MN)$ where M denotes the number of objective functions and N represents the population size. The computational complexity of associating solutions with reference vectors within the population is $O(MKN)$, with K being the dimension of the reference vectors. The computational complexity of the EoD model is $O(N^2D)$, where D signifies the dimension of the decision vector. In summary, the overall computational complexity of the proposed MOEA/D-EoD algorithm is $O(N^2D)$.

Algorithm 4 Offspring-Generation

Input:
m_c : Set of continuous probability models
d_c : Set of probability density functions
m_d : Set of discrete probability models
s_c : Indices of continuous variables
s_d : Indices of discrete variables
Output:
x_{new} : new individual
1. **for** each $i \in s_d$ do
2. $r = \text{Uniform}(0,1)$;
3. According to m_d and, sampling a value d from $\{0, 1, \cdots, x_{i,j}^u\}$;
4. $x_{new,i} = d$;
5. **end**
6. **for** each $i \in s_c$ do
7. $r = \text{Uniform}(0,1)$;
8. According to m_c and r, selecting a bin k;
9. According to $d_{i,k}^c(x)$, sampling a value c;
10. $x_{new,i} = c$;
11. **end**
12. **return** x_{new};

TABLE 2. Parameters.

Parameter	Value
The number of objectives in DTLZ1-DTLZ7, M	3
The number of objectives in ZDT1-ZDT6, M	2
The number of decision variables in DTLZ1, D	M+4
The number of decision variables in DTLZ2-DTLZ7, D	M+9
The number of decision variables in ZDT1-ZDT3, D	30
The number of decision variables in ZDT4 and ZDT6, D	10
Population size	100
Number of divided sub-channels, α	10
The number of evaluation	10000
Run times	30

V. SIMULATION STUDY

A. EXPERIMENTAL RESULTS ON BENCHMARKS

First, we conduct comparative experiments on the proposed MOEA/D-EoD algorithm using benchmark test functions. The algorithms compared include NSGA-III [46], ANSGA-III [47], MOEAD [48], MOEADAWA [49], and RVEA [50]. The benchmark test sets include DTLZ and ZDT, with the objective functions and decision variable dimensions used in the tests shown in Table 2. In addition, the population size for all algorithms was uniformly set to 100, and the maximum number of function evaluations was 10,000. Each algorithm is independently run 30 times on each instance, and the performance metric used is the inverted generational distance (IGD) [51].

We conduct comparison experiments to validate the performance of the proposed MOEA/D-EoD algorithm in solving multi-objective optimization problems on the DTLZ and ZDT benchmark test sets. Table 3 presents the IGD results corresponding to the Pareto fronts obtained from 30 runs of six algorithms, with the best results highlighted in

TABLE 3. Performance results on the DTLZ and ZDT benchmark sets.

Problem	NSGA-III	ANSGA-III	MOEA/D	MOEADAWA	RVEA	MOEA/D-EoD
DTLZ1	2.1894e-1 (1.99e-1) =	1.9046e-1 (2.19e-1) =	3.8128e-1 (6.58e-1) =	2.8675e-1 (1.89e-1) -	4.8784e-1 (2.80e-1) -	**1.7419e-1 (1.44e-1)**
DTLZ2	**5.4823e-2 (9.91e-5) +**	5.8644e-2 (1.59e-3) -	5.4930e-2 (2.43e-4) +	5.7116e-2 (7.22e-4) -	5.5902e-2 (7.19e-4) -	5.5096e-2 (2.87e-4)
DTLZ3	9.1883e+0 (3.76e+0) =	9.3684e+0 (4.59e+0) =	1.6251e+1 (1.32e+1) -	1.0209e+1 (6.16e+0) =	1.5940e+1 (6.58e+0) -	**8.5405e+0 (5.64e+0)**
DTLZ4	2.1732e-1 (2.33e-1) =	1.7053e-1 (2.08e-1) =	3.4107e-1 (3.12e-1) =	1.3926e-1 (1.83e-1) =	**5.5990e-2 (6.29e-4) =**	3.2239e-1 (3.13e-1)
DTLZ5	1.2568e-2 (1.48e-3) -	1.1387e-2 (1.14e-3) -	3.2554e-2 (8.74e-4) -	1.7977e-2 (1.24e-3) -	8.7047e-2 (1.54e-2) -	**5.7716e-3 (1.89e-4)**
DTLZ6	1.8992e-2 (2.95e-3) -	1.4666e-2 (4.66e-3) -	8.6560e-2 (2.11e-1) -	5.0078e-2 (1.66e-1) -	1.1806e-1 (1.24e-1) -	**6.1726e-3 (5.98e-3)**
DTLZ7	9.8026e-2 (7.94e-2) =	**8.5095e-2 (5.47e-2) =**	2.3952e-1 (2.24e-1) =	1.4437e-1 (7.36e-3) =	1.2217e-1 (1.05e-2) =	2.5190e-1 (2.54e-1)
ZDT1	1.9919e-2 (8.97e-3) -	1.9315e-2 (5.92e-3) -	1.2692e-1 (6.14e-2) -	3.9578e-2 (1.17e-2) -	1.1399e-1 (1.90e-2) -	**1.5841e-2 (8.09e-3)**
ZDT2	5.5522e-2 (7.56e-2) +	4.7227e-2 (5.35e-2) +	5.1776e-1 (7.54e-2) -	**3.1577e-2 (8.79e-3) +**	1.7279e-1 (4.08e-2) +	2.6565e-1 (1.65e-1)
ZDT3	1.8237e-2 (5.64e-3) =	**1.8108e-2 (7.08e-3) =**	1.5036e-1 (5.43e-2) -	2.9029e-2 (1.13e-2) -	1.5641e-1 (1.76e-2) -	2.1972e-2 (1.16e-2)
ZDT4	5.5296e-1 (3.06e-1) =	5.3224e-1 (2.93e-1) =	5.0829e-1 (2.88e-1) =	**4.6606e-1 (1.85e-1) =**	1.3113e+0 (4.37e-1) -	4.7787e-1 (2.48e-1)
ZDT6	1.7224e-1 (6.30e-2) -	1.9749e-1 (7.52e-2) -	7.8840e-2 (2.44e-2) =	1.0391e-1 (3.81e-2) -	3.4540e-1 (1.00e-1) -	**7.3210e-2 (4.51e-2)**
+/-/=	2/4/6	1/5/6	1/6/5	1/7/4	1/9/2	

bold. Statistical analysis was performed using the Wilcoxon rank-sum test with a significance level of 0.05. The symbols "+", "−" and "=" indicate that the performance of the comparison algorithm is significantly better than, significantly worse than or not significantly different from that of the proposed MOEA/D-EoD algorithm, respectively.

Based on the experimental results shown in Table 3, it is evident that the decomposition-based MOEA/D-EoD algorithm achieves the best IGD mean values on more than half of the benchmark test functions compared to other decomposition-based algorithms, namely MOEA/D, MOEADAWA, and RVEA. When compared with all algorithms, MOEA/D-EoD ranks first on 6 out of the 12 benchmark test functions. These results demonstrate that MOEA/D-EoD exhibits superior optimization performance compared to NSGA-III, ANSGA-III, MOEAD, MOEADAWA and RVEA.

Fig. 4 illustrates the Pareto front distributions of various algorithms on the DTLZ and ZDT benchmark test sets. The analysis indicates that, compared to other algorithms, MOEA/D-EoD achieves convergence close to the true Pareto front in two-objective optimization problems, with solutions widely distributed across the feasible solution space. In three-objective optimization problems, while other algorithms struggle to converge effectively, MOEA/D-EoD enables solutions to be evenly distributed along the true Pareto front. This demonstrates that the solution set obtained by MOEA/D-EoD is very uniformly distributed, indicating that the proposed estimation distribution model not only maintains population diversity but also adapts to different Pareto fronts during the optimization process.

B. SENSITIVITY ANALYSIS

K-means clustering, an integral component of the EoD model, exerts a significant impact on the algorithm's optimization performance. The setting of the K value directly influences both population diversity and algorithm convergence. To investigate the impact of K value on algorithm performance, we conducted comparative experiments on the DTLZ and ZDT test suites while maintaining other parameters constant. As illustrated in Table 4, the MOEA/D-EoD algorithm achieved the best performance on both test suites when K was set to 10, indicating that the algorithm's optimization performance is maximized at $K = 10$.

TABLE 4. Sensitivity analysis of K.

Parameter	Rank sum
The K value in k-means is set to 2	0
The K value in k-means is set to *5*	2
The K value in k-means is set to 10	6
The K value in k-means is set to 15	3
The K value in k-means is set to 20	1

C. SIMULATION DESIGN

Next, we apply the proposed MOEA/D-EoD algorithm to a real-world scenario to address task offloading and resource allocation issues in MEC systems, thereby further validating the effectiveness of the proposed method. We assume that the wireless base station has a coverage of 100 meters, which is similar with [23], [25]. Then, we express the uplink channel gain between the mobile device u_n and its connected base station bs_m as follows: $h_{n,m} = L_{n,m}^{-\beta}$, where β denote the path loss factor. Due to the variability in computing power across mobile devices, F^L is assigned from the set [0.5, 1.0] GHz. Likewise, the computational resources allocated from edge servers F^C are limited to the range of [5], [10] GHz. Cost for mobile users to perform tasks with edge servers sets within [1], [5]. Table 5 lists the additional experimental variables [52].

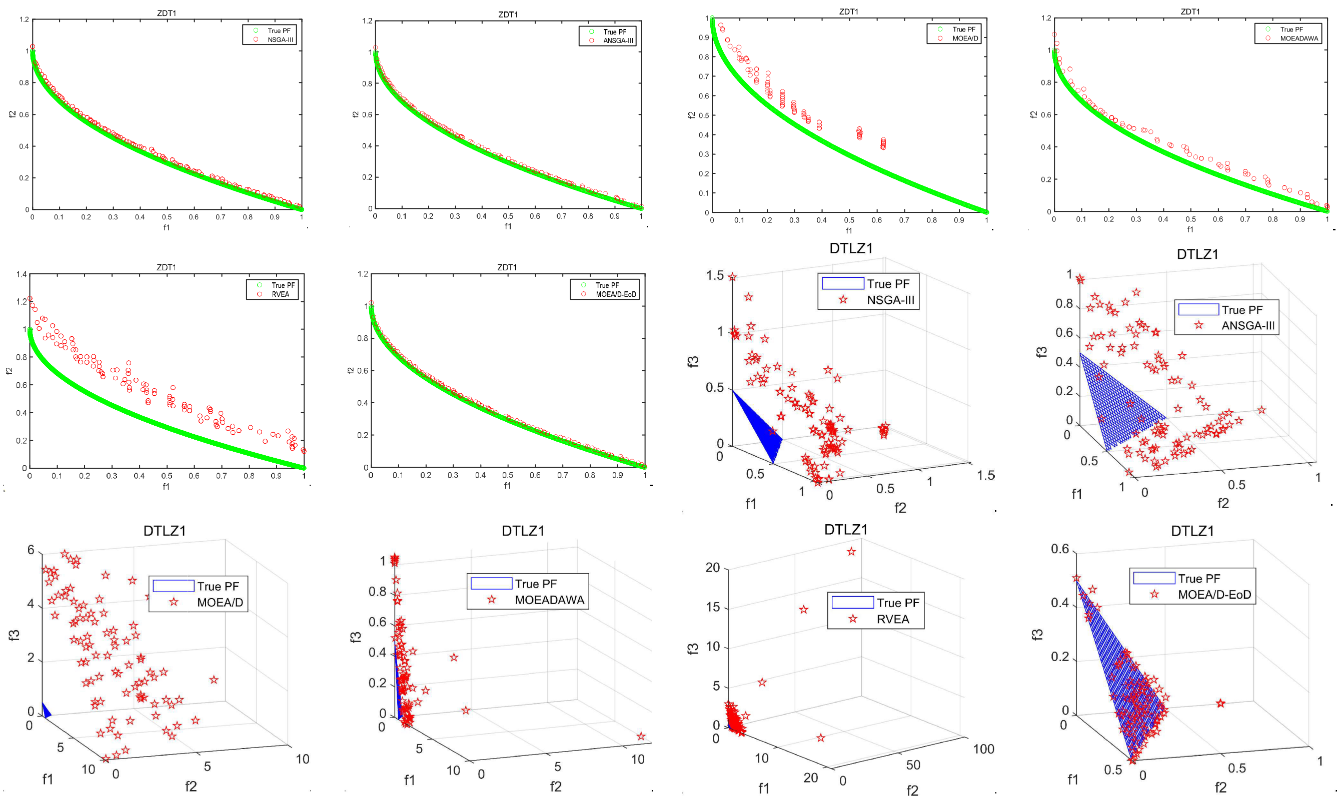

FIGURE 4. Pareto front on the DTLZ and ZDT benchmark sets.

TABLE 5. Parameters.

Parameter	Value
Number of mobile users, N	30
Number of base stations, M	5
Number of divided sub-channels, α	10
Transmission bandwidth of mobile users, B	10 MHz
Background noise, σ^2	-106 dBm
Path loss factor, β	3
Maximum of transmission power, $p_{\max}$	1 W
Data size for the offloading, s_n	5 MB
Number of CPU cycles for a task, c_n	1 Gigacycles

D. COMPARISON WITH OTHER MECHANISMS

The comparison involves evaluating MOEA/D-EoD against three computation offloading strategies, as follows:

(1) Random Offloading Scheme (ROS): Mobile users decide whether its computational task is to be executed locally or on the nearby edge server in a random way. Similarly, if mobile users decide to unload, the connected base stations are assigned orthogonal sub-channels for communication by a random process. We use this scenario to evaluate the offloading decision in the MOEA/D-EoD algorithm and the effect of the assigned wireless sub-channels.

(2) Fixed Transmission Power Scheme (FTPS): Mobile users are offloaded using the same offloading scheme as MOEA/D-EoD, however, all users who decide to perform offloading use a fixed transmission power. We employ this scenario to assess the effectiveness of the transmission power control in the MOEA/D-EoD algorithm.

(3) Fixed Computing Resource Scheme (FCRS): Mobile users are offloaded using the same offloading scheme as MOEA/D-EoD, but these edge servers allocate a fixed amount of computational resources to each user. We employ this scenario to assess the impact of the MOEA/D-EoD algorithm on the allocation of computational resources within edge servers.

E. PERFORMANCE EVALUATION

We focus on three main metrics in the comprehensive performance investigation: (i) the total energy consumption of mobile devices; (ii) the total task delay of tasks; (iii) the total cost of mobile users. That is, the three objectives we established.

1) IMPACT OF DEVICE AMOUNT ON SYSTEM EFFICIENCY

First of all, we consider this MEC scenario in which the mobile devices' amount varies from 5 to 50, with a constant data size of 5MB per task. The comparative results to the total energy consumption, the total task delay and the total cost are depicted in Fig. 5, 6 and 7, respectively.

As seen in Fig. 5 and 6, MOEA/D-EoD shows a clear advantage over ROS, FTPS and FCRS. When the mobile users' amount is relatively small, all four mechanisms remain

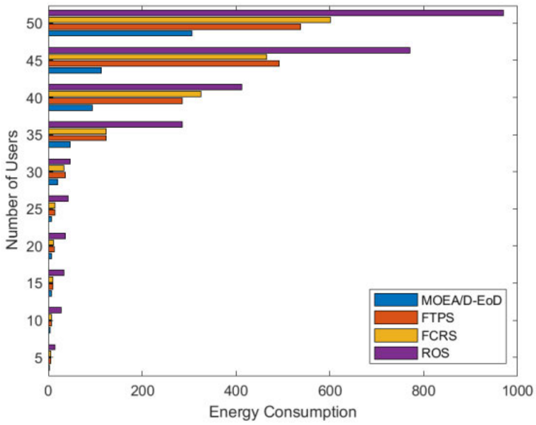

FIGURE 5. Effect of device amount on the total energy consumption.

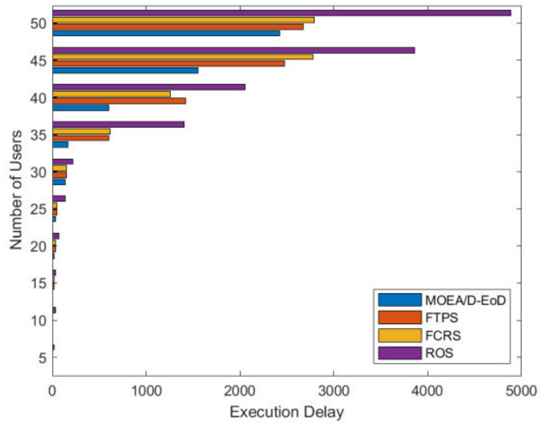

FIGURE 6. Effect of device amount on the total task delay.

low in terms of energy consumption and task latency. However, when the mobile users' amount grows to 35, the remaining three strategies exhibit a notable upsurge in energy consumption and latency. This is because the increasing mobile users are competing for the limited edge cloud resources, and in order to be able to successfully offload tasks to nearby base stations, the mobile users have to expand their transmission power as much as possible, but this will intensify the mutual interference in the transmission process and even lead to communication congestion. The ROS performs the worst because the random offloading approach does not have a dynamic adjustment mechanism. Similarly, although FTPS and FCRS use the same offloading scheme as MOEA/D-EoD, both of them either fix the transmission power or the computational power allocated to the server during offloading, which cannot be adjusted to the actual situation and is less effective than our MOEA/D-EoD. When the users' amount increases to 50, the whole MEC system is at saturation because there are only 5 wireless base stations in the system and each base station can accept up to 10 users. At this point, energy consumption and delay are at their maximum. In Fig. 7, as the mobile users' amount increases, tasks in mounting numbers are offloaded to the edge for execution, and the total cost rises. As ROS randomly determines whether to offload, it results in the highest total cost. Since FTPS and MOEA/D-EoD use the same offloading strategy, the total cost is the same for both. From Eq. (7), we can learn that the spending on task execution is directly linked to the allocation of computing resources by the edge server. To ensure that the tasks offloaded can be completed successfully, sufficient computing power needs to be guaranteed in FCRS, and the fixed allocation of computing resources cannot be too small, which costs more in FCRS.

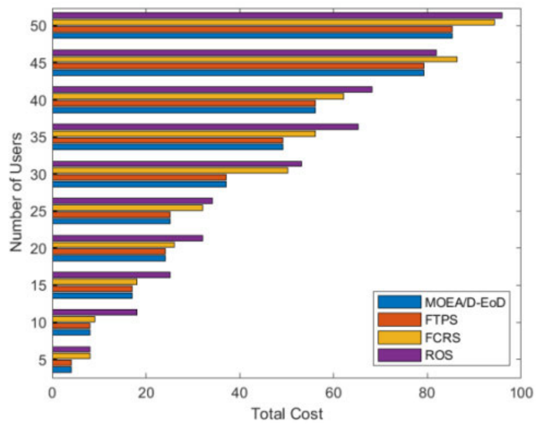

FIGURE 7. Effect of device amount on the total cost.

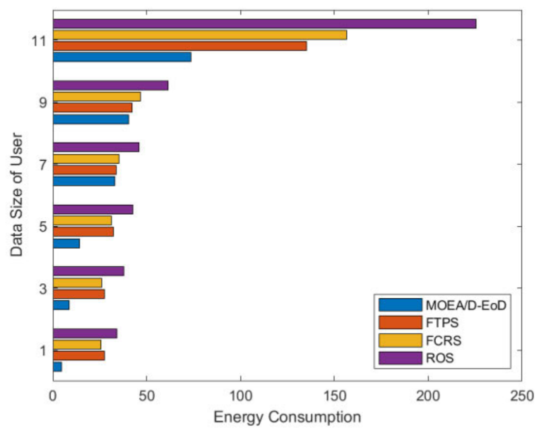

FIGURE 8. Effect of data size on the total energy consumption.

2) IMPACT OF DATA SIZE ON SYSTEM EFFICIENCY

Secondly, our focus lies in assessing the influence of data size on offloading performance, while maintaining a constant number of 30 mobile users. The data size of tasks varies from 1MB to 11MB. Fig. 8, Fig. 9 and Fig. 10 present a comparative analysis of the total energy consumption, total task delay, and total cost concerning different sizes of offloading data, respectively. As indicated in Fig. 8 and 9, we can deduce that as the date size of tasks grows, the energy consumption and latency required to complete them also rises. Also we can find that MOEA/D-EoD is significantly better

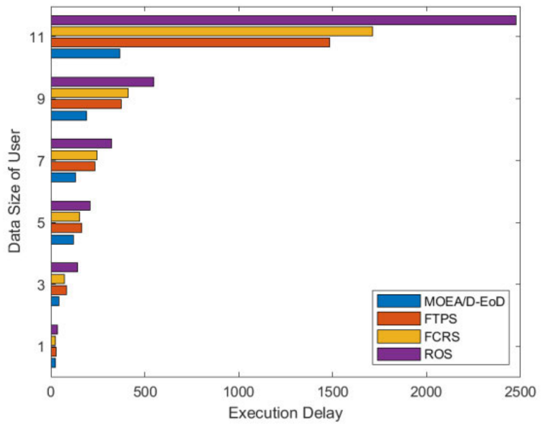

FIGURE 9. Effect of data size on the total task delay.

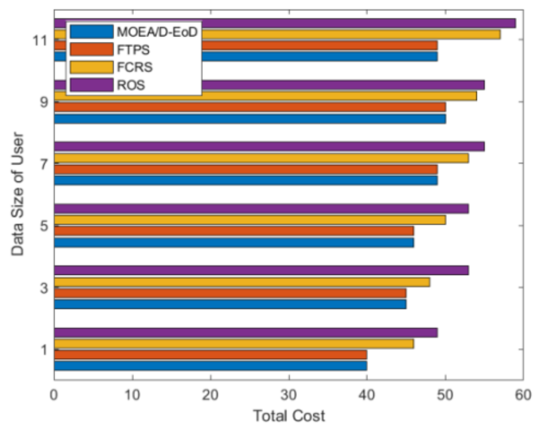

FIGURE 10. Effect of data size on the total cost.

than ROS, FTPS and FCRS. This is because the five base stations in the scenario are adequate for supporting the tasks carried by 30 mobile devices. When the data size of task reaches 11, both energy consumption and latency peak. As revealed in Fig. 10, as the amount of task data increases, edge servers need to allocate more computational resources to offloading tasks to ensure completion, so the cost is increasing.

3) OFFLOADING STRATEGY VALIDATION

In the end, we evaluate the effectiveness of the offloading strategy in MOEA/D-EoD. As shown in Table 6, MOEA/D-EoD will adjust the offloading strategies dynamically with the number of mobile users. The MOEA/D-EoD algorithm possesses the capability of dynamically adjusting task offloading and resource allocation strategies, enabling flexible adaptation to real-world scenarios. This eliminates the resource wastage and performance bottlenecks inherent in fixed strategies. By integrating histogram estimation and kernel density estimation methods, MOEA/D-EoD achieves more accurate handling of both continuous and discrete decision variables, leading to a significant enhancement in optimization performance. Furthermore, leveraging the decomposition

TABLE 6. Comparison of offloading strategies with different numbers of mobile users.

Number of Users	Location	ROS	FTPS	FCRS	MOEA/D-EoD
10	Local	3	6	6	6
	Edge Cloud	7	4	4	4
30	Local	9	12	12	12
	Edge Cloud	21	18	18	18
50	Local	13	18	18	18
	Edge Cloud	37	32	32	32

algorithm framework, MOEA/D-EoD effectively addresses multi-objective optimization problems, striking a balance among multiple metrics such as energy consumption, execution delay and total cost.

VI. CONCLUSION

We investigate the multi-objective task offloading problem within MEC for a scenario involving multiple users and multiple base stations, with joint take account of the impact of transmission power and the resource allocation problem during offloading. We designed a scheme MOEA/D-EoD to solve the above joint problem. Our developed algorithm can solve for the optimal offloaded strategy and resource allocation while satisfying the constraints. The proposed MOEA/D-EoD algorithm exhibits remarkable performance on the DTLZ and ZDT benchmark test suites, outperforming existing multi-objective optimization algorithms. Furthermore, its application to task offloading and resource allocation models demonstrates its effectiveness in real-world scenarios. However, the algorithm's computational complexity is relatively high due to its involvement in multi-objective optimization and mixed variables. Additionally, its adaptability under varying network environments and user demands warrants further refinement.

REFERENCES

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[2] V. Petrov, M. A. Lema, M. Gapeyenko, K. Antonakoglou, D. Moltchanov, F. Sardis, A. Samuylov, S. Andreev, Y. Koucheryavy, and M. Dohler, "Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 485–501, Mar. 2018.

[3] L. Ma, H. Kang, G. Yu, Q. Li, and Q. He, "Single-domain generalized predictor for neural architecture search system," *IEEE Trans. Comput.*, vol. 73, no. 5, pp. 1400–1413, May 2024.

[4] L. Ma, N. Li, P. Zhu, K. Tang, A. Khan, F. Wang, and G. Yu, "A novel fuzzy neural network architecture search framework for defect recognition with uncertainties," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 5, pp. 3274–3285, May 2024.

[5] L. He, B. Dong, and P. Jiang, "A heuristic grafting strategy for manufacturing knowledge graph extending and completion based on nature language processing: KnowTree," *IEEE Access*, vol. 9, pp. 90847–90862, 2021.

[6] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert, "Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 12, pp. 3490–3502, Dec. 2020.

[7] B. Yang, J. Yamazaki, N. Saito, Y. Kokai, and D. Xie, "Big data analytic empowered grid applications—Is PMU a big data issue?" in *Proc. 12th Int. Conf. Eur. Energy Market (EEM)*, May 2015, pp. 1–4.
[8] W.-C. Fang, K.-Y. Wang, N. Fahier, Y.-L. Ho, and Y.-D. Huang, "Development and validation of an EEG-based real-time emotion recognition system using edge AI computing platform with convolutional neural network system-on-chip design," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 4, pp. 645–657, Dec. 2019.
[9] W. Wu, Y. Gao, T. Zhou, Y. Jia, H. Zhang, T. Wei, and Y. Sun, "Deep reinforcement learning-based video quality selection and radio bearer control for mobile edge computing supported short video applications," *IEEE Access*, vol. 7, pp. 181740–181749, 2019.
[10] Y. Yu, "Mobile edge computing towards 5G: Vision, recent progress, and open challenges," *China Commun.*, vol. 13, no. 2, pp. 89–99, 2016.
[11] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.
[12] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4125–4138, Nov. 2022.
[13] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.
[14] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[15] Q.-V. Pham, L. B. Le, S.-H. Chung, and W.-J. Hwang, "Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation," *IEEE Access*, vol. 7, pp. 16444–16459, 2019.
[16] L. Ma, N. Li, Y. Guo, X. Wang, S. Yang, M. Huang, and H. Zhang, "Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12698–12711, Dec. 2022.
[17] S. Kumar, N. Panagant, G. G. Tejani, N. Pholdee, S. Bureerat, N. Mashru, and P. Patel, "A two-archive multi-objective multi-verse optimizer for truss design," *Knowl.-Based Syst.*, vol. 270, Jun. 2023, Art. no. 110529.
[18] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6723–6742, Nov. 2021.
[19] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6684–6696, Jul. 2022.
[20] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 1996, pp. 178–187.
[21] L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, and Y. Jin, "Pareto-wise ranking classifier for multiobjective evolutionary neural architecture search," *IEEE Trans. Evol. Comput.*, vol. 28, no. 3, pp. 570–581, Jun. 2024, doi: 10.1109/TEVC.2023.3314766.
[22] N. Li, L. Ma, G. Yu, B. Xue, M. Zhang, and Y. Jin, "Survey on evolutionary deep learning: Principles, algorithms, applications, and open issues," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–34, Feb. 2024.
[23] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 369–392, 2013.
[24] H. Flores and S. Srirama, "Mobile code offloading: Should it be a local decision or global inference?" in *Proc. 11th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2013, pp. 539–540.
[25] A. Ravi and S. K. Peddoju, "Handoff strategy for improving energy efficiency and cloud service availability for mobile devices," *Wireless Pers. Commun.*, vol. 81, no. 1, pp. 101–132, Mar. 2015.
[26] A. Ravi and S. K. Peddoju, "Energy efficient seamless service provisioning in mobile cloud computing," in *Proc. 7th IEEE Int. Symp. Service-Oriented Syst. Eng.*, Mar. 2013, pp. 463–471.
[27] Z. Sanaei, S. Abolfazli, A. Gani, and M. Shiraz, "SAMI: Service-based arbitrated multi-tier infrastructure for mobile cloud computing," in *Proc. 1st IEEE Int. Conf. Commun. China Workshops (ICCC)*, Aug. 2012, pp. 14–19.
[28] M.-H. Chen, M. Dong, and B. Liang, "Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2868–2881, Dec. 2018.
[29] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2016.
[30] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.
[31] J. Liu, P. Li, J. Liu, and J. Lai, "Joint offloading and transmission power control for mobile edge computing," *IEEE Access*, vol. 7, pp. 81640–81651, 2019.
[32] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[33] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
[34] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.
[35] M. Feng, X. Wang, Y. Zhang, and J. Li, "Multi-objective particle swarm optimization for resource allocation in cloud computing," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst.*, Oct. 2012, pp. 1161–1165.
[36] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25536–25545, Dec. 2022.
[37] K. N. A. Prethi and M. Sangeetha, "A multi-objective optimization of resource management and minimum batch VM migration for prioritized task allocation in fog-edge-cloud computing," *J. Intell. Fuzzy Syst.*, vol. 43, no. 5, pp. 5985–5995, Sep. 2022.
[38] J. Xue and Y. An, "Joint task offloading and resource allocation for multi-task multi-server NOMA-MEC networks," *IEEE Access*, vol. 9, pp. 16152–16163, 2021.
[39] A. Umer, M. Ali, A. I. Jehangiri, M. Bilal, and J. Shuja, "Multi-objective task-aware offloading and scheduling framework for Internet of Things logistics," *Sensors*, vol. 24, no. 8, p. 2381, Apr. 2024.
[40] N. Panagant, S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Many-objective meta-heuristic methods for solving constrained truss optimisation problems: A comparative analysis," *MethodsX*, vol. 10, Jan. 2023, Art. no. 102181.
[41] S. Kumar, P. Jangir, G. G. Tejani, and M. Premkumar, "A decomposition based multi-objective heat transfer search algorithm for structure optimization," *Knowl.-Based Syst.*, vol. 253, Oct. 2022, Art. no. 109591.
[42] S. Kumar, G. G. Tejani, N. Pholdee, S. Bureerat, and P. Jangir, "Multi-objective teaching-learning-based optimization for structure optimization," *Smart Sci.*, vol. 10, no. 1, pp. 56–67, Jan. 2022.
[43] S. Kumar, P. Jangir, G. G. Tejani, M. Premkumar, and H. H. Alhelou, "MOPGO: A new physics-based multi-objective plasma generation optimizer for solving structural optimization problems," *IEEE Access*, vol. 9, pp. 84982–85016, 2021.
[44] J. Zhang, B. Gong, M. Waqas, S. Tu, and Z. Han, "A hybrid many-objective optimization algorithm for task offloading and resource allocation in multi-server mobile edge computing networks," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 1–14, Oct. 2023.
[45] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
[46] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
[47] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach. Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
[48] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[49] Y. Qi, X. Ma F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[50] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
[51] N. Riquelme, C. V. Lücken, and B. Baran, "Performance metricss in multi-objective optimization," in *Proc. Latin Amer. Comput. Conf. (CLEI)*, 2015, pp. 1–11.
[52] Y. Chen, X. Wang, L. Ma, and P. Zhou, "Multi-objective optimization-based task offloading and power control for mobile edge computing," in *Proc. 17th Int. Conf. Intell. Comput. Theories Appl. (ICIC)*, Shenzhen, China. Springer, 2021, pp. 3–17.

CHUNYANG YU received the Ph.D. degree in computer system architecture from Northeastern University, Shenyang, China, in 2011. His current research interests include multi-objective optimization methods and Bayesian optimization algorithm.

YIBO YONG received the M.S. degree in software engineering from Northeastern University, Shenyang, China, in 2021, where he is currently pursuing the Ph.D. degree with the College of Software. His research interests include federated optimization, evolutionary computation, and multi-objective optimization.

YANG LIU received the B.Sc. degree in information and computing science from Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2021. He is currently pursuing the M.Sc. degree in software engineering with the College of Software, Northeastern University, Shenyang, China. His current research interests include multi-objective optimization methods and Bayesian optimization algorithm.

JIAN CHENG received the B.Sc. degree in automation, the M.Sc. degree in control theory and control engineering, and the Ph.D. degree in communication and information system from China University of Mining and Technology, Xuzhou, China, in 1997, 2003, and 2008, respectively. He has been a Postdoctoral Fellow at Tsinghua University and the University of Birmingham from 2009 to 2013. He is currently a Professor and the Chief Scientist with the Research Institute of Mine Artificial Intelligence in Chinese Institute of Coal Science (Tiandi Science and Technology Company Ltd.), State Key Laboratory for Intelligent Coal Mining and Strata Control, Beijing, China. His current research interests include machine learning and pattern recognition, data mining and big data, as well as imbalance learning and image processing and their applications in industrial fields.

QIANG TONG received the Ph.D. degree in computer application technology from Northeastern University, Shenyang, China, in 2015. He is currently an Associate Professor with the School of Software, Northeastern University. His research interests include natural language processing and machine learning.