

# A Hybrid Many-Objective Optimization Algorithm for Task Offloading and Resource Allocation in Multi-Server Mobile Edge Computing Networks

Jiangjiang Zhang, Bei Gong, *Member, IEEE*, Muhammad Waqas, *Senior Member, IEEE*,  
Shanshan Tu, *Member, IEEE*, Zhu Han, *Fellow, IEEE*,

**Abstract**—Mobile edge computing (MEC) is an effective computing tool to cope with the explosive growth of data traffic. It plays a vital role in improving the quality of service for user task computing. However, the existing solutions rarely address all the significant factors that impact the quality of service. To challenge this problem, a trusted many-objective model is built by comprehensively considering the task time delay, server energy consumption, trust metrics between task and server, and user experience utility factors in multi-server MEC networks. We decompose the original problem into task offloading (TO) and resource allocation (RA) to address the model. Then a novel hybrid many-objective optimization algorithm based on cascading clustering and incremental learning is designed to optimize the TO decision solutions. A low-complexity heuristic method is adopted based on the optimal TO decision solutions to optimize the RA problem continuously. To verify the model's validity and the optimisation algorithm's superiority, five other advanced many-objective algorithms are used for comparison. The results show that our algorithm has more than half the number of the superior values for the benchmark problem. And the obtained model solution shows good performance on different indicators metrics for the decomposition problem.

**Index Terms**—Mobile edge computing, task offloading, resource allocation, many-objective optimization

## I. INTRODUCTION

WITH the vigorous development of Internet of things (IoT) technology and various intelligent terminal devices, the data traffic in the communication network is increasing exponentially [1]. A higher requirement for faster computing efficiency and better quality of service (QoS) is put forward in promoting the development of mobile communication technology [2]. Mobile edge computing (MEC), as a novel distributed computing mode, can sink the computing function to the edge of the mobile network and effectively use the limited resources of edge devices to provide users with corresponding computing services [3]. Therefore, MEC

is attracting more and more attention because of its excellent computing mode in the complex and changeable network environment.

Unlike the traditional cloud computing mode, it is difficult for MEC servers limited by computing resources to meet the computing services expected by all tasks at the same time [4]. And the explosive growth of IoT terminals makes the contradiction with limited computing resources more prominent [5]. Typically, when the offloading task does not match the computing load distribution of the edge server, the server computing resource utilization will be reduced, and the task time delay and server energy consumption will be increased [6]. And some malicious behaviors are likely to affect the QoS of MEC [7]. For example, larger computing tasks are deliberately and continuously offloaded to edge servers with smaller computing resources, which may seriously damage the processing of normal computing services. Naturally, it is essential to formulate a safe and efficient task offloading (TO) decision-making scheme to reasonably offload computing tasks to edge servers. And the TO decisions are usually closely related to resource allocation (RA) plans, i.e., how to allocate resources for the MEC server after a task with a fixed offloading decision [8], [9]. When the MEC server is allocated more network and less computing resources, the computing tasks can be quickly offloaded to the server [10]. However, insufficient computing resources make the offloaded tasks unable to be processed in time, leading to additional task time delay and server energy consumption. And the MEC server with less network resources is likely to be idle, resulting in low resource utilization [11]. For a server with more network and computing resources simultaneously, it can only accept a limited number of task computing requests, which needs the assistance of other servers to handle more tasks [12]. Therefore, it is also essential to formulate an effective RA strategy in the multi-server MEC networks (MSMECN) to handle as many task requests as possible and meet the QoS requirements of different offloading tasks.

Recently, some research methods have been developed on handling task offloading and resource allocation (TORA) problems in the MSMECN environment. Wang *et al.* [13] proposed a novel TO model to describe the users' willingness to contribute their resources to the public and designed two dynamic RA algorithms based on the Markov decision process framework to handle the basic trade-off between task time delay and server energy consumption when providing mobile

J. Zhang, B. Gong and S. Tu are with the Faculty of Information Technology, Beijing University of Technology, 100124 Beijing, China, e-mail: (zhangjiangjiang@emails.bjut.edu.cn, gongbei@bjut.edu.cn, sstu@bjut.edu.cn).

M. Waqas is with the Computer Engineering Department, College of Information Technology, University of Bahrain 32038, Bahrain and also with the School of Engineering, Edith Cowan University, Perth WA 6027, Australia, e-mail: (enr.waqas2079@gmail.com).

Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701. e-mail: (hanzhu22@gmail.com).

Corresponding Author: Muhammad Waqas

services in-vehicle networks. Liu *et al.* [14] used the developed computational efficiency analysis model to evaluate three different versions of TO methods and proved that the TO method with cooperation between multi-servers could minimize the time delay. Ho *et al.* [15] designed a method based on deep reinforcement learning (DRL) to solve formulated non-convex TORA problems, which can minimize task time delay and server energy consumption in a complex network environment. Apostolopoulos *et al.* [16] formulated the non-cooperative game between users to determine the corresponding pure Nash equilibrium TO decision-making, and adopted a distributed low complexity RA strategy, which realizes the trust metrics between task and server and timely processing of user tasks.

However, these existing methods only target one or few individual performance factors, such as task time delay, server energy consumption, trust metrics between task and server, and user experience utility, and rarely do they address all the significant factors that impact on the QoS of MEC system [5]. The factors affecting the QoS of the MEC system come from many aspects in the MSMECN environment [10]. Moreover, these factors are inextricably linked and influence each other in the MSMECN environment. Therefore, multiple conflicting QoS of MEC system metrics [5]. Generally, a computing task can be processed with a small delay, giving the user an excellent experience [10]. On the other hand, some malicious MEC servers induce users to offload a large number of tasks to themselves but cannot provide them with timely computing services (that is, the offloaded tasks exceed the computing capacity of the MEC server itself), which will not only increase the task processing time delay and energy consumption but also brings a very poor user experience utility [7], [11]. How to comprehensively address the impact of these factors on the QoS and effectively balance the conflicting metrics is very challenging, which motivates our work.

Against this background, we formulate the TORA problem as a complex many-objective optimization problem (MaOP) [17]. MaOPs have more than three objective functions, which pose a huge challenge to the convergence and diversity (CaD) maintenance of the algorithm [18], [19]. The many-objective evolutionary algorithms (MaOEA) can be used to solve MaOPs [18], [19]. And its various functions are not just the effect of a single factor but the result of the integration and coordination of interdependent and interacting factors through appropriate mechanisms [20]. During addressing a MaOP, the early evolution of the population needs to focus on the convergence of solutions, and the late evolution needs to focus on the diversity [18]. Considering the importance of CaD in different stages of evolution is also the key challenge in designing MaOEA to achieve high-performance [19]. Following is a list of the notable contributions made in this study.

- 1) A trusted many-objective TORA model is built to describe the problem in detail. In our trusted many-objective TORA model, the task time delay, server energy consumption, trust metrics between task and server, and user experience utility are considered comprehensively as the four objectives to be optimized.
- 2) We decompose the TORA problem into a TO problem that optimizes the offloading decision solutions and a

RA problem after fixing the offloading decision solution. Concretely, a novel hybrid MaOEA based on cascading clustering and incremental learning (MaOEA-CCIL) is designed to obtain an optimized TO decision solution, a hybrid interacting process. The cascading clustering mechanism is employed to divide the offloading decision space solutions into elite and ordinary decision solutions. And the incremental learning selection mechanism is introduced into the reference point redistribution to improve the CaD of elite decision solutions, which will guide the evolution of the entire TO decision solution in a better direction. Based on the optimal offloading decision solutions, the RA problem continues to be optimized by employing the low-complexity heuristic optimization method with Karush-Kuhn-Tucker (KKT) condition [21].

- 3) Two extensive simulations are performed to verify the effectiveness of the design model and algorithm. On the one hand, MaOEA-CCIL is compared with other advanced MaOEAs on the benchmark function. On the other hand, the involved MaOEAs are combined with the heuristic optimization method to handle the TORA problem and are extensively measured under different performance indicators. Simulation results show that our algorithm has more than half the number of the superior values for the benchmark problem. And we achieve relatively stable performance on each model objective and obtain excellent results on different indicators metrics for the TORA problem.

The rest of this paper is organized as follows. After the introduction in Section I, the related work is illustrated in Section II. The concrete model construction process and the corresponding objective model expression form are stated in Section III. To handle the model, a novel hybrid MaOEA principle is expressed in Section IV. Two extensive simulation experiments have been conducted to verify the model's effectiveness in Section V. Finally, the paper is concluded in Section VI.

## II. RELATED WORK

With the emergence of various smart IoT applications, the demand for higher QoS has become more and more intense [22]. Typically, more real-time computing, task analysis, and processing capabilities are required in autonomous driving, smart home, and smart medical scenarios [23]. And the MEC mode can not only provide high-quality computing services and effectively utilize edge resources to improve resource utilization by sinking computing functions to the edge of the network [24]. Hence, using MEC mode to improve the QoS is receiving more and more attention [4]. However, it is challenging to handle all user offload tasks simultaneously due to the limited service resources of MEC servers [25]. Optimizing the factors involved in the TORA problem is very important to improve the QoS of MEC.

Recently, scholars have carried out many research work on optimizing the factors involved in the TORA problem [26]. And some typical partial works are categorized and listed in Table I.

TABLE I  
TYPICAL PART OF RESEARCH WORK ON OPTIMIZING THE FACTORS INVOLVED IN THE TORA PROBLEM

Literature	Time delay	Energy consumption	Trust metrics	User experience utility
[1], [12], [13], [15], [18], [23], [26], [27], [28], [29], [38]	✓	×	×	×
[8], [10], [22], [25], [30], [31], [32], [33]	×	✓	×	×
[2], [3], [9]	×	×	×	✓
[7], [21], [8], [34]	✓	✓	×	×
[24], [41]	✓	×	✓	×
[4]	✓	×	×	✓
[11]	✓	✓	✓	×
[28]	✓	✓	×	✓
Our approach	✓	✓	✓	✓

*Optimization on time delay.* Lin *et al.* [27] developed a resource-constrained market-oriented TO scheme based on a delay-guaranteed double auction of resources, which can ensure that the computing tasks of the terminal device minimize the time delay and achieve high efficiency of resource utilization. Sorkhoh *et al.* [28] designed a RA algorithm based on Lagrangian relaxation to achieve low latency and high reliability of vehicle networks within the coverage of roadside units. Lin *et al.* [29] employed deep reinforcement learning to reduce the time delay of the complex MEC job shop scheduling problem.

*Optimization on energy consumption.* Mao *et al.* [30] proposed an energy-saving strategy that considers delays and offloading failures by introducing energy harvesting technology into the MEC system. Chen *et al.* [31] employed an optimal energy harvesting strategy based on the Lyapunov optimization method and greedy algorithm to deal with the problem of multi-user and multi-task mobile edge cloud computing offloading. Zhang *et al.* [32] used the Lyapunov optimization method to derive the optimal strategy composed of CPU frequency and mobile device transmit power. Tran *et al.* [21] studied energy harvesting based on small cell

networks, joint load management and RA in mobile edge cloud systems and designed an algorithm to maximize the number of offloaded users. Wan *et al.* [33] proposed an energy-aware load balancing and a RA method to obtain the optimal solution, which can successfully handle the problem of optimal energy consumption of a hybrid robot on a production line in a candy packaging factory.

*Joint optimization on both time delay and energy consumption.* Xu *et al.* [8] realized the safe operation of the dual UAV-assisted MEC system by optimizing the delay of computing tasks and the energy consumption of the server. Song *et al.* [7] applied energy harvesting technology to the MEC system and reduced the time delay and service failure rate by optimizing the computational offloading strategy. Li *et al.* [34] proposed a two-stage algorithm based on greed and threshold to optimize MEC's TO scheme and RA strategy to minimize time delay and energy consumption in smart manufacturing.

Through the above analysis, it can be observed that the current research results mainly handle the TORA problem from the perspective of optimizing the task time delay and the server energy consumption to improve the QoS of MEC [35]. Many factors affect the TORA problem in improving the

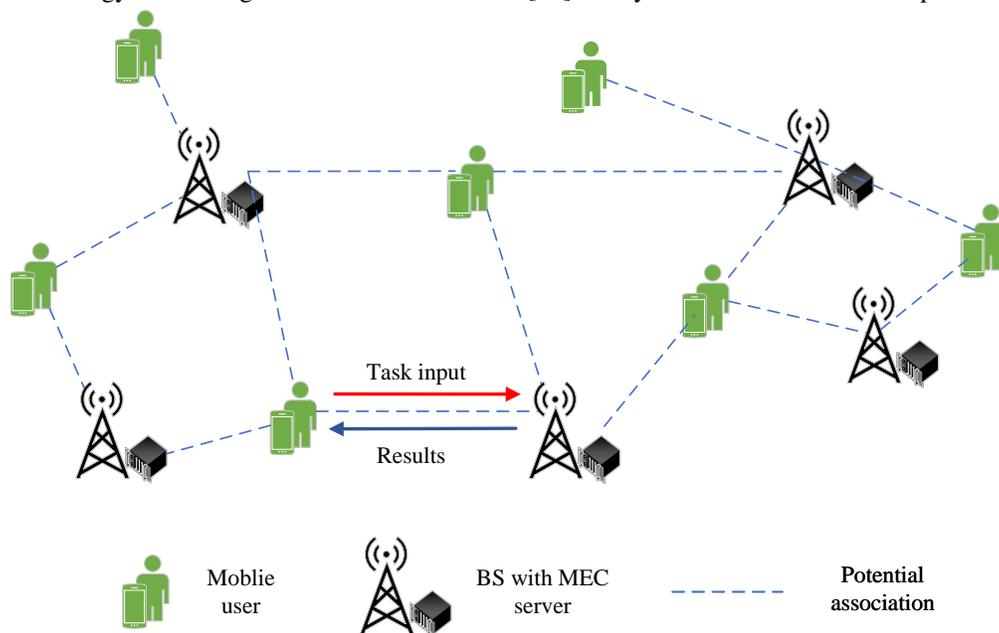


Fig. 1. A diagram of TORA problem in MSMECN environment.

QoS of MEC [21], [36]. Excepting the task time delay and the server energy consumption, some malicious computing tasks are continuously offloaded on a server with less computing power to make the rest of the servers idle, resulting in a waste of resources. Or some malicious MEC servers induce users to offload many tasks to themselves. However, it cannot provide these tasks with timely computing services, which will not only increase the task processing time delay and energy consumption but also brings a very poor user experience utility [21], [37]. These may have a significant impact on dealing with the TORA problem and improving the QoS of MEC.

Unlike the above research work, we will comprehensively consider the factors affecting the TORA problem in the MSMECN environment, including the task time delay, server energy consumption, trust metrics between task and server, and user experience utility. And a trusted many-objective TORA model is built. The specific model-building process will be described in the following text.

### III. PROBLEM DESCRIPTION AND MODEL BUILDING

In this section, we will discuss in detail the TORA problem in the MSMECN environment and the model building process.

#### A. Problem Description

To visualize the TORA problem, a diagram of the MSMECN environment is shown in Fig.1. It can be observed that each edge network base station (BS) is equipped with a MEC server, which is used to provide computation offloading services to resource-constrained mobile users such as smartphones. The communication between BS is realized through a wired link connection. And each user can access the BS through the wireless channels link. They can freely offload the task to be computed to the MEC server from one of the nearby BSs it can connect to [21]. To formally describe the operation process of MEC system, the set of users and MEC servers are denoted as  $\mathcal{U} = \{u_1, u_2, \dots, u_U | i \in [1, U]\}$  and  $\mathcal{S} = \{s_1, s_2, \dots, s_S | j \in [1, S]\}$ , respectively. Due to each BS being equipped with a MEC server, the MEC server  $s$  and BS  $s$  are represented by the same symbol. The MEC server and BS are used interchangeably to describe conveniently. And other key parameters involved in the TORA problem can be found in Table II.

Assumed that each user  $u_i$  has one computation task at a time under normal task computing, denoted as  $T_{u_i}$ , which is atomic and cannot be divided into subtasks. And we can describe the computational task by a two-tuple parameter,  $T_{u_i} = (D_{u_i}, C_{u_i})$  [7]. Where  $D_{u_i}$  describes the amount of input data required to transfer user computing tasks from the local device to the MEC server, including system settings, program codes, and input parameters.  $C_{u_i}$  describes the workload to complete user computing tasks. The two-tuple parameter value can be obtained by carefully analysing the corresponding computational tasks. And each computing task of the user can be processed in the local device or the relevant BS with the MEC server. When the user chooses to offload the task to the MEC server for processing, the computing task must be transmitted to the MEC server through the uplink. In our work,

TABLE II  
SYMBOLIC MEANING OF RELATED MODEL PARAMETERS

Name	Description
$\mathcal{U}$	Set of users
$\mathcal{S}$	Set of BS/MEC servers
$\mathcal{N}$	Set of available sub-bands of each BS
$B$	The operational frequency band
$T_{u_i}$	Computing tasks for $u_i$
$D_{u_i}$	Input data size of computing task $T_{u_i}$
$C_{u_i}$	Workload of complete computing task $T_{u_i}$
$R_{u_i, s_j}^{n_k}$	Uplink transmission rate of user $u_i$ link to server $s_j$
$\gamma_{u_i, s_j}^{n_k}$	SINR from user $u_i$ to server $s_j$ on sub-band $n_k$
$\sigma^2$	Signal processing noise power
$p_{u_i}$	Transmission power of user $u_i$
$P_{u_i}$	Maximum transmission power of user $u_i$
$x_{u_i, s_j}^{n_k}$	Task offloading indicator
$h_{u_i, s_j}^{n_k}$	Uplink channel gain between user $u_i$ to server $s_j$ on sub-band $n_k$
$f_{u_i, s_j}$	Computing resources that server $s_j$ allocates to task of user $u_i$
$f_{s_j}$	Maximum resources of server $s_j$

we consider the MEC system with [7] as the multi-access scheme in the uplink, in which the operational frequency band  $B$  is divided into  $N$  equal sub-bands of size  $W = B/N[\text{Hz}]$ . To ensure orthogonality of uplink transmissions between users associated with the same BS, each user is assigned to a sub-band [7]. Then each BS can service  $N$  users at most at the same time, and the set of available sub-bands of each BS is  $\mathcal{N} = \{n_1, n_2, \dots, n_N | k \in [1, N]\}$ . Then the uplink transmission rate  $R_{u_i, s_j}$  for user  $u_i$  link to server  $s_j$  can be expressed as follows.

$$R_{u_i, s_j} = W \log(1 + \gamma_{u_i, s_j}), \forall u_i \in \mathcal{U}, s_j \in \mathcal{S}, \quad (1)$$

where  $\gamma_{u_i, s_j} = \sum_{n_k \in \mathcal{N}} \gamma_{u_i, s_j}^{n_k}$ , and the signal-to-Interference-plus-Noise Ratio (SINR)  $\gamma_{u_i, s_j}^{n_k}$  from user  $u_i$  to MEC server  $s_j$  on sub-band  $n_k$  can be computed as

$$\gamma_{u_i, s_j}^{n_k} = \frac{p_{u_i} h_{u_i, s_j}^{n_k}}{\sum_{r \in \mathcal{S} \setminus \{s_j\}} \sum_{u_l \in \mathcal{U}_r} x_{u_l, r}^{n_k} p_{u_l} h_{u_l, s_j}^{n_k} + \sigma^2}, \quad (2)$$

$$\forall u_i \in \mathcal{U}, s_j \in \mathcal{S}, n_k \in \mathcal{N},$$

where  $p_{u_i}$  denotes the transmission power of user  $u_i$ .  $h_{u_i, s_j}^{n_k}$  denotes the uplink channel gain between user  $u_i$  and server  $s_j$  on sub-band  $n_k$ , which captures the effect of path-loss, shadowing, and antenna gain.  $x_{u_l, r}^{n_k}$  denotes the accumulated intra-cell interference from all the users associated with other servers on the same sub-band  $n_k$ .  $\mathcal{U}_r$  denotes the set of user  $u_i$  that offload their task to server  $s_j$ .  $\sigma^2$  is the background noise variance.

According to the link transmission mode in the MSMECN environment, each task  $T_{u_i}$  that the user  $u_i$  offload to the MEC server  $s_j$  will be executed. While the task processing mode of the MEC server provides users with convenient computing services, it is also restricted by many factors [21], [36]. Time delay and energy consumption are essential QoS indicators of MEC in the MSMECN environment. Since the user does not participate in the task processing process after offloading the computing task to the MEC server, it will cause the user to be unable to confirm whether the obtained computing task results meet their needs. In the complex MSMECN computing

environment, MEC servers and tasks may encounter some malicious behaviors to make the offloaded computing results untrust, which will make the computing task results obtained by the user untrust and the poor QoS of the MEC system [37]. Therefore, we also need to take the trust metrics between task and server and user experience utility as essential indicators to improve the QoS of MEC in handling the TORA problem [5].

Due to these factors may restrict and influencing each other, we can regard the TORA problem as a complex MaOP [5], [17], [21]. To describe this problem more clearly, we build a trust many-objective optimization model, comprehensively considering the task time delay, server energy consumption, trust metrics between task and server, and user experience utility factors. And specific model construction details will be described in the following text.

### B. Many-objective TORA Model Building

1) *Task Time Delay (Obj<sub>1</sub>)*: In the MSMECN environment, the time delay refers to the total time delay for processing user tasks, including TO transmission time delay, task execution time delay on the local or MEC server, and result return time delay [38]. Usually, because the amount of data of the task execution result is much smaller than the amount of data input by the task, the result return delay is often negligible [7]. To describe the delay, the TO transmission delay and the task execution delay on the MEC server need to be calculated [7]. Each user and BS have a separate antenna for uplink transmission for the TO transmission delay. When the computing task is selected to execute locally, the user  $u_i$  needs to offload the computing task  $T_{u_i}$  to the BS-associated MEC server via the wireless link. The corresponding transmission delay will occur during the uploading of the computing task  $T_{u_i}$ . According to the communication transmission mode, the uplink transmission delay of user  $u_i$  can be calculated as

$$Time_{T_{u_i}}^{up} = \sum_{s_j \in \mathcal{S}} \frac{x_{u_i, s_j} D_{u_i}}{R_{u_i, s_j}}, \forall u_i \in \mathcal{U}, \quad (3)$$

where  $x_{u_i, s_j} = \sum_{n_k \in \mathcal{N}} x_{u_i, s_j}^{n_k}$ . Due to the limited service resources, all computing tasks offloaded by users may not be executed immediately, generating processing delays for computing tasks. And the task  $T_{u_i}$  execution delay of user  $u_i$  is shown as

$$Time_{T_{u_i}}^{exe} = \sum_{s_j \in \mathcal{S}} \frac{x_{u_i, s_j} C_{u_i}}{f_{u_i, s_j}}, \forall u_i \in \mathcal{U}, \quad (4)$$

where  $f_{u_i, s_j}$  is the service computing resource that server  $s_j$  allocates to task of user  $u_i$ . Therefore, the total time delay of computing tasks for user  $u_i$  is shown as

$$\begin{aligned} \min Obj_1 = Time_{T_{u_i}}^{up} + Time_{T_{u_i}}^{exe} &= \sum_{s_j \in \mathcal{S}} x_{u_i, s_j} \left( \frac{D_{u_i}}{R_{u_i, s_j}} + \frac{C_{u_i}}{f_{u_i, s_j}} \right), \\ &\forall u_i \in \mathcal{U}, \end{aligned} \quad (5)$$

2) *Server Energy Consumption (Obj<sub>2</sub>)*: To calculate the server energy consumption of tasks, the cycle energy consumption calculation model is employed, i.e.,  $\mathcal{E} = \xi f^2$  [21], where,  $\xi$  denotes energy coefficient of chip structure and  $f$  denotes

CPU frequency. Therefore, the energy consumption of server  $s_j$  executing tasks can be calculated as follows.

$$Energy_{s_j}^{exe} = \xi f^2 \sum_{u_i \in \mathcal{U}} C_{u_i}, \forall s_j \in \mathcal{S}, \quad (6)$$

Each computing task of user  $u_i$  can be executed locally on the user's local device or offloaded to the MEC server. Due to the user's local device being generally equipped with fixed power supply equipment, the energy consumption of local computing tasks is generally not calculated [21]. When the task is offloaded to the MEC server for processing. The energy consumption will be generated in the uplink task inputs transmission [21], which can be described as

$$Energy_{s_j}^{up} = \sum_{u_i \in \mathcal{U}} p_{u_i} Time_{u_i}^{up}, \forall s_j \in \mathcal{S}, \quad (7)$$

Therefore, the total server energy consumption of computing tasks for server  $s_j$  is shown as follows.

$$\min Obj_2 = Energy_{s_j}^{up} + Energy_{s_j}^{exe}, \forall s_j \in \mathcal{S}, \quad (8)$$

3) *Trust Metrics between Task and Server (Obj<sub>3</sub>)*: Due to the service of the MEC malicious server being hidden, the user does not participate in the specific process of task executing after uploading the offloading task to the MEC server in practice [7]. The result will be caused that the user cannot predict whether the MEC server successfully processes the offloaded task or may do malicious service. And malicious servers may have a certain probability of successfully executing tasks. At the same time, normal servers can also fail to process the offloaded tasks due to some uncontrollable external reasons [41]. It is uncertain and random for the server to process the offloaded tasks successfully, which has a negative impact on the whole MEC system [7]. Consequently, it is essential to measure the service trust of the task and server to ensure that the task time delay and server energy consumption are within tolerance, which is important in improving the QoS of MEC. Specifically, When there is no historical task contact, all servers are regarded as the normally trusted servers at the beginning, i.e., the trust value of all servers is set as  $Obj_3 = 1$ . And the trust value of the server should dynamically change during the contact process between the task  $u_i$  and the server  $s_j$  [7]. The specific trust metrics model can be described as

$$\max Obj_3 = \begin{cases} Obj_3 - Trust_{T_{u_i}, s_j}^1, rand \geq Prob_{u_i, s_j} \\ Obj_3 + Trust_{T_{u_i}, s_j}^2, rand < Prob_{u_i, s_j} \end{cases}, \quad (9)$$

$\forall u_i \in \mathcal{U}, s_j \in \mathcal{S}$ ,

where  $rand$  is a random number in the  $[0, 1]$ .  $Prob_{u_i, s_j}$  is a dividing line between offloading success and failure. It is worth noting that the probability of offloading success as a malicious MEC server should be greater than the failure probability. And the success probability is less than the failure probability for the normal MEC server.  $Trust_{T_{u_i}, s_j}^1$  and  $Trust_{T_{u_i}, s_j}^2$  denote the reward and penalty values for successful and unsuccessful task processing, respectively.

4) *User Experience Utility (Obj<sub>4</sub>)*: After the user sends the offloading task request, it is expected that the task to be processed can be offloaded computing safely and timely. And the results of offloading computing are fed back in time.

However, encountering a malicious server will lead to failure in TO, which will affect the stability of the whole MEC system [21]. Generally, the user experience utility mainly manifests in task time delay, energy consumption and trusted TO result. If the requested task cannot be processed in time or has a large time delay or energy consumption in case of successful offload processing, it will cause users to have a bad service experience. Naturally, minimizing the time delay and energy consumption and maximizing the trust metrics will be of great significance in gaining a better user experience. Based on the above task time delay, server energy consumption and trust metrics between task and server analysis, their relative improvement should be  $Time_{T_{u_i}} = \frac{Time_{T_{u_i}}^{exe} - Time_{T_{u_i}}^{up}}{Time_{T_{u_i}}^{exe}}$ ,  $Energy_{s_j} = \frac{Energy_{s_j}^{exe} - Energy_{s_j}^{up}}{Energy_{s_j}^{exe}}$  and  $Trust_{T_{u_i}, s_j} = Obj_3$ , respectively. And the user experience utility can be described as

$$\max Obj_4 = \sum_{s_j \in \mathcal{S}} (\lambda_{u_i}^1 Time_{T_{u_i}} + \lambda_{u_i}^2 Energy_{s_j} + \lambda_{u_i}^3 Trust_{T_{u_i}, s_j}),$$

$$\forall u_i \in \mathcal{U}. \quad (10)$$

where  $\lambda_{u_i}^1, \lambda_{u_i}^2, \lambda_{u_i}^3 \in [0, 1]$  denote the user preferences of user  $u_i$  for time delay, energy consumption and trust metrics, and  $\lambda_{u_i}^1 + \lambda_{u_i}^2 + \lambda_{u_i}^3 = 1$ .

### C. Model constraints

For each task  $T_{u_i}$  of the user  $u_i$ , it can be either executed locally or offloaded to at most one MEC server [7]. The binary TO variable of incorporating the uplink sub-band scheduling can be described as  $x_{u_i, s_j}^{n_k}, \forall u_i \in \mathcal{U}, s_j \in \mathcal{S}, n_k \in \mathcal{N}$ . Noting that  $x_{u_i, s_j}^{n_k} = 1$  indicates that task  $T_{u_i}$ , from user  $u_i$  is offloaded to MEC server  $s_j$  on sub-band  $n_k$ , and  $x_{u_i, s_j}^{n_k} = 0$  otherwise. And the TO decision of user  $u_i$  should meet the constraints  $\sum_{u_i \in \mathcal{U}} \sum_{s_j \in \mathcal{S}} x_{u_i, s_j}^{n_k} \leq 1, \forall u_i \in \mathcal{U}$ . Additionally, the set of users offloading their tasks to server  $s_j$  is  $\mathcal{U}_r = \{u_i \in \mathcal{U} | \sum_{n_k \in \mathcal{N}} x_{u_i, s_j}^{n_k} = 1\}$ .

In particular, we consider that each user and BS have a single antenna for uplink transmissions in this paper [7]. And the user-server association usually occurs on a large time scale, much larger than the time scale of small-scale fading. Hence, the influence of fast fading is average in the correlation process [21]. Assumed that  $P_{u_i}$  is the maximum transmission power of current user  $u_i$  offload task to MEC server, and  $\mathcal{U}_{off} = \bigcup_{s_j \in \mathcal{S}} \mathcal{U}_r$  is the set of users that offload their tasks.

The constraints should be satisfied,  $0 < p_{u_i} \leq P_{u_i}, \forall u_i \in \mathcal{U}_{off}$ . And  $p_{u_i} = 0, \forall u_i \notin \mathcal{U}_{off}$ .

After the TO decision of the user's computing task is determined, the MEC system will start to perform the RA strategy. The user will offload the computing task to the server through the uplink according to the TO decision. After receiving the offloaded task from the user,  $u_i$ , the server will execute the task  $T_{u_i}$  and return the output result to the user [39], [40]. For the RA strategy of each computing task,  $f_{u_i, s_j}$  is the amount of computing resource that MEC server  $s_j$  allocates to task  $T_{u_i}$  offloaded from user  $u_i$ . Clearly,  $f_{u_i, s_j} = 0, \forall u_i \notin \mathcal{U}$ .

And a feasible RA strategy should meet the resource constraint requirement. That is, the allocated computing resources by the BS/MEC server should be greater than 0 but cannot be greater than the maximum number of resources  $f_{s_j}$  owned by the server  $s_j$ , which expressed as,  $f_{u_i, s_j} > 0$  and  $\sum_{u_i \in \mathcal{U}} f_{u_i, s_j} \leq f_{s_j}$ .

After describing the built many-objective TORA model, how to adequately address the TORA model is also very important in improving the QoS of MEC. In this paper, we decompose the original TORA problem into TO, and RA problem [42]. For the TO problem, the user uploads the task to the corresponding server for calculation through the uplink [7]. The RA problem is based on the TO decision determined by the user task to the MEC server. It allocates the corresponding service resources for the computing task through the downlink, such as computing, cache and network transmission resources [7]. To address and obtain the required optimization solution, the MaOEA-CCIL is designed. TO and RA are treated for optimization in the algorithm, respectively. Initially, a clustering mechanism divides the TO decision into elite and ordinary decision solutions. Then, to improve the CaD of elite decision solutions, the incremental learning selection mechanism is introduced into the reference point redistribution. Finally, based on the optimal offloading decision, the RA problem continues to be optimized by employing the KTT condition method. The specific explanation will be expanded in the following text.

## IV. DESIGNED ALGORITHM

In this paper, we regard the TORA problem as a MaOP. And many scholars have proposed various excellent algorithms to solve MaOPs in the past decade [18]. These algorithms aim to solve the balance problem in CaD based on the distance or angle selection mechanism. However, traditional distance or angle evaluation shows that individuals have weak selection pressure in the later stage of population evolution, such as PBI [18], [19] and PDM [20]. Sometimes, they will naturally prefer concave optimization problems and even negatively influence convex optimization problems. Therefore, whether the distance or angle selection mechanism chooses an elite solution, it is to make the final population solution distributed near the real front as much as possible [19]. At the same time, the CaD of the solutions should be considered to meet the decision-making needs of the actual problems.

Different from the distance and angle selection mechanism for selecting an optimal solution to the TORA problem, a fitness selection mechanism is adopted in this paper. The factors affecting the TORA problem restrict and contradict each other, regarded as a MaOP. The factors of time delay, energy consumption, trust metrics between task and server, and user experience utility are considered comprehensively as the four objectives to be optimized. To address and obtain the required optimization solution, the MaOEA-CCIL is designed. TO and RA are treated for optimization in the algorithm, respectively. In the beginning, a clustering mechanism divides the TO decision into elite and ordinary decision solutions. Then, the incremental learning selection mechanism is introduced into the reference point redistribution to improve the CaD of elite

decision solutions in the real Pareto-optimal front (PF) direction [19]. Finally, based on the optimal offloading decision, the RA problem continues to be optimized by employing the KTT condition method.

### A. Cascading Clustering Selection Mechanism (CCSM)

In CCSM, a simple identification will be realized to divide the population into elite and ordinary solutions by employing a non-dominated sort (NS) mechanism [43]. Specifically, the elite solution is the individual of the first front, close to the real PF [44]. Other individuals are regarded as ordinary solutions. The role of the elite solution is to guide the evolution of ordinary solutions in a better direction. The purpose of identification is the ordinary solution keeps learning under the guidance of the elite solution to become the elite solution eventually or close to the elite solution [19].

After identification of elite and ordinary solutions, the selected elite solution should have been attached to the nearest reference vector in principle, which shows good CaD. However, due to the tired selection pressure of the NS mechanism at the late stage of population evolution, it has been unable to meet the requirements of CaD in solving MaOPs [19], [43]. In other words, the identified elite solutions will likely gather in the local PF and even remotely from the real PF. To overcome the challenges, the idea of clustering is introduced to make up for the deficiency [44]. Concretely, a clustering mechanism is implemented inside the elite solutions. A leader should be promoted in each cluster as the sole individual to evolve in the real PF direction, which can realize the learning from ordinary solutions to elite solutions. To find the leader in each cluster, an effective selection mechanism needs to be used in the CCSM strategy.

Compared with other selection mechanisms, balanced fitness estimation (BFE) achieves better performance balancing CaD because of the novel fitness selection mechanism [19], [43], [44]. Unlike the traditional distance and angle selection mechanism, the BFE can eliminate the impact of drastic changes in dimensions and objectively reflect the CaD state of solutions. Assumed that  $\mathcal{G} = \{g_1, g_2, \dots, g_G\}$ ,  $t \in [1, G]$  is the population solution set with  $G$  individuals. The BFE is described as

$$Value(g_t, \mathcal{G}) = \omega_1 \cdot D_{cv}(g_t, \mathcal{G}) + \omega_2 \cdot D_{cd}(g_t, \mathcal{G}), \quad (11)$$

where  $D_{cv}$  and  $D_{cd}$  are used to describe the solutions degree of CaD, respectively;  $\omega_1$  and  $\omega_2$  are two dynamic adjustment factors, which ensure that CaD is considered constantly in the process of searching for solutions. Their principles can be described in [43]. For the calculation of  $D_{cv}$ , it can be explained as

$$D_{cv}(g_t, \mathcal{G}) = 1 - \text{sqrt}\left(\frac{\sum_{m=1}^M (F'_m(g_t))^2}{M}\right), \quad (12)$$

where  $F'_m(g_t)$  is the normalized objective value of  $F_m(g_t)$ .  $M$  is the number of objectives. From [19], the  $D_{cd}(g_t, \mathcal{G})$  can be

described by the normalized value of SDE. Assumed that the  $CD$  express the SDE value, it can be calculated as follows.

$$CD(g_t) = \min_{g_t, l \in \mathcal{G}, l \neq t} \text{sqrt}\left(\sum_{m=1}^M cd(F'_m(g_t), F'_m(g_l))^2\right), \quad (13)$$

$$D_{cd}(g_t, \mathcal{G}) = \text{Norm}(CD(g_t))$$

where  $\text{Norm}(\cdot)$  is a normalization operation [19].  $g_t$  and  $g_l$  denote two individuals in a population  $\mathcal{G}$ , and

$$cd(\cdot) = \begin{cases} F'_m(g_l) - F'_m(g_t), & \text{if } F'_m(g_l) > F'_m(g_t), \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

At the same time, the elite solution with the best BFE value is taken as the center of the corresponding cluster. All ordinary solutions are assigned to a cluster with the nearest cluster center. For each cluster, the ordinary solutions are sorted in descending order according to their BFE values in the corresponding cluster, which achieve an elite solution to lead the evolution of the ordinary solution. The pseudo-code of the CCSM strategy is described in Algorithm 1, where the  $\mathcal{G}$  and  $\mathcal{Z}$  are described as population and reference vectors, respectively.  $G$  is the required population size of solutions.  $\mathcal{A}$  is used to store the selected elitist solution sets. It is noteworthy that the cluster center of each cluster is not fixed. When the BFE values of all individuals in the cluster are not better than the current cluster center, the current cluster center elitist solution is still preserved, which means that the current cluster center elitist solution is still the closest individual to PF and retained as a leader to guide all ordinary solution in the corresponding cluster to evolve closer to the real PF while ensuring CaD. Meanwhile, suppose an elite solution appears to challenge the current cluster center successfully. In that case, the current cluster center will be replaced with a successful elite solution to the challenge and will continue to take responsibility as the cluster center. In addition, to ensure that the elite solution in each cluster is retained and the genetic next generation, a round-robin selection method is adopted. For each selection round, the best individual in each cluster is selected and added to the next generation's solution set based on the best BFE value. This process until the next generation's population size reaches the requirement.

### B. Reference Point Incremental Learning (RPIL)

The selected individuals have been almost all elite solutions and approximately evenly distributed near the current PF after performing the CCSM strategy. However, reaching the final state where all elite solutions are attached to the corresponding reference vector [43] isn't easy. Some reference points are not activated in the process of population evolution, which weakens the CaD ability of the population to some extent. Therefore, to further improve the CaD of the population, the RPIL strategy continues to be implemented [20]. The RPIL strategy aims to generate more evenly distributed reference points inside these practical areas and reduce the outsiders; there will be more reference vectors intersecting with the true PF while the efficiency can be maintained by reducing the ineffective outsiders [45]. Based on the principle that reference vector activities can reflect the accurate PF distribution in the

---

**Algorithm 1** CCSM ( $\mathcal{G}, \mathcal{Z}$ )

---

**Begin**  
 Individual identification  $\mathcal{G}$  is divided into elite  $\mathcal{G}_E$  and ordinary  $\mathcal{G}_O$  by employing NS mechanism;  
 Cluster and attach  $\mathcal{G}_E$  to nearest reference vector  $\mathcal{Z}$ ;  
**For** each cluster **do**  
     Calculate and sort descending BFE value of each  $\mathcal{G}_E$ ;  
     Select the best value as the corresponding cluster center;  
**End for**  
 Make  $\mathcal{G}_O$  attach to the nearest cluster;  
**For** each  $\mathcal{G}_O$  **do**  
     Calculate and sort descending BFE value of each  $\mathcal{G}_O$ ;  
     Compare and select the best value as the corresponding cluster center;  
**End for**  
 $\mathcal{A} = \phi$ ;  
**While**  $|\mathcal{A}| < G$  **do**  
     Select each cluster with the best BFE value into  $\mathcal{A}$ ;  
**End while**  
 Output  $\mathcal{A}$ ;  
**End**

---

objective space, a classification selector in RPIL is employed to distinguish between effective and invalid reference points. When the classifier identifies the reference point as an effective reference point, the reference point will be evaluated and get a high score, which is regarded as an effective tool to distinguish positive reference points with a higher reference point density space [20].

In the RPIL strategy, the state sampler is used to iteratively learn the reference vector's current and historical activity samples. In addition, if the activity of all reference vectors does not change within a certain number of iterations, the state sampler considers it stable, which means that the current PF has been stably distributed around the active reference vector [44]. Then, sampling based on steady-state is employed in the same way of [45] to train and generate more reference points. It is noted that the RPIL strategy has been proved that highly compatible with the adaptation methods that adapt to the curvature of the true PF. The incremental learning deals with the problems with partial PFs, combined with other adaptation techniques, which can boost the performance of the reference vector-based algorithms [45].

The whole process pseudocode of the RPIL strategy can be described in Algorithm 2, where  $\mathcal{Z}_A$  and  $\mathcal{Z}_{IA}$  are described as an active and inactive reference point, respectively.  $G$  is the required population size of solutions.  $\mathcal{Z}_R$  is used to store the selected reference points. And  $\delta$  is a dynamic threshold parameter. On the one hand, the threshold  $\delta$  of dynamic change can effectively improve the accuracy of a classifier to identify the effective region and eliminate the influence of the decrease of the reference vector caused by the increase of training sample density. On the other hand, the threshold  $\delta$  of dynamic change will not lead to the excessive decrease of elite solution scores with the incremental learning of the classifier. It is helpful to improve the probability of reference

---

**Algorithm 2** RPIL ( $\mathcal{Z}, G$ )

---

**Begin**  
**While** Sampler is not stable state **do**  
     Train a classifier model for identifying the  $\mathcal{Z}_A$  and  $\mathcal{Z}_{IA}$  by employing the SVM method based on reference point  
     sample  $\mathcal{Z}$ ;  
      $\mathcal{Z}_R = \phi$ ;  
     **If**  $|\mathcal{Z}_A| < G$  **do**  
         Generate  $G$  new reference points  $\mathcal{Z}^*$ ;  
         **For**  $t = 1 : G$   
             Employ the trained classifier model to score the high-density  $\mathcal{Z}_t^*$ ;  
             **If**  $Score_{\mathcal{Z}_t^*} > \delta$   
                 Add the  $\mathcal{Z}_t^*$  into the  $\mathcal{Z}_R$ ;  
             **End if**  
         **End for**  
     **End if**  
     **End while**  
      $\mathcal{Z} = \mathcal{Z}_A \cup \mathcal{Z}_R$ ;  
     Output the new reference point set  $\mathcal{Z}$ ;  
**End**

---

vector activation and enhance the robustness of incremental learning [20].

### C. Algorithm Framework

Our algorithm initializes a series of related parameters, including the populations  $\mathcal{G}$  with  $G$  individuals and corresponding reference point  $\mathcal{Z}$ . In the start, the CCSM and RPIL strategy is executed in turn. Next, the offspring and new reference points are generated. Then, general genetic operations improve population diversity, including simulated binary crossover (SBX) and Polynomial mutation (PM). Finally, the CCSM strategy is again employed to select elite solutions based on collecting parents, offspring, and new reference points. The process is cycled until the stop condition is satisfied.

A relatively good offloading solution to the TO problem can be obtained through the above cyclic iteration. Based on the optimal offloading decision, the RA problem continues to be optimized by employing the KTT condition method. Due to space limitations, the operation principle of KKT will not be described in detail here. The specific principle can be found in [21]. Instead, more description about the principle of MaOEA-CCIL is described in detail. The pseudo-code of MaOEA-CCIL is described in Algorithm 3, where  $\mathcal{G}$  and  $\mathcal{Z}$  are described as population and reference points.  $G$  is the required population size of solutions, i.e.,  $Q_1$  and  $Q_2$  denote the offspring generated by different operations.

### D. Complexity analysis

The designed algorithm's stop condition is set to the maximum iteration number. Some leading operators include CCSM, RPIL and general genetic operations. The first elite solution is selected for the CCSM strategy to use the NS

---

**Algorithm 3** Principle of MaOEA-CCIL
 

---

**Begin**

 Initialize the populations  $\mathcal{G}$  with  $G$  individuals, reference vector  $\mathcal{Z}$  and the related parameters;

**While** not meeting the stopping criteria **do**
 $Q_1 = \text{CCSM}(\mathcal{G}, \mathcal{Z});$ 
 $New\_Z = \text{RPIL}(\mathcal{Z}, G);$ 
 $Q_2 = \text{SBX\_PM}(Q_1);$ 
 $\mathcal{G} = \text{CCSM}(Q_1 \cup Q_2, New\_Z);$ 
**End while**

 Output  $\mathcal{G}$ ;

**End**


---

method. The time complexity of the NS method is  $O(MG)^2$ . Then the fitness selection mechanism is adopted to continue selecting elite solutions in their respective cluster. The time complexity of the fitness selection mechanism is  $O(MN)^2$  [19]. Therefore, the time complexity of the whole selection mechanism is  $O(MN)^2$ . In addition, the time complexity of residual operations has been analyzed in literature [20], [44], and their value, in turn, are  $O(MN)^2$ ,  $O(MN)^2$  and  $O(MN)^2$ . In final, it's evident that the time complexity of our approach is  $O(MN)^2$ .

## V. SIMULATION EXPERIMENT

### A. Environment and parameter setting

TABLE III  
RELATED PARAMETER VALUE

Name	Value
Number of users	$U=6$
Number of servers	$S=4$
Number of sub-band	$N=2$
Task input size	$D_{u_i} = 420KB$
Workload to complete user computing tasks	$C_{u_i} = 1000u_i$
Energy coefficient	$\xi = 5 \times 10^{-4}$
CPU capability of each MEC server and of each user	$f_{s_j} = 20GHz, \forall s_j \in S, f_{u_i} = 1GHz$
Maximum transmission power	$P_{u_i} = 20dBm$

Simulation experiments are presented to evaluate the performance of our proposed approach in the TORA problem. In this paper, the experiment is carried out on the windows 10 system, equipped with GPU RTX 3090 Ti and CPU AMD 5950x, with a main frequency of 3.4GHz. And it is run in an environment equipped with Python version 3.7. Some function parameters and problem parameters are described in detail.

1) *Function parameters*: To effectively verify the performance of the designed algorithm, some advanced MaOEAs are introduced and compared with our approach, including MaOEA based on coordinated selection strategy (MaOEA-CSS) [46], MaOEA based on improved decomposition strategy (IDBEA) [47], MaOEA based on fitness assessment mechanism (NMPSO) [43], tri-goal evolution framework for constrained MaOEA (TiGE2) [48], strength Pareto evolutionary algorithm based on reference direction (SPEA/R) [49]. In addition, all original parameters of the involved comparison algorithms will be employed to ensure a fair result. The crossover and mutation probability is set as 1 and  $1/D$  (where

$D$  is the dimension of the objective variable), respectively. The distribution indexes of mutation and crossover are set to be 20. Meanwhile, the threshold  $\delta$  is essential for selecting potentially effective reference points in our approach. For the threshold  $\delta(t)$  for potentially effective reference points, the  $t = 2G$  is proved to be valid, i.e., after the reduction, there will be at least  $2G$  points left (if the total number is below  $2G$ , all will be kept) [20].

For the benchmark problem, the test set MaF [50] is employed to measure algorithm performance, which is an improved version of the widely used test functions DTLZ [51]. These test functions have been proved to effectively verify the algorithm performance under complicated Pareto solutions, irregular PF and multi-modal. Specifically, each problem is tested and executed on different objectives,  $M = 4, 6, 8, 10$ . The population sizes are unified, set to  $G = 240$ . Each test problem is run independently 20 times to make the results convincing. The maximum number of iterations is 10000.

2) *Problem parameters*: To describe the TORA problem in the MSMECN environment, a multi-cell cellular system consisting of multiple hexagonal cells with a BS in the center of each cell is considered [21]. The distance of adjacent BS is set at  $1km$ . And the users and BS/MEC server use a single antenna for uplink transmission and reception, respectively. The multipath loss of uplink data transmission is generated using the model  $L[dB] = 140.7 + 36.7 \log_{10} d_{[km]}$ , the path loss index is 4, and the log-normal shadowing standard deviation is set to 8dB [21]. For computing tasks, the application of face detection and recognition in airport security and surveillance is considered in this paper [36], which highly benefits from collaboration between local devices and the MEC server. The MEC system bandwidth is set to  $B = 20MHz$  and the background noise variance is assumed to be  $\sigma^2 = -100dBm$ . The preference parameters of model can be set as  $D_{u_i} = 420KB, \lambda_{u_i}^1 = 0.4, \lambda_{u_i}^2 = 0.3, \lambda_{u_i}^3 = 0.3$  [21]. And the  $Trust_{T_{u_i}, s_j}^1 = 0.4$  and  $Trust_{T_{u_i}, s_j}^2 = 0.1, \forall u_i \in \mathcal{U}$ .  $Prob_{u_i, s_j} = 0.8$  for malicious server, but when server is normal,  $Prob_{u_i, s_j} = 0.1$  [21]. Users are randomly and evenly distributed within the coverage, and the number of sub-bands  $N$  equals the number of users per unit. The relevant parameters of computing resource can be found in Table III.

### B. Performance metrics

To effectively measure the superiority of the results obtained, it is necessary to introduce some evaluation indicators. These performance metrics reflect the algorithm's quality and are widely employed in performance assessment for MaOPs [19], [44]. In general, the excellent algorithm is hoped that each individual can converge as soon as possible and approach the true PF. Meanwhile, the whole population is expected to be evenly distributed and can cover the entire true PF at the end of the iteration. The description can be found in the following content.

1) *Inverse generation distance (IGD)*: IGD is a comprehensive evaluation index widely employed to measure the CaD of the solution obtained by MaOEAs [44]. Supposed  $PF^*$

TABLE IV  
IGD VALUE OF DIFFERENT ALGORITHMS ON THE MAF USING IGD

Problem	$M$	MaOEA-CSS	IDBEA	NMPSO	TiGE2	SPEA/R	MaOEA-CCIL
MaF1	4	<b>6.3757e-2 (5.39e-4)</b> +	1.0350e-1 (1.41e-3) -	7.7837e-2 (1.67e-3) +	1.7526e-1 (1.17e-2) -	1.7080e-1 (4.68e-3) -	9.6171e-2 (5.26e-3)
	6	<b>1.3022e-1 (1.01e-3)</b> +	2.1643e-1 (2.09e-3) +	1.7881e-1 (7.43e-3) +	3.5183e-1 (2.03e-2) -	2.8628e-1 (3.53e-2) -	2.2447e-1 (7.41e-3)
	8	<b>1.8530e-1 (2.42e-3)</b> +	2.6268e-1 (6.87e-3) +	2.4438e-1 (1.14e-2) +	3.8809e-1 (2.54e-2) -	4.7873e-1 (3.96e-2) -	3.6751e-1 (2.87e-2)
	10	<b>2.2165e-1 (1.19e-2)</b> +	3.1699e-1 (1.48e-2) -	2.8870e-1 (1.03e-2) -	4.3167e-1 (5.24e-2) -	4.6310e-1 (5.12e-2) -	2.4601e-1 (3.09e-3)
MaF3	4	3.1163e+3 (1.21e+3) -	5.6976e+3 (2.44e+3) -	1.4533e+6 (3.10e+6) -	2.1379e+7 (6.65e+7) -	2.6745e+4 (2.29e+4) -	<b>2.1605e+3 (1.76e+3)</b>
	6	5.1337e+3 (1.82e+3) -	2.7865e+5 (8.09e+5) -	1.6895e+7 (2.36e+7) -	8.0325e+6 (2.70e+7) -	1.6041e+5 (5.55e+5) -	<b>2.5226e+3 (9.32e+2)</b>
	8	<b>5.9244e+3 (2.01e+3)</b> ≈	5.3214e+6 (1.36e+7) -	5.1008e+7 (4.79e+7) -	2.8659e+7 (7.17e+7) -	7.0959e+6 (2.52e+7) -	7.6988e+3 (2.92e+3)
	10	<b>6.5725e+3 (1.91e+3)</b> +	3.7299e+6 (6.21e+6) -	5.4332e+7 (3.53e+7) -	6.7197e+7 (1.12e+8) -	1.4589e+7 (6.44e+7) -	3.3947e+4 (1.99e+4)
MaF4	4	2.1412e+2 (5.26e+1) -	5.9293e+2 (8.32e+1) -	1.1108e+3 (7.54e+1) -	8.2788e+2 (2.76e+2) -	2.3096e+2 (6.22e+1) -	<b>1.3268e+2 (4.85e+1)</b>
	6	8.1008e+2 (2.24e+2) -	1.2649e+3 (3.17e+2) -	5.1845e+3 (4.39e+2) -	4.5203e+3 (1.28e+3) -	1.1212e+3 (3.14e+2) -	<b>2.9767e+2 (1.14e+2)</b>
	8	3.4538e+3 (1.02e+3) -	7.6116e+3 (1.59e+3) -	2.2994e+4 (3.39e+3) -	2.6098e+4 (8.97e+3) -	7.0537e+3 (3.14e+3) -	<b>1.6609e+3 (5.18e+2)</b>
	10	1.1891e+4 (3.77e+3) -	2.6136e+4 (5.90e+3) -	1.0104e+5 (1.03e+4) -	1.0335e+5 (3.15e+4) -	3.7600e+4 (1.58e+4) -	<b>4.6148e+3 (1.71e+3)</b>
MaF5	4	1.0796e+0 (1.06e-1) -	9.8663e-1 (6.47e-1) -	8.6049e-1 (2.75e-2) -	1.3484e+0 (1.21e-1) -	8.1911e-1 (2.49e-2) -	<b>7.3472e-1 (9.91e-3)</b>
	6	1.0328e+1 (1.52e+0) -	6.6458e+0 (2.85e+0) -	5.1286e+0 (1.89e+0) ≈	6.9691e+0 (4.92e-1) -	5.0896e+0 (3.27e-1) -	<b>4.8332e+0 (1.44e+0)</b>
	8	5.6368e+1 (6.73e+0) -	2.0616e+1 (7.49e+0) -	<b>1.6011e+1 (1.73e+0)</b> ≈	2.6598e+1 (2.17e+0) -	1.9300e+1 (1.21e+0) -	1.6520e+1 (4.31e+0)
	10	2.4498e+2 (2.21e+1) -	8.7662e+1 (2.29e+1) -	<b>6.4197e+1 (2.95e+1)</b> +	1.1326e+2 (9.92e+0) -	9.8290e+1 (5.93e+0) -	8.1425e+1 (4.04e+1)
MaF6	4	6.9572e-1 (2.10e-1) -	1.9069e+0 (5.76e-1) -	<b>1.1409e-2 (2.93e-3)</b> +	3.5798e+0 (1.45e+0) -	1.3691e-1 (4.76e-2) -	2.7032e-2 (1.62e-2)
	6	1.3198e+0 (3.87e-1) -	1.2258e+0 (4.27e-1) -	4.3717e+0 (3.45e+0) -	6.3160e+0 (2.48e+0) -	1.3074e-1 (3.11e-2) -	<b>2.2833e-2 (6.97e-3)</b>
	8	1.9501e+0 (6.51e-1) -	1.7309e+0 (5.99e-1) -	5.6005e+0 (2.54e+0) -	9.0014e+0 (3.89e+0) -	2.7718e-1 (2.19e-1) -	<b>4.7963e-2 (3.67e-2)</b>
	10	2.3762e+0 (7.12e-1) -	2.9565e+0 (1.22e+0) -	6.1643e+0 (2.10e+0) -	1.0253e+1 (3.91e+0) -	8.8213e-1 (5.29e-1) -	<b>6.9063e-2 (8.79e-2)</b>
+/-/≈		5/14/1	2/18/0	5/13/2	0/20/0	0/20/0	

represents the approximate frontier solution set obtained by the algorithm. And the IGD value of  $PF^*$  is defined as

$$IGD(PF^*) = \frac{\sqrt{\sum_{t=1}^{G'} Ed_{g_t}^2}}{PF}, \quad (15)$$

where  $G'$  is the number of solutions in the true  $PF$ , and  $Ed_{g_t}$  represents the Euclidean distance from the solution  $g_t$  of true  $PF$  to the closest solution of the approximated  $PF^*$ . And the value of IGD is smaller. The performance is better.

2) *Hypervolume (HV)*: As another comprehensive evaluation index, HV is employed to measure the CaD of a solution obtained by MaOEAs. Assumed that  $Z^r = (Z_1^r, \dots, Z_M^r)$  is the reference point set in the objective space, where  $M$  is the number of objectives,  $m \in [1, M]$ . Let  $F_m(g_t)$  denote the  $m$ -th fitness value of solution  $g_t$ , which is dominated by all the Pareto-optimal objective vectors, and  $[F_m(g_t), Z_m^r]$  denote the hypercube, which can be constructed with the reference point  $Z_m^r$  and the solution value  $F_m(g_t)$  as two diagonal corners of the hypercube. Further let  $PF^*$  denote the approximation set of solutions  $g_t$  [44]. Then, the HV metric of the approximate front-surface solutions in  $PF^*$  and the reference point set  $Z^r$  can be computed as follows

$$HV(PF^*) = vol\left(\bigcup_{g_t \in PF^*} [F_1(g_t), Z_1^r] \times \dots \times [F_m(g_t), Z_m^r]\right), \quad (16)$$

where  $vol(\cdot)$  is the Lebesgue measure which calculates the hypervolume of all the objectives' hypercubes. And the  $HV$  value is larger, so the approximation set is more favourable.

3) *Coverage over the Pareto-optimal front (CPF)*: The CPF describes the coverage of solution sets by evaluating the proportion of the dominated solutions [52]. That is, by determining how many the evolutionary algorithms can search more Pareto solutions in the whole solution space, which can be defined as follows.

$$CPF(PF_X^*, PF_Y^*) = \frac{|\{g_t \in PF_Y^* \mid \exists g_t \in PF_X^* : g_t > g_t\}|}{|PF_Y^*|}. \quad (17)$$

where  $PF_X^*$  and  $PF_Y^*$  are two approximate Pareto solutions sets, respectively.  $g_t$  and  $g_t$  are individuals in  $PF_X^*$  and  $PF_Y^*$ , respectively. Symbols  $>$  describe the dominant relationship between individuals. Noted that the value of  $CPF(PF_X^*, PF_Y^*) = 1$ , all the solutions in  $PF_X^*$  are dominated individuals. Otherwise (i.e.,  $CPF(PF_X^*, PF_Y^*) = 0$ ), all the solutions in  $PF_Y^*$  are non-dominated individuals. A larger CPF value will obtain the evenness and spread of the solution set.

### C. Simulation result

In this section, the simulation results are divided into two subsections. First, to verify the superiority of the algorithm, the designed algorithms are compared and analyzed on benchmark function with the existing five MaOEAs. In addition, to verify the effectiveness of the TORA model, they are used separately to handle the TORA problem. And these results are described in detail in the following text.

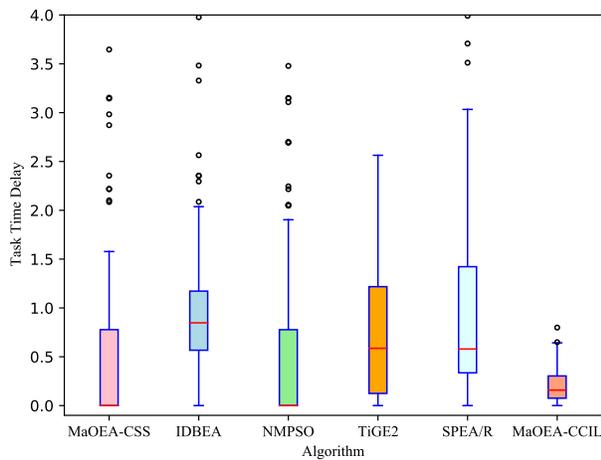
1) *Performance comparison on benchmark function*: Table IV presents the IGD values gained by MaOEA-CSS, IDBEA, NMPSO, TiGE2, SPEA/R, and MaOEA-CCIL on the MaFs. The best result on different test instances is highlighted. Based on Wilcoxon's rank-sum and Friedman statistical test, the labels '+', '-', and '≈' show that the results acquired by different algorithms are significantly superior, worse, or equal to those obtained by our approach, respectively. The results show that the proposed MaOEA-CCIL performs better on the MaF benchmark set. For MaOEA-CSS and NMPSO, they all obtained 5 better results compared with our method in the whole comparison. Especially, MaOEA-CSS shows the best performance on MaF1 with 4, 6, 8 and 10 objectives. This may be attributed to the fact that the coordinated selection mechanism of MaOEA-CSS is better at dealing with multimodal problems like MaF1, which conflicts with the clustering selection mechanism. Compared with IDBEA, TiGE2 and SPEA/R, the MaOEA-CCIL achieves relatively good performance results on all objectives, proving that the CCSM and RPIL of MaOEA-CCIL are more suitable than

another selection mechanism to solve the concave and convex optimization problem. The number of MaOEA-CCIL that obtained better performance results is more than half of other MaOEA. Therefore, MaOEA-CCIL can gain a superior performance on the MaF test functions.

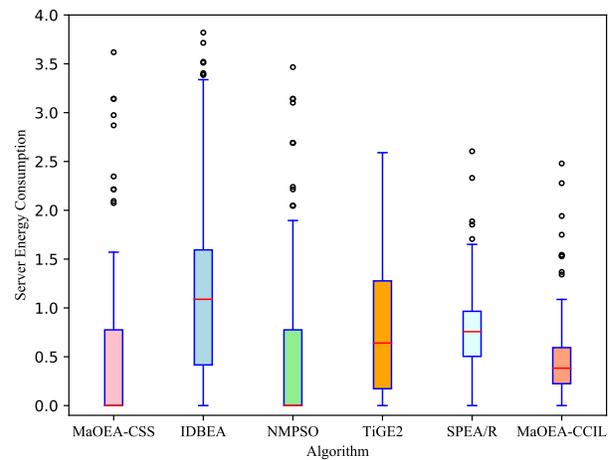
2) *Performance comparison on TORA problem:* Similar to performance comparison on benchmark function, MaOEA-CCIL is compared with MaOEA-CSS, IDBEA, NMPSO, TiGE2 and SPEA/R for handling the TORA problem in the MSMECN environment. However, since the characteristics of the true PF are not known prior, they are hard to capture. Also, the effective areas may be disconnected, irregular or more sophisticated. Therefore, fitting them into a definitive model is also challenging. To tackle this problem, the PF which results from the union of PFs of all methods is considered as the true PF [53]. To intuitively compare the performance of different algorithms for handling the TORA problem in the MSMECN environment, Fig. 2 shows the performance comparison box for different algorithms on different objectives. Some outliers in all the algorithms of Fig. 2 are caused by some solutions, not in the upper and lower quartile value range. However, we can observe that the algorithms perform differently from their overall distributions of upper and lower quartile solutions and the median values. MaOEA-CSS and NMPSO obtained similar

upper and lower quartiles values for task time delay, and their solution distribution is less concentrated than the MaOEA-CCIL algorithm. By comparing the median values of the involved algorithm, their performance can be sorted according to  $\text{MaOEA-CSS} \approx \text{NMPSO} > \text{MaOEA-CCIL} > \text{TiGE2} \approx \text{SPEA/R} > \text{IDBEA}$ . Note that the solution distribution of the MaOEA-CCIL algorithm is more concentrated than other algorithms, making it easier to obtain a solution with a smaller task time delay. Similar to task time delay, MaOEA-CSS and NMPSO obtained similar upper and lower quartiles values in the objective of server energy consumption, and their solution distribution is less concentrated than the *MaOEA-CCIL* algorithm. And the performance ranking of algorithm based median value is followed:  $\text{MaOEA-CSS} \approx \text{NMPSO} > \text{MaOEA-CCIL} > \text{TiGE2} > \text{SPEA/R} > \text{IDBEA}$ .

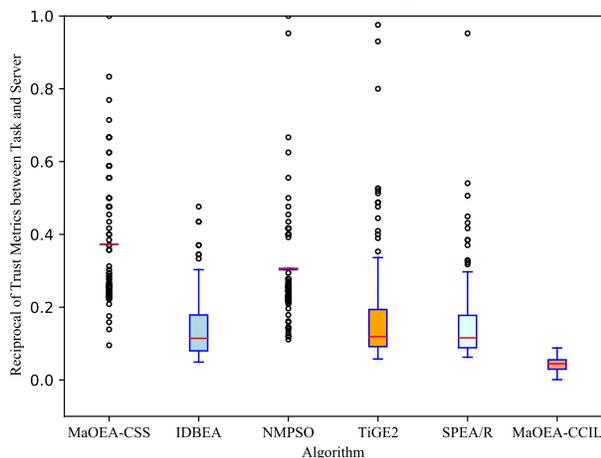
Due to the minimum value being calculated for each objective, the reciprocal of trust metrics between task and server, i.e.,  $\frac{1}{Ob_{j3}}$ , and the reciprocal of user experience utility, i.e.,  $\frac{1}{Ob_{j4}}$ , are shown in Fig. 2(c) and (d), respectively. It's observed from Figs. 2(c) that the MaOEA-CCIL algorithm has the lowest median value and smaller upper and lower quartiles than other algorithms. This result means that the MaOEA-CCIL algorithm is likelier to obtain the TO decision scheme with a higher trust value between task and server. As



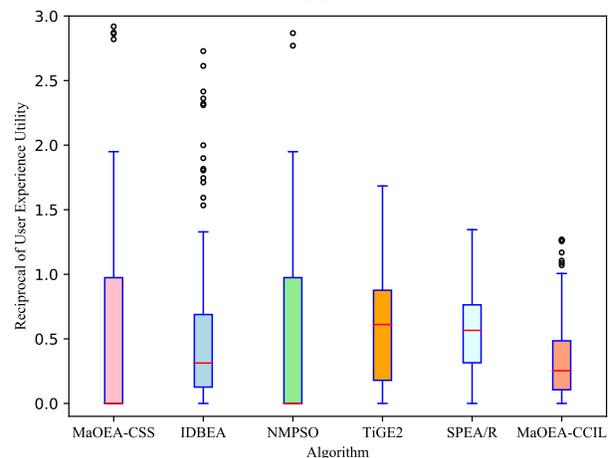
(a) Task Time Delay



(b) Server Energy Consumption



(c) Reciprocal of Trust Metrics between Task and Server



(d) Reciprocal of User Experience Utility

Fig. 2. Performance comparison box for different algorithms on different objectives

TABLE V  
DIFFERENT ALGORITHMS WITH DIFFERENT INDICATORS METRIC ON TORA PROBLEM

Metric	MaOEA-CSS	IDBEA	NMPSO	TiGE2	SPEA/R	MaOEA-CCIL
IGD	1.7663e+0 (2.01e-1) -	8.4974e-1 (1.00e-1) ≈	1.7297e+0 (2.10e-1) -	1.5765e+0 (2.62e-1) -	9.1332e-1 (2.19e-1) ≈	<b>8.2400e-1 (1.23e-1)</b>
HV	0.0000e+0 (0.00e+0) -	3.8274e-2 (4.39e-2) ≈	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	3.6490e-2 (3.79e-2) ≈	<b>4.9267e-2 (6.59e-2)</b>
CPF	4.1667e-3 (9.33e-19) -	1.0508e-1 (1.79e-2) ≈	1.7407e-2 (5.18e-3) -	1.0753e-1 (6.20e-2) ≈	8.3889e-2 (2.70e-2) -	<b>1.0923e-1 (1.93e-2)</b>

seen from Fig. 2(d), the MaOEA-CCIL algorithm has a more concentrated solution distribution than other algorithms. The MaOEA-CSS and NMPSO have a similar performance with similar median values, and upper and lower quartiles. And according to the median value, a follow performance order can be arranged as MaOEA-CSS ≈ NMPSO > MaOEA-CCIL ≈ IDBEA > SPEA/R > TiGE2. Therefore, it can be found that different algorithms show advantages for various objectives. It is not difficult to find that our algorithm can perform relatively stable on each objective.

Meanwhile, MaOEA-CCIL is compared with the other five MaOEAAs based on IGD, HV and CPF indicators for further statistical analysis. Table V shows different algorithms with different indicators metrics in the MSMECN environment. It is clear from the statistical results that the IDBEA and SPEA/R obtained similar performance on IGD and HV. The performance of other algorithms is worse than our approach on all indicators. However, these MaOEAAs involving comparison have shown substantial advantages in handling practical problems, which will be proved indirectly that the MaOEA-CCIL has superiority in handling TORA problems in the MSMECN environment.

Our many-objective TORA optimization model in the MSMECN environment is reasonable from the above analysis. Moreover, the proposed MaOEA-CCIL can achieve promising performance and outperforms other algorithms in this model. Based on the obtained non-dominated solutions, it can provide good decision-making for TORA problems in the MSMECN environment.

## VI. CONCLUSION

In this paper, we view the TORA problem in the MSMECN environment as a MaOP. A trusted many-objective TORA model is built to describe the problem. In the built trusted model, the four objectives to be optimized are considered comprehensively, including TASK time delay, server energy consumption, trust metric between task and server, and user experience utility. To obtain the trusted TORA model solution, we decompose the TORA problem into a TO problem that optimizes the offloading decision solutions and a RA problem after fixing the offloading decision solution. And the MaOEA-CCIL is designed to optimize the TO decision-making. Mainly, the CCSM strategy is employed to class the solution into elite and ordinary solutions. Meanwhile, the RPIL strategy is introduced into the reference point redistribution to improve the CaD of elite solutions, guiding the evolution of the entire TO decision solution in a better direction. Based on the optimal offloading decision solutions, the RA problem continues to be optimized by employing the low-complexity heuristic optimization method with the KKT condition. To

verify the superiority of the MaOEA-CCIL and the model's effectiveness, the MaOEA-CCIL with other advanced MaOEAAs are compared with benchmark function and TORA problem, respectively. The simulation results show that no matter which problem, MaOEA-CCIL has shown superior performance to other MaOEAAs. For the benchmark function, the designed algorithm obtains more than half the number of the superior values by comparing with each MaOEAAs. For the TORA problem in the MSMECN environment, the MaOEA-CCIL has obtained a relatively medium performance ranking position for every model objective. In addition, MaOEA-CCIL has achieved the promised results and got the best performance compared with other MaOEAAs on IGD, HV and CPF indicators metrics for handling the TORA problem.

To further improve QoS and service life, our future work will construct an enhanced TORA model by considering more influencing factors. It is also evident that the effectiveness of MaOEA-CCIL is not limited to addressing the TORA problem but can readily be applied to other fields, such as privacy protection in IoT.

## ACKNOWLEDGEMENT

This work was supported in parts by National Key Research and Development Program of China (Grant No. 2019YFB2102303), National Natural Science Foundation of China (Grant Nos. 61971014 and 11675199) and NSF CNS-2107216, CNS-2128368, CMMI-2222810, US Department of Transportation, Toyota and Amazon.

## REFERENCES

- [1] X. Jiang, N. Guan, H. Du, W. Liu, and Y. Wang, "On the Analysis of Parallel Real-Time Tasks With Spin Locks", *IEEE Transactions on Computers*, vol. 70, no. 2, pp. 199-211, 2021.
- [2] X. Q. Pham, T. D. Nguyen, V. Nguyen, and E. N. Huh, "Joint Service Caching and Task Offloading in Multi-Access Edge Computing: A QoE-Based Utility Optimization Approach," *IEEE Communications Letters*, vol. 25, no. 3, pp. 965-969, 2021.
- [3] W. Chen, Y. Yaguchi, K. Naruse, Y. Watanabe, and K. Nakamura, "QoS-Aware Robotic Streaming Workflow Allocation in Cloud Robotics Systems," *IEEE Transactions on Services Computing*, vol. 14, pp. 544-558, 2021.
- [4] J. Zou, T. Hao, C. Yu, and H. Jin, "A3C-DO: A Regional Resource Scheduling Framework Based on Deep Reinforcement Learning in Edge Scenario," *IEEE Transactions on Computers*, vol. 70, no. 2, pp. 228-239, 2021.
- [5] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-Based Computation Offloading for IoT Devices With Energy Harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930-1941, 2019.
- [6] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924 - 4938, 2017.
- [7] Z. Song, Y. Liu, and X. Sun, "Joint Task Offloading and Resource Allocation for NOMA-Enabled Multi-Access Mobile Edge Computing," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1548-1564, 2021.

- [8] Y. Xu, T. Zhang, D. Yang, Y. Liu, and M. Tao, "Joint Resource and Trajectory Optimization for Security in UAV-Assisted MEC Systems," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 573-588, 2021.
- [9] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green Resource Allocation Based on Deep Reinforcement Learning in Content-Centric IoT," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 781-796, 2020.
- [10] M. Liu, T. Song, and G. Gui, "Deep Cognitive Perspective: Resource Allocation for NOMA-Based Heterogeneous IoT With Imperfect SIC," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2885-2894, 2019.
- [11] M. Min *et al.*, "Learning-Based Privacy-Aware Offloading for Healthcare IoT With Energy Harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307-4316, 2019.
- [12] L. Li, J. Zhou, T. Wei, M. Chen, and X. Sharon Hu, "Learning-Based Modeling and Optimization for Real-Time System Availability," *IEEE Transactions on Computers*, vol. 70, no. 4, pp. 581-594, 2021.
- [13] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and Computation Co-Offloading With Reinforcement Learning in Fog Computing for Industrial Applications," in *IEEE Transactions on Industrial Informatics* vol. 15, no. 2, pp. 976-986, 2019.
- [14] L. Liu, S. Chan, G. Han, M. Guizani, and M. Bandai, "Performance Modeling of Representative Load Sharing Schemes for Clustered Servers in Multiaccess Edge Computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4880 - 4888, June 2019.
- [15] T. M. Ho and K.-K. Nguyen, "Joint Server Selection, Cooperative Offloading and Handover in Multi-Access Edge Computing Wireless Network: A Deep Reinforcement Learning Approach," *IEEE Transactions on Mobile Computing* vol. 21, no. 7, pp. 2421 - 2435, July 2022.
- [16] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-Aware Data Offloading in Multi-Server Multi-Access Edge Computing Environment," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405 - 1418, June 2020.
- [17] X. Cai, J. Zhang, Z. Ning, Z. Cui, and J. Chen, "A Many-objective Multistage Optimization-based Fuzzy Decision-making Model for Coal Production Prediction," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3665 - 3675, 2021.
- [18] S. Yu, R. Langar, X. Fu, L. Wang and Z. Han, "Computation Offloading With Data Caching Enhancement for Mobile Edge Computing," in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11098-11112, Nov. 2018.
- [19] J. Zhang *et al.*, "Privacy protection based on many-objective optimization algorithm," in *Concurrency and Computation Practice and Experience*, vol. 31, no. 20, p. e5342, 2019.
- [20] H. Ge *et al.*, "A many-objective Evolutionary Algorithm With Two Interacting Processes: Cascade Clustering and Reference Point Incremental Learning," in *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 572-586, 2019.
- [21] T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856-868, 2019.
- [22] B. Yang, X. Cao, J. Bassey, X. Li, and L. Qian, "Computation Offloading in Multi-Access Edge Computing: A Multi-Task Learning Approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2745 - 2762, Sep. 2021.
- [23] S. Wan, S. Ding, and C. Chen, "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles," *Pattern Recognition*, vol. 121, p. 108146, 2022.
- [24] K. Jiang, H. Zhou, X. Chen, and H. Zhang, "Mobile Edge Computing for Ultra-Reliable and Low-Latency Communications," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 68-75, 2021.
- [25] T. Bahreini, H. Badri, and D. Grosu, "Mechanisms for Resource Allocation and Pricing in Mobile Edge Computing Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 667-682, 2022.
- [26] M. Hu, L. Zhuang, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Learning-Driven Computation Offloading for Asymmetrically Informed Edge Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 8, pp. 1802-1815, 2019.
- [27] J. Lin, L. Huang, H. Zhang, X. Yang, and P. Zhao, "A novel Latency-Guaranteed based Resource Double Auction for market-oriented edge computing," *Computer Networks*, vol. 189, p. 107873, 2021.
- [28] I. Sorkhoh, D. Ebrahimi, R. Atallah, and C. Assi, "Workload Scheduling in Vehicular Networks With Edge Cloud Capabilities," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8472-8486, 2019.
- [29] C. C. Lin, D. J. Deng, Y. L. Chih, and H. T. Chiu, "Smart Manufacturing Scheduling with Edge Computing Using Multi-class Deep Q Network," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4276-4284, 2019.
- [30] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590-3605, 2016.
- [31] W. Chen, D. Wang, and K. Li, "Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726 - 738, 2019.
- [32] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-Delay Tradeoff for Dynamic Offloading in Mobile-Edge Computing System With Energy Harvesting Devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4642-4655, 2018.
- [33] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4548-4556, 2018.
- [34] X. Li, J. Wan, H. N. Dai, M. Imran, M. Xia, and A. Celesti, "A Hybrid Computing Solution and Resource Scheduling Strategy for Edge Computing in Smart Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4225-4234, 2019.
- [35] B. Yang, X. Cao, X. Li, Q. Zhang, and L. Qian, "Mobile-Edge-Computing-Based Hierarchical Machine Learning Tasks Distribution for IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2169 - 2180, March 2020.
- [36] T. Soyata, R. Muralidharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-Vision: Real-time Face Recognition Using a Mobile-Cloudlet-Cloud Acceleration Architecture," in *IEEE Symposium on Computers and Communications (ISCC)*, Cappadocia, Turkey, 2012.
- [37] Q. Qi *et al.*, "Knowledge-Driven Service Offloading Decision for Vehicular Edge Computing: A Deep Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4192-4203, 2019.
- [38] W. Feng, H. Liu, Y. Yao, D. Cao, and M. Zhao, "Latency-Aware Offloading for Mobile Edge Computing Networks," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2673-2677, 2021.
- [39] K. Peng *et al.*, "Joint optimization of service chain caching and task offloading in mobile edge computing," in *Applied Soft Computing*, vol. 103, p. 107142, 2021.
- [40] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, and W. Saad, "Joint Communication, Computation, Caching, and Control in Big Data Multi-Access Edge Computing," in *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359-1374, 1 June 2020.
- [41] J. Mills, J. Hu, and G. Min, "Multi-Task Federated Learning for Personalized Deep Neural Networks in Edge Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630-641, 2022.
- [42] S. Hu and G. Li, "Dynamic Request Scheduling Optimization in Mobile Edge Computing for IoT Applications," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1426 - 1437, 2020.
- [43] Q. Lin *et al.*, "Particle Swarm Optimization With a Balanceable Fitness Estimation for Many-Objective Optimization Problems," in *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32-46, 2018.
- [44] Z. Cui *et al.*, "Hybrid Many-Objective Particle Swarm Optimization Algorithm for Green Coal Production Problem," in *Information Sciences*, vol. 518, pp. 256-271, 2020.
- [45] 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," in *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, 2014.
- [46] Z. He and G. G. Yen, "Many-Objective Evolutionary Algorithms Based on Coordinated Selection Strategy," in *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 220-233, 2017.
- [47] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based Evolutionary Algorithm for Many Objective Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445-460, 2015.
- [48] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, and J. Zhang, "Tri-Goal Evolution Framework for Constrained Many-Objective Optimization," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 3086-3099, 2020.
- [49] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329-346, 2017.
- [50] R. Cheng *et al.*, "A benchmark test suite for evolutionary many-objective optimization," in *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67-81, 2017.

- [51] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," in *Evolutionary Computation*, 2002, pp. 825-830.
- [52] Y. Tian, R. Cheng, X. Zhang, M. Li, and Y. Jin, "Diversity Assessment of Multi-Objective Evolutionary Algorithms: Performance Metric and Benchmark Problems," in *IEEE Computational Intelligence Magazine*, vol. 14, no. 3, pp. 61-74, 2019.
- [53] S. L. Mohamed Abd Elaziz, "Many-objectives multilevel thresholding image segmentation using Knee Evolutionary Algorithm," in *Expert Systems with Applications*, vol. 125, pp. 305-316, 2019.



**Shanshan Tu** (Member, IEEE) received his PhD degree from Computer Science Department at Beijing University of Posts and Telecommunications in 2014. From 2013 to 2014, he visited University of Essex for National Joint Doctoral Training. He worked in the Department of Electronic Engineering at Tsinghua University as a postdoctoral researcher from 2014 to 2016. He is currently an Associate Professor and Deputy Dean at Faculty of Information Technology, Beijing University of Technology, China. His research interests are in the areas of cloud

computing, MEC and information security techniques.

**Jiangjiang Zhang** is pursuing his Doctor degree in Beijing University of Technology, China. His research interest includes data security, privacy protection, computational intelligence, combinatorial optimization and modelling.



**Bei Gong** (Member, IEEE) received the BS degree from Shandong University, Qingdao, China, in 2005, and the PhD degree from the Beijing University of Technology, Beijing, China, in 2012. He has six National invention patents and one monograph textbook. He is the principal investigator of eight national projects such as the National Natural Science Foundation grants and six provincial and ministerial projects such as the General Science and Technology Program of Beijing Municipal Education Commission. Over the past five years, he has authored

or coauthored more than 30 articles in top-tier journals and prestigious conferences in relevant research fields. His research interests include trusted computing, Internet of Things security, mobile Internet of Things, and mobile edge computing.



**MUHAMMAD WAQAS** (Senior Member, IEEE, ACM) received his PhD degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China in 2019. From Oct. 2019 to Sept. 2021, he was a Research Associate at the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Currently, he is an Assistant Professor at the Computer Engineering Department, College of Information Technology, University of Bahrain, Bahrain. He is also an Adjunct Senior Lecturer at the School of Engineering, Edith Cowan

University, Australia. He has more than 100 research publications in reputed Journals and Conferences. He is an Associate Editor of the International Journal of Computing and Digital Systems. His current research interests are in the areas of Wireless Communication, vehicular networks, Fog/Mobile Edge Computing, Internet of Things and Machine Learning. He is recognised as a Global Talent in the area of Wireless Communications by UK Research and Innovation and Professional Member of Engineer Australia. He is senior member of IEEE, Professional Member of ACM, IEEE Young Professional, Member of Pakistan Engineering Council and PhD approved supervisor by Higher Education Commission of Pakistan.



**Zhu Han** (S'01–M'04 - SM'09 - F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores

Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. Dr. Han's main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, security and privacy. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks".