Abstract—Due to the limited computing resources of swarm of drones, it is difficult to handle computation-intensive tasks locally, hence the cloud based computation offloading is widely adopted. However, for the business which requires low latency and high reliability, the cloud-based solution is not suitable, because of the slow response time caused by long distance data transmission. Therefore, to solve the problem mentioned above, in this paper, we introduce fog computing into swarm of drones (FCSD). Focusing on the latency and reliability sensitive business scenarios, the latency and reliability is constructed as the constraints of the optimization problem. And in order to enhance the practicality of the FCSD system, we formulate the energy consumption of FCSD as the optimization target function, to decrease the energy consumption as far as possible, under the premise of satisfying the latency and reliability requirements of the task. Furthermore, a heuristic algorithm based on genetic algorithm is designed to perform optimal task allocation in FCSD system. The simulation results validate that the proposed fog based computation offloading with the heuristic algorithm can complete the computing task effectively with the minimal energy consumption under the requirements of latency and reliability.

Index Terms—Swarm of drones, fog computing, computation offloading, latency and reliability, genetic algorithm

I. INTRODUCTION

Swarm of drones, which are considered as an intensely promising development direction of Unmanned Aerial Vehicles (UAVs), has made great progress in recent years. Swarm of drones are composed by numerous small and low-cost UAVs, through collaborating with each other, the drones can show strong ability to accomplish the tasks which are difficult for a single large UAV. As a consequence, swarm of drones are widely used for a variety of applications, such as agriculture, smart city, search and rescue, remote sensing, military, etc [1]. For most of these applications, drones are brought to deal with computation-intensive tasks, such as path planning, pattern recognition, etc [2]. However, due to its limited resources (e.g., battery power, computing capability), the single drone is too difficult to handle the complicated task locally [3]. Therefore, to address the computation-intensive tasks mentioned above, some researches considered computation offloading to a cloud server, and then obtain the result from the cloud [4]. In this manner, the capability of swarm of drones is greatly enhanced in a virtual way. And it is suitable for some business (e.g., topographic mapping, resource exploration, environmental monitoring, etc.) which are not sensitive to latency and reliability. But in practice, quite a few computing tasks which the drones need to process, have low latency and high reliability requirements, such as military object recognition, disaster rescue, emergency obstacle avoidance, etc [5] [6]. However, cloud servers are generally located far away from the drones, and long-distance data transmission will lead to high latency, even in some harsh environments, there is no working wireless infrastructure to connect the drones and cloud. Hence, the cloud based computation offloading is not suitable to address the latency and reliability sensitive business.

In order to further enhance the ability of swarm of drones to cope with computation-intensive tasks, specifically focusing on those tasks with low latency and high reliability requirements, we introduce fog computing [7] into swarm of drones. The drones which close to the initiator drone are thought to be fog computing nodes to complete the computing task collaboratively.

Fog computing is a novel computing paradigm which is not intended to replace cloud computing but to compliment it. Recently, there are many researches about fog computing enhancing cloud computing. In [8], authors considered leveraging buses as fog computing servers to provide fog computing services for the mobile users on bus and share the pressure of roadside cloudlets. The authors in [9] proposed combined fog-cloud architecture to reduce the latency of service. In [10], authors formulated a computation offloading game to improve the quality of experience of IoT users in hierarchical fog-cloud computing architecture. But there is no existing research to introduce fog computing into the work of swarm of drones, as a supplementary solution for the computation offloading of cloud computing.

In practice, swarm of drones usually work in harsh environments, and inevitable disturbances (e.g., hardware damage, software breakdown, communication link failure, etc.) will lead to the failure of the task. Hence, besides considering the latency guarantee, a proper reliability-guarantee mechanism is especially needed. However, there are few existing researches about fog (or edge) based computation offloading both considering the latency and reliability guarantee [11]. Therefore, in order to complete the computing task within low latency and high reliability requirements, we construct a
certain mathematical model for system’s latency and reliability during task execution, and take the latency and reliability as the constraints of the optimization problems formulated, thus the computation offloading scheme must be able to meet the requirements of the business on latency and reliability.

A big challenge for swarm of drones utilizing fog computing to deal with computing task is their limitation of battery endurance, hence, under the premise of ensuring the completion of the task within low latency and high reliability requirements, we formulated the energy consumption of whole swarm of drones as the target function of optimization problem, to reduce the overall energy consumption and extend the working time of the swarm of drones as far as possible. Since the formulated problem is NP-hard, we design a heuristic algorithm based on genetic algorithm to solve the problem.

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

- To enhance the ability of swarm of drones to handle complicated tasks, we introduce fog computing into swarm of drones (FCSD), as a supplementary solution for the cloud based computation offloading.
- To meet the latency and reliability requirements of the computing tasks, we construct a certain mathematical model for system's latency and reliability during task execution.
- To improve the practicality of the FCSD system in practice, the energy consumption of FCSD is constructed as the optimization target function, to reduce the energy consumption so that extend the working time of swarm of drones.
- To solve the NP-hard problem formulated, we design a latency and reliability constrained minimum energy consumption algorithm based on genetic algorithm (LRGA-MIE).

The rest of the paper is organized as follows. The system model and problem formulation are presented in Section II. Section III presents the offloading algorithm proposed. The simulation results and analyses are given in Section IV. Finally, Section V conclude the paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In order to enhance the ability of swarm of drones addressing computation-intensive tasks which are sensitive to latency and reliability, the FCSD system is proposed. 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 define \( E_0 \) as the total required amount CPU cycles to complete the task \( \psi_0 \). 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 \([12]\). The drone \( dr_0 \) requests nearby drones \( dr_i \) that can serve as the fog nodes to complete the task \( \psi_0 \) collaboratively. These drones available nearby, denoted by a set \( D = \{ dr_1, dr_2, \ldots, dr_p \} \), are equipped with storage and computation resources. We define \( f_0 \) as the CPU frequency of the drone \( dr_0 \). Similarly, the CPU frequency of the drones available nearby, denoted by a set \( F = \{ f_1, f_2, \ldots, f_p \} \). The coordinate of the drone \( dr_0 \) is \( (x_0, y_0, z_0) \). The \( C = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_p, y_p, z_p)\} \) are the three-dimensional coordinates of the drones available nearby, respectively.

The distance between the drone \( dr_i \) in \( D \) and the drone \( dr_0 \) can be given by

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

where \( r_i \) is the maximum communication radius of individual drones. According to [8] [13], the uplink rate from \( dr_0 \) to \( dr_i \) can be given as

\[
R_{UL}(0,i) = W_{UL} \log_2 \left( 1 + \frac{P_{Tx} (g_{0,i} - |h_{0,i}|)}{N_0} \right), \tag{2}
\]

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

The task \( \psi_0 \) would be partitioned into several subtasks by the drone \( dr_0 \) and distributed to multiple drones. In practice, how to partition a task depends on not only application, but also the requirements, which is worth studying further. Therefore, for simplicity, it can be assumed that the task can be divided into any proportion with arbitrary precision, and there is no overlap between any two subtasks. According to the status of the \( dr_0 \) and \( dr_i \) in \( D \), the initiator drone \( dr_0 \) can make different task offloading and allocation decisions.\( ^{\dagger} \) We define \( \rho \) \( (0 \leq \rho \leq 1) \) as the offloading coefficient, therefore, the part of the task \( \psi_0 \) which need to be executed locally, 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

\(^{\dagger}\)In practice, the drone interacts with the surrounding drones to complete the formation, networking, collaborative works, etc., which enables the drone to be aware of the status of the surrounding drones.
\( \lambda_i (1 - \rho) \Psi_0 \), where \( \lambda_i \in [0, 1] \), and \( \sum_{i=1}^{p} \lambda_i = 1 \). We define 
\[ \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_p]^T \] as the task allocation vector.

After decision, the drone \( dr_0 \) and the drones \( dr_i \in D \) are orchestrated to perform distributed computing to complete the task \( \Psi_0 \) collaboratively. These low-cost drones fly slowly and tend to form a relatively stable formation, rather than constantly changing \( \Psi_0 \). In the meantime, the transmission latency of the subtask is extremely short when the data size is small. Therefore, we assume that the relative distance and the states of the drones will not change during the task assignment process, and each assigned subtask will be executed immediately on the drone \( dr_i \) and \( dr_0 \).

### A. Latency Model

The latency of the drone \( dr_0 \) processing the subtask \( \rho \Psi_0 \) is defined as
\[ T_{Local} = \frac{\rho a_0 D_0}{f_0}. \] (3)

When the drone \( dr_0 \) offloading 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 \( \beta (\beta \geq 1) \) represents a ratio of the transmitted data size to the original task data size due to transmission overhead \( [12] \). Thus, the transmission latency of the subtask from the drone \( dr_0 \) to the drone \( dr_i \) is
\[ T_i^{UL} = \frac{\beta \lambda_i (1 - \rho) D_0}{R^{UL}(0, i)}. \] (4)

And the computation latency of the subtask addressed on the drone \( dr_i \) is
\[ T_i^{Comp} = \frac{\alpha_0 \lambda_i (1 - \rho) D_0}{f_i}. \] (5)

Due to the data size of the result of each subtask is much smaller than the input one, the latency caused by downlink transmission can be neglected \( [15] \). The total execution latency of the subtask completed on the drone \( dr_i \) is given by
\[ T_i = T_i^{UL} + T_i^{Comp} = \frac{\beta \lambda_i (1 - \rho) D_0}{R^{UL}(0, i)} + \frac{\alpha_0 \lambda_i (1 - \rho) D_0}{f_i}. \] (6)

Therefore, the total execution latency of the task \( \Psi_0 \) can be described as
\[ T_{Total} = \max_{i \in [p]} \{ T_{Local}, T_i \} = \max_{i \in [p]} \left( \frac{\rho a_0 D_0}{f_0}, \frac{\beta \lambda_i (1 - \rho) D_0}{R^{UL}(0, i)} + \frac{\alpha_0 \lambda_i (1 - \rho) D_0}{f_i} \right). \] (7)

To meet the latency requirement of the task \( \Psi_0 \), the total execution latency \( T_{Total} \) should meet the constraint \( T_{Total} \leq T_0 \).

### B. Reliability Model

The swarm of drones usually work in harsh environments, and inevitable disturbances (e.g., hardware damage, software breakdown, communication link failure, etc.) will lead to the failure of the whole task, and always with serious consequence. Therefore, a proper reliability-guarantee capability is especially needed to ensure the successful completion of the mission.

According to the widely accepted reliability model proposed by Shatz \( [16] \), 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 failure of the drones and communication links follow a Poisson process \( [16] \), further, the failure rates of the drone \( dr_0 \) and \( dr_i \) are defined as \( \nu_0 \) and \( \nu_i \), respectively, and the failure rate of the communication links between \( dr_0 \) and \( dr_i \) is defined as \( \lambda_{0,i} \). Therefore, the computation reliability of the drone \( dr_0 \) and \( dr_i \) can be represented as \( e^{-\nu_i \frac{\alpha_0 D_0}{f_i}} \) and \( e^{-\nu_i \lambda_i (1 - \rho) D_0} \), respectively. And the communication reliability between \( dr_0 \) and \( dr_i \) can be represented as \( e^{-\lambda_{0,i} \frac{D_0}{m_{UL}(0,i)}} \). The reliability of the subtask which executed locally can be represented as
\[ R_{Local} = e^{-\nu_i \frac{\alpha_0 D_0}{f_i}}. \] (8)

Then, the reliability of the subtask which distributed to the drone \( dr_i \) can be represented as
\[ R_i = e^{-\nu_i \lambda_i (1 - \rho) D_0} - \mu_{0,i} \lambda_{0,i} \frac{D_0}{m_{UL}(0,i)} \cdot \] (9)

Therefore, the reliability of the swarm of drones during the execution time of the task \( \Psi_0 \) can be given by
\[ R_{Total} = R_{Local} \prod_{i=1}^{p} R_i = e^{-\nu_0 \frac{\alpha_0 D_0}{f_0} + \sum_{i=1}^{p} \left( -\nu_i \frac{\lambda_i (1 - \rho) D_0}{f_i} - \mu_{0,i} \lambda_{0,i} \frac{D_0}{m_{UL}(0,i)} \right)} \cdot \] (10)

To meet the reliability requirement of the task \( \Psi_0 \), the total reliability \( R_{Total} \) should meet the constraint \( R_{Total} \geq R_0 \).

### C. Energy Consumption Model

To improve the practicality of the FCSD system, how to minimize the energy consumption, under the premise of ensuring the completion of the task within latency and reliability requirements, must be taken into account. Therefore, a mathematical model that minimizes the energy consumption of FCSD processing a single task is constructed.

1) **Computational energy consumption:** The computational energy consumption of the drone \( dr_0 \) and \( dr_i \) can be given by
\[ E_{Local}^{Comp} = k_f^{\sigma} \; T_{Local}; \] (11)
\[ E_{i}^{Comp} = k_f^{\sigma} \; T_{i}^{Comp}; \] (12)
respectively, where \( k_f^{\sigma} \) and \( k_f^{\sigma} \) are the computation power of the drone \( dr_0 \) and \( dr_i \). According to \( [17] \), the \( k > 0 \) and the \( \sigma \geq 2 \) (which usually close to 3), are the positive constant. As in \( [18] \), the \( k \) and the \( \sigma \) can be set as \( 1.25 \times 10^{-26} \) and 3, respectively.

Therefore, the total computational energy consumption of the swarm of drones is represented as
\[ E_{Total}^{Comp} = k_f^{\sigma} \; T_{Local} + \sum_{i=1}^{p} k_f^{\sigma} T_{i}^{Comp}; \] (13)
\[ = k_f^{\sigma} \frac{\rho a_0 D_0}{f_0} + \sum_{i=1}^{p} k_f^{\sigma} \frac{\alpha_0 \lambda_i (1 - \rho) D_0}{f_i}. \]
2) Transmission energy consumption: The transmission energy consumption of the drone $dr_0$ and the drone $dr_i$ can be given as

$$E_{\text{Trans}}^{\text{Local}} = P_{\text{Tx}} T_{\text{UL}};$$

$$E_{\text{Trans}}^{i} = P_{\text{Rx}} T_{\text{UL}},$$

respectively, where $P_{\text{Tx}}$ and $P_{\text{Rx}}$ denote the transmitting and receiving power of the drone $dr_0$ and $dr_i$, respectively, which are regarded as constant [18]. Therefore, the total transmission energy consumption of the FCSD system can be given by

$$E_{\text{Trans}} = \sum_{i=1}^{p} E_{\text{Trans}}^{i} + \sum_{i=1}^{p} E_{\text{Trans}}^{\text{Local}} = \sum_{i=1}^{p} P_{\text{Tx}} \beta \lambda_i (1 - \rho) D_0 / f_{0} + \sum_{i=1}^{p} P_{\text{Rx}} \beta \lambda_i (1 - \rho) D_0 / f_{1}$$

$$\sum_{i=1}^{p} P_{\text{Tx}} \beta \lambda_i (1 - \rho) D_0 / R^{\text{UL}}(0, i) + \sum_{i=1}^{p} P_{\text{Rx}} \beta \lambda_i (1 - \rho) D_0 / R^{\text{UL}}(0, i).$$

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

$$E_{\text{Total}} = E_{\text{Comp}} + E_{\text{Trans}} = k f_0 \beta \rho \alpha_0 \lambda_i (1 - \rho) D_0 / f_0 + \sum_{i=1}^{p} k f_i \alpha_0 \lambda_i (1 - \rho) D_0 / f_i$$

$$\sum_{i=1}^{p} P_{\text{Tx}} \beta \lambda_i (1 - \rho) D_0 / R^{\text{UL}}(0, i) + \sum_{i=1}^{p} P_{\text{Rx}} \beta \lambda_i (1 - \rho) D_0 / R^{\text{UL}}(0, i).$$

D. Problem Formulation

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

$$P : (\rho, \lambda_i) = \text{arg min} E_{\text{Total}}$$

$$\text{s.t.} \quad T_{\text{Total}} \leq T_0 \quad (19a)$$

$$R_{\text{Total}} \geq R_0 \quad (19c)$$

$$0 \leq \lambda_i, \rho$$

III. LRGA-MIE ALGORITHM

To find the optimal solution of the problem $P$, we design a latency and reliability constrained minimum energy consumption algorithm based on the real-code genetic algorithm (LRGA-MIE) [19].

Genetic algorithm (GA) is a kind of widely used heuristic algorithm due to its advantages of better global searching capability, strong robustness, parallel processing capability, etc. In the real-coded GA, each individual $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ip+1}\}$ in the population represents a possible solution of the optimization problem, which would be initially set to a random value. And then, through the constant evolution of selecting, crossing over and mutating the initial population, an optimal individual is found. Different from the unconstrained optimization problem, the problem $P$ formulated has several constraints including equality and inequality constraints (i.e., Eq. (19a), (19b), (19c) and (19d)). However, GA cannot solve constrained optimization problem directly.

Therefore, we adopt exterior penalty function method [20] to transform the constrained problem into an unconstrained optimization problem.

In the following, the design details of LRGA-MIE algorithm are explained.

The fitness function of LRGA-MIE is reconstructed as follows:

$$f(X) = \begin{cases} E_{\text{Total}}(X) & X \in F; \\ E_{\text{Total}}(X) + h(g) \sum_{j=1}^{p+4} E_j(X) + \xi(X, g) & X \in S - F, \end{cases}$$

where $F$ is the feasible region in the search space $S$, and $S - F$ denotes the infeasible region. $h(g)$ represents the penalty factor, which is a large number and usually taken a strictly increasing positive sequence that tends to infinity as the number of iterations increases. $E_j(X)$ is the constraint violation value of the infeasible individuals for the $j$th constraint, and $\xi(X, g)$ indicates an additional heuristic value for infeasible individuals in the $g$th generation. $E_j(X)$ and $\xi(X, g)$ can be expressed as

$$E_j(X) = \begin{cases} \max(0, -X(j)) & 1 \leq j \leq p + 1; \\ |X(1) + \sum_{i=2}^{p+4} X(i)(1 - X(1)) - 1| & j = p + 2; \\ \max(0, T_{\text{Total}} - T_0) & j = p + 3; \\ \max(0, R_0 - R_{\text{Total}}) & j = p + 4, \end{cases}$$

$$\xi(X, g) = W_{\text{or}}(g) - \min_{X \in S - F} \left\{ E_{\text{Total}}(X) + h(g) \sum_{j=1}^{p+4} E_j(X) \right\}.$$
Crossover: Crossover is to passed the original good genes onto the offspring. Where two new children individuals (i.e., \( X'_1, X'_2 \)) are generated by a linear combination of the two parent individuals (i.e., \( X_1, X_2 \)). The relationship between offspring and parents can be described as
\[
\begin{align*}
X'_1 &= \delta X_1 + (1 - \delta) X_2; \\
X'_2 &= \delta X_2 + (1 - \delta) X_1,
\end{align*}
\] (24)
respectively, where \( \delta \) is a random number on interval \((0, 1)\).

Mutation: Mutation operation determines the local search capability of the LRGA-MIE and improves the diversity of individuals in the population. In this paper, the non-uniform mutation operator is applied. When the individual \( X_i = \{x_{i1}, x_{i2}, \ldots, x_{il}, \ldots, x_{ip+1}\} \) mutates into the new individual \( X'_i = \{x'_{i1}, x'_{i2}, \ldots, x'_{il}, \ldots, x'_{ip+1}\} \), the new gene \( x'_{il} \) can be calculated as
\[
x'_{il} = \begin{cases} 
  x_{il} + (1 - \delta_x) (1 - q^{(1 - q/G)b}) & \text{if } \text{random}(0, 1) = 0, \\
  x_{il} + \delta_x (1 - q^{(1 - q/G)b}) & \text{if } \text{random}(0, 1) = 1.
\end{cases}
\] (25)

Where \( q \) is a random number in the range of \([0, 1]\) with uniform distribution. \( g \) is the current evolution generation, and \( G \) represents the maximum evolution generation. \( b \) is a system parameter, which determines the degree of the random number perturbations on the evolution generation of \( g \). The value of \( b \) is in the range of \([2, 5]\). \( \text{random}(0, 1) \) denotes any value of 0 or 1 with equal probability.

To minimize the energy consumption in Eq. (17), the basic steps of LRGA-MIE are shown in Algorithm 1.

### Algorithm 1 LRGA-MIE algorithm

**Require:** \( p, \mu_0, \nu_1, \nu_0, D, F, \Psi_0, G, S \)

**pc:** Crossover probability, \( pm: \) Mutation probability.

**Ensure:** BestFitness, BestSolution

1. Randomly initialize each individual \( X_i \) in the Population
2. globalBestFitness = 0
3. for Generation 1 to \( G \) do
4. localBestFitness = 0
5. for each individual \( X_i \) \( \in \) Population do
6. Calculate the value of \( E_i(X) \) using equation (23)
7. end for
8. Calculate the the value of \( \xi(X, g) \) and \( WOR(g) \) using equation (22) and (23)
9. for each individual \( X_i \) \( \in \) Population do
10. Calculate the fitness value \( f(X_i) \) using equation (23)
11. if \( f(X_i) < \) localBestFitness then
12. localBestFitness = \( f(X_i) \)
13. localBestSolution = \( X_i \)
14. end if
15. end for
16. if localBestFitness < globalBestFitness then
17. globalBestFitness = localBestFitness
18. BestSolution = localBestSolution
19. end if
20. Select individuals from the Population;
21. if rand < \( \Psi_0 \) then
22. crossopop = crossover(Population, \( pc \))
23. end if
24. if rand < \( pm \) then
25. mutatetop = mutate(crossopop, \( pm \))
26. end if
27. Update the Population: Population = mutatetop
28. end for
29. return BestSolution, globalBestFitness

### IV. PERFORMANCE AND EVALUATION

In this section, to testify the performance of the FSCD system with LRGA-MIE algorithm, a set of simulation results are presented. Referring to [8, 18, 23–27], the system parameters of FSCD are summarized in TABLE 1.

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| \( N_h \) | \(-100 \text{dBm}\) | \( E_r \) | \(0.8\) |
| \( N_f \) | \(-3 \text{dBm}\) randomly in \([0.001, 0.3]\) | \( r \) | \(200 \text{MB} \) |
| \( P_f \) | \(1.25 \text{W}\) | \( s \) | \(2500 \text{MHz}\) |
| \( P_{ef} \) | \(1.18 \text{W}\) | \( x \) | \(2 \times 10^{-5}\) |
| \( S \) | \(100 \text{Mbps}\) | \( y \) | \(0.0001\) |
| \( x' \) | \(1.258 \text{W}\) | \( z \) | \(1.258 \text{W}\) |

The parameters of the algorithm are set as follows: The maximum number of iterations is 300. The population size (i.e., \( G \)) is 100. The crossover and mutation probability (i.e., \( pc \) and \( pm \)) are set as 0.8 and 0.1, respectively. And the value of \( WOR(0) \) is set as \( 10^3 \).

In the following simulations, the parameters of the task \( \Psi_0 \) are set as follows, unless otherwise specified or used as variables. \( D_0 \) is set as 1 MB. \( \alpha_0 \) is set as 1900/8 to represent a computational intensive task [17]. \( T_0 \) and \( R_0 \) are set as 0.8 s and 0.99, respectively. And we assumed that there are a total of 10 drones available nearby that can serve as fog computing nodes to help \( d_{rg} \) achieved the computing task \( \Psi_0 \), i.e., \( p = 10 \). The numerical results in this section are based on an average value over 3000 Monte Carlo simulations.

### A. Latency Performance Comparison of Three Computation Architectures

Fig. 2 shows the latency performance comparison of three computation architectures, i.e., cloud computing, local computing and fog computing. With the increasing of the input data size, the cloud computing curve is intensely higher than the local computing curve and fog computing curve. The reason is that the transmission latency of cloud computing architecture is growing linearly with the increasing of the input data size, due to the long distance data transmission and limited bandwidth between cloud and the drone \( d_{rg} \).

Furthermore, the local computing curve is relatively lower than the cloud computing curve but relatively higher than the fog computing curve. This is because that the drone \( d_{rg} \) has a certain amount of computing ability. When the computing task is relatively smaller, it is able to handle the task locally in a relatively low computation latency and without transmission...
latency. But, when the input data size increased, limited by its computing ability, the local computing manner cannot complete the computing task with lower latency. The fog computing latency, as we can see, is always lower than other two kinds of computing manners, this is because that the transmission latency of fog computing is relatively lower due to the surrounding fog nodes are intensely close, and meantime the computation latency is relatively lower as well, because of it integrates the computational ability of numerous fog nodes. When the input data size of the task $\Psi_0$ is 0.5 MB, we can observe that the latency performance of fog computing improved by 93.12% and 85.42% compared with the cloud computing and local computing respectively. Therefore, the fog computing based computation offloading is suitable for latency sensitive business of swarm of drones.

**B. Reliability Performance Comparison of Different Algorithms**

In this section, we analyze the high efficiency of the LRGA-MIE algorithm in improving reliability in FCSD by comparing it with the Max-Min [28], Weighted Round Robin (WRR) [29] and Min-Min [30] algorithm, and also the random task assignment. The simulation results are shown in Fig. 3. As we can see, when the input data size is relatively smaller, the Max-Min, WRR, Min-Min and LRGA-MIE algorithms all have good performance, compared to the random task assignment. Specifically, when the input data size is 0.1 MB, the reliability of these algorithms are all higher than 0.95. However, with the increasing of the input data size, the reliability performance of the Max-Min, WRR, Min-Min, LRGA-MIE algorithms decrease in different degree. This is because the increase of input data size will increase the processing latency of some fog nodes and even the entire FCSD system, according to Eq. (7). Furthermore, no matter which fog nodes increase in transmission latency or computation latency, the total reliability of the FCSD system will be decreased, according to Eq. (7). But for LRGA-MIE algorithm, as we can see from the Fig. 3 it has always maintains intensely higher reliability. As the input data size increase from 0.2 MB to 0.6 MB, the reliability of LRGA-MIE only decrease from 0.9997 to 0.9989. And when the input data size is 0.6 MB, we can observe that the reliability of LRGA-MIE is higher than that of Max-Min, WRR, Min-Min algorithm and random task allocation strategy by 0.4674, 0.529, 0.5041 and 0.683, respectively. Therefore, the LRGA-MIE algorithm is better suited for optimizing reliability with global consideration of computation capability, transmission capability and failure rate.

**C. Energy Consumption Performance Impact by Latency Constraint and Reliability Constraint**

In this section, we discuss the impact of latency and reliability constraints on total energy consumption of FCSD. The simulation results are shown in Fig. 4. As we can see, when the latency constraint is fixed, the total energy consumption will gradually decrease with the reducing of reliability constraint. Similarly, when the reliability constraint is fixed, the total energy consumption will gradually decrease with the improving of latency constraint. This is because whether the reliability constraint is reduced or the latency constraint is increased, the feasible domain of the optimization problem is extended, and thus more solutions with low energy consumption performance can be obtained. However, we can observe that when the reliability is fixed, e.g., $R_0 = 0.99$, as the latency constraint increases gradually, the energy consumption curve will drop rapidly first, i.e., $T_0$ ranges from 2 s to 2.5 s, then the rate of decline will slow down and finally the curve will be tend to be gentle, i.e., $T_0$ ranges from 2.5 s to 2.9 s. It is because that, in this reliability constraint, the computing task $\Psi_0$ can be completed within 2.5 s. When the latency constraint is greater than 2.5 s, the latency constraint will not have a significant impact on the energy consumption performance of FCSD.

**D. Energy Consumption Performance Comparison of Different Algorithms**

In this section, we analyze the energy consumption performance of different algorithms in FCSD. The simulation results are shown in Fig. 5. We can observe that the random
task assignment has the worst energy consumption performance. When the input data size is relative smaller, although not as good as LRGA-MIE algorithm, the performance of the WRR, Max-Min and Min-Min algorithms are not bad, compared to the random task assignment. However, with the increasing of the input data size, the energy consumption of the Max-Min, Min-Min and WRR algorithms and random task assignment increase rapidly. As we can see, the energy consumption of Max-Min algorithm approaches that of the WRR algorithm, and the Min-Min algorithm is slightly better than the Max-Min and WRR algorithms. But the LRGA-MIE algorithm maintains good performance all along, due to its strong global search ability. It shows that under the premise of ensuring low latency and high reliability requirements of the computing task, the LRGA-MIE algorithm has intensely than the Max-Min and WRR algorithms. But the LRGA-MIE algorithm maintains good performance in reducing the energy consumption in FCSD, and it has strong adaptability and stability to the growth of the input data size. When the input data size is 0.5 MB, the energy consumption performance of the LRGA-MIE algorithm improved by 44.87%, 41.16%, 41.12% and 34.77%, compared with the random task assignment, Max-Min, WRR and Min-Min algorithm, respectively.

V. CONCLUSION

In this paper, to solve the problem that the cloud based computation offloading is not suitable for addressing the latency and reliability sensitive task, we introduced the fog computing based computation offloading into swarm of drones (FCSD). And specifically focusing on the latency and reliability business scenarios and improving the practicality of FCSD, an optimization problem to minimize the energy consumption of FCSD within latency and reliability constraints is constructed. In order to solve the NP-hard problem we formulated, the LRGA-MIE algorithm was proposed. The simulation results demonstrated that the LRGA-MIE can minimize the energy consumption of FCSD, on the basis of completing the computing task within latency and reliability requirements. In future research work, two things are on our agenda. One is to reduce the algorithm complexity of LRGA-MIE to further improve its practicability. The other is to utilize some new technologies (e.g., cognitive radio [31], orbital angular momentum [32], etc.) to solve the problem of spectrum resource shortage when the number of UAV increases rapidly.

REFERENCES

[1] A. Majd, A. Ashraf, E. Troubtysnya, and M. Daneshtablub, "Integrating learning, optimization, and prediction for efficient navigation of swarms of drones," in Parallel, Distributed and Network-based Processing (PDP), 2018 26th Euromicro International Conference. IEEE, 2018, pp. 101–108.
[2] M.-A. Messous, A. Arfaoui, A. Alioua, and S.-M. Senouci, "A sequential game approach for computation-offloading in an uav network," in GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE, 2017, pp. 1–7.
[3] M.-A. Messous, H. Sedd妖maci, N. Houari, and S.-M. Senouci, "Computing offloading game for an uav network in mobile edge computing," in Communications (ICC), 2017 IEEE International Conference. IEEE, 2017, pp. 1–6.
[4] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer, vol. 43, no. 4, pp. 51–56, 2010.
[5] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, “A data dissemination scheme based on clustering and probabilistic broadcasting in vanets,” Vehicular Communications, vol. 13, pp. 78–88, 2018.
[6] J. Guo, B. Song, Y. Chi, L. Jayasinghe, C. Yuen, Y. L. Guan, X. Du, and M. Guizani, “Deep neural network-aided gaussian message passing detection for ultra-reliable low-latency communications,” Future Generation Computer Systems, 2019.
[7] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for internet of things and analytics,” in Big data and internet of things: A roadmap for smart environments. Springer, 2014, pp. 169–186.
[8] D. Ye, M. Wu, S. Tang, and R. Yu, “Scalable fog computing with service offloading in bus networks,” in Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference. IEEE, 2016, pp. 247–251.
[9] V. B. C. d. Souza, W. Ramirez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashkoh, “Handling service allocation in combined fog-cloud scenarios,” in Communications (ICC), 2016 IEEE International Conference. IEEE, 2016, pp. 1–5.
[10] H. Shah-Mansouri and V. W. Wong, “Hierarchical fog-cloud computing for iot systems: A computation offloading game,” IEEE Internet of Things Journal, 2018.
[11] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” IEEE Access, vol. 6, pp. 12 825–12 937, 2018.
[12] O. Munoz, A. Pascual-Iserete, and J. Vidal, “Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading,” IEEE Transactions on Vehicular Technology, vol. 64, no. 10, pp. 4738–4755, 2015.
[13] W. Cheng, X. Zhang, and H. Zhang, “Statistical-qos driven energy-efficiency optimization over green 5g mobile wireless networks,” IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3092–3107, 2016.
[14] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, “Flying ad-hoc networks (fanets): A review of communication architectures, and routing protocols,” in Electrical Engineering and Computing Technologies (INTELECT), 2017 First International Conference on Latest trends in IEEE. IEEE, 2017, pp. 1–9.
[15] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2795–2808, 2016.
[16] S. M. Shatz and J.-P. Wang, “Models and algorithms for reliability-oriented task-allocation in redundant distributed-computer systems,” IEEE Transactions on Reliability, vol. 38, no. 1, pp. 16–27, 1989.
[17] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” HotCloud, vol. 10, pp. 4–4, 2010.
[18] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, “Offloading in mobile edge computing: Task allocation and computational frequency scaling,” IEEE Transactions on Communications, vol. 65, no. 8, pp. 3571–3584, 2017.
[19] D. Simon, Evolutionary Optimization Algorithms. Wiley, 2013. [Online]. Available: https://books.google.com/books?id=gwUwIEPqk30C.
[20] O. Yeniay, “Penalty function methods for constrained optimization with genetic algorithms,” Mathematical and computational Applications, vol. 10, no. 1, pp. 45–56, 2005.
[21] A. E. Eiben, E. H. Aarts, and K. M. Van Hee, “Global convergence of genetic algorithms: A markov chain analysis,” in International Conference on Parallel Problem Solving from Nature. Springer, 1990, pp. 3–12.
[22] H. Zhu, Y. He, X. Wang, and E. C. Tsang, “Discrete differential evolutions for the discounted {0-1} knapsack problem,” International Journal of Bio-Inspired Computation, vol. 10, no. 4, pp. 219–238, 2017.
[23] Y. Qi, H. Wang, L. Zhang, and B. Wang, “Optimal access mode selection and resource allocation for heterogeneous networks,” IET Communications, vol. 11, no. 13, pp. 2012–2019, 2017.
[24] W. Cheng, X. Zhang, and H. Zhang, “Optimal power allocation with statistical qos provisioning for d2d and cellular communications over underlaying wireless networks,” IEEE Journal on Selected Areas in Communications, vol. 34, no. 1, pp. 151–162, 2016.
[25] C. Jiang, Y. Chen, K. R. Liu, and Y. Ren, “Renewal-theoretical dynamic spectrum access in cognitive radio network with unknown primary behavior,” IEEE Journal on Selected Areas in Communications, vol. 31, no. 3, pp. 406–416, 2013.
[26] C. Jiang, Y. Chen, Y. Gao, and K. R. Liu, “Joint spectrum sensing and access evolutionary game in cognitive radio networks," IEEE transactions on wireless communications, vol. 12, no. 5, pp. 2470–2483, 2013.
[27] X. Zhu, C. Jiang, L. Kuang, N. Ge, and J. Lu, “Non-orthogonal multiple access based integrated terrestrial-satellite networks,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2253–2267, 2017.

[28] U. Bhoi, P. N. Ramana et al., “Enhanced max-min task scheduling algorithm in cloud computing,” *International Journal of Application or Innovation in Engineering and Management (IJAICM)*, vol. 2, no. 4, pp. 259–264, 2013.

[29] D. C. Devi and V. R. Uthariaraj, “Load balancing in cloud computing environment using improved weighted round robin algorithm for non-preemptive dependent tasks,” *The scientific world journal*, vol. 2016, 2016.

[30] T. Kokilavani, D. G. Amalarethinam et al., “Load balanced min-min algorithm for static meta-task scheduling in grid computing,” *International Journal of Computer Applications*, vol. 20, no. 2, pp. 43–49, 2011.

[31] S. Haykin et al., “Cognitive radio: brain-empowered wireless communications,” *IEEE journal on selected areas in communications*, vol. 23, no. 2, pp. 201–220, 2005.

[32] W. Cheng, W. Zhang, H. Jing, S. Gao, and H. Zhang, “Orbital angular momentum for wireless communications,” *IEEE Wireless Communications*, 2018.