IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2019, accepted December 25, 2019, date of publication January 6, 2020, date of current version January 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964073

Distributed Fog Computing for Latency and Reliability Guaranteed Swarm of Drones

XIANGWANG HOU[1], (Student Member, IEEE), ZHIYUAN REN[1], (Member, IEEE), JINGJING WANG[2], (Member, IEEE), SHUYA ZHENG[1], WENCHI CHENG[1], (Senior Member, IEEE), AND HAILIN ZHANG[1], (Member, IEEE)

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[2]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Zhiyuan Ren (zyren@s-an.org)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1803301, Grant SQ2016YFHZ021501, and Grant 2018YFE0126000, in part by the National Natural Science Foundation of China under Grant 61571338, in part by the Key Research and Development Program of Shaanxi Province under Grant 2017ZDCXL-GY-05-01, in part by the Xi'an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36, and in part by the 111 Project in Xidian University of China under Grant B08038.

ABSTRACT Swarm of drones, as an intensely significant category of swarm robots, is widely used in various fields, e.g., search and rescue, detection missions, military, etc. Because of the limitation of computing resource of drones, dealing with computation-intensive tasks locally is difficult. Hence, the cloud-based computation offloading is widely adopted, nevertheless, for some latency-sensitive tasks, e.g., object recognition, path planning, etc., the cloud-based manner is inappropriate due to the excessive delay. Even in some harsh environments, e.g., disaster area, battlefield, etc., there is no wireless infrastructure existed to combine the drones and cloud center. Thus, to solve the problem encountered by cloud-based computation offloading, in this paper, Fog Computing aided Swarm of Drones (FCSD) architecture is proposed. Considering the uncertainty factors in harsh environments which may threaten the success of FCSD processing tasks, not only the latency model, but also the reliability model of FCSD is constructed to guarantee the high reliability of task completion. Moreover, in view of the limited battery life of the drone, we formulated the problem as the task allocation problem which minimized the energy consumption of FCSD under the constraints of latency and reliability. Furthermore, to speed up the process of the optimization problem solving to improve the practicality, relying on the recent advances in distributed convex optimization, we develop a fast Proximal Jacobi Alternating Direction Method of Multipliers (ADMM) based distributed algorithm. Finally, simulation results validate the effectiveness of our proposed scheme.

INDEX TERMS Swarm of drones, distributed fog computing, latency, reliability, energy consumption.

I. INTRODUCTION

Swarm of drones, which consists of several small and low-cost drones, has drawn great attention both of academia and industry, especially in military [1]. Through working collaboratively, drones swarm has demonstrated great capabilities, and gained significant advantages in some tasks which are difficult for single large drone to accomplish. Thanks to the low-cost and superior performance characteristics, swarm of drones has been used in various fields, e.g., military, search-and-rescue, intelligent agriculture, etc [2]–[9].

Most of the tasks that the drones swarm need to cope are computation-intensive, e.g., topographic mapping, object recognition, etc [10]. Nevertheless, the limitations of resources (e.g., battery life, computing capability, etc.) that a single low-cost drone equipped limit its ability to handle the tasks alone [11]. Therefore, to deal with the computation-intensive tasks, the cloud-based computation offloading is widely adopted [12]. Through offloading the computation-intensive tasks to the remote cloud server, the computing results will be obtained. For the tasks which are insensitive

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

to the latency, the cloud-based working manner is intensely appropriate, but in practice, quite a few tasks which the drones need to process are sensitive to latency, e.g., dynamic object recognition, emergency obstacle avoidance, etc. However, because of the long distance between the cloud server and the drones, the cloud-based approach may cause the excessive delay. Even in some harsh environments, e.g., disaster area, battlefield, etc., there is no available wireless infrastructure to combine drones with a cloud server. Hence, the cloud-based computation offloading is unsuitable for latency-sensitive tasks.

Fog computing [13]–[15], which extends the computing service closer to the users, motivates us to introduce it into swarm of drones to make up the shortcomings of cloud computing in dealing with latency-sensitive services. The drones which are close to the task initiator drone are thought to be fog computing nodes to complete the computing task collaboratively for reducing the latency. Besides, in practice, swarm of drones usually works in harsh environments, and inevitable disturbances (e.g., hardware damage, software breakdown, communication link failure, etc.) are likely to result in the failure of the tasks. Hence, besides considering the latency guarantee, a proper reliability-guarantee mechanism is especially needed. Nevertheless, most researches about task allocation among drones swarm are focusing on non-real-time tasks [16]–[22], and there lacks full knowledge on the task allocation which both considering the latency and reliability [23]. Otherwise, although the energy consumption of drone is mostly caused by the movements, the energy consumption of processing continuous computing tasks cannot be ignored. Hence, considering the limited battery capacity of drones, the energy consumption both of computing and communication in dealing with computing tasks to which should be paid attention.

Therefore, in this paper, focusing on the latency and reliability sensitive computing tasks, we introduce fog computing into swarm of drones, and construct a task allocation optimization problem which jointly considering the latency, reliability and energy consumption, in order to minimize the energy consumption of the swarm of drones when the latency and reliability requirements are met. Considering the stringent latency requirements of the latency and reliability sensitive tasks, a fast and efficient algorithm is intensely needed. Therefore, benefit by the recent advances in distributed convex optimization, a fast Proximal Jacobi Alternating Direction Method of Multipliers (ADMM) based distributed task allocation algorithm is proposed, which decompose the optimization problem into several subproblems, and each drone can solve the subproblem using their local status information separately. Furthermore, we compare it with the centralized convex optimization algorithm and the heuristic algorithm proposed in our conference version, i.e., latency and reliability constrained minimum energy consumption algorithm based on genetic algorithm (LRGA-MIE) [24].

In summary, the main contributions of this paper are as follows:

- To enhance the capability of drones swarm handling the computation-intensive tasks, the Fog Computing aided Swarm of Drones (FCSD) architecture is proposed, which makes up for the shortcomings of cloud-based computation offloading in processing latency and reliability sensitive tasks of drones swarm.
- Focusing on the latency and reliability sensitive tasks, we construct a task allocation optimization problem which jointly considering the latency, reliability and energy consumption. Moreover, to solve the formulated problem fastly and efficiently, a Proximal Jacobi ADMM based distributed task allocation algorithm is designed.
- Extensive simulations show that the proposed distributed algorithm is beneficial in terms of global optimization capability, expansibility, convergence rate, etc., in comparison to the state-of-the-art algorithms, e.g., centralized convex optimization algorithm, LRGA-MIE algorithm, dual decomposition algorithm, etc.

The rest of the paper is organized as follows. In section II we present the related work. Section III presents the system model and problem formulation. Section IV demonstrates the algorithm design in detail. The simulation results are given in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK

The task allocation problem among swarm of drones has been explored extensively with different situations.

Fu *et al.* [17] analyzed the task allocation problem among drones swarm under the limitation of communication bandwidth and proposed a BW-ACCBBA algorithm to conduct task assignment with fewer overall messages passed in the network. Tang *et al.* [25] studied the task allocation problem among drones swarm in uncertain circumstances and constructed a collaborative task assignment mechanism both utilizing the advantages of centralized algorithm and distributed algorithm. Kopeikin *et al.* [26] proposed a task allocation mechanism with communication control to utilize the underutilized drones to play a role as a relay nodes to help the network support the services. Fu *et al.* [27] comprehensively considered the security of the drones, both of the collisions among drones swarm, network attacks and sudden problems, and proposed the corresponding methods to conduct the drones swarm task allocation, and the authors do several real physical flying experiments invalidate the effectiveness of their method. Cui *et al.* [28] considered the requirements of quality of service (QoS) among drones, and proposed a dynamic task allocation model with the principle of intermittent asynchronous communication to achieve the tasks with lower communication overhead. However, to the best of our knowledge, there is no research existed which both considering the latency and reliability.

Furthermore, various algorithms have been researched to solve the task allocation problem among drones swarm, which can be classified into centralized algorithms and distributed algorithms.

Centralized algorithms mainly include convex optimization algorithm, heuristic algorithms, swarm intelligence algorithm. Among them, the basic method is the centralized convex optimization, the authors in [29], [30] proposed a mixed integer linear programming (MILP) based algorithm to obtain the global optimal task allocation scheme but with poor scalability and low search speed. To get an acceptable solution with fast speed, heuristic algorithms have been explored. Gao *et al.* [31] formulated the task allocation problem among multiple drones into multi-objective optimization problem, and proposed particle swarm optimization to solve it. Zhao *et al.* [32] and Li *et al.* [33] introduced the ant algorithm to solve the task allocation problem. Zhao *et al.* [34] designed an improved K-means clustering algorithm of simulated annealing algorithm, which makes the task assignments of drones swarm balanced. Chen *et al.* [35] introduced the genetic algorithm into the task allocation problem of multiple drones. Nevertheless, although the heuristic algorithms are with relatively fast speed and high scalability, but the results are near-optimal and cannot give the concrete gap between the final result and the optimal solution. In addition, the heuristic algorithms are easily falling into the local optimum, which is a troublesome issue to solve.

The research on distributed algorithm is relatively fewer than that of centralized algorithm. Jošilo *et al.* [36] developed a decentralized algorithm for task allocation among multiple fog computing nodes based on game theory. Capitan *et al.* [37] proposed a decentralized mechanism for multiple drones cooperation based on partially observable Markov decision processes. Zhao *et al.* [38] introduced a distributed heuristic task allocation algorithm, and the simulation results validate that the designed method can achieve excellent performance compared with the consensus-based bundle algorithm. However, these algorithms either converge slowly or cannot guarantee the convergence to the optimal solution. For the latency and reliability sensitive tasks we focused, the existing algorithms are not suitable.

III. SYSTEM MODEL AND PROBLEM FORMULATION

To improve the capability of drones swarm handling the computation-intensive tasks, the FCSD architecture is proposed, which aims to make up for the shortcomings of the cloud-based computation offloading in coping with latency and reliability sensitive tasks. The architecture of FCSD is shown in Fig. 1.

The drone dr_0 has a computing task $\Psi_0 \triangleq \{D_0, \alpha_0, T_0, R_0\}$, where D_0 denotes the input size of the total task; T_0 and R_0 represent the latency and the reliability constraints, respectively. We denote by E_0 the total required amount CPU cycles to complete the task Ψ_0. The number of CPU cycles E_0 is modeled as $E_0 = \alpha_0 D_0$, where $\alpha_0(\alpha_0 > 0)$ depends on the computational complexity of the task [39]. The drone dr_0 requests nearby drones dr_i that can serve as the fog nodes to complete the task Ψ_0 collaboratively. These drones available nearby, denoted by a set $\mathcal{D} = \{dr_1, dr_2, \ldots, dr_N\}$, are equipped with storage and computation resources. We denote by f_0 the CPU frequency of the drone dr_0. Similarly, the CPU frequency of the drones available nearby, denoted by a set $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$. The coordinate of the drone dr_0 is (x_0, y_0, z_0). The $\mathcal{C} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \cdots, (x_N, y_N, z_N)\}$ are the three-dimensional coordinates of the drones available nearby, respectively.

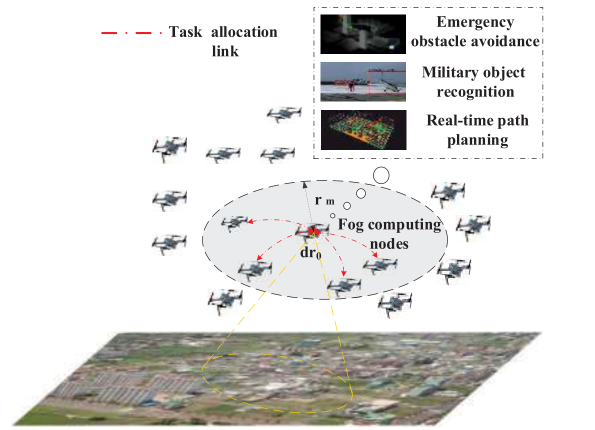


FIGURE 1. The architecture of Fog Computing aided Swarm of Drones (FCSD).

According to [40], the distance between the drone $dr_i \in \mathcal{D}$ and the drone dr_0 can be given by

$$g_{0,i} = [(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2]^{\frac{1}{2}}, \quad g_{0,i} \le r, \tag{1}$$

where r is the maximum communication radius of individual drones.

According to [41], [42], the uplink rate from dr_0 to dr_i can be given as

$$R^{\mathrm{UL}}(0, i) = W^{\mathrm{UL}} \log_2 \left(1 + \frac{P_{\mathrm{Tx}} \left(g_{0,i}{}^{-\gamma} |h_0|\right)}{N_0}\right), \tag{2}$$

where W^{UL} represents the uplink bandwidths between the drone dr_0 and dr_i; P_{Tx} denotes the transmission power of the drone dr_0; γ is the path loss exponent which ranges from $2 \le \gamma \le 5$; h_0 is the complex Gaussian channel coefficient which follows the complex normal distribution CN(0,1); N_0 is the additive white Gaussian noise(AWGN).

In FCSD, the task Ψ_0 will be divided into several subtasks and assigned to multiple drones. In practice, how to divide the task depends not only on the structure of the task, but also on the requirements, which deserves further study. Hence, for simplicity, it can be assumed that task Ψ_0 can be divided into arbitrary proportions with arbitrary precision and there is no overlap existed between any two subtasks, according to [23]. We denote by ρ $(0 \le \rho \le 1)$ the offloading coefficient, therefore, the part of the task Ψ_0 which need to be executed locally by drone dr_0, can be described as $\rho\Psi_0$, and the part of the task which need to be offloaded to the drones available nearby is $(1 - \rho)\,\Psi_0$. Then, we denote the subtask offloaded to the drone dr_i as $\lambda_i(1-\rho)\Psi_0$, where $\lambda_i \in [0,1]$, and $\sum_{i=1}^{N}\lambda_i = 1$. We denote by $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^{\mathrm{T}}$ the task allocation vector.

According to the optimal task allocation scheme, the drone dr_0 and the drones $dr_i \in \mathcal{D}$ are orchestrated to perform distributed computing to complete the task Ψ_0 collaboratively. Note that the flying speeds of the low-cost drones are relatively slow, and the relative positions of them are relatively stable, instead of constantly changing [43], and meantime, the tasks which we studied is latency-sensitive, which is in general processed within a ultra-low duration. Hence, the status of the entire drones swarm will not change, during the extremely short slot from the task initiation to the completion of the task processing. Furthermore, it is reasonable to neglect the impact of the dynamics of the environment on our proposed scheme.

A. LATENCY MODEL

We denote T_{Local} as the latency of the drone dr_0 dealing with the subtask $\rho\Psi_0$ locally, which can be represented by

$$T_{\mathrm{Local}} = \frac{\rho\alpha_0 D_0}{f_0}. \tag{3}$$

When the drone dr_0 offloads the subtask $\lambda_i(1-\rho)\Psi_0$ to the drone dr_i, the size of the transmitted data will be $\beta\lambda_i(1-\rho)D_0$, where β ($\beta \geq 1$) represents a ratio of the transmitted data size to the original task data size due to transmission overhead [39]. Thus, the transmission latency of the subtask from the drone dr_0 to the drone dr_i can be given by

$$T_i^{\mathrm{UL}} = \frac{\beta\lambda_i(1-\rho)D_0}{R^{\mathrm{UL}}(0,i)}. \tag{4}$$

And the computation latency of the subtask addressed on the drone dr_i is expressed as

$$T_i^{\mathrm{Comp}} = \frac{\alpha_0\lambda_i(1-\rho)D_0}{f_i}. \tag{5}$$

According to [44], due to the fact that for many applications (e.g., object recognition), the size of the computation result is much smaller than the size of input data, the transmission latency for the drones to send the computation result back to the initiator drone dr_0 is neglected in general. Hence, the total execution latency of the subtask completed on the drone dr_i can be denoted by

$$\begin{aligned} T_i &= T_i^{\mathrm{UL}} + T_i^{\mathrm{Comp}} \\ &= \frac{\beta\lambda_i(1-\rho)D_0}{R^{\mathrm{UL}}(0,i)} + \frac{\alpha_0\lambda_i(1-\rho)D_0}{f_i}. \end{aligned} \tag{6}$$

Therefore, the total execution latency of the task Ψ_0 can be described as

$$\begin{aligned} T_{\mathrm{Total}} &= \max_{i\in N}\{T_{\mathrm{Local}}, T_i\} \\ &= \max_{i\in N}\left\{\frac{\rho\alpha_0 D_0}{f_0}, \frac{\beta\lambda_i(1-\rho)D_0}{R^{\mathrm{UL}}(0,i)} + \frac{\alpha_0\lambda_i(1-\rho)D_0}{f_i}\right\}. \end{aligned} \tag{7}$$

To meet the latency requirement of the task Ψ_0, the total execution latency T_{Total} should meet the constraint $T_{\mathrm{Total}} \leq T_0$.

B. RELIABILITY MODEL

Considering the various disturbances that the drones swarm might encounter in its working environment, which may result in the failure of the task, reliability is a significant indicator to which must be paid attention.

According to the widely accepted reliability model proposed by Shatz [45], the system reliability is that " the product of the probability that each processor is operational during the time of processing the tasks assigned to it, and the probability that each communication link is operational during the period of the data transmission." The failures of the drones and communication links follow the Poisson process [45], further, the failure rates of the drone dr_0 and dr_i are denoted as ν_0 and ν_i, respectively, and the failure rate of the communication links between dr_0 and dr_i is denoted as $\mu_{0,i}$. Therefore, the computation reliability of the drone dr_0 and dr_i can be represented as $e^{-\nu_i\frac{\rho\alpha_0 D_0}{f_0}}$ and $e^{-\nu_i\frac{\lambda_i(1-\rho)\alpha_0 D_0}{f_i}}$, respectively. And the communication reliability between dr_0 and dr_i can be represented as $e^{-\mu_{0,i}\frac{\lambda_i(1-\rho)\beta D_0}{R^{\mathrm{UL}}(0,i)}}$. The reliability of the subtask which executed locally can be represented as

$$R_{\mathrm{Local}} = e^{-\nu_0\frac{\rho\alpha_0 D_0}{f_0}}. \tag{8}$$

Then, the reliability of the subtask which distributed to the drone dr_i can be represented as

$$R_i = e^{-\nu_i\frac{\lambda_i(1-\rho)\alpha_0 D_0}{f_i} - \mu_{0,i}\frac{\lambda_i(1-\rho)\beta D_0}{R^{\mathrm{UL}}(0,i)}}. \tag{9}$$

Therefore, the reliability of the swarm of drones during the execution time of the task Ψ_0 can be given by

$$\begin{aligned} R_{\mathrm{Total}} &= R_{\mathrm{Local}}\prod_{i=1}^{N} R_i \\ &= e^{-\nu_0\frac{\rho\alpha_0 D_0}{f_0} + \sum_{i=1}^{N}\left(-\nu_i\frac{\lambda_i(1-\rho)\alpha_0 D_0}{f_i} - \mu_{0,i}\frac{\lambda_i(1-\rho)\beta D_0}{R^{\mathrm{UL}}(0,i)}\right)}. \end{aligned} \tag{10}$$

To meet the reliability requirement of the task Ψ_0, the total reliability R_{Total} should meet the constraint $R_{\mathrm{Total}} \geq R_0$.

C. ENERGY CONSUMPTION MODEL

Flight endurance is the bottleneck of swarm of drones, although the energy consumption of drone is mainly caused by the movements, the energy consumption caused by continuous computing tasks processing should also be concerned. Thus, in this paper, we established the energy consumption model both considering the computing and the communication of FCSD in dealing with a single computing task, so as to minimize the energy consumption as far as possible on the premise of ensuring the latency and reliability requirements of the task.

1) COMPUTATIONAL ENERGY CONSUMPTION

The computational energy consumption of the drone dr_0 and dr_i can be given by

$$E_{\text{Local}}^{\text{Comp}} = kf_0^{\sigma} T^{\text{Local}}, \tag{11}$$

and

$$E_i^{\text{Comp}} = kf_i^{\sigma} T_i^{\text{Comp}}, \tag{12}$$

respectively, where kf_0^{σ} and kf_i^{σ} are the computation power of the drone dr_0 and dr_i. According to [46], the $k > 0$ and the $\sigma \geq 2$ (which usually close to 3), are the positive constant. As in [47], the k and the σ can be set as 1.25×10^{-26} and 3, respectively.

Therefore, the total computational energy consumption of swarm of drones is represented as

$$\begin{aligned} E_{\text{Total}}^{\text{Comp}} &= kf_0^{\sigma} T_{\text{Local}} + \sum_{i=1}^{N} kf_i^{\sigma} T_i^{\text{Comp}} \\ &= kf_0^{\sigma} \frac{\rho\alpha_0 D_0}{f_0} + \sum_{i=1}^{N} kf_i^{\sigma} \frac{\alpha_0 \lambda_i (1-\rho) D_0}{f_i}. \end{aligned} \tag{13}$$

2) TRANSMISSION ENERGY CONSUMPTION

The transmission energy consumption of drone dr_0 and drone dr_i can be given by

$$E_{\text{Local}}^{\text{Trans}} = P_{\text{Tx}} T_i^{\text{UL}}, \tag{14}$$

and

$$E_{\text{i}}^{\text{Trans}} = P_{\text{Rx}} T_i^{\text{UL}}, \tag{15}$$

respectively, where P_{Tx} and P_{Rx} denote the transmitting and receiving power of drone dr_0 and dr_i, respectively, which are regarded as constant [47]. Therefore, the total transmission energy consumption of the FCSD system can be given by

$$\begin{aligned} E_{\text{Total}}^{\text{Trans}} &= \sum_{i=1}^{N} E_{\text{Local}}^{\text{Trans}} + \sum_{i=1}^{N} E_{\text{i}}^{\text{Trans}} \\ &= \sum_{i=1}^{N} P_{\text{TR}} \frac{\beta\lambda_i(1-\rho)D_0}{R^{\text{UL}}(0,i)} + \sum_{i=1}^{N} P_{\text{SR}} \frac{\beta\lambda_i(1-\rho)D_0}{R^{\text{UL}}(0,i)}. \end{aligned} \tag{16}$$

In summary, the total energy consumption of swarm of drones can be represented as

$$\begin{aligned} E_{\text{Total}} &= E_{\text{Total}}^{\text{Comp}} + E_{\text{Total}}^{\text{Trans}} \\ &= kf_0^{\sigma} \frac{\rho\alpha_0 D_0}{f_0} + \sum_{i=1}^{N} kf_i^{\sigma} \frac{\alpha_0 \lambda_i (1-\rho) D_0}{f_i} \\ &\quad + \sum_{i=1}^{N} P_{\text{Tx}} \frac{\beta\lambda_i(1-\rho)D_0}{R^{\text{UL}}(0,i)} + \sum_{i=1}^{N} P_{\text{Rx}} \frac{\beta\lambda_i(1-\rho)D_0}{R^{\text{UL}}(0,i)}. \end{aligned} \tag{17}$$

D. PROBLEM FORMULATION

To sum up, a problem to minimize the energy consumption of FCSD within latency and reliability constraints, is modeled as follows:

$$\mathcal{P}1: \min_{\rho,\boldsymbol{\lambda}} E_{\text{Total}} \tag{18}$$

$$\text{s.t. } \rho + \sum_{i=1}^{N} \lambda_i(1-\rho) = 1, \tag{19a}$$

$$T_{\text{Total}} \leq T_0, \tag{19b}$$

$$R_{\text{Total}} \geq R_0, \tag{19c}$$

$$0 \leq \lambda_i, \rho. \tag{19d}$$

IV. ALGORITHM DESIGN

In this section, a centralized Linear Programming (LP) based convex optimal algorithm is constructed as a benchmark, i.e., always with optimal solution, but with high computational complexity. Therefore, to accelerate the decision making process and enhance the scalability, we propose a distributed algorithm based on the Proximal Jacobi ADMM.

A. CENTRALIZED LP-BASED ALGORITHM (BENCHMARK)

To deal with $\mathcal{P}1$, firstly, we simplify the hard constraints of it, and further linearize it into a LP problem for convenient solving.

Constraints (19b) and (19c) are obstacles for problem solving, thus equivalent substitutions are adopted to convert these them into the linear constraints. Constraint (19b) can be replaced by

$$\frac{\rho\alpha_0 D_0}{f_0} \leq T_0, \tag{20}$$

and

$$\frac{\beta\lambda_i(1-\rho)D_0}{R^{\text{UL}}(0,i)} + \frac{\alpha_0\lambda_i(1-\rho)D_0}{f_i} \leq T_0. \tag{21}$$

And due to the monotonically increasing property of exponential functions, constraint (19c) is equivalent to

$$\begin{aligned} \sum_{i=1}^{N} \left(-\nu_i \frac{\lambda_i(1-\rho)\alpha_0 D_0}{f_i} - \mu_{0,i} \frac{\lambda_i(1-\rho)\beta D_0}{R^{\text{UL}}(0,i)} \right) \\ - \nu_0 \frac{\rho\alpha_0 D_0}{f_0} \geq \ln R_0. \end{aligned} \tag{22}$$

Substituting formula (20), (21), and (22) into problem $\mathcal{P}1$, we can obtain a standard convex QP problem $\mathcal{P}2$.

$$\begin{aligned} \mathcal{P}2: \ &\min_{\rho,\boldsymbol{\lambda}} E_{\text{Total}} \\ &\text{s.t. } (19\text{a}), (19\text{d}) \\ &\quad (20), (21), \text{and } (22). \end{aligned} \tag{23}$$

Furthermore, to linearize the problem, we use ρ', $\boldsymbol{\lambda}' = [\lambda_1', \lambda_2', \ldots, \lambda_N']^{\text{T}}$ to substitute the quadratic terms in $\mathcal{P}2$, i.e.,

$$\begin{cases} \rho = \rho'; \\ \lambda_i(1-\rho) = \lambda_i', \end{cases} \tag{24}$$

Through the above process, a convex LP problem is obtained, as shown as:

$$\mathcal{P}3: \min_{\rho',\boldsymbol{\lambda}'} kf_0^{\sigma}\frac{\rho'\alpha_0 D_0}{f_0}+\sum_{i=1}^{N} kf_i^{\sigma}\frac{\alpha_0\lambda_i' D_0}{f_i} + \sum_{i=1}^{N} P_{\mathrm{Tx}}\frac{\beta\lambda_i' D_0}{R^{\mathrm{UL}}(0,i)}+\sum_{i=1}^{N} P_{\mathrm{Rx}}\frac{\beta\lambda_i' D_0}{R^{\mathrm{UL}}(0,i)} \tag{25}$$

$$\text{s.t. } \rho'+\sum_{i=1}^{N}\lambda_i' = 1, \tag{26a}$$

$$\frac{\rho'\alpha_0 D_0}{f_0} \le T_0, \tag{26b}$$

$$\frac{\beta\lambda_i' D_0}{R^{\mathrm{UL}}(0,i)}+\frac{\alpha_0\lambda_i' D_0}{f_i} \le T_0, \tag{26c}$$

$$\sum_{i=1}^{N}\left(\nu_i\frac{\lambda_i'\alpha_0 D_0}{f_i}+\mu_{0,i}\frac{\lambda_i'\beta D_0}{R^{\mathrm{UL}}(0,i)}\right) + \nu_0\frac{\rho'\alpha_0 D_0}{f_0} \le -\ln R_0, \tag{26d}$$

$$0 \le \lambda_i', \rho'. \tag{26e}$$

The problem $\mathcal{P}3$ can be solved by numerous convex optimization methods, e.g., simplex method, dual simplex method, etc. In this paper, we adopt a interior point method here, as a benchmark algorithm for comparing with following distributed algorithm. Algorithm 1 summarized the LP-based algorithm in detail.

Algorithm 1 LP-Based Algorithm

Input: $\Psi_0, \mathcal{D}, \mathcal{F}, \mathcal{C}, f_0, (x_0, y_0, z_0), N, \nu_0, \nu_i, \mu_0, \mu_i, \alpha, \beta$.
Output: $\rho^*, \boldsymbol{\lambda}^*$
1: Solve $\mathcal{P}3$ to obtain ρ' and $\boldsymbol{\lambda}' = [\lambda_1', \lambda_2', \dots, \lambda_N']^{\mathrm{T}}$
2: Substitute ρ' and $\boldsymbol{\lambda}'$ for ρ^* and $\boldsymbol{\lambda}^*$ using Eq. (24) and Eq. (28);
3: **return** $\rho^*, \boldsymbol{\lambda}^*$.

B. DISTRIBUTED ALGORITHM BASED ON PROXIMAL JACOBI ADMM

However, due to the signaling overhead and relatively large computation pressure on the initiator drone, caused by centralized algorithm, especially when the amounts of participating drones are excessive, a decentralized algorithm executed on each participant is intensely needed for practically implementing. Alternating direction method of multipliers (ADMM) [48] is an efficiently distributed algorithm with superior convergence property and robustness, which has drawn great attention in machine learning, image processing, etc., recent years. Therefore, in this paper, we are motivated to introduce the ADMM to solve the computation offloading and task allocation optimization problem distributedly. Unfortunately, the traditional ADMM is proved that it will not converge when extend it directly to solve multi-block problem [49]. The typical method is to introduce new global variables to convert the optimization problem into two-block problem [48], nevertheless, this method will increase the number of global variables, making the optimization problem troublesome. Moreover, traditional ADMM is Gauss-Seidel type, i.e., the variable blocks are updated one after another, which is inappropriate to solve the problem we formulated in parallel. Therefore, to solve the problem mentioned above, we propose a novel distributed computation offloading and task allocation algorithm, based on the latest development of Jacobi type ADMM, named Proximal Jacobi ADMM [50], which update the decision variables in parallel and can converge to the optimal solution with fast rate.

To satisfy the ADMM, we introduce relaxation variables λ_{N+1}' to transform the inequality constraint (26d) into equality constraint, i.e.,

$$\sum_{i=1}^{N}\left(\nu_i\frac{\lambda_i'\alpha_0 D_0}{f_i}+\mu_{0,i}\frac{\lambda_i'\beta D_0}{R^{\mathrm{UL}}(0,i)}\right) + \nu_0\frac{\rho'\alpha_0 D_0}{f_0}+\lambda_{N+1}' = -\ln R_0, \tag{27}$$

in which $\lambda_{N+1}' \ge 0$. For brevity, we denote $\boldsymbol{x} = [x_0, x_1, x_2, \dots, x_{N+1}]^{\mathrm{T}}$, where

$$x_i = \begin{cases} \rho', & i = 0, \\ \lambda_i', & 1 \le i \le N+1. \end{cases} \tag{28}$$

Define that χ_i is the feasible set of x_i, thus, according to constraints (26c), (26c), (26e), the feasible set of x_i is outlined as follows

$$\chi_i = \begin{cases} \left\{x_i \middle| \frac{x_i\alpha_0 D_0}{f_0} \le T_0, x_i \ge 0\right\}, & i = 0, \\ \left\{x_i \middle| \frac{\beta x_i D_0}{R^{\mathrm{UL}}(0,i)}+\frac{\alpha_0 x_i D_0}{f_i} \le T_0, x_i \ge 0\right\}, & 1 \le i \le N, \\ \{x_i | x_i \ge 0\}, & i = N+1. \end{cases} \tag{29}$$

Obviously, $\chi_i \subseteq \mathbb{R}$ is a nonempty closed convex set. With the task allocation vector $\boldsymbol{x}$, the optimization target function of $\mathcal{P}3$ is composed by multiple sub-functions, i.e., the energy consumption of drone dr_0 and dr_i, which can be represented as

$$E_i = \begin{cases} kf_0^{\sigma}\frac{x_i\alpha_0 D_0}{f_0}, & i = 0, \\ kf_i^{\sigma}\frac{\alpha_0 x_i D_0}{f_i}+P_{\mathrm{Tx}}\frac{\beta x_i D_0}{R^{\mathrm{UL}}(0,i)} + P_{\mathrm{Rx}}\frac{\beta x_i D_0}{R^{\mathrm{UL}}(0,i)}, & 1 \le i \le N, \\ 0, & i = N+1. \end{cases} \tag{30}$$

For further simplicity, indicator function is introduced to incorporate the constraints $x_i \in \chi_i$ to the sub-function E_i, i.e.,

$$I_{\chi_i}(x_i) = \begin{cases} 0, & x_i \in \chi_i, \\ +\infty, & x_i \notin \chi_i. \end{cases} \tag{31}$$

For each drone, the sub-function E_i is replaced by:

$$E_i^+ = E_i(x_i) + I_{\chi_i}(x_i). \tag{32}$$

The problem of $\mathcal{P}3$ can be represented as a standard multi-block ADMM problem, i.e.,

$$\mathcal{P}4: \min_{\boldsymbol{x}} \sum_{i=0}^{N+1} E_i^+ \tag{33}$$

$$\text{s.t.} \sum_{i=0}^{N+1} \boldsymbol{A_i} x_i = \boldsymbol{c}. \tag{34}$$

where $\boldsymbol{A_i}$ is

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_0}^{\mathrm{T}} \\ \boldsymbol{A_1}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{A_i}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{A_N}^{\mathrm{T}} \\ \boldsymbol{A_{N}c_1}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} 1 & \dfrac{\nu_0\alpha_0 D_0}{f_0} \\ 1 & \dfrac{\nu_1\alpha_0 D_0}{f_1} + \dfrac{\mu_{0,1}\beta D_0}{R^{\mathrm{UL}}(0,1)} \\ \vdots & \vdots \\ 1 & \dfrac{\nu_i\alpha_0 D_0}{f_i} + \dfrac{\mu_{0,i}\beta D_0}{R^{\mathrm{UL}}(0,i)} \\ \vdots & \vdots \\ 1 & \dfrac{\nu_N\alpha_0 D_0}{f_N} + \dfrac{\mu_{0,N}\beta D_0}{R^{\mathrm{UL}}(0,N)} \\ 0 & 1 \end{pmatrix}^{\mathrm{T}},$$

and $\boldsymbol{c}$ is represented as

$$\boldsymbol{c} = \begin{pmatrix} 1 \\ -\ln R_0 \end{pmatrix}.$$

Dual decomposition methods [51] is a simple distributed algorithm to solve $\mathcal{P}4$. Consider the Lagrangian function of $\mathcal{P}4$:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i=0}^{N+1} E_i^+ - \boldsymbol{u}^{\mathrm{T}} \left(\sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right), \tag{35}$$

where $\boldsymbol{u} \in \mathbb{R}^2$ is the Lagrange multiplier variable. We can decompose the problem $\mathcal{P}4$ into several subproblems for drones swarm to execute it in parallel:

$$\begin{cases} x_0^{k+1} = \arg\min_{x_0} \mathcal{L}_\rho(x_0, x_1, \ldots, x_{N+1}, \boldsymbol{u}^k), \\ \quad\vdots \\ x_i^{k+1} = \arg\min_{x_i} \mathcal{L}_\rho(x_0, x_1, \ldots, x_{N+1}, \boldsymbol{u}^k), \\ \quad\vdots \\ x_{N+1}^{k+1} = \arg\min_{x_{N+1}} \mathcal{L}_\rho(x_0, x_1, \ldots, x_{N+1}, \boldsymbol{u}^k), \\ \boldsymbol{u}^{k+1} = \boldsymbol{u}^{k+1} - \theta_k \left(\sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right), \end{cases} \tag{36}$$

where $\theta_k > 0$ is step-size. Due to the components x_i of task allocation vector $\boldsymbol{x}$ are separable, the updates of x_i are independent. However, in practice, the convergence of dual decomposition method is intensely slow [52]. Therefore, Jocabi type ADMM which integrates the decomposability of dual decomposition method and the superior convergence of multiplier method is presented to solve the computation offloading and task allocation problem. Compared with dual method, ADMM utilizes the augmented Lagrangian function for $\mathcal{P}4$:

$$\mathcal{L}_\eta(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i=0}^{N+1} E_i^+ - \boldsymbol{u}^{\mathrm{T}} \left(\sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right) + \frac{\eta}{2} \left\| \sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right\|_2^2, \tag{37}$$

where $\frac{\eta}{2} \left\| \sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right\|_2^2$ is a quadratic penalty of the constraints to enhance the convexity of Lagrangian function Eq. (35) with a parameter η. Similar to dual decomposition method, Jacobi ADMM can also decompose the problem into several sub-problems, i.e.,

$$\begin{cases} x_0^{k+1} = \arg\min_{x_0} \mathcal{L}_\rho(x_0, x_1^k, \ldots, x_i^k \ldots, x_{N+1}^k, \boldsymbol{u}^k), \\ \quad\vdots \\ x_i^{k+1} = \arg\min_{x_i} \mathcal{L}_\rho(x_0^k, x_1^k, \ldots, x_i \ldots, x_{N+1}^k, \boldsymbol{u}^k), \\ \quad\vdots \\ x_{N+1}^{k+1} = \arg\min_{x_{N+1}} \mathcal{L}_\rho(x_0^k, x_1^k, \ldots, x_{N+1}^k, x_{N+1}, \boldsymbol{u}^k), \\ \boldsymbol{u}^{k+1} = \boldsymbol{u}^{k+1} - \theta \left(\sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right). \end{cases} \tag{38}$$

Expanding the ith sub-problem, it can be represented as

$$x_i^{k+1} = \arg\min_{x_i} E_i^+ + \frac{\eta}{2} \left\| \boldsymbol{A_i} x_i + \sum_{i=0, j\neq i}^{N+1} \boldsymbol{A_j} x_j - \boldsymbol{c} - \frac{\boldsymbol{u}^{\mathrm{T}}}{\eta} \right\|_2^2. \tag{39}$$

However, if we implement the update of x_i directly according to Eq. (38), (39) with any variants [53], it will not converge.

Therefore, to obtain the global optimal solution, a proximal term $\frac{1}{2} \left\| x_i - x_i^k \right\|_{\boldsymbol{P_i}}^2$ for the update of x_i and a damping factor $\xi > 0$ for the update of $\boldsymbol{u}$ are added, hence the updates of x_i and $\boldsymbol{u}$ are replaced by

$$x_i^{k+1} = \arg\min_{x_i} E_i^+ + \frac{\eta}{2} \left\| \boldsymbol{A_i} x_i + \sum_{i=0, j\neq i}^{N+1} \boldsymbol{A_j} x_j - \boldsymbol{c} - \frac{\boldsymbol{u}^{\mathrm{T}}}{\eta} \right\|_2^2 + \frac{1}{2} \left\| x_i - x_i^k \right\|_{\boldsymbol{P}_i}^2, \tag{40}$$

and

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^{k+1} - \theta\xi \left(\sum_{i=0}^{N+1} \boldsymbol{A_i} x_i - \boldsymbol{c} \right), \tag{41}$$

respectively. Where $\boldsymbol{P_i}$ is a symmetric positive semi-definite matrix, and $\frac{1}{2} \left\| x_i - x_i^k \right\|_{\boldsymbol{P_i}}^2$ is defined as

$$\frac{1}{2} \left\| x_i - x_i^k \right\|_{\boldsymbol{P_i}}^2 := \frac{1}{2} (x_i - x_i^k)^{\mathrm{T}} \boldsymbol{P}_i (x_i - x_i^k). \tag{42}$$

According to [54], the choice of $\boldsymbol{P}_i$ is adopted as

$$\boldsymbol{P}_i = \tau \boldsymbol{I}_{\boldsymbol{x}_i} - \eta {\boldsymbol{A}_i}^{\mathrm{T}} \boldsymbol{A}_i, \tag{43}$$

where $\boldsymbol{I}_{\boldsymbol{x}_i}$ is an identity matrix with the same dimension of x_i, in this paper, the dimension is 1. And τ is a key weight parameter that controls the relative weight or balance between the Lagrangian function and the proximal term, i.e., Eq. (42) [55].

Let define

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}, \tag{44}$$

and

$$\boldsymbol{Y} = \begin{pmatrix} \tau \boldsymbol{I}_{\boldsymbol{x}} & \\ & \frac{1}{\eta} \boldsymbol{I}_{\boldsymbol{u}} \end{pmatrix}. \tag{45}$$

To guide the update of τ in the iterations, a judgement function is defined here:

$$\begin{aligned} \Gamma := \; & 2(\boldsymbol{u}^k - \boldsymbol{u}^{k+1})^{\mathrm{T}} \boldsymbol{A} (\boldsymbol{x}^k - \boldsymbol{x}^{k+1}) + \left|\left| \boldsymbol{X}^k - \boldsymbol{X}^{k+1} \right|\right|_{\boldsymbol{Y}}^2 \\ = \; & 2\left(\boldsymbol{u}^k - \boldsymbol{u}^{k+1}\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{x}^k - \boldsymbol{x}^{k+1}\right)^{\mathrm{T}} \\ & + \tau \left|\left| \boldsymbol{x}^k - \boldsymbol{x}^{k+1} \right|\right|_2^2 + \frac{1}{\eta} \left|\left| \boldsymbol{u}^k - \boldsymbol{u}^{k+1} \right|\right|_2^2. \end{aligned} \tag{46}$$

In the iteration, τ updates according to the following rules:

$$\tau^{k+1} = \begin{cases} \tau^k, & \text{if } \Gamma > \varpi \left|\left| \boldsymbol{X}^k - \boldsymbol{X}^{k+1} \right|\right|_{\boldsymbol{Y}}^2, \\ \varrho \tau^k, & \text{otherwise,} \end{cases} \tag{47}$$

where $\varrho > 1$, and $\varpi \in \mathbb{R}^+$ which usually set as small as enough. According to [55], when $\tau^{k+1} \neq \tau^k$, the updates in the $(k+1)$th iteration both of primal variable $\boldsymbol{x}$ and dual variable $\boldsymbol{u}$ should be discarded and recover the values to that of the kth iteration, i.e., $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$, $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k$. According to [50], the value of τ should be assigned a relatively small value initially, and with the iteration, the value of τ increase to make the proportion of proximal term to Eq. (40) larger and larger.

In FCSD, the initiator drone dr_0 takes charge of the updates of x_0 and x_{N+1}, and meantime, other participating drones, i.e., $dr_i \in \mathcal{D}$, are responsible for the updates of x_i, $i \in \{1, 2, \ldots, N\}$, according to Eq. (40). After all the drones finish their subproblems and report their results to the drone dr_0, the drone dr_0 will update the dual variable $\boldsymbol{u}$ according to Eq. (41). Repeat the process mentioned in the above, until the convergence conditions are met or the maximum number of iterations is reached, and the optimal solution will be obtained. According to the optimal solution, i.e., the optimal task allocation strategy, the task will be divided and assigned to the drones swarm for distributed computing collaboratively. More details are summarized in Algorithm 2.

Algorithm 2 Proximal Jacobi ADMM Algorithm

Input: $\Psi_0, \mathcal{D}, \mathcal{F}, \mathcal{C}, f_0, (x_0, y_0, z_0), N, \nu_0, \nu_i, \mu_0, \mu_i, \alpha, \beta, \theta, \xi, \varrho, \varpi, \boldsymbol{x}^0, \boldsymbol{u}^0, \tau^0, MaxIter$
Output: $\rho^*, \boldsymbol{\lambda}^*$

for k = 1 to *MaxIter* **do**
  Each drone Update x_i for $i = \{0, 1, \ldots, N, N+1\}$ in parallel by Eq. (40)
    dr_0: Update x_0^{k+1} and x_{N+1}^{k+1};
    dr_i: Update x_i^{k+1} and send the result to dr_0;
  After all the the x_i^{k+1} are received, dr_0 do:
    Update dual variable $\boldsymbol{u}^{k+1}$ using Eq. (41);
    Update parameter τ^{k+1} using Eq. (47);
  if Stopping criteria are satisfied **then**
    break;
  end if
  dr_0 returns $\boldsymbol{x}^{k+1}$, $\boldsymbol{u}^{k+1}$ and τ^{k+1} to each drone dr_i;
end for
Obtain the optimal solution $\boldsymbol{x}^{k+1}$;
Substitute $\boldsymbol{x}^{k+1}$ for ρ^* and $\boldsymbol{\lambda}^*$ using Eq. (24) and Eq. (28);
return $\rho^*, \boldsymbol{\lambda}^*$.

C. COMPUTATIONAL COMPLEXITY ANALYSIS OF THE ALGORITHMS

For the centralized optimal algorithm, i.e., the LP-based algorithm, the computational complexity is typically $O\left((2N+2)^{3.5} * (N+3)^2\right)$. Although it can achieve the optimal solution, the scalability is poor. Thus, to obtain an acceptable solution with relatively fast speed, an heuristic algorithm, i.e., LRGA-MIE algorithm is proposed in our conference version [24], the computational complexity of LRGA-MIE is $\mathcal{O}(G * S * (N+1))$, where S and G represent the population size and maximum iterations number, respectively. According to [56], S and G are linear functions with respect to $(N+1)$, hence the complexity of LRGA-MIE can be represented as $\mathcal{O}\left((N+1)^3\right)$. However, the heuristic algorithms are easily falling into local optimum, and the selection of algorithm parameters is also troublesome. Therefore, the proposed Proximal Jacobi ADMM based distributed algorithm is a better method, which can achieve the optimal solution as well as with fast speed. The computational complexity on the drone dr_0 is $O(N+1)$, and solving each subproblem in each drone $dr_i \in \mathcal{D}$ is $O(1)$. Furthermore, according to [50] and [55], if τ is adjusted as demonstrated in Eq. (47), the Proximal Jacobi ADMM based distributed algorithm will converge to the optimal solution with an $o(1/k)$ convergence rate.

V. SIMULATION RESULTS

In this section, simulation results of the proposed algorithms are presented.

A. PARAMETER SETTINGS

Unless otherwise specified, referring to [41], [47], [57], the system parameters of FCSD are set as follows:

TABLE 1. System parameters of FCSD.

Parameter	Value	Parameter	Value
W^{UL}	0.5 MHz	f_0, f_i	U([0.3, 0.8] GHz)
N_0	-100 dBm	(x_0, y_0, z_0)	(0 m, 0 m, 0 m)
P_{Tx}	1.258 W	(x_i, y_i, z_i)	randomly in 100 m^3 area
P_{Rx}	1.181 W	ν_0, ν_i	U([0, 0.005])
γ	3	μ_i	U([0, 0.005])
h_0	CN(0,1)	f_c	1 GHz
k	1.25×10^{-26}	W^c	2 MHz
σ	3	μ_c	0.17
r	100 m	ν_c	0.00001
β	1	(x_c, y_c, z_c)	(2000 m, 2000 m, 2000 m)

1) DEFAULT SYSTEM PARAMETERS OF FCSD

$W^{\mathrm{UL}} = 0.5$ MHz, $N_0 = -100$ dBm, $P_{\mathrm{Rx}} = 1.181$ W, $P_{\mathrm{Tx}} = 1.258$ W, $\gamma = 3$. h_0 follows the complex normal distribution CN(0,1). $k = 1.25 \times 10^{-26}$, $\sigma = 3$, $r = 100$ m. And due to the communication overhead is much small to neglect, the β is set to 1 [23]. The CPU-frequencies of swarm of drones including dr_0 and dr_i, are assumed to be uniformly distributed, i.e., $f_0, f_i \sim$ U([0.3, 0.8] GHz), The coordinates of the master drone dr_0 are set to the origin, i.e., (0 m, 0 m, 0 m), and the coordinates of the drones nearby available are distributed randomly in 100 m^3 area. The failure rates of the drones dr_0 and dr_i are assumed to be uniformly distributed, i.e., $\nu_0, \nu_i \sim$ U([0, 0.005]). And the failure rates of the communication links between dr_0 and dr_i are also assumed to be uniformly distributed, i.e., $\mu_i \sim$ U([0, 0.005]). We assume that the fog nodes are 10. i.e., $N = 10$. For convenient reference, the main default system parameters of FCSD are summarized in Table. 1.

2) DEFAULT TASK PARAMETERS

Unless otherwise specified or used as variables, in the following simulations, the parameters of the task Ψ_0 are set as follows. D_0 is 4 Mb. α_0 is $\frac{1900}{8}$ cycles/bit to denote a computation-intensive task, i.e., x264 CBR encode task [46]. T_0 and R_0 are set as 0.8 s and 99%, respectively.

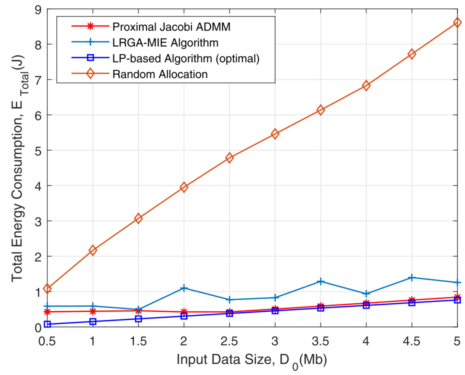
3) DEFAULT ALGORITHM PARAMETERS

The parameters of the Proximal Jacobi ADMM are set as follows: The step-size of θ is set as 10. The damping factor ξ is 1. ϱ is 2. ϖ is 10^{-26}. $\boldsymbol{x}^0$ is random initialized range in (0,1). $\boldsymbol{u}^0$ is initialized as $\boldsymbol{Ax^0} - \boldsymbol{c}$. τ^0 is set as $0.4*(N+1)*\theta$. *MaxIter* is 50. The primal residual convergence accuracy is set as 10^{-2}. The parameters of the LRGA-MIE algorithm are set as follows: The maximum number of iterations is 50. The population size is 100. The crossover and mutation probabilities are set as 0.2 and 0.2, respectively.

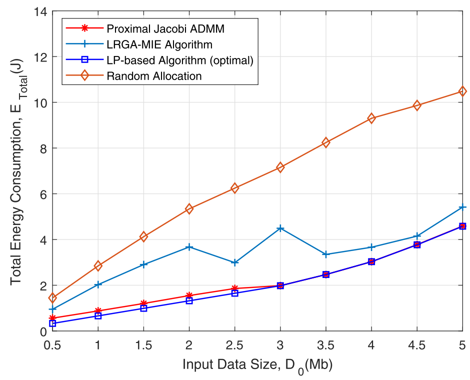
B. EXPERIMENTS ANALYSIS

1) ENERGY CONSUMPTION PERFORMANCE

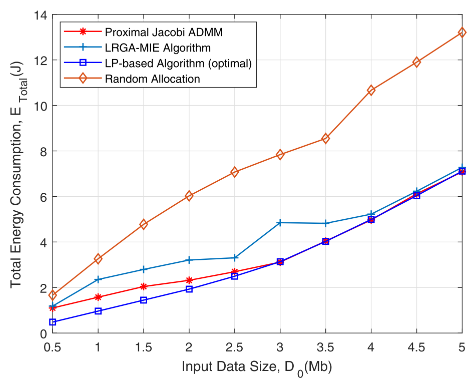
Fig. 2 shows the energy consumption performance comparison of different algorithms when processing different tasks, i.e., gzip ASCII compress task, x264 VBR encode task, x264 CBR encode task. Correspondingly, the computational complexities of the tasks mentioned above are set as $\frac{330}{8}$ cycles/bit, $\frac{1300}{8}$ cycles/bit, $\frac{1900}{8}$ cycles/bit, respectively [46]. As we can see, with the increasing of input data size, the system energy consumption with different algorithms are increased gradually but with different rates. When assigning the tasks without optimization, i.e., random allocation, the processing energy consumption of FCSD system is pretty large, compared with that of the optimized ones. But different



(a) Energy consumption performance comparison of different algorithms towards gzip ASCII compress task, i.e., computational complexity $\alpha_0 = \frac{330}{8}$ cycles/bit.

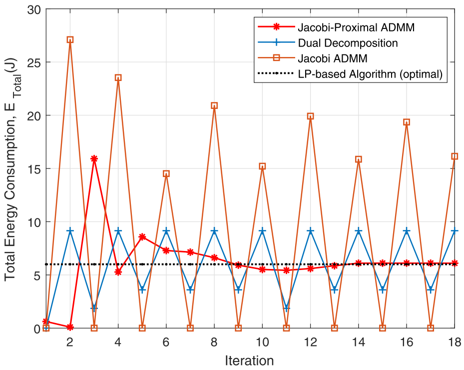


(b) Energy consumption performance comparison of different algorithms towards x264 VBR encode task, i.e., computational complexity $\alpha_0 = \frac{1300}{8}$ cycles/bit.

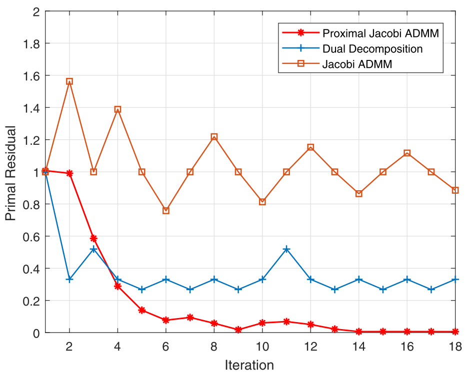


(c) Energy consumption performance comparison of different algorithms towards x264 CBR encode task, i.e., computational complexity $\alpha_0 = \frac{1900}{8}$ cycles/bit.

FIGURE 2. Energy consumption performance comparison of different algorithms towards different kinds of tasks versus input data size.



(a) Energy consumption values with the number of iterations.



(b) Primal residual values with the number of iterations.

FIGURE 3. Convergence performance of different distributed algorithms for the formulated multiple-block optimization problem.

algorithms are with different performance. The LP-based algorithm (optimal algorithm) achieves the lowest energy consumption all along. And the performance of Proximal Jacobi ADMM based algorithm is always close to the optimal one, especially when the processing tasks with large input data size. Although the LRGA-MIE algorithm achieves relatively good performance compared with random task assignment, the performance is poor and unstable compared with LP-based algorithm and Proximal Jacobi ADMM based algorithm. The reason is that the genetic algorithm is easy falling into the local optimal solution although with fast search speed.

The convergence and the convergence rate of distributed algorithm are the key factors whether the algorithm can be adopted in practice. Thus in Fig. 3, we show the good convergence performance of Proximal Jacobi ADMM, and compared with dual decomposition algorithm, which is the origin of the ADMM, as well as the Jacobi ADMM without proximal term. We can see that the Proximal Jacobi ADMM converge fast, and after the 14th iteration, the energy consumption performance of Proximal Jacobi ADMM is gradually fitted to the optimal value (see Fig. 3(a)), and the primal residual (see Fig. 3(b)) is gradually fitted to 0, which indicate that the algorithm is converged. The dual decomposition algorithm converges intensely slowly, in fact, it may takes hundreds of iterations for the algorithm to converge. But for

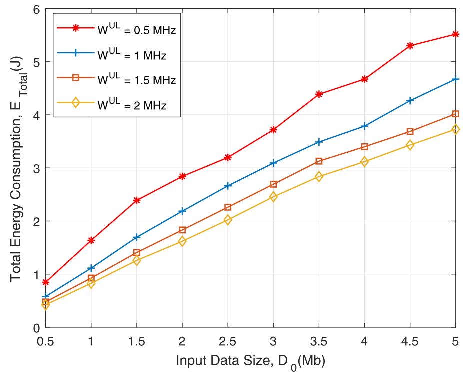


FIGURE 4. Impact of transmission bandwidth on the energy consumption.

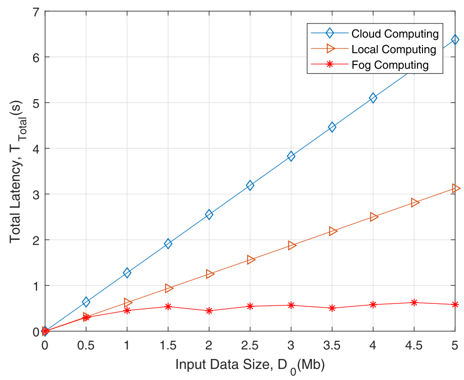


FIGURE 5. Comparison of latency performance of different computing paradigms.

Jacobi ADMM without proximal term, it cannot converge, although hundreds of iterations we took. Similar conclusions can be found in [50] and [55].

Within the same requirements of latency and reliability. Fig. 4 show the impact of bandwidth on the system energy consumption. As we can see that the smaller the bandwidth, the lower the energy consumption. For instance, when the input data size is 5 Mb, the energy consumption of the FCSD system with a bandwidth of 0.5 MHz is 48 % greater than that of the FCSD system with a bandwidth of 2 MHz, 37 % greater than that of the FCSD system with a bandwidth of 1.5 MHz, 18 % greater than that of the system with a bandwidth of 1 MHz. However, with the increasing of system bandwidth, the impact of bandwidth on the system energy consumption will become insignificant.

2) LATENCY PERFORMANCE

Fig. 5 shows the comparison of different computing paradigms, i.e., cloud computing, local computing, and fog computing. We can observe that when the input data size is relatively small, all of three computing paradigms are with small latency, which can satisfy the latency requirement of the task well. For instance, when the input data size is less than 0.5 Mb, the latency of three computing paradigms are all less than 1 s. However, with the increasing of input data size, the latency of cloud computing increases rapidly. The reason is that the cloud computing is far away from the drone

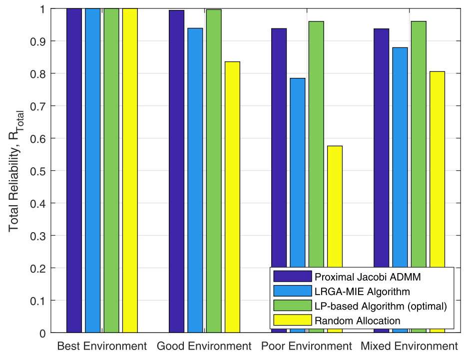


FIGURE 6. Reliability performance comparison of different algorithms in different environments.

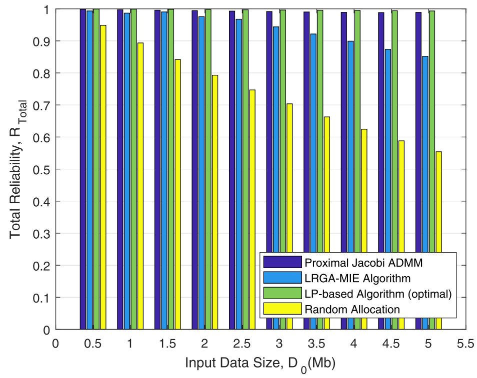
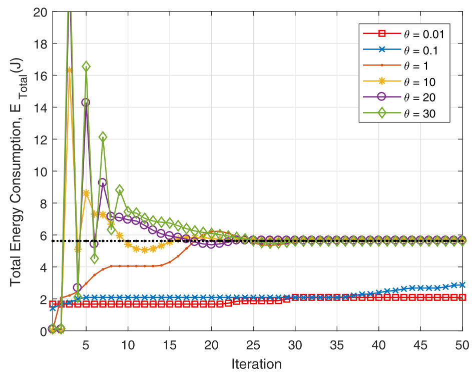
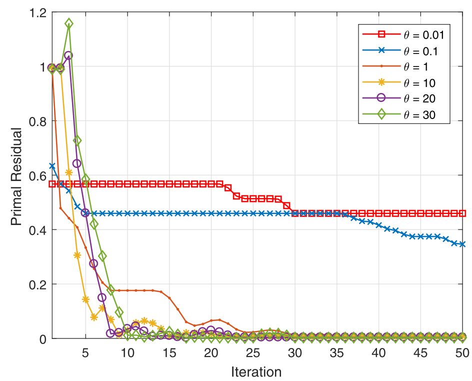


FIGURE 7. Reliability performance of different algorithms versus input data size.



(a) Energy consumption values with the number of iterations.



(b) Primal residual values with the number of iterations.

FIGURE 8. Convergence performance of Proximal Jacobi ADMM with different step size θ.

dr_0, which leads to high transmission latency, hence although it has powerful computing capability, but with the increasing of input data size, the transmission latency of cloud computing paradigm increases linearly. As for local computing approach, due to the drone dr_0 is with certain computing capability, it is manageable for dealing the tasks with small data size. However, when the data size of the tasks increase, the processing latency of local computing approach increases linearly due to the limited computing capability. But for fog computing manner, benefit by the huge computing capability of the whole drones swarm, without long transmission due to the close distance among drones, although the input data size increases, the fog computing based manner can handle the computing tasks in a relatively smaller latency.

3) RELIABILITY PERFORMANCE

In Fig. 6, we compared the reliability performance of different algorithms in different environments. In the best environment, the failure rates of fog nodes and links are set as 0, i.e., $\nu_i, \mu_i = 0$. In the good environment, the failure rates of fog nodes and links are assumed to be uniformly distributed, i.e., $\nu_i, \mu_i \sim \mathrm{U}([0, 0.03])$. In the poor environment, $\nu_i, \mu_i \sim \mathrm{U}([0.1, 0.5])$. In the mixed environment, some nodes and links have $\nu_i, \mu_i \sim \mathrm{U}([0, 0.03])$, and other nodes and links have $\nu_i, \mu_i \sim \mathrm{U}([0.1, 0.5])$. As we can see, in the best environment, the reliability performance of any algorithm including random allocation is 100 %, because there won't be any node or link failures in such a ideal environment. But in practice, such ideal environment is not existed, therefore, we studied the reliability performance in three experiments which are close to the real world, i.e., the good environment, the poor environment, and the mixed environment. We can observe, with any optimization, the reliability performance of random allocation scheme is intensely low, especially in poor environments, the reliability of random allocation scheme is only 57 %, which is unacceptable for a task with low latency and high reliability requirements. As comparison, the reliability performance of LRGA-MIE algorithm, LP-based algorithm, and Proximal Jacobi ADMM based algorithm are 78 %, 94 %, and 96 %, respectively, which is considerable in a environment with high failure rates, e.g., in the battlefield with complex electromagnetic disturbances.

Fig. 7 shows the reliability performance of algorithms versus the input data size. We can observe that the LP-based algorithm and Proximal Jacobi ADMM algorithm are with high adaptability to the data size. With the increasing of the input data size, the reliability of these two algorithms still maintain a high level more than 99 %. Although the reliability performance of LRGA-MIE algorithm is inferior to the LP-based and Proximal Jacobi ADMM algorithm, it still keeps the reliability more than 85 %. As a contrast, without optimization, the random task allocation scheme has the worst reliability performance. When the input data size is 5 Mb, the reliability of random task allocation scheme is only 55 %.

4) IMPACT OF PARAMETER SETTING ON THE PERFORMANCE OF ADMM

In Fig. 8, we explored the impact of step size on the convergence performance of Proximal Jacobi ADMM based algorithm. We can observe that the larger the step size, the faster the algorithm converges. When the step size is relatively smaller, e.g., $\theta = 0.01$ or 0.1, the algorithm cannot converge within 50 iterations. When the step size θ increase to 1, the algorithm converges rapidly. However, when the step size increases from 1 to a higher value, the convergence speed first increases gradually, and then reaches a saturation value. From then on, the convergence speed does not increase significantly with the increase of step size.

VI. CONCLUSION

In this paper, to enhance the capability of the drones swarm dealing with the latency and reliability sensitive tasks, the FCSD architecture is proposed, which can make up for the shortcomings of the cloud-based architecture. The drones which close to the task initiator drone are thought to be fog computing nodes to complete the task with the initiator drone cooperatively. Considering the limited battery capacity of drones, the task allocation optimization problem is constructed as an energy consumption minimization problem with the latency and reliability constraints. Furthermore, due to the stringent latency requirements of the latency and reliability sensitive tasks, a fast and efficient algorithm is intensely needed, therefore, benefit by the recent advances of distributed convex optimization, a distributed task allocation algorithm with low computation complexity is designed based on Proximal Jacobi ADMM. Finally, extensive simulations invalidate the effectiveness of the proposed scheme.

ACKNOWLEDGMENT

The author, Dr. Jingjing Wang, would like to acknowledge the financial support of the Shuimu Tsinghua Scholar Program. This article was presented in part at the X. Hou, *et al.*, "Fog based computation offloading for swarm of drones," in the IEEE International Conference on Communications (ICC), Shanghai, China, IEEE, 2019, pp. 1-7.

REFERENCES

[1] R. P. Otto, "Small unmanned aircraft systems (SUAS) flight plan: 2016–2036. Bridging the gap between tactical and strategic," Air Force Deputy Chief of Staff, Washington, DC, USA, Accession no. AD1013675, 2016.

[2] J. A. Shaffer, E. Carrillo, and H. Xu, "Hierarchal application of receding horizon synthesis and dynamic allocation for UAVs fighting fires," *IEEE Access*, vol. 6, pp. 78868–78880, 2018.

[3] J. Zhang, S. Chen, R. G. Maunder, R. Zhang, and L. Hanzo, "Adaptive coding and modulation for large-scale antenna array-based aeronautical communications in the presence of co-channel interference," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1343–1357, Feb. 2018.

[4] J. Wang, C. Jiang, Z. Wei, T. Bai, H. Zhang, and Y. Ren, "UAV aided network association in space-air-ground communication networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[5] J. Zhang, T. Chen, S. Zhong, J. Wang, W. Zhang, X. Zuo, R. G. Maunder, and L. Hanzo, "Aeronautical ad hoc networking for the Internet-above-the-clouds," *Proc. IEEE*, vol. 107, no. 5, pp. 868–911, May 2019.

[6] L. Huang, H. Qu, and L. Zuo, "Multi-type UAVs cooperative task allocation under resource constraints," *IEEE Access*, vol. 6, pp. 17841–17850, 2018.

[7] J. Zhang, S. Chen, R. G. Maunder, R. Zhang, and L. Hanzo, "Regularized zero-forcing precoding-aided adaptive coding and modulation for large-scale antenna array-based air-to-air communications," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2087–2103, Sep. 2018.

[8] J. Wang, C. Jiang, Z. Wei, C. Pan, H. Zhang, and Y. Ren, "Joint UAV hovering altitude and power control for space-air-ground IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1741–1753, Apr. 2019.

[9] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Jan. 2018.

[10] M.-A. Messous, A. Arfaoui, A. Alioua, and S.-M. Senouci, "A sequential game approach for computation-offloading in an UAV network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[11] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an UAV network in mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[12] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[14] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," 2019, *arXiv:1908.06849*. [Online]. Available: https://arxiv.org/abs/1908.06849

[15] X. Hou, Z. Ren, K. Yang, C. Chen, H. Zhang, and Y. Xiao, "IIoT-MEC: A novel mobile edge computing framework for 5G-enabled IIoT," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[16] S. Karaman, T. Shima, and E. Frazzoli, "A process algebra genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 489–503, Aug. 2012.

[17] X. Fu, J. Pan, X. Gao, B. Li, J. Chen, and K. Zhang, "Task allocation method for multi-UAV teams with limited communication bandwidth," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1874–1878.

[18] Z. Jia, J. Yu, X. Ai, X. Xu, and D. Yang, "Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm," *Aerosp. Sci. Technol.*, vol. 76, pp. 112–125, May 2018.

[19] X. Guangtong, L. Li, T. Long, Z. Wang, and M. Cai, "Cooperative multiple task assignment considering precedence constraints using multi-chromosome encoded genetic algorithm," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2018, p. 1859.

[20] M. Zhu, X. Du, X. Zhang, H. Luo, and G. Wang, "Multi-UAV rapid-assessment task-assignment problem in a post-earthquake scenario," *IEEE Access*, vol. 7, pp. 74542–74557, 2019.

[21] F. Liu, S. Liang, and X. Xian, "Multi-robot task allocation based on utility and distributed computing and centralized determination," in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, Qingdao, China, May 2015, pp. 3259–3264.

[22] X. Hu, H. Ma, Q. Ye, and H. Luo, "Hierarchical method of task assignment for multiple cooperating UAV teams," *J. Syst. Eng. Electron.*, vol. 26, no. 5, pp. 1000–1009, Oct. 2015.

[23] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[24] X. Hou, Z. Ren, W. Cheng, C. Chen, and H. Zhang, "Fog based computation offloading for swarm of drones," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

[25] T. Shuyan, Q. Zheng, and X. Jiankuan, "Collaborative task assignment scheme for multi-UAV based on cluster structure," in *Proc. 2nd Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, Nanjing, China, vol. 2, Aug. 2010, pp. 285–289.

[26] A. Kopeikin, S. S. Ponda, L. B. Johnson, and J. P. How, "Multi–UAV network control through dynamic task allocation: Ensuring data-rate and bit-error-rate support," in *Proc. IEEE Global Commun. Conf. Workshops (GLOBECOM Workshops)*, Anaheim, CA, USA, Dec. 2012, pp. 1579–1584.

[27] Z. Fu, Y. Mao, D. He, J. Yu, and G. Xie, "Secure multi-UAV collaborative task allocation," *IEEE Access*, vol. 7, pp. 35579–35587, 2019.

[28] Y. Cui, J. Ren, W. Du, and J. Dai, "UAV target tracking algorithm based on task allocation consensus," *J. Syst. Eng. Electron.*, vol. 27, no. 6, pp. 1207–1218, Dec. 2016.

[29] E. I. Grøtli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! and MILP," *J. Intell. Robot. Syst.*, vol. 65, nos. 1–4, pp. 265–282, 2012.

[30] M. Alighanbari, Y. Kuwata, and J. How, "Coordination and control of multiple UAVs with timing constraints and loitering," in *Proc. Amer. Control Conf.*, Denver, CO, USA, vol. 6, 2003, pp. 5311–5316.

[31] Y. Gao, Y. Zhang, S. Zhu, and Y. Sun, "Multi-UAV task allocation based on improved algorithm of multi-objective particle swarm optimization," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Zhengzhou, China, Oct. 2018, pp. 443–447.

[32] K. Zhao, "Multiple-agent task allocation algorithm utilizing ant colony optimization," *J. Netw.*, vol. 8, no. 11, pp. 2599–2606, Nov. 2013.

[33] J. J. Li, R. B. Zhang, and Y. Yang, "Meta-heuristic ant colony algorithm for multi-tasking assignment on collaborative AUVs," *Int. J. Grid Distrib. Comput.*, vol. 8, no. 3, pp. 135–144, Jun. 2015.

[34] Z. Junwei and Z. Jianjun, "Study on multi-UAV task clustering and task planning in cooperative reconnaissance," in *Proc. 6th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, Hangzhou, China, vol. 2, Aug. 2014, pp. 392–395.

[35] C. Jianping, Y. Yimin, and W. Yunbiao, "Multi-robot task allocation based on robotic utility value and genetic algorithm," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Shanghai, China, vol. 2, Nov. 2009, pp. 256–260.

[36] S. Josilo and G. Dan, "Decentralized algorithm for randomized task allocation in fog computing systems," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 85–97, Feb. 2019.

[37] J. Capitan, M. T. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned POMDPs," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 650–671, May 2013.

[38] W. Zhao, Q. Meng, and P. W. H. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 902–915, Apr. 2016.

[39] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[40] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[41] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Bejiing, China, Jun. 2016, pp. 247–251.

[42] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Trans. Ind. Informat.*, early access, doi: 10.1109/tii.2019.2954334.

[43] A. Majd, A. Ashraf, E. Troubitsyna, and M. Daneshtalab, "Integrating Learning, Optimization, and Prediction for Efficient Navigation of Swarms of Drones," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Mar. 2018, pp. 101–108.

[44] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[45] S. M. Shatz and J.-P. Wang, "Models and algorithms for reliability-oriented task-allocation in redundant distributed-computer systems," *IEEE Trans. Rel.*, vol. 38, no. 1, pp. 16–27, Apr. 1989.

[46] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. HotCloud*, vol. 10, 2010, p. 4.

[47] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Apr. 2017.

[48] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[49] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, nos. 1–2, pp. 57–79, 2016.

[50] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with O(1/k) convergence," *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, May 2017.

[51] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, no. 3, pp. 399–417, Jun. 1963.

[52] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

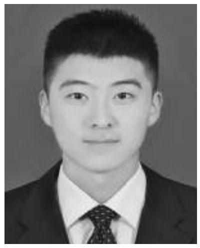
[53] B. He, L. Hou, and X. Yuan, "On full jacobian decomposition of the augmented lagrangian method for separable convex programming," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2274–2312, Jan. 2015.

[54] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, no. 3, pp. 889–916, Mar. 2016.

[55] Y. Li, G. Shi, W. Yin, L. Liu, and Z. Han, "A distributed ADMM approach with decomposition-coordination for mobile data offloading," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2514–2530, Mar. 2018.

[56] H. Zhu, Y. He, X. Wang, and E. C. Tsang, "Discrete differential evolutions for the discounted 0–1 knapsack problem," *Int. J. Bio-Inspired Comput.*, vol. 10, no. 4, p. 219, 2017.

[57] Y. Qi, H. Wang, L. Zhang, and B. Wang, "Optimal access mode selection and resource allocation for cellular-VANET heterogeneous networks," *IET Commun.*, vol. 11, no. 13, pp. 2012–2019, Sep. 2017.



XIANGWANG HOU (Student Member, IEEE) received the B.S. degree (Hons.) in electronic information engineering from the Shandong University of Technology, Shandong, China, in 2017. He is currently pursuing the M.S. degree with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. His current research interests include edge/fog computing, Industrial Internet of Things, and vehicular networks. He has served as a TPC Member of the IEEE ICC 2019 Workshop, and also as a Reviewer of the IEEE ICC, the IEEE WCNC, the IEEE ICCC, JCIN, and IEEE ACCESS.

ZHIYUAN REN (Member, IEEE) received the M.S. and Ph.D. degrees in communication and information system from Xidian University, in 2007 and 2011, respectively. He is currently an Associate Professor with the School of Telecommunication Engineering, Xidian University. He has published more than 20 journal publications and conference publications. His major research interests are distributed computing, mobile edge computing, and network virtualization.

JINGJING WANG (Member, IEEE) received the B.S. degree (Hons.) in electronic information engineering from the Dalian University of Technology, Liaoning, China, in 2014, and the Ph.D. degree (Hons.) in information and communication engineering from Tsinghua University, Beijing, China, in 2019. From 2017 to 2018, he visited the Next Generation Wireless Group, University of Southampton, U.K., chaired by Prof. Lajos Hanzo. He is currently a Postdoctoral Researcher at the Department of Electronic Engineering, Tsinghua University. His research interests include resource allocation and network association, learning theory-aided modeling, analysis and signal processing, as well as information diffusion theory for mobile wireless networks. He received the China Postgraduate National Scholarship Award, in 2017, the Best Journal Paper Award from the IEEE Technical Committee on Green Communications and Computing, in 2018, the Beijing Distinguished Graduated Student Award, the Tsinghua Outstanding Distinguished Doctoral Dissertation, and the Best Paper Award from the IEEE ICC, in 2019.

SHUYA ZHENG received the B.S. degree (Hons.) in electronic information engineering from the Shandong University of Technology, Shandong, China, in 2017. She is currently pursuing the M.S. degree with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. Her current research interests include edge/fog computing and the Industrial Internet of Things.

WENCHI CHENG (Senior Member, IEEE) received the B.S. and Ph.D. degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2008 and 2013, respectively. He was a Visiting Scholar with the Networking and Information Systems Laboratory, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA, from 2010 to 2011. He is currently a Professor with Xidian University. His current research interests include 5G wireless networks and orbital-angular-momentum-based wireless communications. He has published more than 80 international journal and conference papers in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE Magazines, the IEEE INFOCOM, GLOBECOM, and ICC. He received the URSI Young Scientist Award (2019), the Young Elite Scientist Award of CAST (2016–2018), the Best Dissertation (Rank 1) of China Institute of Communications, the Best Paper Award for the IEEE ICCC 2018, the Best Paper Nomination for the IEEE GLOBECOM 2014, and the Outstanding Contribution Award for Xidian University. He has served or serving as the IoT Session Chair for the IEEE 5G Roadmap, the Wireless Communications Symposium Co-Chair for the IEEE GLOBECOM 2020, the Publicity Chair for the IEEE ICC 2019, the Workshop Chair for the IEEE INFOCOM 2020 IWECN, the Next Generation Networks Symposium Chair for the IEEE ICCC 2019, the Workshop Chair for the IEEE ICC 2019 Workshop on Intelligent Wireless Emergency Communications Networks, the Workshop Chair for the IEEE ICCC 2017 Workshop on Internet of Things, and the Workshop Chair for the IEEE GLOBECOM 2019 Workshop on Intelligent Wireless Emergency Communications Networks. He has served or serving as an Associate Editor for the IEEE COMMUNICATIONS LETTERS and IEEE ACCESS.

HAILIN ZHANG (Member, IEEE) received the B.S. and M.S. degrees from Northwestern Polytechnic University, Xi'an, China, in 1985 and 1988, respectively, and the Ph.D. degree from Xidian University, Xi'an, in 1991. In 1991, he joined the School of Telecommunications Engineering, Xidian University, where he is currently a Senior Professor and the Dean of the school. He is also the Director of the Key Laboratory in Wireless Communications, sponsored by the China Ministry of Information Technology, a Key Member of the State Key Laboratory of Integrated Services Networks, one of the state government specially compensated scientists and engineers, a Field Leader in Telecommunications and Information Systems at Xidian University, and the Associate Director of the National 111 Project. He has published more than 150 articles in journals and conferences. His current research interests include key transmission technologies and standards on broadband wireless communications for 5G and 5G-beyond wireless access systems.