J. Liu, G. Li, Q. Huang, M. Bilal, X. Xu and H. Song, "Cooperative Resource Allocation for Computation-Intensive IIoT Applications in Aerial Computing," in *IEEE Internet of Things Journal*, 2022, doi: 10.1109/JIOT.2022.3222340.

https://doi.org/10.1109/JIOT.2022.3222340

Cooperative Resource Allocation for Computation-Intensive IIoT Applications in Aerial Computing

Jialei Liu, *G*uosheng Li, *Q*uanzhen Huang, *M*uhammad Bilal, *Senior Member, IEEE,* Xiaolong Xu, *Senior Member, IEEE,* and Houbing Song, *Senior Member, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) will be a vital part of the massive Industrial Internet of Things (IIoT) in the 5G and 6G paradigms. The UAVs are required to collaborate with each other to deal with some computation-intensive IIoT applications in an autonomous UAV system. However, due to the limited processing capacity of UAVs, they are occasionally unable to handle certain tasks adequately (e.g., crowd sensing). Therefore, it is an important issue to realize efficient offloading of these computation-intensive IIoT applications. In this paper, we first partition the computation-intensive IIoT application into a directed acyclic graph with multiple collaborative tasks. Then, we establish a joint optimization problem based on the models of the processor resources and energy consumption for the task offloading scheme. Thirdly, we propose a cooperative resource allocation approach to optimize the joint optimization problem under the constraints of resource and communication latency, and then can migrate more computation-intensive tasks to the edge clouds. Finally, we build an aerial computing simulation system, and make a comparative evaluation and analysis of our proposed cooperative resource allocation approach in terms of effectiveness and performance. The experimental results show that our proposed approach performs better than other related approaches.**

***Index Terms*—Aerial computing, unmanned aerial vehicle, IIoT application, resource efficiency, communication latency**

I. Introduction

WITH the rapid advancements in Internet of Things (IoT) and wireless communication technologies, the Unmanned Aerial Vehicles (UAVs) are widely exploited for some emerging Industrial Internet of Things (IIoT) applications in several critical scenarios, e.g., parcel delivery, weather monitoring, aerial access networks, and agriculture [1], [2]. Based on realistic projections, the scale of the UAV industry is huge. For example, the size of the U.S. economy alone is $80 billion, and tens of thousands of new jobs are expected to be created over the next decade [3]. Furthermore, a new paradigm of UAV-enabled aerial computing has drawn extensive attention due to its mobility, flexibility, and maneuverability. However, since all operations of storage and computation of the UAVs are carried out onboard, the extensive application of these UAVs faces inherent resource (e.g., storage, computation, and communication) constraints. Meanwhile, there are many computation-intensive IIoT applications in UAV-enabled aerial computing environments, these applications require large amounts of communication resources to transmit the stream of data traffic generated by cameras or sensors on the UAVs [4], [5], [6], or powerful processing power to accomplish artificial intelligence processing (e.g., crowd sensing in smart cities), navigation, and object recognition [7], [8]. Therefore, it is challenging for these UAVs to effectively perform these computation-intensive IIoT applications.

In order to alleviate the limitation of UAVs, mobile edge computing has appeared as a new computing paradigm that utilizes resources near UAVs to provide timely services in conjunction with cloud servers. In mobile edge computing, the computation-intensive IIoT application from the edge of the mobile network will usually be split into a series of tasks, which can be independent design, development, deployment and operational [9]. These tasks coordinating with each other are allocated onto the UAVs, the edge clouds, or the remote cloud [10],[11], and turn over a cluster consisting of multiple virtual machines or containers for collaborative processing [12]. Furthermore, the UAVs reduce the energy consumption and accelerate the calculation process speed, and also make it possible to run emerging IIoT applications on the UAVs. In the process of computing offloading, multiple tasks split by these computation-intensive IIoT applications need to select the best computing node for collaboratively processing by multiple virtual machines or containers [13]. Meanwhile, given the consumption of energy, processor resources, and bandwidth resources by these tasks with the required communication latency, it has become an urgent problem how to optimally allocate tasks of these computation-intensive IIoT applications to edge clouds.

Jialei Liu is with the Computer School, Hubei University of Arts and Science, Xiangyang, 441053, China; and also with the School of Software Engineering, Anyang Normal University, Anyang, 454001, China (e-mail: jialeiliu@hbuas.edu.cn).

Guosheng Li is the with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, Weifang, 261000, China (e-mail:sdsgliguosheng@wfust.edu.cn).

Quanzhen Huang is with the School of Electrical Information Engineering, Henan University of Engineering, Zhengzhou, 450000, China (e-mail: huang2004_susu@haue.edu.cn).

Muhammad Bilal is with the Department of Computer Engineering, Hankuk University of Foreign Studies, Yongin-si, 17035, South Korea (e-mail: m.bilal@ieee.org).

Xiaolong Xu is with the School of Computer and Software, Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing, 210000, China (e-mail:xlxu@nuist.edu.cn).

Houbing Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA (email: h.song@ieee.org)

Compared with the resources in the remote cloud, the resources in the edge cloud are usually: 1) resource constrained–limited computing resources, because the UAVs have small processors and limited power budget; 2) Heterogeneity–processors with different architectures; 3) Dynamics–their computation-intensive IIoT applications dynamically change and compete for limited resources. These factors cause various resources such as CPU, bandwidth, and storage in the edge cloud to be scarcer and more expensive than the remote cloud [14], [15]. Therefore, the cost of leasing edge servers becomes a big challenge for application operators as they rent large amount of edge servers and allocate massive tasks to provide a better user experience. According to RightScale's report [16], 26% of enterprises with more than 1,000 employees spend more than $6 million on cloud every year, but 35% of which is wasted, indicating that users may always overestimate resource utilization. In addition, application operators always have specific requirements for their applications, which are to maintain some key performance indicators, such as average response time [17]. Thus, it is urgent to investigate how to find the optimal resource allocation scheme for a computation-intensive IIoT application, while ensuring the actual performance indicators such as communication latency, energy consumption, processor resource consumption and bandwidth resource consumption of the computation-intensive IIoT application.

To solve the above issues, this paper studies the joint optimization problem based on the models of the processor resources and energy consumption to obtain a near-optimal task offloading scheme.Therefore, a cooperative resource allocation approach is introduced to optimize the joint optimization problem under the constraints of resource and communication latency, and then can migrate more computation-intensive tasks to the edge clouds.

The **main contributions** are as follows:

- We formulate two sub-problems of energy consumption and processor resource consumption during the offloading of computation-intensive IIoT applications, and introduce energy consumption model, resource wastage model, resource load imbalance level model, and bandwidth resource consumption model.
- Based on these models, we first construct a joint optimization model to measure the consumption of processor resources, energy, and bandwidth resources of computation-intensive IIoT application offloading scheme. Then, we propose a cooperative resource allocation approach to simultaneously minimize the consumption of energy and processor resources in the edge clouds under the constraints of resource and communication latency.
- We build an aerial computing simulation system, and make a comparative evaluation and analysis of our proposed cooperative resource allocation approach in terms of effectiveness and performance.

The rest of this paper is described as follows. Section 2 presents the related work. Section 3 presents the system model. Section 4 proposes the allocation model of computation-intensive IIoT application. Section 5 proposes the design and implementation of cooperative resource allocation approach. Section 6 provides the performance evaluation including experimental parameter settings, comparison results, and parameter study. Finally, we conclude this paper with the future work and recommendations in Section 7.

II. Related Work

In recent years, many scholars have carried out researches on the IIoT application allocation, and made many research achievements. For example, Tong et al. [18] designed edge clouds as a tree hierarchy of regionally distributed edge servers to improve cloud resource utilization efficiency. Meng et al.[19] proposed an online algorithm to greedy scheduling of newly arrived tasks to satisfy new deadlines. Chen et al. [20] proposed a data-intensive service edge allocation scheme based on genetic algorithm to minimize the response time of data-intensive service allocation. Dai et al. [21] jointly optimized user association and computation offload to minimize the overall energy consumption. Chen et al. [22] proposed an energy efficient dynamic offloading algorithm to minimize the energy consumption of task offloading scheme. Liu et al. [23] presented a collaborative computation offloading scheme while considering the limited computation capabilities of UAVs to maximize the total utility of the UAVs. The above researches [18-23] are basically carried out by taking a single indicator as the optimization objective. It is difficult to achieve the purpose of reducing the system utility and network resources of the aerial computing environment with ensuring the communication latency.

At present, there are also massive researches on multi-objective optimization deployment of IIoT applications. For example, Chen et al. [24] formulated a data-intensive applications deployment strategies to minimize the delay of the mobile devices and minimize the monetary cost of application service providers. Deng et al. [16] proposed an approach to generate an appropriate deployment scheme at the minimum cost under the on-demand billing model. Wu et al. [25] proposed an efficient offloading framework with intelligent decision-making ability to jointly minimize system utility and bandwidth allocation of each mobile device. Pallewatta et al. [26] proposed an IIoT application layout strategy to minimize latency and network utilization. Goudarzi et al. [27] proposed a parallel IIoT batch application layout decision approach based on memetic algorithm to minimize the energy consumption and execution time of IIoT applications. Yang et al. [28] proposed an asynchronous advantage actor-critic based resource allocation and UAVs placement approach to maximize the network capacity. Adhikari et al. [29] proposed a new Cybertwin-driven resource supply strategy to analyze the Internet of everything applications on 6G-enabled edge networks, thus reducing the delay and energy consumption of these applications. The above studies [16, 24-29] all solve the corresponding joint optimization problem through the corresponding calculation offloading algorithm. However, their disadvantage lies in that the designed cost does not take into account the bandwidth resource consumption between tasks

TABLE I
KEY NOTATIONS

Notation	Description
N	the number of virtual machines or containers.
M	the total number of edge servers.
E	the total number of started edge servers in *e*-th EC.
Q	the number of IIoT applications
$M^{'}$	the startup edge servers
U_j^*	the utilization of one resource in the j-th edge server.
R_i^*	the resource demand of the i-th virtual machine or container.
C_j^*	the total number of one resource type in the *j*-th edge server
Ω	the set of various resources of edge servers.
P_j^{idle}	the power consumption of the *j*-th edge server without load.
M_e	the total number of edge clouds of the aerial computing.
CR	the crossover probability.
F	the scaling factor.

allocated on different edge servers and the energy consumption of edge clouds.

According to the analysis above, we first establish the energy consumption model, resource wastage model, resource load imbalance level model, and bandwidth resource consumption model. Then, these models are integrated to construct a joint optimization model. Finally, this model is exploited to obtain the near-optimal resource allocation scheme of computation-intensive IIoT applications by our proposed approach. That is, when the multiple collaborative computation-intensive applications are offloaded to the resource-constrained edge clouds, how to approximately minimize consumption of energy and processor resources in the edge clouds under the constraints of resource and communication latency.

III. SYSTEM MODEL

The key enabler of various IIoT applications and services is known as aerial access networks of 5G and 6G wireless systems (as shown in Fig.1). The global coverage of comprehensive access network and diverse QoS provisioning can be realized by the aerial access network connecting with satellite and terrestrial infrastructures. Moreover, various computation-intensive and low-latency IIoT applications can be better supported by pushing the cloud resources such as computing and storage to the edge of the mobile network. Therefore, the convergence of aerial access networks and mobile edge computing is expected to provide the services of computing, caching, sensing, and control as well as traditional communication services for a huge number of IIoT devices worldwide.

As shown in Fig.1, the system model is mainly composed of five parts: 1) UAVs that work with the edge clouds or the remote cloud to collaborate on the computation-intensive tasks; 2) Edge Clouds (ECs) including a wireless cellular base station and multiple edge servers; 3) Remote Cloud (RC) providing cloud services; 4) Edge server and Cloud server accommodating the virtual machines (VMs) or containers (As); 5) Tasks (Ts) assigned to the virtual machines or containers. The total number of edge servers is M in the aerial computing system. Please note that each edge server or cloud server is turned off or startup at some point. The startup edge server handles some collaborative computation-intensive tasks through the virtual machines or containers on it, and these collaborative tasks communicate with each other and constitute a directed acyclic graph, i.e., a computation-intensive IIoT application. For instance, a computation-intensive IIoT application for crowd sensing in smart cities is divided into multiple artificial intelligence-related tasks, which are offloaded onto the UAVs, the edge server or cloud server and handed over to these virtual machines or containers for processing. Please note that we treat the IIoT application as the computation-intensive IIoT application in subsequent sections to focus on the main point.

When an UAV makes an application offloading request, a virtual machine or container on the edge server or cloud server assists the device in handling the tasks offloaded onto it, and feeds the result back to the UAV. Considering that high-rise buildings and other factors in the city have great interference to wireless signals, all edge clouds are connected by a fiber optic backhaul network based on a full network topology (as shown in Fig.2), and the propagation delay between edge clouds is load-independent [30]. Each edge cloud can gain some computing and storage capacity by deploying a number of heterogeneous edge servers interconnected by switches. In addition, it can receive, process, and forward the offloading requests from the UAVs via a wireless cellular base station. Since the UAVs appear randomly, the number of offloading requests for IIoT applications will vary from time to time [31]. Meanwhile, each IIoT application usually has a certain deadline and is modeled as a directed acyclic graph reflecting the task dependence. Therefore, the cellular base stations of communication facility providers are rented by application service providers to allocate and handle these collaborative tasks. To better understand this context, some key notations are listed in Table I.

IV. ALLOCATION MODEL OF IIOT APPLICATION

This section first introduces the resource wastage model, resource load imbalance model, energy consumption model and bandwidth resource consumption model of IIoT application allocation scheme; and then proposes a joint optimization objective function for the IIoT application allocation.

A. Resource wastage model

Since edge cloud has scarce and expensive processing resources compared with remote cloud, how to maximize the resource utilization ratio of edge server is the focus of current attention while deploying IIoT applications to edge clouds. For each edge server, the utilization of certain resources (e.g., CPU, memory, disk, and bandwidth) refers to the ratio of the resources used to the total resources [32], as shown in Eq.(1). The total resource utilization of the edge cloud can be defined as the average utilization of each resource type of all startup edge servers. The mean value further reflects the idle resources in the edge clouds, that is, the larger the mean value is, the less the idle resources in the edge clouds will be, and conversely, the more. Therefore, the idle resources (i.e., resource wastage) W in the edge clouds can be calculated by Eq.(2).

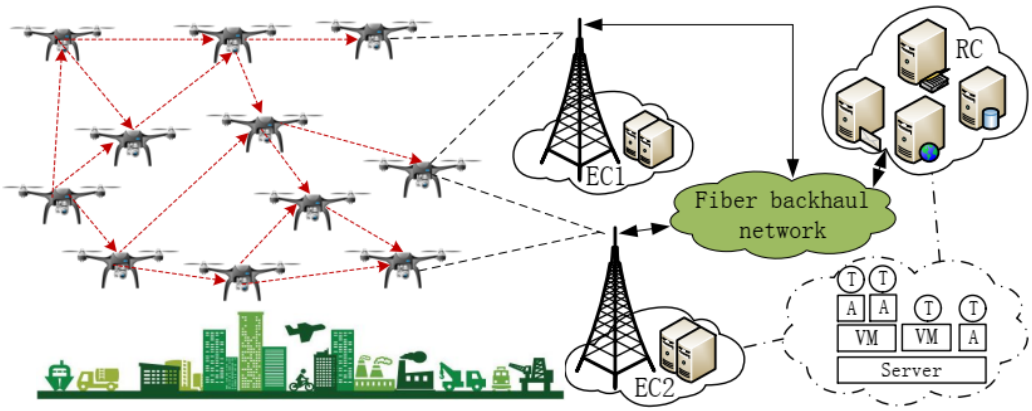


Fig. 1. Aerial computing system model

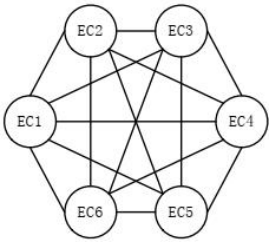


Fig. 2. A full network topology

$$\begin{cases} U_j^{CPU} = \sum_{i=1}^{N} z_{ij} \cdot R_i^{CPU} / C_j^{CPU} \\ U_j^{RAM} = \sum_{i=1}^{N} z_{ij} \cdot R_i^{RAM} / C_j^{RAM} \\ U_j^{disk} = \sum_{i=1}^{N} z_{ij} \cdot R_i^{disk} / C_j^{disk} \\ U_j^{BW} = \sum_{i=1}^{N} z_{ij} \cdot R_i^{BW} / C_j^{BW} \end{cases} \tag{1}$$

$$W = \frac{4M' - \sum_{j=1}^{M'} \left(U_{j=1}^{CPU} + U_{j=1}^{RAM} + U_{j=1}^{disk} + U_{j=1}^{Bw} \right)}{4M'} \tag{2}$$

where $M'(M' \leq M)$ and N denote the number of startup edge servers and virtual machines or containers, respectively; U_j^* represents the utilization of one resource in the *j*-th edge server; R_i^* represents the resource demand of the *i*-th virtual machine or container; C_j^* represents the total number of one type of resource in the *j*-th edge server; the Boolean variable z_{ij} denotes whether the *i*-th virtual machine or container is assigned to the *j*-th edge server, in other words, if the *i*-th virtual machine or container is assigned to the *j*-th edge server, then $z_{ij} = 1$, or else, $z_{ij} = 0$.

B. Resource load imbalance model

While improving the utilization rate of various resources of edge servers, the load balancing degree of these edge servers should also be taken into account. The resource load imbalance level *IB* for all edge clouds can be obtained by averaging the various resource load imbalance levels [32], as shown in Eq.(3).

$$IB = \frac{1}{|\Omega|} \sum_{q \in \Omega} \sqrt{\sum_{j=1}^{M'} \left(U_j^q - \sum_{j=1}^{M'} U_j^q / M' \right)^2} \tag{3}$$

where Ω represents the set of various resources such as CPU, bandwidth, memory, and disk of edge servers.

C. Energy consumption model

Some studies show that CPU is the most important component of energy consumption, and the energy consumption of edge server is linearly related to its CPU utilization, or piecewise linear function [33], [34]. However, there are also some literatures that the comprehensive utilization of CPU, memory, disk and network interface directly determines the energy consumption of edge servers [35], [36]. The energy consumption models of different edge servers are different and cannot be represented by the simple linearity [35]. On the basis of literature [36], the fitting error of linear model and piecewise linear model is larger than that of polynomial model. The energy consumption model of the edge server established by the quadratic polynomial model is better suited for the actual edge server, as shown in Eq.(4).

$$P = \sum_{j=1}^{M} \left(P_j^{idle} + \omega_1 \cdot U_j^{CPU} + \omega_2 \cdot (U_j^{CPU})^2 \right) \tag{4}$$

where ω_1 and ω_2 represent the positive fixed polynomial coefficients; P_j^{idle} represents the power consumption of the *j*-th edge server without any load.

D. Bandwidth resource consumption model

Considering that multiple tasks of the IIoT applications are often offloaded onto the edge servers and handed over to a virtual machine or container cluster for collaboratively processing. This means that there will be communication between virtual machines or containers in the same cluster. Therefore, the location of these virtual machines or containers of the same cluster decides directly the bandwidth and communication latency consumed by them. When some virtual machines or containers are placed on the same edge server, there are not the bandwidth and communication latency consumed by the

communication between virtual machines or containers. On the contrary, there are certain amount of bandwidth consumption and communication latency. Thus, the bandwidth consumption *BW* after processing a group of IIoT applications can be calculated by Eq.(5).

$$BW = \sum_{e=1}^{E}\sum_{k=1}^{M_e}\sum_{l=1}^{m_k}\sum_{t=1}^{V_l} y_{lt} \cdot bw_{lt} \tag{5}$$

where E and M_e represent the total number of edge clouds and started edge servers in the *e*-th edge cloud of the aerial computing environment, respectively; m_k represents the number of virtual machines or containers on the *k*-th edge server; V_l represents the number of virtual machines or containers working with the *l*-th virtual machine or container on the *k*-th edge server for an IIoT application; bw_{lt} represents the bandwidth value from the *l*-th virtual machine or container to the *t*-th virtual machine or container, which is specified as a range of random values; the binary variable y_{lt} indicates whether the *l*-th virtual machine or container is a data sender, if so, then $y_{lt} = 1$ otherwise $y_{lt} = 0$.

E. Joint optimization formulation

When multiple IIoT applications are offloaded onto the edge cloud, it is the key issue how to adopt a cooperative resource allocation approach to improve the energy efficiency and system utility of these edge clouds under the constraints of resource and communication latency. Since the heterogeneous virtual machines or containers that handle these IIoT applications are assigned to different heterogeneous edge servers [37], different allocation schemes have different effects on the system utility and bandwidth resource consumption of the aerial computing system. To solve the problem of inconsistent and approximate optimization objectives, it is necessary to obtain a near-optimal resource allocation scheme by a cooperative resource allocation approach of IIoT application. Furthermore, the resource load imbalance level, resource wastage, energy consumption and bandwidth consumption of edge clouds are minimized as much as possible under the constraint of resource regulation and communication latency. The joint optimization objective function of the optimization phase can be expressed by Eq.(6).

$$\min : \theta_1 * W + \theta_2 * IB + \theta_3 * P + \theta_4 * BW \tag{6}$$

s.t.

$$\sum\nolimits_{i=1}^{N} z_{ij} \cdot R_i^{CPU} < C_j^{CPU} \tag{7}$$

$$\sum\nolimits_{i=1}^{N} z_{ij} \cdot R_i^{RAM} < C_j^{RAM} \tag{8}$$

$$\sum\nolimits_{i=1}^{N} z_{ij} \cdot R_i^{disk} < C_j^{disk} \tag{9}$$

$$\sum\nolimits_{i=1}^{N} z_{ij} \cdot R_i^{BW} < C_j^{BW} \tag{10}$$

$$\sum\nolimits_{j=1}^{M} z_{ij} = 1, z_{ij} = 0 or 1 \tag{11}$$

$$\sum_{e=1}^{E}\sum_{k=1}^{M_e}\sum_{l=1}^{m_k}\sum_{t=1}^{V_l} b_{lt} \cdot \frac{data_l}{bw_{lt}} < L \tag{12}$$

where θ_1, θ_2, θ_3, and θ_4 are the tunable positive weights ($0 < \theta_1, \theta_2, \theta_3, \theta_4 < 1$, and $\theta_1 + \theta_2 + \theta_3 + \theta_4 = 1$), and their value will be assigned in Section VI-A; the Eqs.(7) to (11) represent the resource constraint conditions. That is, the resource capacity of the edge server must be more than the total resources required by the virtual machines or containers, and one virtual machine or container can be deployed onto only one edge server. The Eq.(12) represents the constraint condition of communication latency, that is, the total communication latency required by the edge servers to process a batch of IIoT applications should be less than the preset threshold value *L*; $data_l$ represents the amount of data that the *l*-th virtual machine or container sends to the *t*-th virtual machine or container. The binary variable b_{lt} indicates whether the *l*-th virtual machine or container is a data sender and has the maximum transfer time in the same layer of the directed acyclic graph, if so, then $b_{lt} = 1$ otherwise $b_{lt} = 0$.

V. Design of Cooperative Resource Allocation Approach

To solve the above joint optimization problem, this section will introduce the implementation details of Differential Evolution [38] based cooperative resource allocation approach of IIoT application (DEOA). Firstly, it briefly introduces the DE algorithm; then proposes the improvement measures of DE algorithm; and finally introduces the detailed implementation scheme of DEOA approach.

A. DE algorithm

DE algorithm designed by Storn and Price in 1997 is a heuristic random search algorithm that exploits population difference to solve Chebyshev polynomials [38]. Different from other algorithms, it can obtain the individual differences by exploiting arithmetic operators to revise the internal expression of them. Furthermore, the current individual will be substituted by a new vector while the fitness value of the new vector is greater than the fitness value of the current individual. Currently, there are many different optimization strategies of DE algorithm via the number of the weighted difference vectors and disturbed individuals [39]. The DE/rand/1/bin strategy is applied to determine the disturbed vectors randomly, and then improve the population diversity.

B. Improvement of DE algorithm

The essence of IIoT application allocation is to establish a reasonable mapping relationship between the task set of application and processing resources of edge server in edge clouds based on the constraints of resources and communication latency, so as to obtain the smallest value of joint optimization objective function. To solve the above issue, it is very necessary to improve the variation operator, crossover operator, selection operator, and relevant control parameters of DE algorithm on the basis of formulating a chromosome coding scheme, and then introduce an improved DE algorithm.

1) Chromosome coding scheme: According to the characteristics of the IIoT application allocation, a more concise and understandable real coding approach is exploited to encode chromosomes. In the encoding scheme, chromosome length is expressed as the total amount of tasks for a batch of IIoT applications. Each gene fragment represents a task number, its bit value represents the edge server number assigned to an edge cloud, and the final chromosome encoding scheme can be shown in Fig.3.

As shown in Fig.3, each edge cloud has M_e heterogeneous edge servers; each IIoT application (APP) can consist of multiple tasks with different quantity and different computing resource requirements, which are represented by a directed acyclic graph; when *Q* IIoT applications are allocated to an aerial computing environment with *E* heterogeneous edge clouds, the edge server in each edge cloud and the task in each IIoT application are both numbered from one. Please note that the total amount of tasks is much larger than the number of resources in the edge clouds and is adopted to represent the length of chromosomes. Then the corresponding decoding can be obtained according to the encoding scheme in Fig.3, as shown in Eq.(13).

$$EC_1(1):\{APP_1:T_1, APP_2:T_1\}, EC_1(2):\{APP_Q:T_1\} \cdots EC_E(M_e):\{APP_Q:T_Q\} \tag{13}$$

2) Improved algorithm: To solve the cooperative resource allocation of IIoT application, the DE/rand/1/bin strategy needs to be redefined by the above coding scheme. Since the adjustment of the scaling factor *F* and crossover probability *CR* can exploit random, constant, and adaptive strategies, these strategies have different influence on the convergence speed, diversity, and search space of DE [40]. Thus, in order to avoid falling into local optimality and low convergence rate, multiple optimization strategies are exploited to improve the diversity and convergence rate of the DE algorithm, and then better solve the cooperative resource allocation of IIoT application. Assumed that the crossover probability *CR* and scaling factor *F* both exploit the adaptive strategies, *F* is positively related to variation search space. That is, the value of *F* will gradually decrease with the operation of the algorithm, thus ensuring the diversity of the population in the initial stage of the algorithm and protecting the optimum solution in the later stage of the algorithm. The process in which the scaling factor F_n adaptively changes over the iteration is shown in Eq.(14).

$$F_n = F_0 \cdot 5^{\lambda},\ \lambda = \mathrm{e}^{\frac{f_n - f_{\max}}{f_n - f_{\min}}} \tag{14}$$

where f_n, $f_{\max}$, and $f_{\min}$ respectively denote the fitness of the individual x_n, the worst individuals and the optimal individuals in the current iteration; F_0 denotes the scaling factor of the initial state.

Since the individuals with low fitness are more likely to be reserved for the next generation by the larger *CR*, and the global searching capacity and diversity of the algorithm can be improved by the lower *CR*, the CR_n of the individual x_n adaptively changes over the iteration is shown in Eq.(15).

$$CR_n = \begin{cases} CR_{\min}\ , \text{ if } \overline{f} \le f_n \\ CR_{\min} + (CR_{\max} - CR_{\min}) \cdot \dfrac{f_n - f_{\min}}{f_{\max} - f_{\min}}, \text{ if } \overline{f} > f_n \end{cases} \tag{15}$$

where $\overline{f}$ presents the average fitness of the current all individuals; $CR_{\max}$ and $CR_{\min}$ respectively present the maximal crossover probability and the minimum crossover probability.

Next, to redefine the variation operator, crossover operator, and selection operator, the population size equals *NP*; the individual with *D* optimization parameters is denoted as a *D*-dimension parameter vector; and there is a population with NP *D*-dimensional parameter vectors $x_{n,G} = (v_{1n,G}, v_{2n,G}, ..., v_{Dn,G})\ \ , n = 1, 2, 3, ..., NP$ in the *G*-th iteration.

3) Variation operator: A variation vector $v_{n,G+1} = (v_{1n,G+1}, v_{2n,G+1}, ..., v_{Dn,G+1})$ for each vector can be obtained by the Eq.(16).

$$v_{n,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \tag{16}$$

where $F \in [0, 1)$ denotes a preset scaling factor controlling the difference vector; $r1 \ne r2 \ne r3 \ne n$ denotes four randomly determined and distinct individuals, and each parameter vector has at least four elements.

4) Crossover operator: A crossover vector $u_{n,G+1} = (u_{1n,G+1}, u_{2n,G+1}, ..., u_{Dn,G+1})$ can be produced by the discrete crossover of the variation vector and the target vector via Eq.(17). To preserve genetic information, another crossover vector $w_{n,G+1} = (w_{1n,G+1}, w_{2n,G+1}, ..., w_{Dn,G+1})$ can be produced by Eq.(18).

$$u_{gn,G+1} = \begin{cases} x_{dn,G}\ if\ \mathrm{d} \ne d_{rand}\ or\ \ (r > CR) \\ v_{dn,G+1}\ if\ \mathrm{d} = d_{rand}\ or\ \ (r \le CR) \end{cases} \tag{17}$$

$$w_{gn,G+1} = \begin{cases} v_{dn,G}\ if\ \mathrm{d} \ne d_{rand}\ or\ (r > CR) \\ x_{dn,G+1}\ if\ \mathrm{d} = d_{rand}\ or\ (r \le CR) \end{cases} \tag{18}$$

where d_{rand} denotes a randomly determined value in the range $[1, D]$; *d* denotes an integer in range $[1, D]$; *d* denotes a randomly selected value from the range $[0, 1]$.

5) Selection operator: In order to further preserve genetic information, the fitness values $f(*)$ of the crossover vector $w_{n,G+1}$, the crossover vector $u_{n,G+1}$, and the variation vector $v_{n,G+1}$ are compared to determine the individual with the minimum fitness by the Eq.(19), which be retained to the *G*+1 generation.

$$x_{n,G+1} = \begin{cases} u_{n,G+1} \text{ if } (\mathrm{f}(x_{n,G}) \ge \mathrm{f}(u_{n,G+1})) \\ v_{n,G+1} \text{ else if } (\mathrm{f}(x_{n,G}) \ge \mathrm{f}(v_{n,G+1})) \\ w_{n,G+1} \text{ else if } (\mathrm{f}(x_{n,G}) \ge \mathrm{f}(w_{n,G+1})) \\ x_{n,G}\ otherwise \end{cases} \tag{19}$$

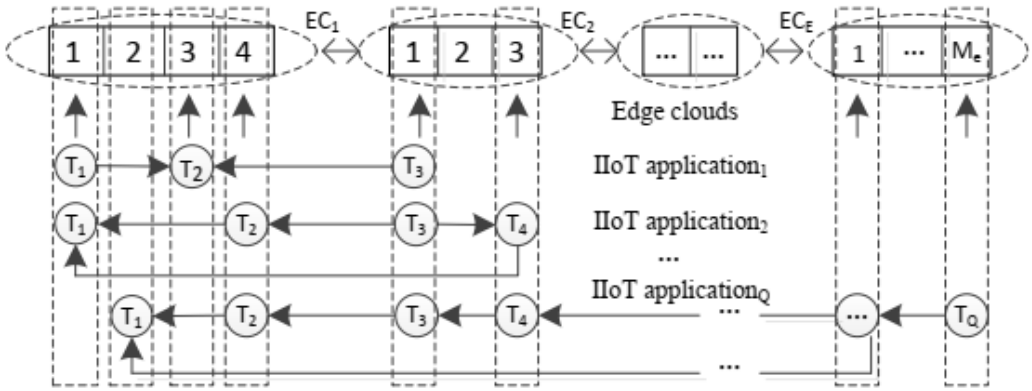


Fig. 3. Chromosome encoding scheme

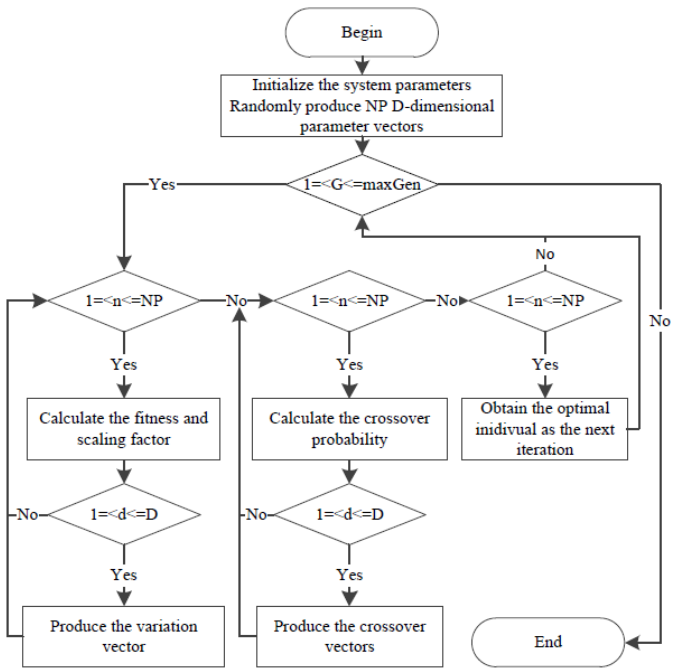


Fig. 4. Diagram flow chart of Algorithm 1

Algorithm 1 Differential Evolution based cooperative resource allocation approach of IIoT application (DEOA)

Input: The system parameters.
Output: The near-optimal allocation scheme.

1: Initialize population size *NP*, chromosome *D*, maximum number of iterations $\max Gen$, maximum and minimum crossover probability $CR_{\max}$ and $CR_{\min}$,initial scaling factor F_0, number of edge clouds *E*
2: Produce *NP* *D*-dimensional vectors Randomly $x_{n,G} = (v_{1n,G}, v_{2n,G}, ..., v_{Dn,G})\,, n = 1, 2, ..., NP$
3: **for** *G*=1 to $\max Gen$ **do**
4: **for** *n*=1 to *NP* **do**
5: Calculate the fitness and the average fitness of related individual and current all individuals
6: Calculate the scaling factor F_n of individual x_n
7: **for** *d*=1 to *D* **do**
8: Produce $v_{n,G+1}$ via Eq.(16)
9: **end for**
10: **end for**
11: **for** *n*=1 to *NP* **do**

Based on the above encoding scheme, we can exploit these redefined operators to improve the DE algorithm, and propose the DEOA approach to better solve the cooperative resource allocation problem of IIoT applications. Algorithm 1 describes the implementation scheme of the DEOA approach. Fig.4 shows the diagram flow chart of Algorithm 1.

As showed in Algorithm 1 and Fig.4, Lines 1-2 execute the initialization of the system parameters. Lines 4-10 first randomly select three different individuals (i.e.,$x_{r1,G}$, $x_{r2,G}$, and $x_{r3,G}$) different from $x_{n,G}$ in the current population and rank these individuals from smallest to largest in terms of fitness, compute F_n of individual x_n, and finally variate the D gene fragments in the current individual by the Eq.(16) to acquire the variation vector $v_{n,G+1}$. Please note that if the assignment Ri of one variated gene fragment exceeds the specified assignment range, the formulation $\lfloor |Ri| \rfloor / \sum_{e=1}^{E} M_e$ is required to map the variation value to the specified range of assignment. Lines 11-16 calculate the crossover probability CR_n of the individual x_n via the Eq.(15), and then obtain the crossover vectors $u_{n,G+1}$ and $w_{n,G+1}$ by the Eqs.(17) and (18). The loop in Lines 17-19 acquires the best individual as the next iteration from the current individual vector $x_{n,G}$, the vector $v_{n,G+1}$, the vectors $u_{n,G+1}$ and $w_{n,G+1}$.

6) Complexity analysis: The time complexity of the Algorithm 1 mainly includes three operations of mutation, crossover, and selection. Based on the previous condition, the variation operator consumes the time complexity $O(NP * D)$ to calculate the fitness and scaling factor, and produce the variation vector. The crossover operator consumes the time complexity $O(D * NP)$ to produce the vectors $u_{n,G+1}$ and $w_{n,G+1}$. The selection operator consumes the complexity $O(NP)$ to obtain the superior individual. Therefore, the maximum time complexity required for the algorithm to iteration once is $O(NP * D)$. The time complexity of the algorithm iteration $\max Gen$ can be expressed as $O(\max Gen * NP * D)$.

VI. Performance Evaluation

This section first introduces setting of simulation experiment environment, then compares the DEOA approach with other related approaches, and finally evaluates and analyzes these approaches from two aspects of performance and effectiveness.

A. Experiment setup

We can set up an aerial computing experiment environment with 25 edge clouds and 20 UAVs by extending the CloudSim simulator in a server with 8GB of memory and Intel i7-7500U [41], [42]. In this experimental environment, a base station is uniquely deployed into an edge cloud and interconnects other base stations by a fiber optic backhaul network based on a full network topology; and each edge cloud also includes a amount of heterogeneous edge servers and multiple UAVs interconnecting via wireless access. The configuration information (i.e., CPU, memory, bandwidth, and hard disk) of each edge server can be a randomly selected value set from the set 3720 MIPS, 10GB, 10GB/s, and 1TB, 5320 MIPS, 10GB, 10GB/s, and 1TB [43], and deployed to multiple edge clouds. When 20 UAVs appear at some point and generate a group of IIoT applications, there are three tasks in each IIoT application, which are allocated to 60 heterogeneous virtual machines based on the task size.

The bandwidth resource requirement of each edge server can randomly get specified value from the collection [10, 50] Mbps; the amount of sending data of each sending task can randomly get specified value from the collection [1, 2] Mb; the CPU and memory demand of each virtual machine can randomly get specified value from the collection 2000MIPS and 3.75GB, 500MIPS and 0.6GB, 1000MIPS and 1.7GB, 2500MIPS and 0.85GB [43]; the disk requirement of each virtual machine is set at 1GB. When these edge servers are started without any load, the power of edge servers is respectively set to 86w and 93.7w [43]. The values of ω_1 and ω_2 are set to 1.30447 and 0.02867, respectively [34]. The initial population size *NP* is set to 100; the chromosome length *D* is set to 60; and the maximum iteration number is set to 300. $CR_{\max}$, $CR_{\min}$, and F_0 are respectively set to 0.90, 0.1 and 0.2, respectively. The communication latency threshold *L* is set to 2.35s. To treat each optimization objective equally, the values of the tunable positive weights θ_1,θ_2,θ_3, and θ_4 are all set to 0.25, Other values of these weights are not the focus of this paper and will be further elaborated in future work. Finally, each optimization objective value close to the true value is obtained through a large number of repeated experiments.

To evaluate and analyze the DEOA approach, we exploit the following benchmark approaches to compare with it.

- Random Deployment (RD): When multiple candidate edge servers meet resource constraints, each virtual machine is ran on a randomly assigned edge server.
- First Fit Deployment (FFD): When there are several candidate edge servers that meet the constraints, the edge server that meets the resource requirements for the first time is selected to run each virtual machine.
- Particle Swarm Optimization (PSO): When a number of candidate edge servers meet the resource constraints, the edge server satisfying the resource demands is selected to run each virtual machine via the particle swarm optimization algorithm [44].
- Genetic Algorithm (GA): When a number of candidate edge servers meet the resource constraints, the edge server that meets the resource requirements is selected via the genetic algorithm to run each virtual machine [16].

B. Experimental results and evaluation

We first compared the resource wastage level, resource load imbalance level, energy consumption and bandwidth resource consumption of the edge clouds while handling a group of IIoT applications under the condition of specified communication latency; and then analyzed and studied the relevant experimental parameters including the number of base stations and the number of IIoT applications.

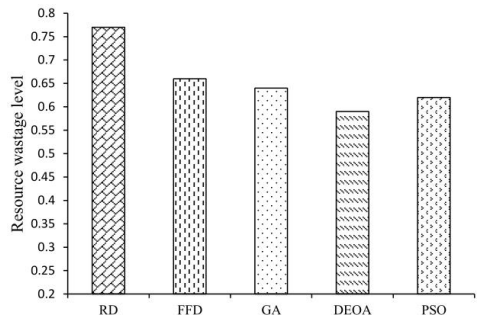


Fig. 5. Comparison of resource wastage level

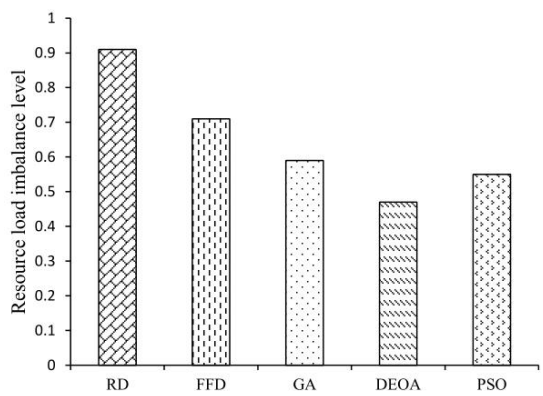


Fig. 6. Comparison of resource load imbalance level

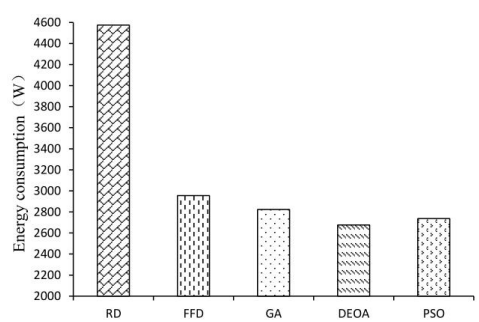


Fig. 7. Comparison of energy consumption

1) Performance analysis: As shown in Figs.5 to 9 and Table II, since RD approach randomly deploys the virtual machines accommodating the tasks, its resource wastage level and resource load imbalance level are both the highest of

TABLE II
THE PERFORMANCE ANALYSIS OF DEOA APPROACH AND OTHER RELATED APPROACHES

	RD	FFD	GA	DEOA	PSO
Resource wastage level	0.77	0.66	0.64	0.59	0.62
Resource load imbalance level	0.91	0.71	0.59	0.47	0.55
Energy consumption	4575W	2955W	2823W	2676W	2737W
Bandwidth resource consumption	1240Mb	1211Mb	1161Mb	1115Mb	1141Mb
Communication latency	2.6s	2.35s	2.28s	2.15s	2.26s

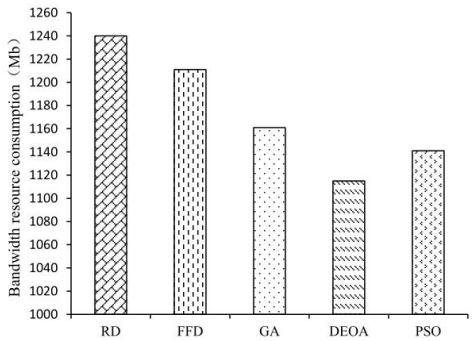


Fig. 8. Comparison of bandwidth resource consumption

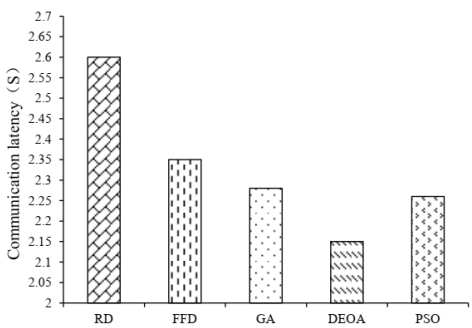


Fig. 9. Comparison of communication latency

four approaches. FFD approach exploits the first adaption to deploy these virtual machines, that is, the next edge server will be selected to host residual virtual machines, only when the first appropriate edge server cannot accommodate them. Although the resources of the edge server can be fully utilized, the resource wastage level and the resource load imbalance level are still very high. GA and PSO both exploit intelligent optimization algorithms to deploy the virtual machines that these tasks are located, but they still have higher resource wastage level and resource load imbalance level than the DEOA approach. The above reasons also result in the decreasing energy consumption of the RD, FFD, GA, PSO, and DEOA. Given that the tasks deployed on the virtual machine need to communicate with other tasks in a cluster, the bandwidth resource consumption of different deployment solutions for these five approaches and the communication latency processing a batch of IIoT applications are also diminishing. Since the communication latency of the RD approach exceeds the communication latency threshold, it is not recommended for the deployment of the IIoT applications.

2) Parameter study: In view of the above approaches, we further analyzed the effect of the number of base stations and the number of IIoT applications on energy consumption and processing resource consumption of the edge clouds, as shown in Fig.10 and Fig.11.

(1) Impact of the number of base stations

Fig.10 shows the effect of the number of base stations on all approaches. For analysis purposes, the number of virtual machines and the IIoT applications was respectively set up to 20 and 60; the corresponding relationship between the count of base stations and the count of edge servers is 15 and 76, 20 and 101, 25 and 127, 30 and 152. The analysis of these figures can be obtained that although the resource wastage level, the resource load imbalance level, energy consumption and bandwidth resource consumption fluctuate with the increase of the count of base stations, it can be thought that there is no such effect. This is due to that a certain number of virtual machines are randomly allocated to the edge servers in edge clouds. Although the count of edge servers has increased, the count of edge servers is still fewer, and then the impact on these indicators is relatively small. In these five approaches, the DEOA approach always has the lowest resource wastage level, the resource load imbalance level, resource load imbalance level, energy consumption and bandwidth resource consumption.

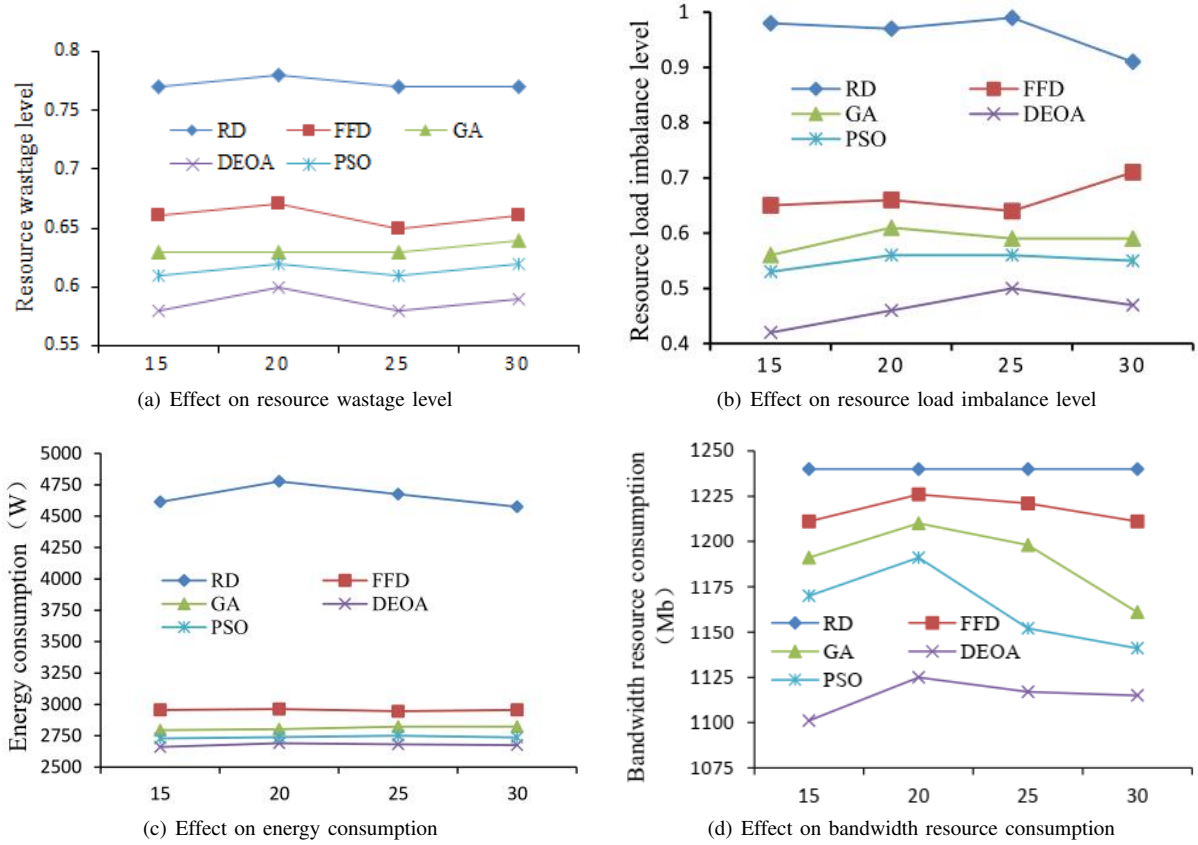
(2) Impact of the number of IIoT applications

Fig.11 shows the effect of the number of IIoT applications on all approaches. For analysis purposes, the count of base stations equals 25, the count of the edge servers equals 127; the number of IIoT applications varied from 5 to 20 by step size 5, and the count of virtual machines changed accordingly to 15, 30, 45, and 60. The analysis of these figures can be found that although the resource wastage level fluctuates with the increase of the number of IIoT applications, it can be thought that there is no such effect. The resource load imbalance level, energy consumption and bandwidth resource consumption of all approaches increase as the number of IIoT applications increases; this is due to that the increase in the number of IIoT applications requires more virtual machines to deal with their tasks. Furthermore, in these five approaches, the DEOA approach always has the lowest resource wastage, energy consumption, resource load imbalance level, and bandwidth resource consumption.

VII. CONCLUSIONS AND FUTURE WORK

With the advent of the 5G and 6G wireless networks, an increasing number of UAVs have adopted 5G or 6G communication technology, and produce a large number of IIoT applications with high computing demand and delay sensitivity. Considering that UAVs cannot meet the computation demand of these IIoT applications, these UAVs offload some

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3222340



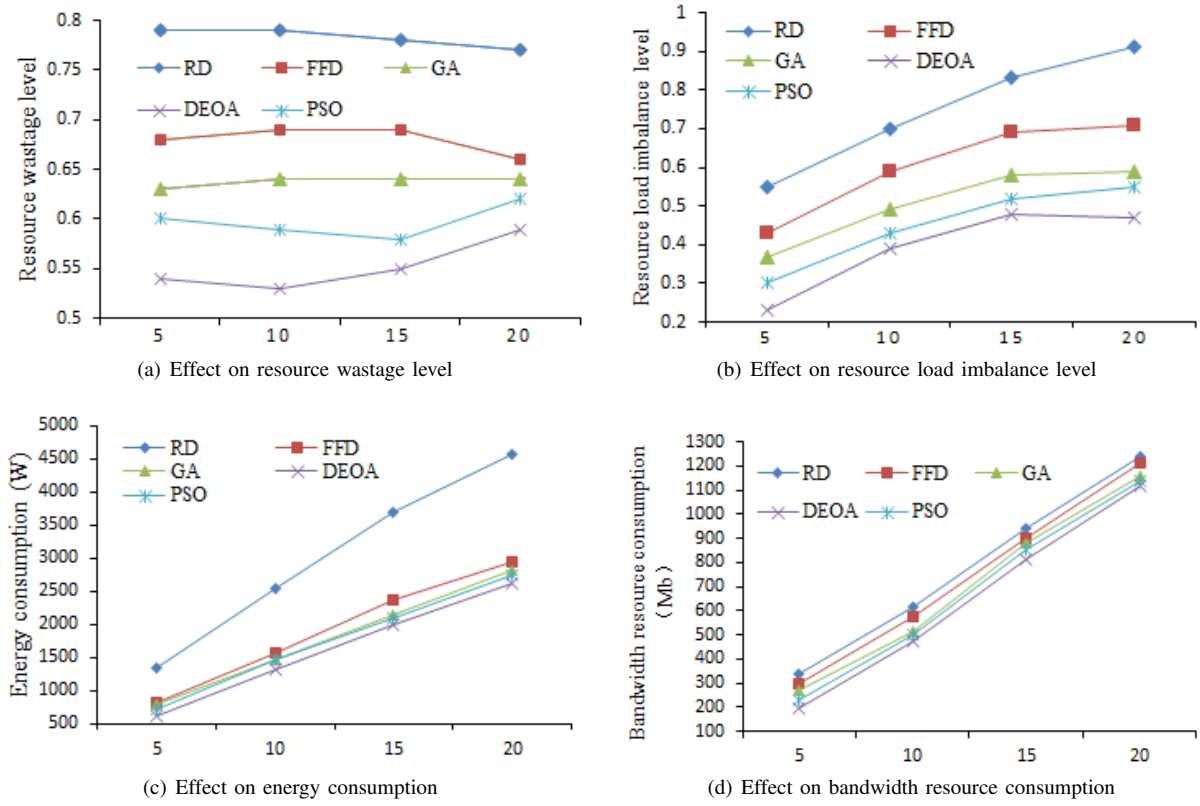
(a) Effect on resource wastage level

(b) Effect on resource load imbalance level

(c) Effect on energy consumption

(d) Effect on bandwidth resource consumption

Fig. 10. Impact of the number of base stations



(a) Effect on resource wastage level

(b) Effect on resource load imbalance level

(c) Effect on energy consumption

(d) Effect on bandwidth resource consumption

Fig. 11. Impact of the number of IIoT applications

tasks of IIoT applications onto the edge clouds or the remote cloud. To improve the system utility and processing efficiency of edge clouds, this paper presented a cooperative resource allocation approach for these IIoT applications to allocate their collaborative computation-intensive tasks. Through the comparative analysis of experimental results, our proposed approach is superior to other related approaches in terms of performance and effectiveness.

In future work, we will need to further reduce the time complexity of our proposed approach. Meanwhile, we will optimize the energy consumption, processor resource consumption for aerial computing environment under the circumstance that IIoT applications are offloaded to UAVs, edge clouds and remote cloud.

ACKNOWLEDGEMENTS

The work of this paper was funded by the National Natural Science Foundation of China under grant No. 62173126, the Science and Technology Innovation Team Support Plan of Henan University under grant No.21IRTSTHN017, the Key Science and Technology Research Project of Henan Province under grant No.222102210047, the Key Science and Technology Research Project of Anyang City under grant No.2021C01GX017, and in part Institute of Information and Communications Technology Planning Promotion (ITTP) Korea No. 2019-0-01816.(Corresponding author: Muhammad Bilal).

REFERENCES

[1] Z. Yong, Z. Rui, and J. L. Teng, "Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges," IEEE Communications Magazine, 2016, 54(5): 36-42.

[2] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges," IEEE Communications Magazine, 2017, 55(3): 22-28.

[3] Z. Yong, Q. Wu, and Z. Rui, "Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond," Proceedings of the IEEE, 2019, 107(12): 2327-2375.

[4] L. Gu, D. Zeng, S. Guo, Y. Xiang, and J. Hu, "A General Communication Cost Optimization Framework for Big Data Stream Processing in Geo-Distributed Data Centers," IEEE Transactions on Computers, 2016, 65(1):19-29.

[5] J. Li, A. Sai, X. Cheng, W. Cheng, Z. Tian, and Y. Li, "Sampling-Based Approximate Skyline Query in Sensor Equipped IoT Networks," Tsinghua Science and Technology, 2021,26(2): 219-229.

[6] Y. Feng, J. A. Zhang, B. Cheng, X. He, and J. Chen, "Magnetic Sensor-Based Multi-Vehicle Data Association," IEEE Sensors Journal, 2021, 21(21):24709-24719.

[7] J. Mabrouki, M. Azrour,D. Dhiba,Y. Farhaoui, and S. Hajjaji, "IoT-Based Data Logger for Weather Monitoring Using Arduino-Based Wireless Sensor Networks with Remote Graphical Application and Alerts," Big Data Mining and Analytics, 2021, 4(1): 25-32.

[8] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. Hajjaji, "Intelligent monitoring system for biogas detection based on the Internet of Things: Mohammedia, Morocco city landfill case," Big Data Mining and Analytics, 2021, 4(1): 10-17.

[9] L. Wang, X. Zhang, R. Wang, C. Yan, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," Knowledge-Based Systems, 2020,204(Sep.27):106196.1-16196.11.

[10] J. Huang, S. Li, and Y. Chen, "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing," Peer-to-Peer Networking and Applications, 2020, 13(2020):1776–1787.

[11] K. Zhang, X. Gui, D. Ren, J. Li, J. Wu, and D. Ren, "Survey on Computation Offloading and Content Caching in Mobile Edge Networks," Journal of Software, 2019, 30(8):2491-2516.

[12] M. Niu, B. Cheng, Y. Feng, and J. Chen, "GMTA: A Geo-Aware Multi-Agent Task Allocation Approach for Scientific Workflows in Container-Based Cloud," IEEE Transactions on Network and Service Management, 2020, 17(3): 1568-1581.

[13] S. Li, J. Huang, J. Hu, and B. Cheng, "QoE-DEER: A QoE-Aware Decentralized Resource Allocation Scheme for Edge Computing," IEEE Transactions on Cognitive Communications and Networking, 2022, 8(2): 1059-1073.

[14] C. Hong, and B. Varghese, "Resource Management in FogEdge Computing A Survey on Architectures, Infrastructure, and Algorithms, "ACM Computing Surveys, 2019, 52(5):97:1-97:37.

[15] S. Li, and J. Huang, "Energy Efficient Resource Management and Task Scheduling for IoT Services in Edge Computing Paradigm," 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC 2017), pp. 846-851, 2017.

[16] S. Deng, Z. Xiang, J. Taheri, K. A. Mohammad, J. Yin, A. Y. Zomaya, and S. Dustdar, "Optimal Application Deployment in Resource Constrained Distributed Edges," IEEE Transactions on Mobile Computing, 2021, 20(5): 1907-1923.

[17] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," in Proceedings of the World Wide Web Conference (WWW 2018), pp. 187-196,2018.

[18] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in Proceedings of the 35th IEEE International Conference on Computer Communications (INFOCOM 2016), 2016, DOI:10.1109/INFOCOM.2016.7524340.

[19] J. Meng, H. Tan, C. Xu, W. Cao, L. Liu, and B. Li, "Dedas: Online task dispatching and scheduling with bandwidth constraint in edge computing," in Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2019), pp. 2287-2295, 2019.

[20] Y. Chen, S. Deng, H. Ma, and J. Yin, "Deploying Data-intensive Applications with Multiple Services Components on Edge," Mobile Networks and Applications, 2020,25: 426-441.

[21] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint Computation Offloading and User Association in Multi-task Mobile Edge Computing," IEEE Transactions on Vehicular Technology, 2018, 67(12):12313-12325.

[22] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things," IEEE Transactions on Cloud Computing, 2021,9(3):1050-1060.

[23] Y. Liu, S. Xie, and Y. Zhang, "Cooperative Offloading and Resource Management for UAV-Enabled Mobile Edge Computing in Power IoT System," IEEE Transactions on Vehicular Technology, 2020, 69(10): 12229-12239.

[24] Y. Chen, S. Deng, H. Zhao, Q. He, Y. Li, and H. Gao, "Data-intensive application deployment at edge: A deep reinforcement learning approach," in Proceedings of the IEEE International Conference on Web Services (ICWS 2019), pp. 355-359,2019.

[25] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate Edge and Cloud Computing with Distributed Deep Learning for Smart City Internet of Things," IEEE Internet of Things Journal, 2020, 7(9):8099-8110.

[26] S. Pallewatta, V. Kostakos, and R. Buyya, "Microservices-based IoT Application Placement within Heterogeneous and Resource Constrained Fog Computing Environments," in Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2019), pp. 71-81, 2019.

[27] M. Goudarzi, H. Wu, M. S. Palaniswami, and R. Buyya, "An Application Placement Technique for Concurrent IoT Applications in Edge and Fog Computing Environments," IEEE Transactions on Mobile Computing, 2021, 20(4): 1298-1311.

[28] H. Yang, J. Zhao, J. Nie, N. Kumar, K. -Y. Lam, and Z. Xiong, "UAV-Assisted 5G/6G Networks: Joint Scheduling and Resource Allocation Based on Asynchronous Reinforcement Learning," in Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2021), 2021, DOI:10.1109/INFOCOMWKSHPS51825.2021.9484604.

[29] M. Adhikari, A. Munusamy, N. Kumar, and S. N. Srirama, "Cybertwin-driven Resource Provisioning for IoE Applications at 6G-enabled Edge Networks," IEEE Transactions on Industrial Informatics, 2022, 18(7): 4850-4858.

[30] J. Oueis, E. Calvanese-Strinati, A. D. Domenico, and S. Barbarossa, "On the impact of backhaul network on distributed cloud computing," in Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW 2014), pp.12-17, 2014.

[31] M. Soualhia, C. Fu, and F. Khomh, "Infrastructure fault detection and prediction in edge cloud environments," in Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC 2019), pp. 222-235,2019.

[32] X. Liu, B. Cheng, and S. Wang, "Availability-aware and Energy-efficient Virtual Cluster Allocation Based on Multi-objective Optimization in Cloud Datacenters," IEEE Transactions on Network and Service Management, 2020, 17(2): 972-985.

[33] C. Lien, Y. Bai, and M. Lin, "Estimation by software for the power consumption of streaming-media servers," IEEE transactions on instrumentation and measurement, 2007, 56(5): 1859-1870.

[34] V. Maio, G. Kecskemeti, and R. Prodan, "An improved model for live migration in data centre simulators," in Proceedings of the IEEE/ACM international conference utility and cloud computing (UCC 2016), pp. 108-117, 2016.

[35] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah,"Analyzing the energy efficiency of a database server," in Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2010), pp. 231-242, 2010.

[36] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-Aware and Performance-Guaranteed Virtual Machine Placement in the Cloud," IEEE Transactions on Parallel and Distributed Systems, 2018, 29: 1385-1400.

[37] R. Householder, S. Arnold, and R. C. Green, "On Cloud-based Oversubscription," international journal of engineering trends and technology, 2014, 8: 425-431.

[38] R. Storn, and K. Price, "Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, 1997, 11: 341-359

[39] R. Storn, "On the usage of differential evolution for function optimization," in Proceedings of the North American Fuzzy Information Processing Society (NAFIPS 1996), pp. 519-523, 1996.

[40] L. Tang, Y. Dong, and J. Liu, "Differential Evolution with an Individual-Dependent Mechanism," IEEE Transactions on Evolutionary Computation, 2015, 19(4): 560-574.

[41] J. Liu, C. Liu, B. Wang, G. Gao, and S. Wang, "Optimized Task Allocation for IoT Application in Mobile Edge Computing," IEEE Internet of Things Journal, 2022, 9(13):10370-10381.

[42] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," Software Practice and Experience, 2016, 47(A): 1275-1296.

[43] A. Beloglazov, and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[44] J. Liu, S. Wang, A. Zhou, S. A. Kumar, F. Yang, and R. Buyya, "Using proactive fault-tolerance approach to enhance cloud service reliability," IEEE Transactions on Cloud Computing, 2018, 6(4): 1191-1202.

Jialei Liu received Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China, in 2018 and M.S. degree in computer science and technology from Henan Polytechnic University, China, in 2008. He is an associate professor at the School of Computer, Hubei University of Arts and Science, China. His major research interests include cloud computing, edge intelligence, and aerial computing.

Guosheng Li received his bachelor degree from Qufu Normal University, China, in 2003. Since 2004, he has been working as a lecturer in Weifang University of Science and Technology, China. He has published some research papers in reputable journals and conferences. His research interests include big data and computer networks.

Quanzhen Huang received the Ph.D. degree in Control Science from Shanghai University, China, in 2012 and the M.S. degree in Control Science from Henan Polytechnic University, China, in 2009. He is a professor at School of Electrical Information Engineering, Henan Institute of Engineering, China. His major research interests include control science and aerial computing.

Muhammad Bilal received the Ph.D. degree in information and communication network engineering from the School of Electronics and Telecommunications Research Institute (ETRI), Korea University of Science and Technology, in 2017. From 2017 to 2018, he was with Korea University, where he was a Postdoctoral Research Fellow with the Smart Quantum Communication Center. In 2018, he joined Hankuk University of Foreign Studies, South Korea, where he is currently working as an Associate Professor with the Division of Computer and Electronic Systems Engineering. He is the author/co-author of 80+ articles published in renowned journals, including IEEE Transactions on Intelligent Transportation System, IEEE IoT Journal, IEEE Systems Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Network and Service Management, and IEEE Transactions on Network Science and Engineering, etc.), 1 book editorship, 2 book chapters, 7 published proceedings papers, 3 issued US patents, and 6 Korean patents. His research interests include Network Optimization, Cyber Security, the Internet of Things, Vehicular Networks, Information-Centric Networking, Digital Twin, Artificial Intelligence, and Cloud/Fog Computing. Dr. Bilal has served as a Technical Program Committee Member on many international conferences, including the IEEE VTC, the IEEE ICC, ACM SigCom, and the IEEE CCNC. He is an editorial board member of IEEE Transactions on Intelligent Transportation Systems, IEEE Future Directions in Technology, Policy, and Ethics Newsletter, Alexandria Engineering Journal (Elsevier), and Intelligent Automation & Soft Computing, Frontiers in Communications and Networks, Frontiers in the Internet of Things, and Co-EiC of the International Journal of Smart Vehicles and Smart Transportation.

Xiaolong Xu received his B.Sc. and M.Sc. degrees in computer science from Nanjing University of Information Science and Technology, in 2010 and 2013, respectively, and the Ph.D. degree from the Nanjing University, China, in 2016. He worked as a research scholar at Michigan State University, USA, from Apr. 2017 to May 2018. He is currently a professor with the school of computer and software, Nanjing University of Information Science and Technology. He has published more than 30 peer review papers in international journals and conferences.

Houbing Song (M'12–SM'14) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in August 2012. He is currently a Tenured Associate Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us), University of Maryland, Baltimore County (UMBC), Baltimore, MD. Prior to joining UMBC, he was a Tenured Associate Professor of Electrical Engineering and Computer Science at Embry-Riddle Aeronautical University, Daytona Beach, FL. He has served as an Associate Technical Editor for IEEE Communications Magazine (2017-2020), an Associate Editor for IEEE Internet of Things Journal (2020-present), IEEE Transactions on Intelligent Transportation Systems (2021-present), and IEEE Journal on Miniaturization for Air and Space Systems (J-MASS) (2020-present). He is the editor of eight books, the author of more than 100 articles and the inventor of 2 patents. His research interests include cyber-physical systems/internet of things, cybersecurity and privacy, and AI/machine learning/big data analytics. His research has been sponsored by federal agencies (including National Science Foundation, US Department of Transportation, and Federal Aviation Administration, among others) and industry. His research has been featured by popular news media outlets, including IEEE GlobalSpec's Engineering360, Association for Uncrewed Vehicle Systems International (AUVSI), Security Magazine, CXOTech Magazine, Fox News, U.S. News & World Report, The Washington Times, and New Atlas. Dr. Song is a senior member of IEEE and ACM, and an ACM Distinguished Speaker. Dr. Song is a Highly Cited Researcher identified by Clarivate™ (2021, 2022) and a Top 1000 Computer Scientist identified by Research.com. Dr. Song was a recipient of 10+ Best Paper Awards from major international conferences, including IEEE CPSCom-2019, IEEE ICII 2019, IEEE/AIAA ICNS 2019, IEEE CBDCom 2020, WASA 2020, AIAA/ IEEE DASC 2021, IEEE GLOBECOM 2021 and IEEE INFOCOM 2022.