Multi-factor Evolution for Large-scale Multi-objective Cloud Task Scheduling

Tianhao Zhao[1], Linjie Wu[1], Di Wu[2], Jianwei Li[1], and Zhihua Cui[1*]
[1] College of Computer Science, Taiyuan University of Science and Technology,
Taiyuan, Shanxi, 030024, China.
[2]Faculty of Information Technology, Beijing University of Technology,
Beijing, 100000, China.
[e-mail: zhaotianhao1015@163.com, wulinjie19971013@163.com, wuxiaodou942@163.com,
1997021@tyust.edu.cn, zhihuacui@gmail.com]
[*]Corresponding author: Zhihua Cui

*Received December 2, 2022; revised February 11, 2023; accepted March 27, 2023;
published April 30, 2023*

Abstract

Scheduling user-submitted cloud tasks to the appropriate virtual machine (VM) in cloud computing is critical for cloud providers. However, as the demand for cloud resources from user tasks continues to grow, current evolutionary algorithms (EAs) cannot satisfy the optimal solution of large-scale cloud task scheduling problems. In this paper, we first construct a large-scale multi-objective cloud task problem considering the time and cost functions. Second, a multi-objective optimization algorithm based on multi-factor optimization (MFO) is proposed to solve the established problem. This algorithm solves by decomposing the large-scale optimization problem into multiple optimization subproblems. This reduces the computational burden of the algorithm. Later, the introduction of the MFO strategy provides the algorithm with a parallel evolutionary paradigm for multiple subpopulations of implicit knowledge transfer. Finally, simulation experiments and comparisons are performed on a large-scale task scheduling test set on the CloudSim platform. Experimental results show that our algorithm can obtain the best scheduling solution while maintaining good results of the objective function compared with other optimization algorithms.

Keywords: Cloud computing, evolutionary algorithm, large-scale, multi-factorial, multi-objective, task scheduling.

1. Introduction

Cloud service is an information processing method in which a cloud provider provides virtual computing resources for all users through a network platform system to achieve large-scale computing [1, 2]. Cloud services use technologies such as distributed computing and virtual resource management to centralize dispersed resources in the network (e.g., virtual machine computing resources, application runtime platforms, and software) to form a shared pool of resources and provide services and billing to users in a dynamic on-demand and measurable manner [3, 4]. With the fast development of science and technology, cloud computing has brought a revolution in information technology and other fields through its efficient and powerful architecture, and computing technology has been extensively applied for personal and business purposes. Cloud computing is known as the primary solution for complex computing and large-scale data operations. The technology brings extreme convenience to users and businesses with its advantages of hyperscale, virtualization, high reliability, versatility, high scalability, and pay-as-you-go. As the tasks handled in cloud computing environments have different processing characteristics and requirements, hybrid, public, and private cloud environments have been derived [5]. **Fig. 1** depicts the classification of the cloud environment and the three typical service models: Saas (Software as a Service), Paas (Platform and Services), and Iaas (Infrastructure as a Service)[6].

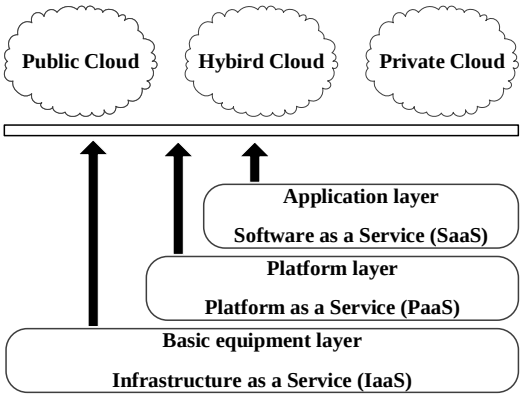

Fig. 1. Types and classifications of cloud services.

The cloud provider provides virtual machine resources on the cloud (VMs consist of hardware computing, such as CPU, RAM, and BandWidth) [7]. These VM resources perform scheduling tasks independently of each other, and the VM in different data centers will execute different operating systems as well as application software. At the same time, virtualization, as a resource management technology, breaks the indivisible barrier of physical structure, which allows a single computer to be virtualized as multiple logical computers, and the abstraction of physical computing resources to achieve simulation, isolation, and sharing of resources. Therefore, it allows multiple VMs to perform their respective tasks simultaneously and without interfering with each other when task scheduling virtual machine resources.

The scheduling problem [8, 9] occurs in various areas of the real world [10, 11], so it has strong research value and practical significance [12, 13]. The fast growth of science and technology has made the size of the Internet grow with each passing day [14, 15], and the massive growth of scheduling tasks on the cloud has increased the load on cloud computing. The large-scale cloud task scheduling problem is a highly researched NP-hard problem

because of using its task complexity and heterogeneity of VM resources. Researchers have conducted extensive research on scheduling problems in cloud environments and have proposed scheduling algorithms that include a combination of mathematical optimization modeling and various heuristics [16, 17]. EA is a population-based metaheuristic method to simulate the iterative course of events of biological evolution. With self-organizing, self-adaptive, and self-learning properties, EAs can handle NP-hard problems effectively without the limitation of solving them [18, 19]. Therefore, EAs and related algorithms have been extensively applied in path planning [20, 21], vehicle scheduling [22, 23], and cloud task resource scheduling [24, 25]. For instance, Geng et al. [26] proposed a multi-objective cloud task scheduling problem with optimization modeling. And in the paper proposes a model developed based on a multi-objective EA solution from a hybrid perspective. Xu et al. [27] proposed a high-dimensional multi-objective cloud environment problem for uncertainty and user satisfaction in the current cloud environment. And proposed an algorithm based on the normal distribution of angular penalty distance. However, as the number of tasks uploaded by users and the level of demand increases, the cloud needs to continue to grow in complexity and the number of VMs and the need for heterogeneity continues to increase. EAs often need to consume more computational resources as well as running time to get feasible scheduling solutions when solving the scheduling problem. Therefore, traditional EAs can become almost useless in solving the constructed model. To better describe the real-life large-scale cloud task scheduling problem, it is crucial to propose an effective scheduling approach to solve the problem.

Multiple optimization problems [28] often occur simultaneously in real life, and there will be some correlation between these problems, making it possible to share knowledge between problems. Motivated by multi-task learning, an algorithm MFEA based on the MFO strategy is proposed in the field of evolutionary algorithms, which expands a new direction for population-based algorithms so that multiple populations can make full use of the parallel search information implied between populations for knowledge migration during the evolutionary process, thus achieving the goal of accelerating the parallel evolution of multiple populations and finally obtaining the optimal solution required for their respective optimization problems.

In this paper, a large-scale multi-objective cloud task scheduling model is proposed. In this model, the appropriate VM resources are scheduled by considering the task requirements submitted by users, while also considering the two objectives of task execution time and cost. According to the characteristics of the constructed model, an MFO-based multi-objective optimization algorithm is proposed to accomplish cloud task scheduling. When the algorithm solves the large-scale cloud task scheduling problem, the solution includes all VM data in the cloud and the virtualization technology allows all VMs to perform tasks without interfering with each other. The main types of tasks we consider are independent tasks, each of which needs to be assigned under a suitable virtual machine. This indicates that a large-scale cloud task scheduling problem can be decomposed into multiple smaller-scale problems, and thus the introduction of MFO techniques allows the EA to solve the divided MFO problem form during the optimization process. The main contributions of the article are as follows:

(1) This paper proposes a large-scale multi-objective cloud task scheduling model considering task execution time and execution cost, decomposes the model constructed above into multiple small-scale scheduling models by a decomposition strategy of dimensionality reduction of the decision space dimensions, and redescribes it in the form of an MFO problem;

(2) An NSGA-III algorithm based on MFO is proposed. The initial population generated by the algorithm is divided into multiple subpopulations, and crossover across subpopulations is performed during iterations to generate higher-quality individuals and improve the performance of the algorithm. The combination of the strategies provides the algorithm with the ability to solve the MFO problem.

The other sections of this paper are organized as follows. Section II introduces the related work to EA-based cloud task scheduling methods and scheduling problems based on MFO techniques. Section III details the multi-objective cloud task scheduling problem proposed in this paper. Section IV describes the processing of the proposed method and the overall framework. Section V discusses the setup for the experimental part and the analysis of the results. Finally, in the conclusion, we have summarized this paper and look forward to future work.

2. Related Work

In this Section, we first present the related work to the cloud task scheduling problem. After that, we present the application of MFO based on EAs in realistic scenarios presented in section 2.2.

2.1 The optimization problem for cloud task scheduling

Scholars are currently concentrating on modeling the scheduling problem as a mathematical problem about multi-objective optimization [29]. This approach aims to describe cloud task scheduling as a mathematical function and to solve the constructed mathematical problem using the EA. Malti et al. [30] proposed a multi-objective task scheduling grey wolf optimization (MOTSGWO) algorithm to optimize important parameter settings such as energy consumption, migration duration, and utilization in cloud services for the linear relationship between applications and workloads in the cloud task scheduler. Experimental results display that the MOTSGWO is outperform other scheduling algorithms. Imene et al. [31] constructed a task scheduling problem in the cloud considering minimized running time (TE), cost (cout), and power consumption (CE), and used the NSGA-III to optimize the above-proposed problem. Peng et al. [32] considered that the traditional GA algorithm can suffer from the shortage of long execution time in solving task scheduling problems with task priorities, and proposed a parallel GA with MapReduce framework, which is optimized in two phases. The first phase combines the GA with heuristic strategies to allocate tasks, and the second phase combines the GA with the MapReduce framework to assign jobs. Experimental results indicate that the method can substantially minimize the total scheduling execution time of cloud tasks. Alghamdi, MI [33] considered that the dramatic increase in the number of tasks and resources in the cloud leads to the problem that current optimization algorithms have high time complexity when solving cloud scheduling problems. It proposes a binary particle swarm optimization (BPSO) algorithm based on artificial neural networks (ANNs). BPSO uses ANNs to determine the variance of virtual machine resources, allowing the algorithm to update particle positions in iterations based on the variance of resources. Zade, B, and Mansouri, N [34] proposed an approach for a highly efficient scheduling method using Fuzzy Improved Red Fox Optimization (FIRFO) algorithm and game theory (EGFIRFO). EGFIRFO is used to solve a cloud scheduling problem considering four objectives such as resource utilization, load balancing, manufacturing span, and execution time. Compared with other scheduling methods, EGFIRFO achieves the best experimental results.

Li et al. [35] proposed an improved multi-objective cuckoo search (IMOCS) algorithm for

optimizing the application scheduling problem in mobile edge computing (MEC). Simulation experiments demonstrate that the IMOCS scheduling algorithm can provide the best scheduling solution for MEC. Xia et al. [19] presented a multi-objective genetic algorithm (MOGA) and used MOGA for a large-scale workflow scheduling problem. MOGA utilizes the longest common subsequence (LCS) approach to preserve gene bits in elite individuals. Malti et al. [30] proposed an optimization algorithm based on flower pollination behavior. And it is used as a multi-objective cloud service scheduling model considering minimizing time span, execution cost, and maximizing task mapping. To improve the algorithm's search capability, the algorithm uses non-dominated sorting and technique for order of preference by similarity to ideal solution (TOPSIS) techniques. It is experimentally demonstrated that the algorithm can efficiently optimize the problem and find the optimal scheduling solutions. Qin et al. [36] proposed a cluster-based cooperative co-evolutionary (CBCC) algorithm. CBCC was applied to a multi-objective workflow scheduling problem, including minimizing cost, time, and risk. Experimental results demonstrate that CBCC has good performance compared with other advanced algorithms. Emami, H [37] proposed an Enhanced Sunflower Optimization (ESFO) algorithm and achieved good performance on a cloud task scheduling model considering energy consumption and time span objectives. The performance of ESFO is improved by 0.73% and 2.24% respectively compared to other algorithms.

2.2 The multi-factor optimization for the optimization problem

In real life, optimization tasks do not exist independently of each other, there will be a degree of similarity and potential connection between these tasks [38]. To meet the need of optimizing multiple tasks simultaneously, Gupta et al. [39] first introduced the concept of evolutionary multitasking (EMT) and proposed a multi-factor evolutionary algorithm (MFEA) framework for solving the Multi-tasking Optimization (MTO) problem. The detailed flow of the MTO problem is illustrated in **Fig. 2**. The EMT algorithm can optimize several different (but potentially similar) tasks simultaneously and exploit the commonalities between the different tasks to accelerate the convergence of each task itself. MFO framework provides a new way of solving the algorithm, which exploits the implicit information between populations for knowledge transfer to accelerate the parallel evolution of the populations and ultimately obtain the optimal solution for each task. The purpose of MFO is to discover potential information that may exist between multiple tasks, allowing algorithms to use this information to facilitate the optimization of each other's effectiveness. This process of using the information to optimize each other is called knowledge transfer. The idea of knowledge transfer is implemented in the population by finding high-quality individuals from different tasks, judging whether they satisfy the transfer conditions, and if they do, we crossover the individuals and consider that the resulting offspring individuals will inherit the good genes from their parents, thus satisfying the needs of different tasks.

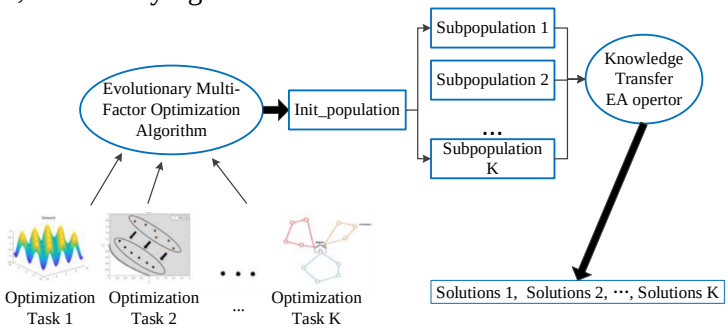

Fig. 2. The MFO algorithm uses the knowledge transfer approach to solve the MTO problem.

The MTO problem exists in various application domains, such as the Internet service domain [40], medical domain [41], etc. The researchers have worked extensively on the multitasking scheduling problem [42]. Zhou et al. [43] proposed an MFEA algorithm with individual gradient measurement (MFEA-IG) for the mobile agent path planning (MAPP) problem, where there would be multiple agents considered for simultaneous optimization, and the performance of MFEA-IG was shown to be much better than that of traditional EA through experiments. Yi et al. [44] proposed a novel interval MFEA (IMFEA), which uses the interval crowding distance of the individual set to ensure diversity in the algorithm evolution process. And IMFEA is applied to the robot path planning problem with multiple terrains. Bali et al. [45] considered the multi-UAV path planning problem, and since different UAV paths need to be optimized for their respective objective values, an MTO problem based on fidelity and time cost was proposed while using a multi-objective multifactor evolutionary algorithm (MO-MFEA) for optimization. Zhang et al. [46] pointed out that dynamic job shop scheduling (DJSS) problems are characterized by dynamic uncertainty in the processing, leading to the problem that traditional GA cannot predict the best solution well, proposed a GA combines MFO method and proved the validity of the MFO strategy was demonstrated experimentally. Rauniyar et al. [47] proposed a pollution-routing problem (PRP) considering multiple paths optimized simultaneously, so to solve the constructed MTO problem simultaneously, the MFO is combined with the NSGA-II algorithm, which gives NSGA-II the ability to optimize multiple optimization problems simultaneously. Experimental results display that the MFO strategy is superior to the single-population EA algorithm. Liu et al. [48] pointed out that the traditional power scheduling problem includes both active and reactive power scheduling problems, and the deficiency that the EA can only solve a single optimal scheduling problem independently, and then proposed an MFO-based power scheduling method, which introduced the MO-MFEA to optimize both active and reactive power scheduling problems. Zheng et al. [49] proposed an MFEA algorithm based on a greedy allocation operator for the large-scale virtual machine placement (LVMP) problem for which the current EA cannot meet the demand, and experimentally demonstrated the excellence of the proposed method in optimizing the LVMP problem.

3. The Formulation of the Problem

The process of scheduling VM resources for tasks in the cloud is illustrated in **Fig. 3**. Suppose the number of users is i. They will upload m tasks to the cloud to request the cloud resources with the number of VM resources n. Where the data center agent collects the tasks in the cloud, forwards them to the resource scheduling operator, and makes a resource scheduling request to the resource provider. Thus, it is a process of scheduling m cloud tasks to n VMs and finally getting the best set of scheduling solutions.

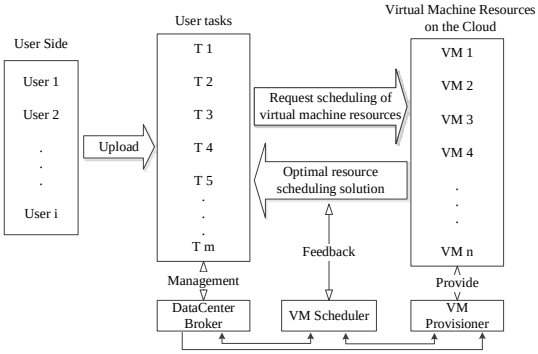

Fig. 3. The execution process of the cloud task scheduling.

In this paper, we consider the overall time for all tasks to complete execution, so we choose the VM with the longest execution time as the scheduling execution time. It is calculated as follows:

$$T_{total} = Max(VM^{i}_{Time})$$
$$VM^{i}_{Time} = \sum_{j=1}^{N}\left(\frac{Task^{j}_{length}}{VM^{i}_{CPU}\cdot VM^{i}_{mip}} + \frac{Task^{j}_{filesize} + Task^{j}_{outputsize}}{VM^{i}_{BW}}\right) \quad (1)$$

Where VM^{i}_{Time} denotes the time taken by the $i-th$ VM to complete execution. $Task^{j}_{length}$, $Task^{j}_{filesize}$, and $Task^{j}_{outputsize}$ denotes the attributes of task length, file upload size, and file output size of the $j-th$ cloud task, respectively. VM^{i}_{CPU}, VM^{i}_{mip}, and VM^{i}_{BW} denotes the computing power attribute of the $i-th$ VM, respectively. The detailed settings of these attributes are given in Section 5.

To better describe cloud services in realistic scenarios, we charge users based on the traffic and bandwidth used by their tasks, and we refer to Tencent Cloud's pricing methodology [26]. The scheduling execution cost is calculated as follows:

$$Cost = \sum_{i=1}^{M} sum(VM^{i}_{BW\_Cost})$$
$$VM^{i}_{BW\_Cost} = \begin{cases} 0.08*(Task_{filesize} + Task_{outputsize}), & U_{bw} \le 10\% \\ 0.25*(Task_{bw} - 5) + 5*0.063, & U_{bw} > 10\% \end{cases} \quad (2)$$

Where $VM^{i}_{BW\_Cost}$ denotes the execution cost of the $i-th$ VM. U_{bw} denotes the bandwidth utilization of the current processing task. When $U_{bw} \le 10\%$, it will be charged at 0.08/Kb. When $U_{bw} > 10\%$, the first 5Kb will be charged at 0.063/Kb, after that it will be charged at 0.25/Kb. $Task_{bw}$ indicates the total bandwidth size used by the task.

For the cloud task scheduling problem, the user requires that the appropriate VM resource scheduling scheme can be selected by consuming the least cost in the shortest time. Thus, the objective function of the cloud task scheduling problem to be solved in this paper is as follows:

$$\begin{cases} \min T_{total} = Max(VM^{i}_{Time}) \\ \min Cost = \sum_{i=1}^{M} sum(VM^{i}_{BW\_Cost}) \end{cases} \tag{3}$$

4. Proposed Method

In this section, we first present the idea of an NSGA-III-based algorithm for solving optimization problems. Second, we propose an MFO-based NSGA-III and describe the general framework of the proposed MFO_NSGA-III algorithm. Finally, we discuss the algorithm to solve the multi-objective cloud task scheduling model constructed above.

4.1 NSGA-III for multi-objective cloud task scheduling problems

The NSGA-III is a reference point-based non-dominated ranking algorithm [50]. It is one of the representatives of multi-objective optimization algorithms, and NSGA-III has been used in various fields for solving optimization problems [51-53] because of its excellent capability in solving high-dimensional multi-objective problems due to the introduction of the reference point mechanism.

Algorithm 1. Pseudocode of NSGA-III

Input: Population size *N;* objective function set: $F(x)=\{f_1(x), f_2(x),\ldots, f_m(x)\}$; number of objective functions: *M;* maximum number of iterations: *MaxIt.*
Output: A set of optimal solutions.
An initial population *P* of size *N* is randomly generated in the search space;
Initialize a set of reference points *Z* of dimension *M*;
For each $i \in N$ **do**
  Evaluate the objective value of individual p_i on $F(x)$;
End for
While the *MaxIt* number of iterations is not reached **do**
  For each $i \in N/2$ **do**
    $Parent_i$, $Parent_{i+1}$ →Tournament selection of parent individuals from *P*;
    $Offspring_i$, $Offspring_{i+1}$ → Crossover and Mutation for $Parent_i$, $Parent_{i+1}$;
    Evaluate the objective value of $Offspring_i$ and $Offspring_{i+1}$ on $F(x)$;
  End for
  Initpopulation = $P \cup Offspring$;
  Select *N* optimal individuals from *Initpopulation* by an environment selection strategy based on reference point *Z*;
  Z→Update the reference point values on each dimension by the objective value of *Initpopulation*;
End while

The flow of the NSGA-III is illustrated in Algorithm 1. The core of the algorithm lies in the reference point mechanism. A set of reference points Z is first obtained by solving the target dimension of the optimization problem. EA includes operators such as matching, environment selection, mutation operation, and crossover operation. The previous generation of the NSGA-II uses the crowding distance environment selection approach to get a new population that satisfies the conditions by the environmental selection of the merged populations, which leads to the problem of insufficient selection pressure of the algorithm as

the objective dimension of the optimization problem rises. NSGA-III provides a new environment selection mechanism for the algorithm by considering the Euclidean distance between reference points and individuals to select individuals. And the reference points are updated based on the objective values of the population, thus accelerating the evolution of the algorithm toward the true Pareto front.

4.2 The overall Framework

The cloud task scheduling problem usually considers the required attributes of user upload tasks, such as task length, file upload size, and file output size. Due to the continuous development of the cloud environment, the increasing demand for cloud computing from individual users and enterprises has led to the expansion of cloud task scheduling into a large-scale optimization problem, and the configuration of VM resources to satisfy the demand for a large number of tasks. EAs tend to consume more computational resources and computation time to obtain a satisfactory set of solutions, which leads to EA being subject to certain limitations. As a result, traditional EA is no longer able to optimize the requirements for solving large-scale optimization problems. We decompose the large-scale cloud task scheduling problem into multiple subproblems and redescribe it as an MTO problem. For a minimization MTO problem, it is described as follows:

$$\begin{aligned} &\min F(x) = \min T_1(x_1) \cup \min T_2(x_2) \cup ... \cup \min T_k(x_k) \\ &s.t. \qquad x = \{x_1, x_2, ..., x_k\} \\ &\qquad T_i(x_i) = \{ f_1(x_i),\ f_2(x_i),\ \ldots ,\ f_m(x_i)\} \end{aligned} \tag{4}$$

Where T represents the optimization task in the MTO problem, k represents the number of optimization tasks, f_i represents the objective function of each optimization task, $i = 1, 2, ..., m$, and m represents the number of objective functions.

The overall framework of our proposed method is shown in Algorithm 2. The MFO strategy is combined with the NSGA-III. The overall flow of the proposed algorithm is shown in **Fig. 4**. After obtaining the initialized population, the optimization problem currently solved needs to be decomposed into multiple subtasks. Assume that there are m cloud tasks and n VMs in the cloud environment. Determine the number of tasks m_i for the $i-th$ cloud task scheduling task. Then the number of cloud task scheduling tasks k is calculated as follows:

$$k = \lfloor m / m_i \rfloor \tag{5}$$

The purpose of the MFO strategy is to optimize multiple problems simultaneously, and this approach can effectively reduce the loss of computational resources and accelerates the convergence of the algorithm. This is because traditional EA has the disadvantage of wasting effective evolutionary information due to re-iterations when solving multiple optimization problems. The empowerment algorithm achieves effective utilization of evolution in evolution by dividing the population into multiple subpopulations and transferring knowledge among them. MFO satisfies multiple optimization problems simultaneously by dividing the population generated by the algorithm into multiple subpopulations and the subpopulations co-evolve with each other using knowledge transfer. In line 11 we use the knowledge transfer parameter random mating probability (RMP) in MFO that controls the transfer. $Parent_i^j$

$Parent_i^{j+1}$ represent the $i-th$ individual from subpopulation j and subpopulation $j+1$, respectively, and when the generated random numbers satisfy the constraints of the RMP parameters, implicit knowledge transfer will be achieved through crossover operations. During the evolutionary process, crossover operations produce offspring individuals that can inherit superior gene loci from parent individuals, resulting in higher-quality individuals. Therefore, individuals generated by using implicit knowledge transfer will have a greater likelihood of performing well on different tasks. If the RMP parameters are not satisfied, the subpopulations of different tasks will maintain an independent evolutionary process to ensure that the subpopulations meet the requirements of the tasks. It is worth noting that we are applying the MFO technique to NSGA-III to provide the algorithm with the ability to handle the MTO problem, while the goal of this paper is still to optimize an optimization problem, so the individual evaluations between different subpopulations still use the same set of objective functions, which does not cause additional computational resources to be wasted.

Algorithm 2. Pseudocode of MFO_NSGA-III

Input: Population size *N;* objective function set: $F(x)=\{f_1(x), f_1(x),\ldots, f_m(x)\}$; number of objective functions: *M;* maximum number of iterations: *MaxIt;* transfer probability: *RMP;* number of tasks: *k*.
Output: A set of optimal solutions.
An initial population *P* of size *N* is randomly generated in the search space;
Initialize a set of reference points *Z* of dimension *M*;
For each $i \in N$ **do**
  Evaluate the objective value of individual p_i on *F(x)*;
End for
Subpopulation^j is obtained by dividing the multitasking according to Eq. 5, *j=1,2,...,k;*
While the *MaxIt* number of iterations is not reached **do**
  Generate random numbers *rand* $\in$ (*0, 1*)*;*
  For each $j \in k$ **do**
    Parent^j_i, *Parent*$^{j+1}_i$ →Tournament selection of parent individuals from *Subpopulation*^{j} and *Subpopulation*$^{j+1}$;
    Parent$^j_{i+1}$, *Parent*$^{j+1}_{i+1}$ →Tournament selection of parent individuals from *Subpopulation*^{j} and *Subpopulation*$^{j+1}$;
    If *rand*<RMP **do**
      Offspring^j_i, *Offspring*$^{j+1}_i$ → Crossover and Mutation for *Parent*^j_i, *Parent*$^{j+1}_i$;
      Offspring$^j_{i+1}$, *Offspring*$^{j+1}_{i+1}$ → Crossover and Mutation for *Parent*$^j_{i+1}$, *Parent*$^{j+1}_{i+1}$;
    Else
      Offspring^j_i, *Offspring*$^j_{i+1}$ → Crossover and Mutation for *Parent*^j_i, *Parent*$^j_{i+1}$;
      Offspring$^{j+1}_i$, *Offspring*$^{j+1}_{i+1}$ → Crossover and Mutation for *Parent*$^{j+1}_i$, *Parent*$^{j+1}_{i+1}$;
    End if
    Evaluate the objective value of *Offspring*^{j} and *Offspring*$^{j+1}$ on *F(x)*;
  End for
  Offspring = *Offspring*$^1 \cup$ *Offspring*$^2 \cup \ldots \cup$ *Offspring*^{k}*;*
  Initpopulation = *P* $\cup$ *Offspring*;
  Select *N* optimal individuals from *Initpopulation* by an environment selection strategy based on reference point *Z*;
  Z→Update the reference point values on each dimension by the objective value of *Initpopulation*;
End while

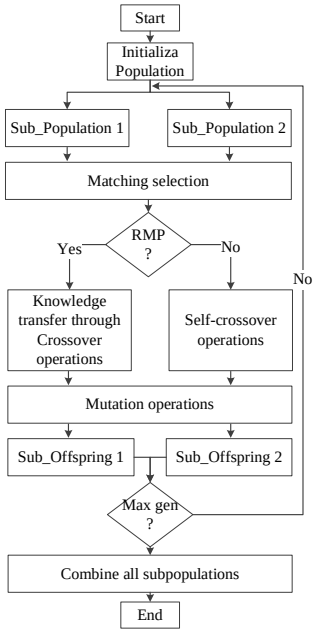

Fig. 4. The overall flow chart of the MFO_NSGA-III algorithm.

Next, this paper considers the solution process of the proposed method for a specific cloud task scheduling problem. Starting from the NSGA-III, when the initial population is generated, the tasks uploaded by the user have all been assigned to the corresponding VM. Individuals in the population each represent a set of feasible scheduling solutions, while the gene bits in this set of solutions represent the VM number to which the current task is scheduled. The optimization problem constructed in this paper needs to consider the task execution time and cost (Eq.1 and Eq.2) as an objective function. Due to the presence of virtualization technology in cloud services, it is possible to run each VM independently from the other, which leads to the fact that the solution can be divided into multiple groups of scheduling solutions of variable length. After dividing all the scheduling solutions using Eq.5, it is passed to the MFO technique where each group of divided scheduling solutions can be considered as a separate optimization problem. And these multiple optimization problems can be optimized simultaneously to find the best cloud task scheduling solution parallel using a common encoding between gene bits. **Fig. 5** represents the scheduling solution generated by the MFO-based NSGA-III algorithm for two optimization problems. First, the solution satisfying the initial decision space requirement is generated by the algorithm, assuming a total of two optimization problems are decomposed, and the decomposition strategy based on dimensionality reduction in the article will decompose this solution into decision variables satisfying optimization problem 1 and optimization problem 2, respectively.

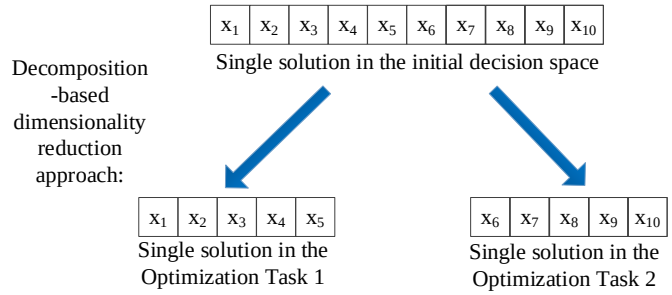

Fig. 5. Divide a complete set of scheduling solutions into two tasks.

5. Experiment

In this section, we first depict the test set utilized in this paper and the simulation setup; secondly, we describe the comparison algorithm used in this paper and the parameter settings. Finally, we analyze the experimental results.

5.1 Description of the simulation test set

The simulation data set for this paper is built by Cloudsim [54] Software, which is a Cloud simulation platform proposed by Grid Lab and the Gridbus project of the University of Melbourne, Australia. And use the PlatEMO platform on Matlab for comparative experiments. Ye Tian et al. [55] proposed a multi-objective optimization tool based on MATLAB, which consists of 50 multi-objective optimization algorithms and 110 multi-objective optimization test sets. The codes of all comparison algorithms used in this paper are integrated into the PlatEMO platform. The experiment was run with the Win11 system, the CPU is Intel(R) Core (TM) i7-12700H@2.8GHz with 16.0 GB of RAM.

In this paper, tasks, and VMs are set up using Clousim. **Table 1** and **Table 2** show the configuration of VMs and the required attributes of cloud tasks, respectively. Since this paper considers large-scale cloud task scheduling, cloud resources consist of 100 VM data sets with different parameter configurations. Meanwhile, taking the test set with 5000 tasks as an example, it is generated by 5000 cloud tasks with discrete and uniform distribution. Therefore, this paper is a random optimization process, and there is no optimal solution in the test data set. A black box test can be carried out. In **Table 2**, we gave a total of 6 test cases, including TS1 to TS6, and the number of tasks in the test set ranged from 5000 to 10000.

Table 1. Parameter settings for virtual machine resources

ID	Mips	Size	Ram	Bw	CPU (core)	ID	Mips	Size	Ram	Bw	CPU (core)
1	300	10240	1024	1536	1	51	1000	10240	3072	3072	6
2	300	10240	1024	1536	1	52	1000	10240	3072	3072	6
3	300	10240	1024	1536	1	53	1000	10240	3072	3072	6
4	300	10240	1024	1536	1	54	1000	10240	3072	3072	6
5	300	10240	1024	1536	1	55	1000	10240	3072	3072	6
6	500	10240	2048	1536	1	56	1500	10240	4096	3072	6
7	500	10240	2048	1536	1	57	1500	10240	4096	3072	6
8	500	10240	2048	1536	1	58	1500	10240	4096	3072	6
9	500	10240	2048	1536	1	59	1500	10240	4096	3072	6
10	500	10240	2048	1536	1	60	1500	10240	4096	3072	6
11	1000	10240	3072	1536	1	61	300	10240	1024	3072	12
12	1000	10240	3072	1536	1	62	300	10240	1024	3072	12
13	1000	10240	3072	1536	1	63	300	10240	1024	3072	12
14	1000	10240	3072	1536	1	64	300	10240	1024	3072	12
15	1000	10240	3072	1536	1	65	300	10240	1024	3072	12
16	1500	10240	4096	1536	1	66	500	10240	2048	3072	12
17	1500	10240	4096	1536	1	67	500	10240	2048	3072	12
18	1500	10240	4096	1536	1	68	500	10240	2048	3072	12

19	1500	10240	4096	1536	1	69	500	10240	2048	3072	12
20	1500	10240	4096	1536	1	70	500	10240	2048	3072	12
21	300	10240	1024	1536	2	71	1000	10240	3072	3072	12
22	300	10240	1024	1536	2	72	1000	10240	3072	3072	12
23	300	10240	1024	1536	2	73	1000	10240	3072	3072	12
24	300	10240	1024	1536	2	74	1000	10240	3072	3072	12
25	300	10240	1024	1536	2	75	1000	10240	3072	3072	12
26	500	10240	2048	2048	2	76	1500	10240	4096	6144	12
27	500	10240	2048	2048	2	77	1500	10240	4096	6144	12
28	500	10240	2048	2048	2	78	1500	10240	4096	6144	12
29	500	10240	2048	2048	2	79	1500	10240	4096	6144	12
30	500	10240	2048	2048	2	80	1500	10240	4096	6144	12
31	1000	10240	3072	2048	2	81	300	10240	1024	6144	16
32	1000	10240	3072	2048	2	82	300	10240	1024	6144	16
33	1000	10240	3072	2048	2	83	300	10240	1024	6144	16
34	1000	10240	3072	2048	2	84	300	10240	1024	6144	16
35	1000	10240	3072	2048	2	85	300	10240	1024	6144	16
36	1500	10240	4096	2048	2	86	500	10240	2048	6144	16
37	1500	10240	4096	2048	2	87	500	10240	2048	6144	16
38	1500	10240	4096	2048	2	88	500	10240	2048	6144	16
39	1500	10240	4096	2048	2	89	500	10240	2048	6144	16
40	1500	10240	4096	2048	2	90	500	10240	2048	6144	16
41	300	10240	1024	2048	6	91	1000	10240	3072	6144	16
42	300	10240	1024	2048	6	92	1000	10240	3072	6144	16
43	300	10240	1024	2048	6	93	1000	10240	3072	6144	16
44	300	10240	1024	2048	6	94	1000	10240	3072	6144	16
45	300	10240	1024	2048	6	95	1000	10240	3072	6144	16
46	500	10240	2048	2048	6	96	1500	10240	4096	6144	16
47	500	10240	2048	2048	6	97	1500	10240	4096	6144	16
48	500	10240	2048	2048	6	98	1500	10240	4096	6144	16
49	500	10240	2048	2048	6	99	1500	10240	4096	6144	16
50	500	10240	2048	2048	6	100	1500	10240	4096	6144	16

Table 2. Parameter settings for Cloud Task Test Suites

	Number of tasks	Length	Filesize	Outputsize	CPU(core)
TS1	5000	1000~10000	100~1000	200~2000	1
TS2	6000	1000~20000	100~2000	200~4000	1
TS3	7000	1000~30000	100~3000	200~6000	1
TS4	8000	1000~40000	100~4000	200~8000	1
TS5	9000	1000~50000	100~5000	200~10000	1
TS6	10000	1000~60000	100~6000	200~12000	1

5.2 Comparison of algorithms and experimental setup

The comparison algorithms selected in this paper include advanced evolutionary algorithms including BiGE [56], GrEA [57], KnEA [58], MOEAD/M2M [59], NSGA-III [50], and VaEA [60]. The parameters of these comparison algorithms were set based on the PlatEMO platform as shown in **Table 3**. All algorithms were executed based on the GA operator, the probability of crossover $proC$ and probability of mutation $proM$ were set to 1, the population size N was set to 100, and the maximum number of iterations $MaxIt$ was set to 10000. All algorithms were executed independently 30 times and the experimental results were obtained.

Table 3. Comparison algorithm parameter setting

ALGORITHMS	Parameters settings
MFO_NSGA-III	Rmp=0.3
GrEA	Div=45
KnEA	Rate = 0.5
MOEAD-M2M	K = 10

5.3 Simulation experiment results and analysis

The experimental results of all compared algorithms on the proposed test sets of large-scale cloud tasks are shown in **Table 4** and **Table 5**. This paper addresses a multi-objective optimization problem that is featured by obtaining an optimal set of Pareto solutions. The

experimental results for all test cases are compared in **Table 4** and **Table 5**. The experimental results in the table are calculated from Eq. 1 and Eq. 2. The optimal results have been marked in bold. The experimental results display that our proposed method MFO_NSGA-III obtains the optimal results regarding task execution time and task execution cost on all simulation test sets such as TS1 to TS6, while other comparison algorithms do not achieve satisfactory results in the optimization process. This is because the processed datasets are all large-scale optimization problems, and the conventional EA is constrained in the process of going for the optimal solution in the ultra-high dimensional search space, thus the problem of not being able to find the optimal solution region arises. MFO_NSGA-III algorithm firstly decomposes the large-scale problem by reducing the dimensionality; secondly, the introduced MFO strategy effectively uses the knowledge transfer in the evolutionary process to optimize the decomposed multiple sub-problems simultaneously; and finally, it also shows significant superior performance in the experimental results.

Table 4. Experimental results of task execution time for different numbers of cloud task test sets

Data set number	Number of tasks	MFO_NSGA-III	BiGE	GrEA	KnEA	MOEAD/M2M	NSGA-III	VaEA
TS1	5000	**5.74E+04**	1.34E+06	2.70E+09	6.75E+08	2.80E+15	2.70E+09	1.85E+12
TS2	6000	**1.79E+07**	1.28E+11	8.93E+07	7.24E+12	3.32E+15	3.63E+13	7.25E+12
TS3	7000	**2.22E+05**	8.16E+08	8.16E+08	1.96E+11	3.18E+15	2.37E+13	1.96E+10
TS4	8000	**1.47E+06**	1.89E+09	5.73E+07	3.19E+13	1.45E+15	5.74E+07	1.56E+13
TS5	9000	**2.61E+06**	1.53E+13	1.20E+08	1.20E+09	1.27E+15	4.09E+10	1.20E+09
TS6	100000	**3.85E+05**	1.20E+12	2.50E+12	1.27E+13	4.48E+15	5.55E+13	2.37E+13

Table 5. Experimental results of task execution cost for different numbers of cloud task test sets

Data set number	Number of tasks	MFO_NSGA-III	BiGE	GrEA	KnEA	MOEAD/M2M	NSGA-III	VaEA
TS1	5000	**7.64E+04**	4.88E+06	8.10E+07	2.54E+09	1.43E+16	1.07E+10	3.84E+12
TS2	6000	**5.36E+05**	4.01E+09	2.69E+06	2.17E+11	5.29E+16	1.09E+12	4.94E+11
TS3	7000	**4.44E+05**	2.45E+07	3.53E+09	5.88E+09	5.41E+16	7.11E+11	4.08E+10
TS4	8000	**4.41E+04**	7.37E+09	2.13E+08	9.57E+11	2.28E+16	1.88E+06	4.68E+11
TS5	9000	**7.83E+04**	4.60E+11	3.60E+06	4.78E+09	1.75E+16	1.38E+11	4.85E+09
TS6	100000	**1.45E+06**	3.59E+10	7.50E+10	3.80E+11	8.00E+16	1.67E+12	7.12E+11

We analyze the convergence of all algorithms on the objective function results using the TS1 test set with 5000 tasks as an example. The results in the other test sets are similar to those of TS1, so only TS1 is presented in this section. In this paper, the maximum number of iterations of the algorithm is set to 100, and the difference between each result is 5 generations. The convergence curves of the MFO_NSGA-III algorithm and the comparison algorithm concerning time and cost objective values are given in **Fig. 6** and **Fig. 7**. We can see from the figure that our algorithm converges by the 3rd (15th) iteration and finds a better region of optimal solutions than the other algorithms. In the early iterations, the MFO_NSGA-III algorithm did not find a good initial population and was in line with the performance of most of the algorithms, because the initial population generation was still too randomized, and a better way of generating the initial population for the problem was not found. However, the inclusion of the MFO technique in the iterations reduces the computational burden of the algorithm, which allows it to find a better range of locally optimal solutions and converge more quickly than other algorithms.

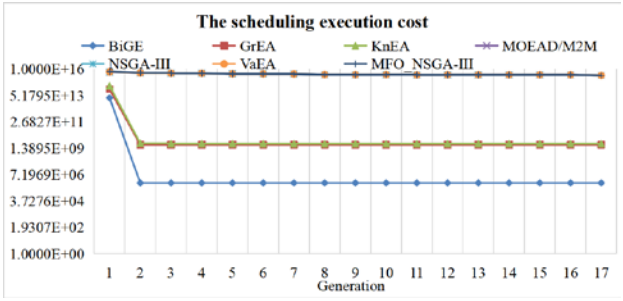

Fig. 6. Convergence plots of all algorithms in terms of task execution time for TS1.

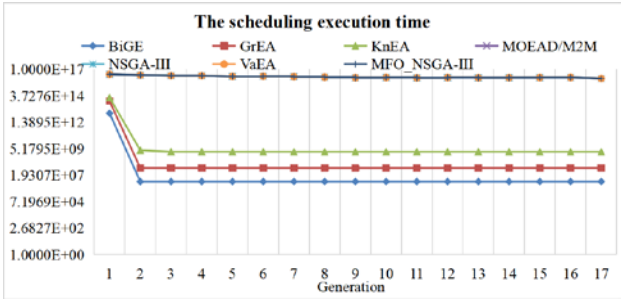

Fig. 7. Convergence plots of all algorithms in terms of task execution cost for TS1.

The histograms of the results of the MFO_NSGA-III algorithm versus all algorithms are displayed in **Fig. 8** and **Fig. 9**. As can be seen from the figure, the proposed method outperforms much well than the traditional EA algorithm on all large-scale test sets TS1 to TS6. The above experimental results prove that the MFO_NSGA-III algorithm has good performance and good competitiveness in dealing with large-scale cloud task scheduling problems. Therefore, the method can better deal with real-life large-scale problems.

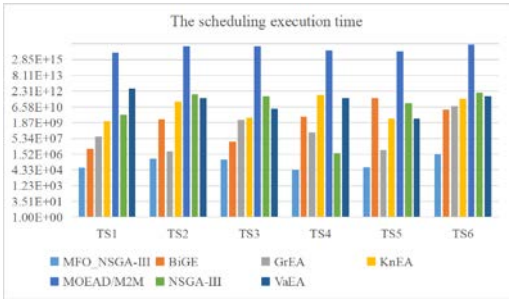

Fig. 8. The scheduling execution time for all comparison algorithms in TS1~TS6.

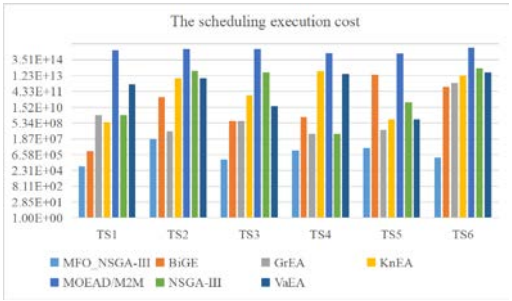

Fig. 9. The scheduling execution cost for all comparison algorithms in TS1~TS6.

5.4 Performance of MFO_NSGA-III on Benchmark

To further demonstrate the effectiveness of the proposed algorithm, this chapter uses a classical multi-objective benchmark test function set for experimental comparison with other optimization algorithms. The test sets used include DTLZ, BT, UF, and ZDT: 1) the DTLZ test benchmark suite has seven functions, of which D=12 has five functions (DTLZ2~DTLZ6); 2) the BT test benchmark suite has nine functions, where D is all 30; 3) the UF test benchmark suite has ten functions, where the dimensions are all D=30. 4) the ZDT test benchmark suite has five functions, where D=30 has three functions (ZDT1~ZDT3), and D=10 has two functions (ZDT4 and ZDT6); the characteristics of these four Benchmarks are given in **Table 6**.

Table 6. Summary of Benchmark test suites

Benchmark test suite	Features
DTLZ [61]	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler
BT [62]	Benchmark MOP with bias feature
UF [63]	Unconstrained benchmark MOP
ZDT [64]	Benchmark MOP proposed by Zitzler, Deb, and Thiele

The following performance metrics are used in the experiments:

Inverted Generational Distance (IGD) Comprehensive Metric can reflect both convergence and distribution. It is by calculating the minimum distance sum between each individual on the True Pareto front surface and the set of individuals obtained by the algorithm. This indicator is calculated as follows:

$$IGD(POF^*,\ POF)=\frac{1}{n}\sum_{p^*\in POF^*}\min_{p\in POF}\| p^*-p\|^2 \tag{6}$$

Where POF^* is the True Pareto Front of a given multi-objective optimization problem, POF is an approximation set obtained by an MOEA and n is the number of individuals in the POF^*.

The comparison algorithms selected for this chapter are MOEAD-M2M, MOPSO, RVEA, and MOEAD. All algorithm settings are set according to the PlatEMO platform. A summary of the algorithm parameter settings is shown in **Table 7**.

Table 7. Comparison algorithm parameter setting in Benchmark experiments

ALGORITHMS	Parameters settings
MOEAD-M2M	K = 10
MOPSO	Div = 10
RVEA	Alpha = 2; Fr = 0.1
MOEA/D	Type = 1

The results of the mean IGD, and standard deviation for all compared algorithms on the four benchmark test suites are given in **Table 8**. The best results in the tables are marked in bold. In **Table 8** we use the Wilcoxon test to verify the difference between the algorithm and other algorithms in terms of experimental results. The Wilcoxon test with a degree of confidence of 95% was used to assess the significance of the differences between the compared algorithms. In the experiments "+", "-" and "=" represent that the performance of the comparison algorithm is better than, weaker than, and equal to the performance of MFO_NSGA-III, respectively.

Table 8. IGD values for algorithms on all Benchmark suites

Problem	M	D	MOEADM2M	MOPSO	RVEA	MOEAD	MFO_NSGAIII
DTLZ1	3	10	**4.3540e+2(6.02e+1) +**	1.6844e+3(5.91e+2) -	1.2066e+3(8.39e+1) +	5.9097e+2(7.57e+1) +	1.2696e+3(8.47e+1)
DTLZ2	3	12	1.5678e-1 (7.08e-3) -	1.0461e-1 (1.17e-2) -	5.5957e-2 (7.22e-4) =	**5.4905e-2 (2.21e-4) +**	5.5706e-2 (2.46e-4)
DTLZ3	3	12	7.2548e+1 (2.27e+1) -	1.4069e+2 (7.14e+1) -	1.6862e+1 (5.95e+0) +	**1.3629e+1 (9.57e+0) +**	3.0813e+1 (8.81e+0)
DTLZ4	3	12	1.2679e-1 (1.43e-2) +	3.7647e-1 (1.47e-1) -	**5.5822e-2 (6.44e-4) +**	4.9289e-1 (3.11e-1) -	1.3705e-1 (1.84e-1)
DTLZ5	3	12	4.5267e-2 (1.15e-2) -	1.2866e-2 (2.49e-3) =	8.2559e-2 (1.21e-2) -	3.2277e-2 (9.22e-4) -	**1.2389e-2 (1.20e-3)**
DTLZ6	3	12	1.0371e+0 (4.81e-1) -	2.8940e+0 (8.42e-1) -	1.3291e-1 (1.41e-1) -	8.5980e-2 (2.00e-1) -	**4.7994e-2 (1.12e-1)**
DTLZ7	3	22	1.0497e+0 (4.79e-1) -	4.4507e+0 (1.02e+0) -	2.1845e-1 (5.41e-2) -	1.6128e-1 (4.01e-2) -	**9.3510e-2 (5.35e-3)**
BT1	2	30	4.2417e+0 (9.18e-2) -	4.6226e+0 (2.03e-1) -	3.6211e+0 (1.15e-1) -	3.8342e+0 (1.09e-1) -	**3.3411e+0 (8.33e-2)**
BT2	2	30	2.2900e+0 (1.17e-1) -	3.1083e+0 (2.61e-1) -	1.1691e+0 (7.33e-2) -	1.2501e+0 (9.93e-2) -	**9.5510e-1 (3.01e-2)**
BT3	2	30	4.4297e+0 (1.04e-1) -	4.7694e+0 (1.98e-1) -	3.2017e+0 (2.34e-1) -	3.3369e+0 (2.18e-1) -	**2.9058e+0 (8.29e-2)**
BT4	2	30	4.2657e+0 (9.18e-2) -	4.6483e+0 (1.52e-1) -	3.1684e+0 (1.28e-1) -	3.4710e+0 (1.29e-1) -	**2.8450e+0 (9.63e-2)**
BT5	2	30	4.3409e+0 (1.07e-1) -	4.8035e+0 (1.90e-1) -	3.6488e+0 (1.34e-1) -	3.8035e+0 (1.30e-1) -	**3.3503e+0 (8.51e-2)**
BT6	2	30	1.1711e+0 (8.96e-1) -	3.1794e+0 (2.53e-1) -	5.3659e-1 (2.10e-1) =	9.0544e-1 (2.96e-1) -	**5.3410e-1 (1.30e-1)**
BT7	2	30	3.0867e+0 (3.26e-1) -	3.2042e+0 (3.79e-1) -	6.2299e-1 (2.12e-1) =	**5.6582e-1 (1.82e-1) =**	6.8995e-1 (3.18e-1)
BT8	2	30	5.0960e+0 (1.33e+0) -	7.5588e+0 (7.33e-1) -	**2.9279e+0 (4.18e-1) =**	4.5378e+0 (9.22e-1) -	2.9744e+0 (4.37e-1)
BT9	3	30	4.6362e+0 (2.77e-1) -	4.1359e+0 (1.88e-1) -	2.7118e+0 (1.14e-1) -	2.6276e+0 (3.39e-1) -	**2.5469e+0 (7.65e-2)**
UF1	2	30	2.5368e-1 (5.43e-2) -	6.4170e-1 (1.20e-1) -	1.5722e-1 (5.08e-2) -	3.5099e-1 (1.16e-1) -	**1.1918e-1 (2.79e-2)**
UF2	2	30	7.6017e-2 (7.34e-3) -	1.2682e-1 (2.14e-2) -	1.0803e-1 (1.13e-2) -	2.0261e-1 (7.00e-2) -	**6.9379e-2 (1.30e-2)**
UF3	2	30	3.7837e-1 (7.72e-2) =	5.5033e-1 (2.52e-2) -	4.3941e-1 (4.50e-2) -	**3.2801e-1 (1.87e-2) +**	3.8511e-1 (4.56e-2)
UF4	2	30	8.4697e-2 (5.40e-3) -	1.1004e-1 (1.27e-2) -	1.4825e-1 (7.39e-3) -	1.2801e-1 (5.32e-3) -	**7.9066e-2 (3.24e-3)**
UF5	2	30	2.0820e+0 (3.43e-1) -	3.3852e+0 (3.83e-1) -	7.7598e-1 (2.31e-1) -	1.3931e+0 (3.45e-1) -	**5.4200e-1 (1.62e-1)**
UF6	2	30	1.0239e+0 (1.91e-1) -	3.0453e+0 (5.15e-1) -	5.2628e-1 (6.11e-2) -	5.9843e-1 (2.25e-1) -	**4.3662e-1 (1.03e-1)**
UF7	2	30	2.6519e-1 (8.83e-2) -	7.0482e-1 (1.26e-1) -	2.6272e-1 (9.38e-2) -	4.7605e-1 (1.23e-1) -	**1.3042e-1 (1.18e-1)**
UF8	3	30	4.9139e-1 (9.18e-2) =	5.1401e-1 (5.71e-2) =	**3.7688e-1 (3.49e-2) +**	5.7882e-1 (2.37e-1) =	4.7848e-1 (5.01e-2)
UF9	3	30	5.9681e-1 (4.19e-2) -	6.0662e-1 (4.66e-2) -	4.2276e-1 (6.83e-2) =	5.4972e-1 (8.65e-2) -	**4.2018e-1 (7.87e-2)**
UF10	3	30	4.4743e+0 (6.39e-1) -	2.7125e+0 (3.25e-1) -	9.0492e-1 (2.60e-1) =	**7.3978e-1 (8.73e-2) =**	8.4915e-1 (2.89e-1)
ZDT1	2	30	9.4270e-2 (5.88e-2) -	9.4105e-1 (2.79e-1) -	1.3351e-1 (2.86e-2) -	1.6453e-1 (8.67e-2) -	**1.4612e-2 (2.14e-3)**
ZDT2	2	30	1.1695e-1 (9.18e-2) -	1.7714e+0 (3.77e-1) -	1.6027e-1 (2.37e-2) -	5.3213e-1 (8.49e-2) -	**2.1459e-2 (3.30e-3)**
ZDT3	2	30	2.3206e-1 (8.74e-2) -	1.1107e+0 (2.36e-1) -	1.6447e-1 (2.69e-2) -	1.5319e-1 (4.45e-2) -	**1.5227e-2 (5.68e-3)**
ZDT4	2	10	1.8534e+1 (5.76e+0) -	1.8643e+1 (8.52e+0) -	1.1143e+0 (4.05e-1) +	**4.9005e-1 (2.01e-1) +**	1.8215e+0 (6.58e-1)
ZDT6	2	10	**7.1544e-3 (1.47e-3) +**	6.8998e-1 (1.32e+0) =	3.4056e-1 (1.05e-1) -	8.0853e-2 (2.47e-2) +	2.8963e-1 (7.03e-2)
+/-/=			3/26/2	0/28/3	5/20/6	6/22/3	

The experimental results of IGD demonstrate that MFO_NSGA-III achieves a large number of best IGD results on the four benchmark test suites. Meanwhile, the MFO_NSGA-III algorithm obtained a total of 20 best results, while the performance of the algorithm was significantly better than the other algorithms compared. The above results demonstrate that MFO_NSGA-III is very competitive in solving the MOPs.

We first compare the experimental comparison of the objective function values obtained on the simulation test set, which aims to demonstrate that the proposed algorithm is able to obtain the best decision variables with limited computational resources, and that the MFO_NSGA-III algorithm is more practical when faced with large-scale optimization problems in real-world scenarios than other algorithms. Secondly, we plot the convergence of the objective function for the convergence performance of the algorithm. The purpose of this experiment is to demonstrate that the proposed algorithm can reach convergence and obtain the scheduling solution at a faster iteration rate, and the MFO_NSGA-III algorithm shows a better competitive performance compared with other algorithms. Finally, to further demonstrate the effectiveness of the MFO_NSGA-III algorithm, we perform an experimental comparison of the evaluation metrics in the proposed benchmark test function suite, which aims to demonstrate that the proposed algorithm is more effective in finding TruePOF, and that the MFO_NSGA-III algorithm shows its performance in several benchmark suites compared to other algorithms. The purpose of this experiment is to demonstrate that the proposed algorithm is more effective in finding TruePOF, and the MFO_NSGA-III algorithm shows its superior performance in several benchmark suites compared to other algorithms.

5. Conclusion

With information technology development, the complexity of tasks and heterogeneity of resources in cloud task scheduling problems are increasing, and large-scale optimization models can better describe the actual problems. In this paper, an MFO-based NSGA-III is proposed and applied to a large-scale scenario optimization problem of cloud task scheduling VM resources. We first construct a multi-objective cloud task scheduling model considering task execution time and cost. And recharacterize it as an MTO problem using a decomposition-based strategy. Second, the MFO strategy is combined with the NSGA-III and used to get the optimal scheduling solution for the large-scale cloud task scheduling problem. In the experimental part, experimental comparisons with other advanced multi-objective optimization algorithms are performed, and the results display that our proposed method obtained the optimal experimental results, demonstrates the effectiveness and competitive shape, and is more adapted for solving large-scale optimization problems than other EAs.

Although our proposed method has a good prospect of solving large-scale optimization problems. But how to generate a higher quality set of initial populations in a population-based algorithm is crucial to guide the entire evolutionary process. We need to take into account the characteristics of solving optimization problems to find the most suitable initial population, which is a good topic for the algorithm to search the region where the best solution is located. In future work, we will continue to drill down on MFO-based optimization algorithms for how to effectively use problem characteristics to generate populations and effectively handle higher dimensional optimization problems, and we design more problem-specific solution strategies and combine them with optimization algorithms to propose more targeted and practically meaningful solution algorithms.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No.61806138); National Natural Science Foundation of China (Grant No.61003053); National Key Research and Development Program of China (Grant No.YDZJSX2021A038); China University Industry-University-Research Collaborative Innovation Fund (Future Network Innovation Research and Application Project) (Grant 2021FNA04014); Postgraduate Joint Training Demonstration Base of Taiyuan University of Science and Technology Fund (Grant NO. JD2022003).

References

[1] M. L. Chiang, H. C. Hsieh, Y. H. Cheng, W. L. Lin, and B. H. Zeng, "Improvement of tasks scheduling algorithm based on load balancing candidate method under cloud computing environment," *Expert Syst. Appl.*, vol. 212, p. 19, 2023, Art no. 118714. Article (CrossRef Link)

[2] K. Mershad, H. Artail, M. A. R. Saghir, H. Hajj, and H. Hajj, "A Study of the Performance of a Cloud Datacenter Server," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 590-603, Oct-Dec 2017. Article (CrossRef Link)

[3] N. J. Navimipour and S. Asghari, "Cloud service composition using an inverted ant colony optimisation algorithm," *Int. J. Bio-Inspired Comput.*, vol. 13, no. 4, pp. 257-268, 2019. Article (CrossRef Link)

[4] P. Azad, N. J. Navimipour, and M. Hosseinzadeh, "A fuzzy-based method for task scheduling in the cloud environments using inverted ant colony optimisation algorithm," *Int. J. Bio-Inspired Comput.*. vol. 14, no. 2, pp. 125-137, 2019. Article (CrossRef Link)

[5] C. L. Li and L. Y. Li, "Optimal scheduling across public and private clouds in complex hybrid cloud environment," *Inf. Syst. Front.*, vol. 19, no. 1, pp. 1-12, Feb 2017. Article (CrossRef Link)
[6] X. R. Song, L. Pan, and S. J. Liu, "An online algorithm for optimally releasing multiple on-demand instances in IaaS clouds," *Futur. Gener. Comp. Syst.*, vol. 136, pp. 311-321, Nov 2022. Article (CrossRef Link)
[7] Z. X. Zhang, M. K. Zhao, H. Wang, Z. H. Cui, and W. S. Zhang, "An efficient interval many-objective evolutionary algorithm for cloud task scheduling problem under uncertainty," *Inf. Sci.*, vol. 583, pp. 56-72, Jan 2022. Article (CrossRef Link)
[8] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, and G. Wu, "A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1463-1474, 2022. Article (CrossRef Link)
[9] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE T. Cybern.*, vol. 51, no. 3, pp. 1430-1442, 2021. Article (CrossRef Link)
[10] S. Fatemi-Anaraki, R. Tavakkoli-Moghaddam, M. Foumani, and B. Vahedi-Nouri, "Scheduling of Multi-Robot Job Shop Systems in Dynamic Environments: Mixed-Integer Linear Programming and Constraint Programming Approaches," *Omega-Int. J. Manage. Sci.*, vol. 115, p. 15, Feb 2023, Art no. 102770. Article (CrossRef Link)
[11] J. Liu, P. Yang, and C. Chen, "Intelligent energy-efficient scheduling with ant colony techniques for heterogeneous edge computing," *J. Parallel Distrib. Comput.*, vol. 172, pp. 84-96, Feb 2023. Article (CrossRef Link)
[12] Z. Xue, W. Guo, Z. Cui, and W. Zhang, "The global evaluation strategy for many‐objective partial collaborative computation offloading problem," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 2, p. e7474, 2023. Article (CrossRef Link)
[13] Z. Cui, Z. Xue, T. Fan, X. Cai, and W. Zhang, "A Many-objective Evolutionary Algorithm Based on Constraints for Collaborative Computation Offloading," *Swarm and Evolutionary Computation*, vol. 77, p. 101244, 2023. Article (CrossRef Link)
[14] Y. K. Liu, Y. Y. Ping, L. Zhang, L. H. Wang, and X. Xu, "Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning," *Robot. Comput.-Integr. Manuf.*, vol. 80, p. 15, Apr 2023, Art no. 102454. Article (CrossRef Link)
[15] S. Tiwari et al., "A smart decision support system to diagnose arrhythymia using ensembled ConvNet and ConvNet-LSTM model," *Expert Syst. Appl.*, vol. 213, p. 13, Mar 2023, Art no. 118933. Article (CrossRef Link)
[16] X. J. Cai, S. J. Geng, D. Wu, J. H. Cai, and J. J. Chen, "A Multicloud-Model-Based Many-Objective Intelligent Algorithm for Efficient Task Scheduling in Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9645-9653, Jun 2021. Article (CrossRef Link)
[17] M. Abdel-Basset, R. Mohamed, W. Abd Elkhalik, M. Sharawi, and K. M. Sallam, "Task Scheduling Approach in Cloud Computing Environment Using Hybrid Differential Evolution," *Mathematics*, vol. 10, no. 21, p. 26, Nov 2022, Art no. 4049. Article (CrossRef Link)
[18] H. F. Li, D. J. Wang, M. C. Zhou, Y. S. Fan, and Y. Q. Xia, "Multi-Swarm Co-Evolution Based Hybrid Intelligent Optimization for Bi-Objective Multi-Workflow Scheduling in the Cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 9, pp. 2183-2197, Sept 2022. Article (CrossRef Link)
[19] X. W. Xia, H. X. Qiu, X. Xu, and Y. L. Zhang, "Multi-objective workflow scheduling based on genetic algorithm in cloud environment," *Inf. Sci.*, vol. 606, pp. 38-59, Aug 2022. Article (CrossRef Link)
[20] X. C. Wang, Z. W. Lyu, Z. C. Wei, L. L. Wang, Y. Lu, and L. Shi, "Multi-objective path planning algorithm for mobile charger in wireless rechargeable sensor networks," *Wirel. Netw.*, vol. 29, pp. 267-283, 2023. Article (CrossRef Link)
[21] X. H. Zhang, Y. Guo, J. Q. Yang, D. L. Li, Y. Wang, and R. Zhao, "Many-objective evolutionary algorithm based agricultural mobile robot route planning," *Comput. Electron. Agric.*, vol. 200, p. 9, Sep 2022, Art no. 107274. Article (CrossRef Link)

[22] W. Q. Zhang, H. R. Li, W. D. Yang, G. H. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm considering combination timing for multi-type vehicle routing problem with time windows," *Comput. Ind. Eng.*, vol. 171, p. 12, Sep 2022. Article (CrossRef Link)
[23] H. Jiang, M. X. Lu, Y. Tian, J. F. Qiu, and X. Y. Zhang, "An evolutionary algorithm for solving Capacitated Vehicle Routing Problems by using local information," *Appl. Soft. Comput.*, vol. 117, p. 16, Mar 2022, Art no. 108431. Article (CrossRef Link)
[24] M. Zhang, H. Q. Li, L. Liu, and R. Buyya, "An adaptive multi-objective evolutionary algorithm for constrained workflow scheduling in Clouds," *Distrib. Parallel Databases*, vol. 36, no. 2, pp. 339-368, Jun 2018. Article (CrossRef Link)
[25] T. P. Pham and T. Fahringer, "Evolutionary Multi-Objective Workflow Scheduling for Volatile Resources in the Cloud," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1780-1791, Jul-Sep 2022. Article (CrossRef Link)
[26] S. J. Geng, D. Wu, P. H. Wang, and X. J. Cai, "Many-Objective Cloud Task Scheduling," *IEEE Access*, vol. 8, pp. 79079-79088, 2020. Article (CrossRef Link)
[27] J. L. Xu, Z. X. Zhang, Z. M. Hu, L. Du, and X. J. Cai, "A many-objective optimized task allocation scheduling model in cloud computing," *Appl. Intell.*, vol. 51, no. 6, pp. 3293-3310, Jun 2021. Article (CrossRef Link)
[28] Y. Du, T. Wang, B. Xin, L. Wang, Y. Chen, and L. Xing, "A data-driven parallel scheduling approach for multiple agile earth observation satellites," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 679-693, 2020. Article (CrossRef Link)
[29] M. Belhor, A. El-Amraoui, A. Jemai, and F. Delmotte, "Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem," *Expert Syst. Appl.*, vol. 213, p. 15, Mar 2023, Art no. 119035. Article (CrossRef Link)
[30] S. Mangalampalli, G. R. Karri, and M. Kumar, "Multi objective task scheduling algorithm in cloud computing using grey wolf optimization," *Cluster Comput., Early Access*, p. 20, 2022. Article (CrossRef Link)
[31] L. Imene, S. Sihem, K. Okba, and B. Mohamed, "A third generation genetic algorithm NSGAIII for task scheduling in cloud computing," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7515-7529, Oct 2022. Article (CrossRef Link)
[32] Z. H. Peng, P. Pirozmand, M. Motevalli, and A. Esmaeili, "Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework," *Math. Probl. Eng.*, vol. 2022, p. 11, Sep 2022, Art no. 4290382. Article (CrossRef Link)
[33] M. I. Alghamdi, "Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO)," *Sustainability*, vol. 14, no. 19, p. 20, Oct 2022, Art no. 11982. Article (CrossRef Link)
[34] B. M. H. Zade and N. Mansouri, "Improved red fox optimizer with fuzzy theory and game theory for task scheduling in cloud environment," *J. Comput. Sci.*, vol. 63, p. 37, Sep 2022, Art no. 101805. Article (CrossRef Link)
[35] J. L. Li et al., "Multiobjective Oriented Task Scheduling in Heterogeneous Mobile Edge Computing Networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8955-8966, Aug 2022. Article (CrossRef Link)
[36] S. Qin, D. C. Pi, Z. S. Shao, and Y. Xu, "A Cluster-Based Cooperative Co-Evolutionary Algorithm for Multiobjective Workflow Scheduling in a Cloud Environment," *IEEE Trans. Autom. Sci. Eng., Early Access*, p. 15, 2022. Article (CrossRef Link)
[37] H. Emami, "Cloud task scheduling using enhanced sunflower optimization algorithm," *ICT Express*, vol. 8, no. 1, pp. 97-100, Mar 2022. Article (CrossRef Link)
[38] Q. L. Dang and J. W. Yuan, "A Kalman filter-based prediction strategy for multiobjective multitasking optimization," *Expert Syst. Appl.*, vol. 213, p. 11, Mar 2023, Art no. 119025. Article (CrossRef Link)
[39] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial Evolution: Toward Evolutionary Multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343-357, Jun 2016. Article (CrossRef Link)

[40] J. N. Chen, S. Y. Chen, Q. Wang, B. Cao, G. Feng, and J. H. Hu, "iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7011-7024, Aug 2019. Article (CrossRef Link)
[41] J. J. Mayl, S. E. Vaala, P. V. Patel, M. B. Ritter, and K. M. Richardson, "Media Multitasking in Medical Students: A Theory-Based Approach to Understanding this Behavior," *Teach. Learn. Med.*, *Early Access*, p. 12, 2022. Article (CrossRef Link)
[42] D. J. Wang, Y. G. Yu, Y. Q. Yin, and T. C. E. Cheng, "Multi-agent scheduling problems under multitasking," *Int. J. Prod. Res.*, vol. 59, no. 12, pp. 3633-3663, Jun 2021. Article (CrossRef Link)
[43] Y. Zhou, T. Wang, and X. Peng, "MFEA-IG: A Multi-Task Algorithm for Mobile Agents Path Planning," in *Proc. of 2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020.
[44] J. Yi, J. R. Bai, H. B. He, W. Zhou, and L. Z. Yao, "A Multifactorial Evolutionary Algorithm for Multitasking Under Interval Uncertainties," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 908-922, Oct 2020. Article (CrossRef Link)
[45] K. K. Bali, A. Gupta, Y. S. Ong, and P. S. Tan, "Cognizant Multitasking in Multiobjective Multifactorial Evolution: MO-MFEA-II," *IEEE T. Cybern.*, vol. 51, no. 4, pp. 1784-1796, Apr 2021. Article (CrossRef Link)
[46] F. F. Zhang, Y. Mei, S. Nguyen, and M. J. Zhang, "Multitask Multiobjective Genetic Programming for Automated Scheduling Heuristic Learning in Dynamic Flexible Job-Shop Scheduling," *IEEE T. Cybern.*, *Early Access*, pp. 1-14, 2022. Article (CrossRef Link)
[47] A. Rauniyar, R. Nath, and P. K. Muhuri, "Multi-factorial evolutionary algorithm based novel solution approach for multi-objective pollution-routing problem," *Comput. Ind. Eng.*, vol. 130, pp. 757-771, Apr 2019. Article (CrossRef Link)
[48] J. W. Liu, P. L. Li, G. B. Wang, Y. X. Zha, J. C. Peng, and G. Xu, "A Multitasking Electric Power Dispatch Approach With Multi-Objective Multifactorial Optimization Algorithm," *IEEE Access*, vol. 8, pp. 155902-155911, 2020. Article (CrossRef Link)
[49] Z. Liang, J. Zhang, L. Feng, and Z. Zhu, "Multi-factorial optimization for large-scale virtual machine placement in cloud computing," *arXiv preprint arXiv:2001.06585*, 2020. Article (CrossRef Link)
[50] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577-601, Aug 2014. Article (CrossRef Link)
[51] Q. Wang, Z. H. Cui, and L. F. Wang, "Charging path optimization for wireless rechargeable sensor network," *Peer Peer Netw. Appl.*, vol. 14, no. 2, pp. 497-506, Mar 2021. Article (CrossRef Link)
[52] Y. Y. Li et al., "Multi-objective optimization of the Atkinson cycle gasoline engine using NSGA III coupled with support vector machine and back-propagation algorithm," *Energy*, vol. 262, Jan 2023, Art no. 125262. Article (CrossRef Link)
[53] F. Xue and D. Wu, "NSGA-III algorithm with maximum ranking strategy for many-objective optimisation," *Int. J. Bio-Inspired Comput*, vol. 15, no. 1, pp. 14-23, 2020. Article (CrossRef Link)
[54] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software Practice & Experience*, vol. 41, no. 1, pp. 23-50, 2011. Article (CrossRef Link)
[55] Y. Tian, R. Cheng, X. Y. Zhang, and Y. C. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization," *IEEE Comput. Intell. Mag.*, Editorial Material vol. 12, no. 4, pp. 73-87, Nov 2017. Article (CrossRef Link)
[56] M. Q. Li, S. X. Yang, and X. H. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45-65, Nov 2015. Article (CrossRef Link)
[57] S. X. Yang, M. Q. Li, X. H. Liu, and J. H. Zheng, "A Grid-Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721-736, Oct 2013. Article (CrossRef Link)
[58] X. Y. Zhang, Y. Tian, and Y. C. Jin, "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761-776, Dec 2015. Article (CrossRef Link)

[59] H. L. Liu, F. Q. Gu, and Q. F. Zhang, "Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450-455, Jun 2014. Article (CrossRef Link)
[60] Y. Xiang, J. Peng, Y. R. Zhou, M. Q. Li, and Z. F. Chen, "An angle based constrained many-objective evolutionary algorithm," *Appl. Intell.*, vol. 47, no. 3, pp. 705-720, Oct 2017. Article (CrossRef Link)
[61] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, IEEE, vol. 1, pp. 825-830, 2002. Article (CrossRef Link).
[62] H. Li, Q. Zhang, and J. Deng, "Biased Multiobjective Optimization and Decomposition Algorithm," *IEEE T. Cybern.*, vol. 47, no. 1, pp. 52-66, 2016. Article (CrossRef Link)
[63] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, vol. 264, pp. 1-30, 2008.
[64] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000. Article (CrossRef Link)

Tianhao Zhao is currently working toward M.S. degree at computer science and technology, Taiyuan University of Science and Technology, Taiyuan, China. His research interests include computational intelligence, combinatorial optimization and multitasking optimization.

Linjie Wu is currently working toward M.S. degree at computer science and technology, Taiyuan University of Science and Technology, Taiyuan, China. His research interests include computational intelligence, combinatorial optimization and dynamic multi-objective optimization.

Di Wu is currently pursuing a Ph.D. degree in the college of computer science and technology at Beijing University of Technology, Beijing, China. Her research interests include many-objective algorithms and knowledge graph embedding.

Zhihua Cui received the Ph.D. degree in control theory and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008. He is currently a professor with the School of Computer Science and Technology, Taiyuan University of Science and Technology, China. He is the editor-in-chief of the International Journal of Bio-inspired Computation. His research interests include computational intelligence, stochastic algorithm, and combinatorial optimization.

Jianwei Li received the M.S. degree from Taiyuan University of Science and Technology, Taiyuan, China, in 2003. He is currently an associate professor with the School of Computer Science and Technology, Taiyuan University of Science and Technology, China.