International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **6**, December **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **2289–2304**

AN IMPROVED MULTI-OBJECTIVE GREY WOLF OPTIMIZER FOR DEPENDENT TASK SCHEDULING IN EDGE COMPUTING

Kaihua Jiang[1,2], Hong Ni[1,2], Rui Han[1,*] and Xu Wang[1]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ jiangkh; nih; wangx }@dsp.ac.cn; *Corresponding author: hanr@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received February 2019; revised June 2019

Abstract. *As a booming computing paradigm, edge computing is considered as an effective promotion to cloud computing. Current edge computing architectures decompose cloud services into sets of dependent subtasks, and then migrate subtasks to the devices at the edge of the network. So the dependent task scheduling strategy plays a decisive role in edge computing. This paper proposes an Improved Multi-Objective Grey Wolf Optimizer algorithm (IMOGWO) to solve the problem. IMOGWO accesses the evolutionary status of the search agents in terms of the entropy of the external archive which saves the approximate optimal solutions, and then adjusts the adaptive parameter as the status evolves. Also, the concepts of the individual density and cell domination are involved in to improve the external archive upgradation and leader selection mechanism respectively. Moreover, a Gaussian perturbation-based elitist learning strategy is adopted to avoid the local optima. The proposed algorithm is tested on twelve multi-objective benchmark problems chosen from CEC2009 test instances and compared with two popular heuristic optimization algorithms: Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) and Multi-Objective Particle Swarm Optimization (MOPSO). The simulation results illustrate that IMOGWO significantly outperforms MOEA/D and MOPSO in convergence, diversity, and coverage. Furthermore, a case study demonstrates that IMOGWO solves the dependent task scheduling problem in edge computing efficiently.*
Keywords: Edge computing, Dependent task scheduling, Optimization problem, Heuristic algorithm, Multi-objective grey wolf optimizer

1. **Introduction.** As the Internet of Things (IoT) permeates people's daily lives, a large quantity of embedded devices have accessed the network and produced massive data [1]. So how to efficiently process the data has become an urgent problem to the IoT network. In current cloud computing architecture, the embedded devices are distributed at the edge of the network. Data produced by the edge devices must be transmitted to the cloud servers via complex networks, and returned to the edge devices after execution. Due to the remote transmission route and restricted bandwidth between cloud servers and edge devices, users suffer from the long response time and bandwidth occupation. Meanwhile, cloud servers struggle with tremendous computational stress and severe traffic peaks in the network as well [2]. Consequently, edge computing is proposed to cooperate with cloud computing and remedy the defects [3].

DOI: 10.24507/ijicic.15.06.2289

In current edge computing architectures, the mainstream approach to processing the edge data is decomposing cloud services into sets of subtasks and distributing the subtasks to the edge devices near the end users. The process reduces the transmission latency between cloud servers and edge devices, and the response time for real-time services. Meanwhile, the computational stress is offloaded from the centralized cloud servers to the separate edge devices; thus, the high concurrency and network congestion are avoided. Furthermore, by transferring computation and communication overhead from the battery-powered devices to the active devices, edge computing extends the lifetime of the battery-powered devices [1].

Since the recourses of edge devices are restricted and insufficient, original cloud service must be decomposed into dependent subtasks to execute on the edge devices. So dependent task scheduling strategy affects the service delay of edge computing decisively. Dependent task scheduling can be abstracted to an optimization problem, which has been proved to be NP-complete [4]. George Amalarethinam and Joyce Mary use the list scheduling to search the solution [5]. Liu et al. adopt a one-dimensional search algorithm based on Markov decision process to find the optimal task scheduling policy [6]. As the feasible region increases exponentially along with the number of tasks and devices, the traditional methods fail to select the approximate optimal solutions within the restricted time; whereas featured with the stochastic jump and rapid convergence, the meta-heuristic algorithm can eliminate the influence of data scale within several iterations. So various meta-heuristic algorithms have been favored by both academic and industry and used to solve the optimization problem. Wang et al. adopt Multi-Objective Evolutionary Algorithm (MOEA) to balance the trade-off between the service delay and power consumption in edge-cloud computing [7]. And Xu et al. involve Genetic Algorithm (GA) to calculate the priority queue in the directed acyclic graph [8]. Moreover, currently maintain dependent task scheduling strategies mostly select only one indicator as the optimization target such as the makespan of all tasks, whereas neglect the other indicators including load balance, resource utilization, energy consumption, thus causing severe degradation of efficiencies in practical use. So it is essential to apply multi-objective meta-heuristic algorithms to the dependent task scheduling problem.

Grey wolf optimizer is a new meta-heuristic algorithm proposed by Mirjalili et al. in 2014. It mimics the leadership hierarchy and hunting mechanism of grey wolves [9]. MOGWO is the multi-objective version of GWO designed by Mirjalili et al. in 2016 [10]. Compared with other meta-heuristic algorithms, MOGWO features some unique characteristics such as fewer parameters, appropriate convergence, flexible and scalable inspired strategy and the capability of balancing exploration and exploitation during the search [11], which match the time-limited dependent task scheduling problem appropriately. Due to the advantages above, MOGWO has been applied in various engineering fields including bioinformatics, machine learning, image processing, welding scheduling, and networking [12]. Based on MOGWO, this paper proposed a multi-objective optimization algorithm to solve the dependent task scheduling problem in edge computing.

The remainder of this paper is organized as follows. Section 2 formulates the mathematical model for the dependent task scheduling problem in edge computing and presents the definitions as well as preliminaries of multi-objective optimization. Section 3 describes the principles and innovations of the proposed IMOGWO. The simulation results and a case study are presented in Section 4. Finally, the conclusions are summarized in Section 5.

2. Problem Statement and Preliminaries.

2.1. The dependent task scheduling model.

In edge computing architecture, an original cloud service is decomposed into a set of dependent subtasks. Edge devices receive the subtasks and execute them in sequence, and then send the results to the end users in order to provide service. The basic architecture of edge computing is shown in Figure 1.

The dependencies between decomposed subtasks can be modeled as a Directed Acyclic Graph (DAG), which is widely used in scheduling, Bayesian network and data compression [13]. DAG makes it possible to find the longest paths from a given starting vertex in linear time, whereas the longest path problem in an arbitrary graph is NP-hard. A DAG $G = \{V, E\}$ consists of a set V of vertices, and a set E of edges, where each vertex $v_i \in V$ represents a subtask and each directed edge $e_{i,j} \in E$ represents a precedence constraint indicating that subtask v_i depends on the subtask v_j. Each subtask executes on one individual edge device, whereas one edge device can process plural subtasks in sequence. The weight below the vertex represents the execution overhead of the subtask, and the weight below the directed edge represents the communication overhead between two subtasks. Figure 2 shows a DAG representing the dependencies between subtasks $T_1 \sim T_5$, in which T_4 depends on T_1 and T_2, and T_5 depends on T_2 and T_3. T_{op} and T_{ed} represent the preparation and conclusion tasks of the original cloud application respectively.

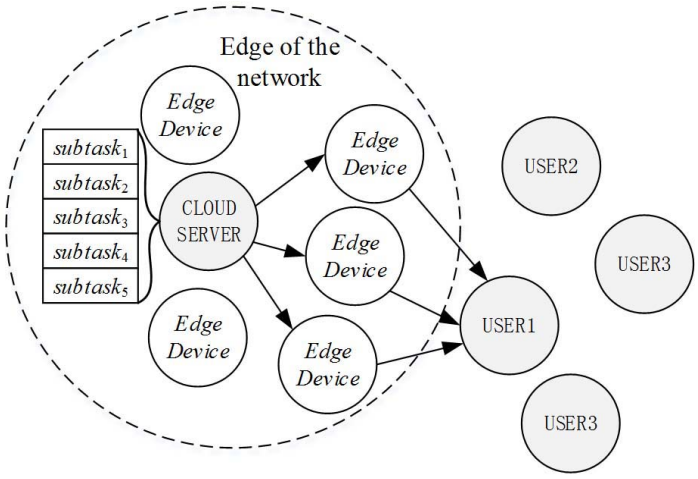

FIGURE 1. Edge computing architecture

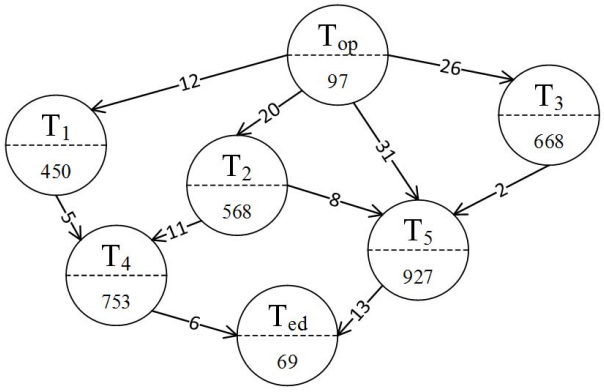

FIGURE 2. DAG of dependent task scheduling

2.2. **Preliminaries of multi-objective optimization.** Without loss of generality, the multi-objective optimization problem can be formulated as a minimization problem as follows:

$$\text{Minimize: } F(\vec{x}) = f_1(\vec{x}), f_2(\vec{x}), \ldots, f_o(\vec{x}) \tag{1}$$

$$\text{Subject to: } g_i(\vec{x}) \geq 0,\ i = 1, 2, \ldots, m;\ h_i(\vec{x}) = 0,\ i = 1, \ldots, p \tag{2}$$

$$L_i \leq x_i \leq U_i,\ i = 1, 2, \ldots, n \tag{3}$$

where n is the number of variables, o is the number of objective functions, m is the number of inequality constraints, p is the number of equality constraints, g_i indicates the i-th inequality constraint, h_i represents the i-th equality constraint, and $[L_i, U_i]$ are the boundaries of the i-th variable.

In single-objective optimization, the better solution can be distinguished easily by comparing the corresponding objective values. However, in multi-objective optimization, plural objectives of equal importance compose a multi-dimensional objective space. Therefore, comparing coordinates in each dimension is unable to prove the overall superiority of one solution over the others. This paper adopted Pareto dominance as the criteria to distinguish the solutions, which is proposed by Pareto in 1964 [14]. The mathematical definition of Pareto dominance is presented as follows.

Definition 2.1. *Pareto domination*

Suppose two vectors: $\vec{x} = (x_1, x_2, \ldots, x_k)$ and $\vec{y} = (y_1, y_2, \ldots, y_k)$. Vector x dominates vector y (denote as $x \succ y$) iff:

$$\forall i \in \{1, 2, \ldots, k\}, \quad f(x_i) \leq f(y_i) \wedge \exists j \in \{1, 2, \ldots, k\}, \quad f(x_j) < f(y_j) \tag{4}$$

Definition 2.2. *Pareto optimal*

A solution $\vec{x} \in X$ is called Pareto optimal iff:

$$\neg \exists \vec{y} \in X | F(\vec{y}) \succ F(\vec{x}) \tag{5}$$

A set containing all non-dominated solution is called Pareto optimal set (denote as P_s).

Definition 2.3. *Pareto optimal front*

A set containing the corresponding objective function values of all the Pareto optimal solution in Pareto optimal set is called Pareto optimal front.

$$P_f := \{F(x) | x \in P_s\} \tag{6}$$

2.3. **Parallel cell coordinates system.** The parallel coordinate is a common method to visualize multi-dimensional data [15]. An n-dimensional point is represented as a polyline with vertices on the parallel axes, and the position of the vertex on the i-th axis corresponds to the i-th coordinate of the point. This paper adopted a parallel coordinate to visualize the multi-objective optimization space. The approximate optimal front obtained so far is transformed into a two-dimensional plane, and each coordinate of the front is transformed into an integer. The mathematical formulation of Parallel Cell Coordinates System (PCCS) is as follows.

Among the plural objectives corresponding to the k-th non-dominated solution, the m-th coordinate $f_{k,m}$ is mapped in a two-dimensional plane with $K \times M$ cells:

$$L_{k,m} = \left\lceil K \cdot \frac{f_{k,m} - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right\rceil \tag{7}$$

where $\lceil x \rceil$ denotes the minimal integer no less than x. K is the number of the obtained solutions in the external archive, $k = 1, 2, \ldots, K$. M is the number of the objective functions, $m = 1, 2, \ldots, M$. $f_m^{\max} = \max_k (f_{k,m})$ and $f_m^{\min} = \min_k (f_{k,m})$ represent the maximum and minimum of the m-th objective of the optimal front separately. $L_{k,m} \in$

$\{1, 2, \ldots, K\}$ is the integer value that $f_{k,m}$ transforms into in PCCS, representing the m-th coordinate of the objectives for the k-th non-dominated solution. When $f_{k,m} = f_m^{\min}$, $L_{k,m} = 1$.

Particularly, the number of non-dominated solutions in the external archive K is not a given constant. Instead, it changes during the search. Meanwhile, the cell in PCCS varies as the $f_m^{\max}$, $f_m^{\min}$ and K change. So PCCS does not need any prior knowledge about the multi-objective optimization and suits the characteristics of MOGWO perfectly.

2.4. **Pareto entropy.** Information entropy is the average rate of information that a stochastic source of data produces. It can describe the distribution of an uncertain system [16] and has been adopted by various multi-objective optimization algorithms to measure the diversity of the Pareto optimal front [17]. The mathematical formula of Pareto entropy is as follows:

$$Entropy(t) = -\sum_{k=1}^{K}\sum_{m=1}^{M} \frac{Cell_{k,m}(t)}{KM} \cdot \log \frac{Cell_{k,m}(t)}{KM} \tag{8}$$

where $Cell_{k,m}(t)$ represents the number of coordinates located in the cell (k, m) when the approximate optimal front transforms into PCCS. Pareto entropy fluctuates when solutions join or quit the archive and keeps constant when the archive remains unchanged, so that the evolutionary status of the search agents can be accessed with Pareto entropy [17].

2.5. **Brief summary.** After describing the dependent task scheduling model, 2.2 expresses the multi-objective optimization problem and related concepts. 2.3 introduces the PCCS to improve the external archive upgradation and the leader selection mechanism. Based on PCCS, 2.4 involves the Pareto entropy to access the evolutionary status of the proposed algorithm.

3. Improved Multi-Objective Grey Wolf Optimizer.

3.1. **Principles of grey wolf optimizer.** As a heuristic optimization algorithm, grey wolf optimizer mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [9]. Grey wolves maintain a strict hierarchical institution in predation behavior, which divides the wolf group into four grades, including α, β, δ and ω. Each wolf represents an agent searching for the candidate solutions. The α wolf is the leader of wolves and represents the fittest solution in the current iteration. Consequently, β and δ wolves represent the second and third best solution respectively. The rest of the wolves are named ω. The i-th ω wolf follows the leaders to search and updates its position during the t-th iteration as follows:

$$\vec{D}_i = \left|\vec{C} \cdot \overrightarrow{X_p(t)} - \overrightarrow{X(t)}\right| \tag{9}$$

$$\overrightarrow{X_i(t+1)} = \overrightarrow{X_p(t)} - \vec{A} \cdot \vec{D}_i \tag{10}$$

where $\overrightarrow{X_p(t)}$ represents the aim position of the prey and $\overrightarrow{X(t)}$ represents the current position of the i-th ω wolf agent.

The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \overrightarrow{rand_1} - \vec{a}; \quad \vec{C} = 2 \cdot \overrightarrow{rand_2} \tag{11}$$

where $\vec{a}$ decreases linearly from 2 to 0, and $\overrightarrow{rand_1}$, $\overrightarrow{rand_2}$ are random vectors in $[0, 1]$.

The position of each ω wolf is updated as follows:

$$\overrightarrow{X(t+1)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

where $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are calculated by taking α, β, δ wolves as the encircled prey respectively.

MOGWO integrated an external archive to maintain the non-dominated solutions, a grid mechanism to improve the non-dominated solutions, and a leader selection mechanism to update and replace the solution in the archive [10].

3.2. **Improvements of IMOGWO.** This paper improved MOGWO by four aspects: external archive upgradation, leader selection mechanism, entropy-based adaptive parameter, and local extremum disturbance.

3.2.1. *External archive upgradation.* MOGWO adopts an external archive and an adaptive grid to store and upgrade the non-dominated solutions obtained so far. In every iteration, each new solution is compared with the member solutions of the external archive. If a new solution is not dominated by any old solution and the archive is not full, it will join the archive. Moreover, if the new solution dominates one or more old solutions, the dominated old solutions will be rejected by the archive. Otherwise, the new dominated solution will be omitted by the archive.

Once the archive is full, the grid mechanism divides the objective space into several segments. When the new solutions join the archive, grid mechanism finds the most crowded segment and omits one solution, then inserts the new solution to the least crowded segment [10], so that the diversity of the approximate optimal front is improved.

The grid mechanism only improves the diversity of the approximate optimal front, and meanwhile neglects the convergence. Furthermore, when the search agents converge highly at a point during the search, omitting one wolf at a time is ineffective. This paper adopted the individual density to measure the non-dominated solutions. After mapping the approximate optimal front in PCCS, the individual density of the corresponding non-dominated solution P_i $(i = 1, 2, \ldots, K)$ is calculated as follows:

$$Density(P_i) = \sum_{i=1, j \neq i}^{K} PCD(P_i, P_j)^{-2} \tag{13}$$

where P_j $(j = 1, 2, \ldots, K,\ j \neq i)$ denotes another solution in the external archive. PCD denotes the parallel cell distance between solution P_i and P_j, which is calculated as follows:

$$PCD(P_i, P_j) = \begin{cases} 0.5 & \forall m, \quad L_{i,m} = L_{j,m} \\ \sum_{m=1}^{M} |L_{i,m} - L_{j,m}| & \exists m, \quad L_{i,m} \neq L_{j,m} \end{cases} \tag{14}$$

where L is the PCCS coordinate calculated by Equation (7).

The external archive sorted the non-dominated solutions by the individual density from small to large, and then rejected the solutions beyond the largest archive size. Therefore, archive was upgraded within one sort. The improved archive upgradation mechanism promoted the efficiency of upgradation significantly, and meanwhile ensured the diversity of the approximate optimal solutions.

3.2.2. *Leader selection mechanism.* In MOGWO, the leader selection mechanism chooses the least crowded segments of the search space and chooses one non-dominated solution in a segment as α, β, or δ wolves. The selection is made by a roulette-wheel method with probability N_i^{-1} for each hypercube.

The original leader selection mechanism emphasizes the diversity of the approximate optimal front and neglects the convergence, meanwhile loses the ability to strike a balance between exploration and exploitation. This paper proposed a new leader selection mechanism based on the concept of cell dominance.

Definition 3.1. *Cell dominance*

Suppose $L_{x,m}$ and $L_{y,m}$ $(m = 1, 2, \ldots, M)$ are PCCS coordinate vectors corresponding to the non-dominated solution P_x and P_y of the external archive separately. Vector $L_{x,m}$ dominates vector $L_{y,m}$ (denote as $x \succ_c y$) iff:

$$\forall m \in \{1, 2, \ldots, M\},\ L_{x,m} \leq L_{y,m} \wedge \exists n \in \{1, 2, \ldots, M\},\ L_{x,n} < L_{y,n} \tag{15}$$

Definition 3.2. *The strength of cell dominance*

The strength of cell dominance $S_c(x)$ of a non-dominated solution $x \in A$ is the number of solutions cell dominated by x in PCCS. The mathematical definition is as follows:

$$S_c(x) = |\{y | y \in A \wedge x \succ_c y\}| \tag{16}$$

The strength of cell dominance reflected the comprehensive degree of an approximate optimal solution prevailing against the others and was adopted to measure the convergence degree of the non-dominated solutions in the external archive.

The new leader selection mechanism sorted the non-dominated solutions by the strength of cell dominance from high to low, and then selected the solutions with the highest strength as the leaders in each iteration. The proposed mechanism guaranteed the maximal probability to approach the true Pareto front, and thus promoted the convergence of IMOGWO.

3.2.3. *Entropy-based adaptive parameter.* This paper adopted the information entropy of the approximate Pareto optimal front as the index for the evolutionary status of wolves. The entropy increased as the new solution joined the archive and decreased as plural dominated solutions dropped out of the archive.

Suppose that in the t-th iteration, the p-th solution is dominated by new solution q. Without loss of generality, assume the other solutions keep unchanged. So the difference of Pareto entropy is calculated as follows:

$$\begin{aligned}\Delta E(t) &= E(t) - E(t-1) \\ &= \frac{1}{HM}\left|\sum_{m=1}^{M} C_{p,m} \log\left(\frac{C_{p,m}}{HM}\right) + \sum_{m=1}^{M} C_{q,m} \log\left(\frac{C_{q,m}}{HM}\right)\right. \\ &\quad \left. - \sum_{m=1}^{M} (C_{p,m} - 1) \log\left(\frac{C_{p,m} - 1}{HM}\right) - \sum_{m=1}^{M} (C_{q,m} + 1) \log\left(\frac{C_{q,m} + 1}{HM}\right)\right|\end{aligned} \tag{17}$$

where $C_{p,m}$ denotes the number of coordinates in the cell (p, m) of PCCS in the t-th iteration. As the function increases monotonically, $|\Delta E(t)|$ obtains the minimum entropy difference $2 \log 2/KM$, when solution p and q distribute in PCCS evenly [17].

This paper separated the evolutionary status of search agents into three phases. When the new solution joined the archive or plural solutions quitted the archive, the approximate optimal solutions approached the true Pareto optimal, which was called convergence status. In convergence status, the number of archive members changed in contiguous iterations; whereas in diversity status, the number of archives kept maximum, new solutions replaced the equivalent old solutions and $|\Delta E(t)| \geq 2 \log 2/KM$. When $|\Delta E(t)| < 2 \log 2/KM$, the approximate optimal solutions in the external archive kept

stagnant, which is called stagnation status. According to the evolutionary status of the wolves, the convergence parameter a self-adjusted as follows:

$$a(t) = \begin{cases} S(t) - 2/t \cdot (1 + |\Delta E(t)|) & \text{Convergence} \\ S(t) & \text{Diversity} \\ S(t) + 1/t \cdot |\Delta E(t)| & \text{Stagnation} \end{cases} \qquad S(t) = 2\cos\left(\frac{t}{T} \cdot \frac{\pi}{2}\right) \tag{18}$$

When the search agents fell into the local optima, the descent speed of parameter a slowed down. Thus, the random parameter C obtained more weight in position upgradation formula. Respectively, the descend speed accelerated as the search agents converge rapidly, emphasizing the weight of the surrounded prey. When the search agents were in diversity status, a followed the original formula of MOGWO. The adaptive parameter benefitted the rapid convergence of IMOGWO and reserved the adequate probability for search agents to execute the domination replacement in order to improve diversity.

3.2.4. *Local extremum disturbance.* Despite the emphasis of diversity in the external archive upgradation mechanism, the search agents still probably fall into local optima at the final phase of iterations. This paper adopted the Gaussian perturbation-based Elitist Learning Strategy (ELS) to jump out of local optima [19]. When the evolutionary status turned stagnant, ELS chose a random solution in the external archive and added Gaussian perturbation on a random dimension of variables. The mathematical formula of Gaussian perturbation-based ELS is as follows.

$$P^d = P^d + Gaussian\left(\mu, \delta^2\right) \cdot \left(U^d - L^d\right), \quad d = randi(1, D) \tag{19}$$

where P^d is the d-th variable of a random solution in the archive. U^d and L^d are the maximum and minimum of the d-th variable respectively.

The local extremum disturbances increased the stochastic behavior of the search agents. Benefitting from the disturbances, the search agents obtained more probability to jump out of the local optima at the final phase of iterations.

3.2.5. *Pseudo of IMOGWO.* Depending on the architecture of MOGWO, this paper improved the external archive upgradation as well as the leader selection mechanism, and then integrated the entropy-based adaptive parameter and the local extremum disturbances. The overall process of improved MOGWO is organized as Algorithm 1.

3.2.6. *Application of IMOGWO in dependent task scheduling.* To apply the proposed IMOGWO algorithm to the dependent task scheduling in edge computing, the following steps are required [20]. Firstly, abstract the mathematical model from the realistic scenarios, including the adjacency matrix, parameters such as execution overhead, and bandwidth. Secondly, select the optimization objectives from the QoS indexes and derive the objective functions from the mathematical model. The makespan, load balance, and energy consumption are often more valued in edge computing. Thirdly, put the objective functions into the IMOGWO and calculate the external archive. The solutions need to be rounded due to the discrete distribution of objective space. Finally, choose one solution from the archive according to the actual condition and put into action. Taking the solution $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 3 & 2 \end{bmatrix}$ as an instance, the makespan of assigning 5 subtasks to 3 devices is calculated in Figure 3.

Algorithm 1: Pseudo of improved multi-objective grey wolf optimizer

```
/* the initialization process */
initialize the scenario parameters
initialize the grey wolf population X_i (i = 1, 2, ..., n)
initialize a, A, and C
calculate the objective values of the initial wolves
choose the best as α wolf, second as β, and the third as δ
/* the main loop of IMOGWO */
while (t < max number of iterations)
  for each search agent
    /* the position upgradation */
    updated the position of search agent by Equations (9)-(12)
    if optimization is in stagnation status
      /* refer to 3.2.4. Local extremum disturbance */
      add local extremum disturbance by Equation (19)
    end if
  end for
  calculate the objective values of each search agent
  find the non-dominated solutions
  /* refer to 3.2.1. External archive upgradation */
  upgrade the external archive by individual density in Equation (13)
  calculate the Pareto entropy of the archive by Equation (17)
  /* refer to 3.2.3. Entropy-based adaptive parameter */
  update parameter a by Equation (18), as well as A and C
  /* refer to 3.2.2. Leader selection mechanism */
  select α, β, and δ leaders by the strength of cell dominance in Equation (16)
  t = t + 1
end while
return archive
```

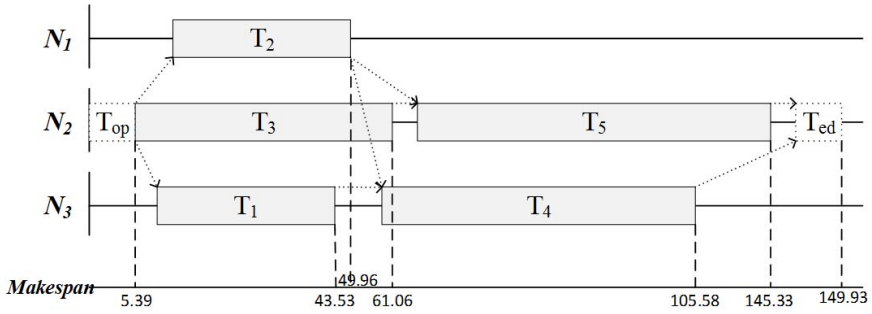

FIGURE 3. A typical makespan example

4. Simulation Results and Case Study.

4.1. **Simulation results.** The proposed IMOGWO was compared with two multi-objective heuristic algorithms: MOPSO [21] and MOEA/D [22]. The parameters of each algorithm were settled according to the corresponding references. For all the problems, this paper adopted a maximum of 30,000 for function evaluations, 1000 for iterations and 100 for search agents. The maximal capacity of the archive was 100 for the bi-objective problems and 150 for the tri-objective problems respectively. And the number of variables was 30 for unconstrained problems and 10 for constrained problems. Each algorithm ran independently for 30 times. All simulation experiments were executed on an 8GB-memory, 2.60GHz-CPU PC with Matlab R2014a.

TABLE 1. Bi-objective test problems

FUNC	Mathematical formulation
UF1	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$, $f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
UF2	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$, $f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	$y_j = \begin{cases} x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + 4j\pi/n\right) + 0.6x_1\right]\cos\left(6\pi x + j\pi/n\right), & \text{if } j \in J_1 \\ x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + 4j\pi/n\right) + 0.6x_1\right]\sin\left(6\pi x + j\pi/n\right), & \text{if } j \in J_2 \end{cases}$
UF3	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos\left(2+20y_j\pi/\sqrt{j}\right)\right)$
	$f_2 = \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_2}\cos\left(2+20y_j\pi/\sqrt{j}\right)\right)$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	$y_j = x_j - x_1^{0.5(1+3(j-2)/(n-2))}$, $j = 2,3,\ldots,n$
UF4	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h\left(y_j\right)$, $f_2 = 1-x_2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h\left(y_j\right)$, $h(t) = \lvert t\rvert/1+e^{2\lvert t\rvert}$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	$y_j = x_j - \sin\left(6\pi x_1 + j\pi/n\right)$, $j = 2,3,\ldots,n$
UF7	$f_1 = \sqrt[5]{x_1} + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$, $f_2 = 1-\sqrt[5]{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	$y_j = x_j - \sin\left(6\pi x_1 + j\pi/n\right)$, $j = 2,3,\ldots,n$
CF1	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left(x_j - x_1^{0.5(1+3(j-2)/(n-2))}\right)^2$
	$f_2(x) = 1-x_1 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left(x_j - x_1^{0.5(1+3(j-2)/(n-2))}\right)^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	Constraint: $f_1 + f_2 - a\left\lvert\sin\left[N\pi\left(f_1 - f_2 + 1\right)\right]\right\rvert - 1 \ge 0$, where N is an interger and $a \ge 1/2N$
CF2	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$, $f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	Constraint: $\frac{t}{1+e^{4\lvert t\rvert}} \ge 0$, where $t = f_2 + \sqrt{f_1} - a\sin\left[N\pi\left(\sqrt{f_1} - f_2 + 1\right)\right] - 1$
CF4	$f_1 = x_1 + \sum_{j\in J_1} h_j\left(y_j\right)$, $f_2 = 1-x_1 + \sum_{j\in J_2} h_j\left(y_j\right)$, $h_j(t) = t^2$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$ $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	$y_j = x_j - \sin\left(6\pi x_1 + j\pi/n\right)$, $j = 2,3,\ldots,n$
	Constraint: $\frac{t}{1+e^{4\lvert t\rvert}} \ge 0$, where $t = x_2 - \sin\left(6\pi x_1 + 2\pi/n\right) - 0.5x_1 + 0.25$
CF6	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$, $f_2 = \left(1-x_1\right)^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$
	$y_j = \begin{cases} x_j - 0.8x_1\cos\left(6\pi x_1 + j\pi/n\right), & \text{if } j \in J_1 \\ x_j - 0.8x_1\sin\left(6\pi x_1 + j\pi/n\right), & \text{if } j \in J_2 \end{cases}$
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}$, $J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$
	Constraint: $x_2 - 0.8x_1\sin\left(6\pi x_1 + 2\pi/n\right) - sign\left[0.5\left(1-x_1\right) - \left(1-x_1\right)^2\right]\sqrt{\left\lvert 0.5\left(1-x_1\right) - \left(1-x_1\right)^2\right\rvert} \ge 0$
	Constraint: $x_4 - 0.8x_1\sin\left(6\pi x_1 + 4\pi/n\right) - sign\left[0.25\sqrt{1-x_1} - 0.5\left(1-x_1\right)\right]\sqrt{\left\lvert 0.25\sqrt{1-x_1} - 0.5\left(1-x_1\right)\right\rvert} \ge 0$

Table 2. Tri-objective test problems

FUNC	Mathematical formulation
UF8	$f_1 = \cos(x_1\pi/2)\cos(x_2\pi/2) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$f_2 = \cos(x_1\pi/2)\sin(x_2\pi/2) + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$f_3 = \sin(0.5x_1\pi) + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_3}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$J_1 = \{j \mid 3 \le j \le n,$ and $j-1$ is a multiplication of $3\}$, $J_2 = \{j \mid 3 \le j \le n,$ and $j-2$ is a multiplication of $3\}$
	$J_3 = \{j \mid 3 \le j \le n,$ and j is a multiplication of $3\}$
UF10	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[4y_j^2 - \cos(8\pi y_j) + 1\right]$
	$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left[4y_j^2 - \cos(8\pi y_j) + 1\right]$
	$f_3 = \sin(0.5x_1\pi) + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_3}\left[4y_j^2 - \cos(8\pi y_j) + 1\right]$
	$J_1 = \{j \mid 3 \le j \le n,$ and $j-1$ is a multiplication of $3\}$, $J_2 = \{j \mid 3 \le j \le n,$ and $j-2$ is a multiplication of $3\}$
	$J_3 = \{j \mid 3 \le j \le n,$ and j is a multiplication of $3\}$
CF8	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$f_3 = \sin(x_1\pi/2) + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_3}(x_j - 2x_2\sin(2\pi x_1 + j\pi/n))^2$
	$J_1 = \{j \mid 3 \le j \le n,$ and $j-1$ is a multiplication of $3\}$, $J_2 = \{j \mid 3 \le j \le n,$ and $j-2$ is a multiplication of $3\}$
	$J_3 = \{j \mid 3 \le j \le n,$ and j is a multiplication of $3\}$, Constraint: $\frac{f_1^2+f_2^2}{1-f_3^2} - a\left\lvert\sin\left[N\pi\left(\frac{f_1^2+f_2^2}{1-f_3^2}+1\right)\right]\right\rvert - 1 \ge 0$

This paper adopted 12 benchmark multi-objective test problems proposed in CEC 2009 as test beds [23], which were provided in Tables 1 and 2. The chosen benchmark problems consisted of 9 bi-objective problems including UF1∼4, 7, CF1, 2, 4, 6 and 3 tri-objective problems including UF8, UF10, and CF8, where UF indicated the unconstrained problems and CF indicated the constrained problems. The chosen test problems introduced different multi-dimensional search spaces to the simulation experiments with various Pareto optimal fronts including convex, non-convex, discontinuous, and multi-modal kinds.

As to the performance metric, this paper adopted the Inverted Generational Distance (IGD) as the criteria. Suppose P^* is a set of uniformly distributed points along the Pareto optimal front which distributes in the multi-objective space, and A is an approximate set to the Pareto optimal front, the average distance from P^* to A is defined as follows:

$$IGD(A, P^*) = \sum_{v\in P^*} d(v, A)/|P^*| \tag{20}$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A. If $|P^*|$ is large enough to cover the Pareto optimal front, $IGD(A, P^*)$ can measure both the diversity and convergence of A in a sense [23]. To obtain a low value of $IGD(A, P^*)$, the set A must be very close to the true Pareto optimal front and not miss any part of the whole front. The statistic results of IGD for each algorithm were provided in Table 3.

Table 3 shows the statistical results of IMOGWO and the comparisons for IGD. In 12 test problems, IMOGWO obtains 9 best mean value and standard deviation of IGD except UF1, 4, 7, in which IMOGWO is marginally inferior to MOGWO. As IGD can evaluate both the diversity and convergence of an algorithm [23], it can be stated that IMOGWO overall outperforms MOGWO, MOPSO, MOEA/D in accuracy and stability.

TABLE 3. Statistical results for IGD

	UF1				UF2			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	0.1219	0.1145	0.1370	0.1871	0.0163	0.0583	0.0604	0.1223
Std.	0.0131	0.0195	0.0441	0.0507	0.0104	0.0074	0.2763	0.0107
	UF3				UF4			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	0.2203	0.2557	0.3140	0.2886	0.0611	0.0587	0.1360	0.0681
Std.	0.0527	0.0807	0.0447	0.0159	0.0044	0.0005	0.0074	0.0021
	UF7				CF1			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	0.1802	0.1604	0.3540	0.4552	0.0164	0.3889	0.3694	0.7758
Std.	0.0787	0.1391	0.2044	0.1898	0.0222	0.1911	0.0953	0.2863
	CF2				CF4			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	0.1835	0.5849	0.6615	0.9943	0.1485	0.6530	0.4603	0.7290
Std.	0.0752	0.2437	0.3051	0.4522	0.0937	0.3151	0.1355	0.4611
	CF6				UF8			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	0.0055	0.6642	0.2408	0.7602	0.6984	2.0578	0.5367	NULL
Std.	0.0032	0.2208	0.0104	0.2375	0.1121	1.1455	0.1826	NULL
	UF10				CF8			
IGD	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*	*IMOGWO*	*MOGWO*	*MOPSO*	*MOEA/D*
Mean	1.2493	3.5945	1.6372	NULL	0.1327	1.0528	0.6904	NULL
Std.	0.1377	3.4883	0.2988	NULL	0.0772	0.4170	0.3070	NULL

Figure 4 shows the obtained Pareto optimal fronts of IMOGWO, MOGWO, MOPSO and MOEA/D for UF1∼3. UF1, 2, 3 obtain the same true Pareto front, but the performance of each algorithm on them varies greatly. In UF1, MOEA/D fits the latter part of true Pareto front perfectly, yet fails to cover the former part. The other three algorithms can only find several solutions near to the true Pareto front. Benefitted from the improved mechanisms, IMOGWO obtains a more scattered optimal front, and thus realizes a smaller IGD. In UF2, IMOGWO, MOGWO and MOPSO obtain similar optimal fronts that cover the complete true Pareto front except for MOEA/D. Whereas the coverage of IMOGWO is broader and more continuous compared to the other two algorithms. As to UF3, MOPSO and MOEA/D can only approach several points of the true Pareto front, whereas IMOGWO and MOGWO cover the entirety. Moreover, IMOGWO apparently approaches the true Pareto front closer, thus proving the convergence and coverage of IMOGWO on UF3.

Figure 5 shows the obtained Pareto optimal fronts of IMOGWO and MOGWO for the other bi-objective test problems. The results of MOPSO and MOEA/D are neglected as they are all inferior to the original MOGWO. In constrained problems CF1, 2, 4, 6, IMOGWO shows visible narrowness to the true Pareto front because of the additive local extremum disturbances, which makes the exploitation to the constrained search space more meticulous. In UF4 and UF7, IMOGWO obtains a larger IGD than MOGWO though, the coverage of IMOGWO is significantly broader than MOGWO.

Figure 6 shows the obtained Pareto optimal fronts of IMOGWO and MOGWO for the tri-objective test problems. In UF8, the coverage of IMOGWO is significantly superior to MOGWO. In UF10, IMOGWO covers a part of the true Pareto front narrowly, whereas

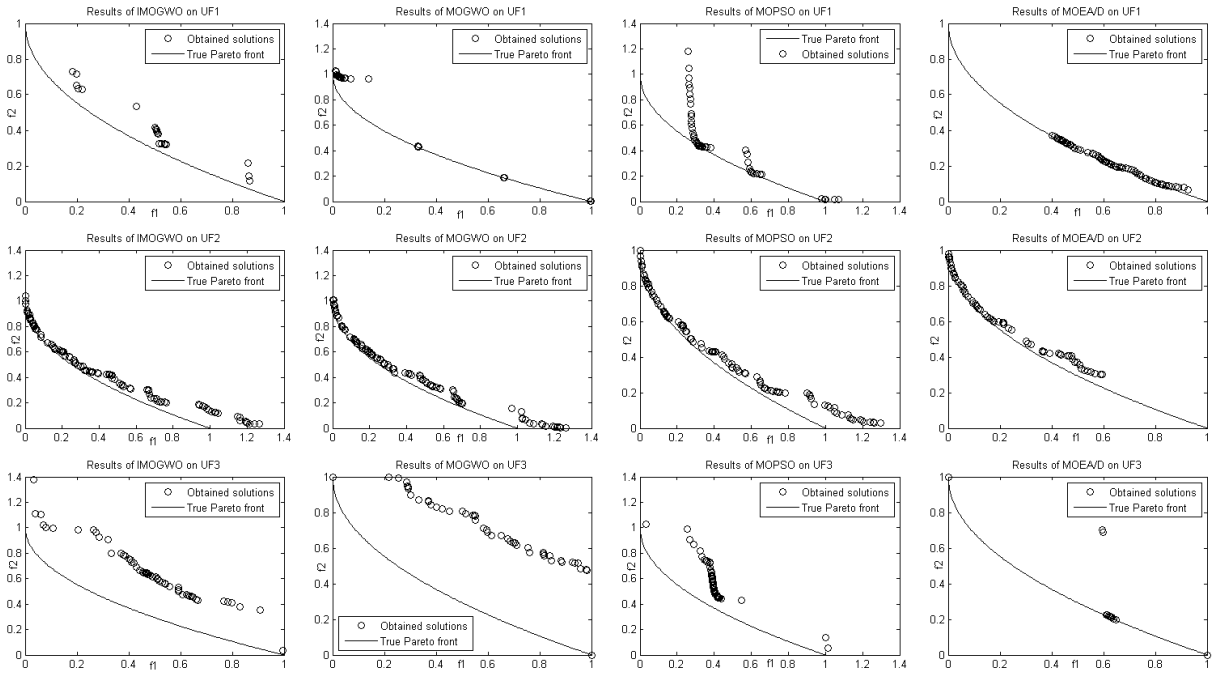

FIGURE 4. Pareto optimal fronts of IMOGWO, MOGWO, MOPSO, and MOEA/D

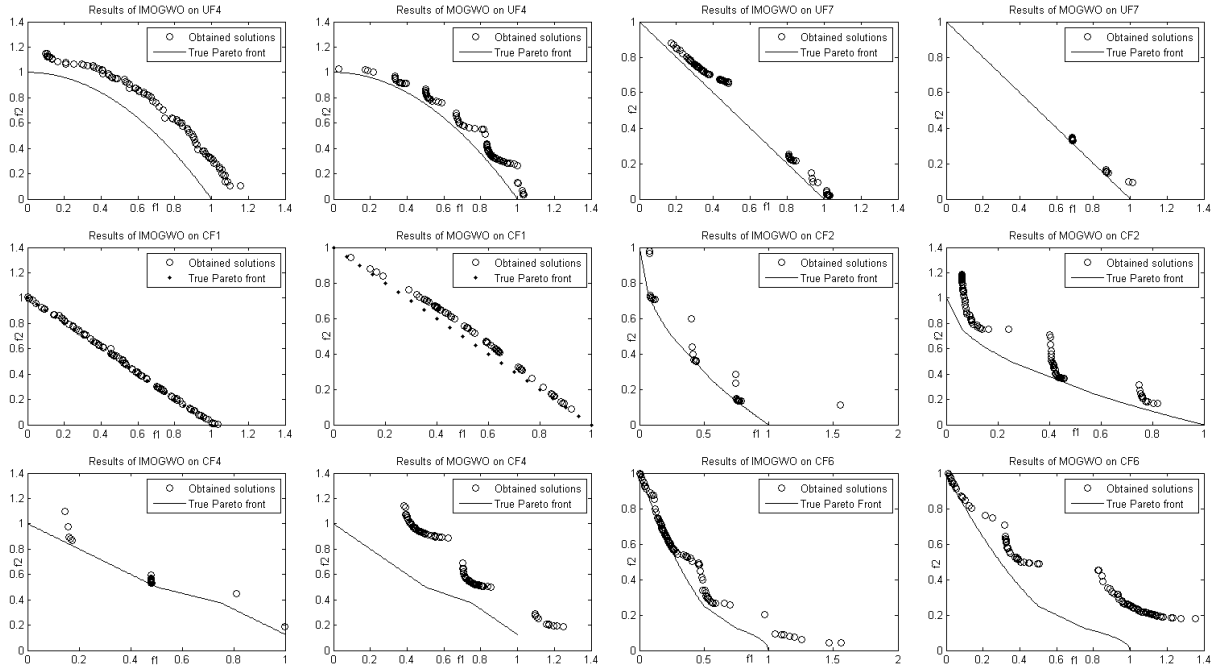

FIGURE 5. Pareto optimal fronts of IMOGWO and MOGWO

MOGWO fails to get close to the true Pareto front. In CF8, MOGWO gets trapped in the local optima, yet IMOGWO distributes evenly near the true Pareto front. Consequently, IMOGWO shows favorable performance on tri-objective test problems.

4.2. **A case study.** To evaluate the actual effect of IMOGWO in reality, this paper realized the proposed algorithm on the smart routers settled at the edge of the network. In the work of [24], the computation and storage resources of edge routers are integrated. By

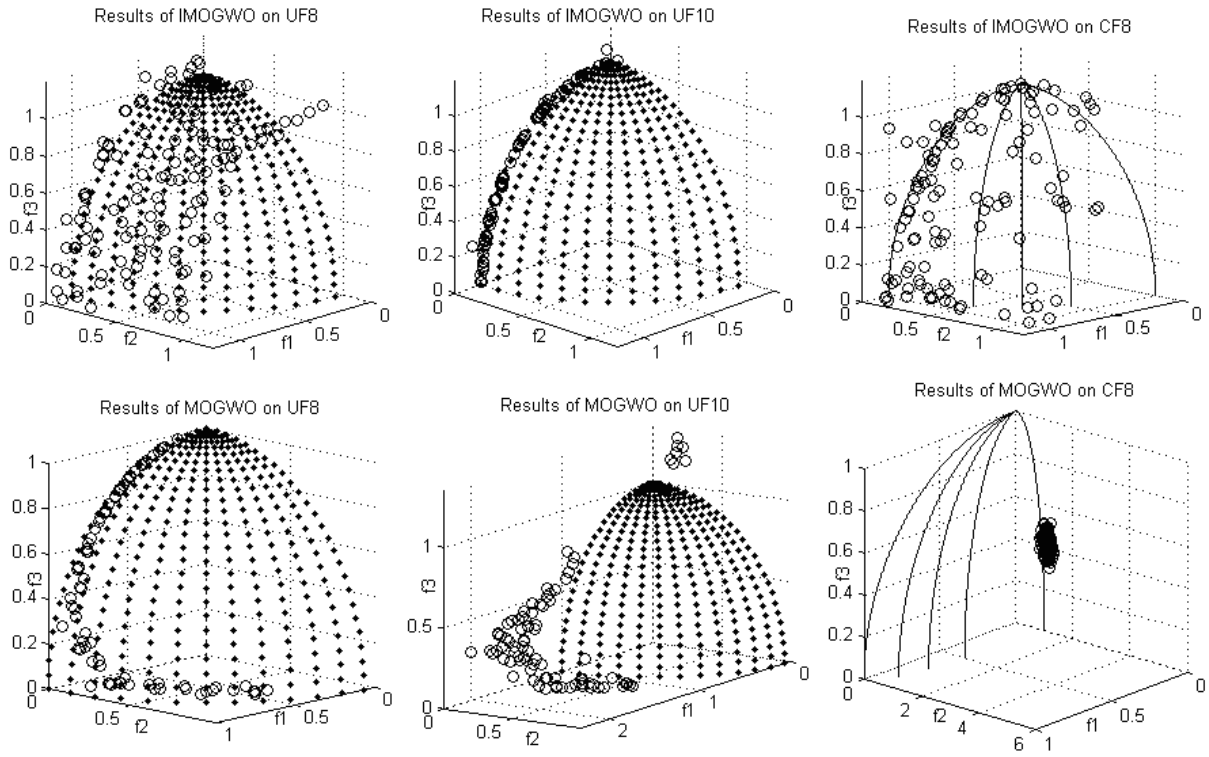

FIGURE 6. Obtained Pareto optimal fronts by IMOGWO, MOGWO for UF8, 10 and CF8

caching the hotspot streaming media data positively, edge routes provide direct streaming service as the users make requests.

Supposing the scenarios of assigning 100, 200, 500 cache subtasks to 20 edge routers separately, this paper adopted IMOGWO, Min-Min, and Round-Robin algorithm to plan the scheduling strategy. The ideal scheduling strategy aimed to obtain a minimal makespan and the load balance among all edge routers, which meant to minimize the standard deviation of the resource utilization.

The makespan of subtask t_i executed on the device n_j is calculated as follows:

$$Makespan(t_i) = \frac{t_i}{v_j} + MAX\left\{\frac{e_{j,k}}{b_{j,n(k)}} + Makespan(t_k)|t_k \in \psi_{(i)}^{pre}\right\} \tag{21}$$

where $\psi_{(i)}^{pre}$ represents the set of subtasks on which the subtask t_i depends. $b_{j,n(k)}$ is the bandwidth from device n_j to $n_{(k)}$. $e_{j,k}/b_{j,n(k)}$ is the transmission latency between device n_j and $n_{(k)}$. And t_i/v_j is the execution delay of the subtask t_i executed on the device n_j.

This paper executed IMOGWO, Min-Min, and Round-Robin on 30 random dependency matrix respectively, and then calculated the average of the objective functions. To IMOGWO, the mean value contained all the non-dominated solutions in the archive. The makespan and standard deviation of resource utilization of each algorithm were shown in Figure 7.

As Figure 7 shows, IMOGWO outperforms Min-Min and Round-Robin in makespan. Also, as the number of subtasks rises, the uptrend of makespan for IMOGWO is weaker than the other two algorithms. Meanwhile, IMOGWO obtains a minimal standard deviation of resource utilization among three algorithms, which illustrates that IMOGWO provides the best load balance to edge routers. Moreover, the standard deviation of IMOGWO slightly drops as the subtasks increase. Whereas the standard deviations of Min-Min and Round-Robin rise oppositely, illustrating that IMOGWO shows superior

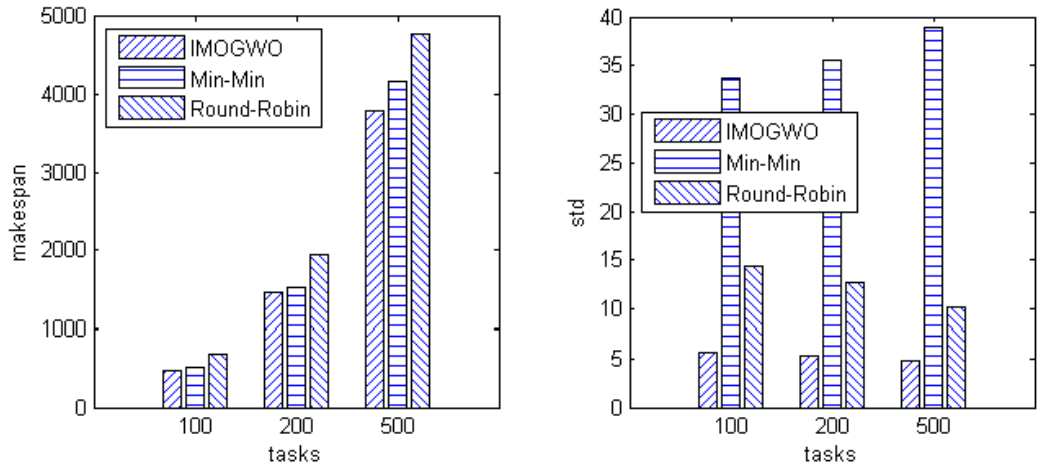

FIGURE 7. Makespan and std. of IMOGWO, Min-Min, and Round-Robin

performance in the large-scale scene. In conclusion, IMOGWO solves the dependent task scheduling problem in edge computing efficiently.

5. **Conclusions.** This paper proposed a multi-objective heuristic algorithm based on Grey Wolf Optimizer (IMOGWO) to solve the dependent task scheduling in edge computing. First, the proposed algorithm accessed the evolutionary status of the wolves according to the entropy of the approximate optimal solutions saved in the archive, and then adjusted the adaptive parameter as the wolves evolved. Secondly, the concept of individual density was adopted to promote the efficiency of external archive upgradation. Thirdly, the new leader selection mechanism involved the strength of cell dominance to guarantee the convergence of IMOGWO. Lastly, a Gaussian perturbation-based elitist learning strategy was involved to avoid the local optima. The improvements above allowed IMOGWO to obtain superior coverage and convergence simultaneously.

The proposed IMOGWO algorithm was applied to twelve multi-objective test problems chosen from CEC 2009 test instances and compared with two well-known algorithms: MOPSO and MOEA/D. The simulation results illustrated that IMOGWO provided very competitive results. The IGD performance quantitatively proved the excellent convergence behavior of IMOGWO. Moreover, the shape of the obtained Pareto fronts and coverage revealed the significant advantages over MOPSO and MOEA/D. The case study demonstrated that IMOGWO outperforms Min-Min and Round-Robin in makespan and load balance.

Acknowledgment. This work is partially supported by the Young Elite Researcher Project of Institute of Acoustics, Chinese Academy of Sciences: Research on Key Technologies of Distributed Computing Framework for SEA Service (Grant No. QNYC201715) and the Strategic Priority Research Program of Chinese Academy of Sciences: SEANET Technology Standardization Research and System Development (Grant No. XDC020107 01). The authors would like to express their appreciation to the reviewers and anonymous referees for their helpful comments and suggestions, which have improved this work.

REFERENCES

[1] W. Yu et al., A survey on the edge computing for the Internet of Things, *IEEE Access*, vol.6, pp.6900-6919, 2018.

[2] C. Lee, D2D proxy caching strategy for 5G-based multi-zone networking services, *ICIC Express Letters*, vol.12, no.11, pp.1177-1182, 2018.
[3] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, Edge computing: Vision and challenges, *IEEE Internet of Things Journal*, vol.3, no.5, pp.637-646, 2016.
[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
[5] D. I. George Amalarethinam and G. J. Joyce Mary, A new DAG based dynamic task scheduling algorithm (DYTAS) for multiprocessor systems, *International Journal of Computer Applications*, vol.19, pp.24-28, 2011.
[6] J. Liu, Y. Mao, J. Zhang and K. B. Letaief, Delay-optimal computation task scheduling for mobile-edge computing systems, *2016 IEEE International Symposium on Information Theory (ISIT)*, Barcelona, pp.1451-1455, 2016.
[7] X. Wang, H. Ni, R. Han and X. Huang, Trade-off between service delay and power consumption in edge-cloud computing, *International Journal of Innovative Computing, Information and Control*, vol.14, no.6, pp.2011-2024, 2018.
[8] Y. Xu, K. Li, J. Hu and K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, *Information Sciences*, vol.270, pp.255-287, 2014.
[9] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, vol.69, pp.46-61, 2014.
[10] S. Mirjalili, S. Saremi, S. M. Mirjalili et al., Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization, *Expert Systems with Applications*, vol.47, pp.106-119, 2016.
[11] H. Faris, I. Aljarah, M. A. Al-Betar and S. Mirjalili, Grey wolf optimizer: A review of recent variants and applications, *Neural Computing and Applications*, pp.1-23, 2018.
[12] S. Zapotecas-Martínez, A. García-Nájera and A. López-Jaimes, Multi-objective grey wolf optimizer based on decomposition, *Expert Systems with Applications*, vol.120, pp.357-371, 2019.
[13] F. Balmas, Displaying dependence graphs: A hierarchical approach, *Proc. of the 8th Working Conference on Reverse Engineering*, Stuttgart, Germany, pp.261-270, 2001.
[14] V. Pareto, *Cours D'économie Politique*, Librairie Droz, Geneva, 1964.
[15] A. Inselberg, The plane with parallel coordinates, *The Visual Computer*, vol.1, pp.69-91, 1985.
[16] J. Kapur and H. Kesavan, *Entropy Optimization Principles with Applications*, Academic Press, San Diego, 1992.
[17] X. Cui, M. Li and T. Fang, Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles, *Proc. of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, Seoul, Korea, pp.1316-1321, 2001.
[18] W. Hu, G. G. Yen and X. Zhang, Multiobjective particle swarm optimization based on Pareto entropy, *Journal of Software*, vol.25, pp.1025-1050, 2014.
[19] Z.-H. Zhan, J. Zhang, Y. Li and H. S.-H. Chung, Adaptive particle swarm optimization, *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.39, no.6, pp.1362-1381, 2009.
[20] W. J. Wang, Y. S. Chang, W. T. Lo et al., Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments, *The Journal of Supercomputing*, vol.66, no.2, pp.783-811, 2013.
[21] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evolutionary Computation*, vol.8, no.3, pp.256-279, 2004.
[22] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evolutionary Computation*, vol.11, no.6, pp.712-731, 2007.
[23] Q. Zhang et al., Multiobjective optimization test instances for the CEC 2009 special session and competition, *Proc. of University of Essex, Colchester, UK, and Nanyang Technological University*, Singapore, 2008.
[24] X. Huang, X. Zeng, C. Liu et al., Design and implementation of smart router's HTTP server in SEA service, *Journal of Network New Media*, vol.6, pp.31-37, 2017.