International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 4758-4772

https://doi.org/10.70517/ijhsa463407

Fuzzy clustering algorithm based multi-objective optimization problem solving strategy in large-scale computing environment

Bojun Liu[1,*]
[1] Faculty of Engineering, University of Sydney, Sydney, Australia
Corresponding authors: (e-mail: xfyjrLBJ0628@163.com).

Abstract With the increase of the number of decision variables in the optimization problem, the problem becomes more and more complex, and when the number of decision variables exceeds a certain value, it is called a large-scale multi-objective optimization problem, and it is difficult for traditional algorithms to satisfy the user's needs. To this end, the fuzzy clustering algorithm is first used to cluster the population particles, and the particle with the highest non-dominated rank is selected as the class globally optimal particle from each class by the non-dominated sorting method, and the class globally optimal solution is applied to the speed updating formula of the multi-objective particle swarm algorithm, which in turn completes the design of the multi-objective optimization problem solution strategy based on the fuzzy clustering-particle swarm combination algorithm, and carries out a Numerical simulation analysis. The IGD values of this paper's algorithm for the test functions are higher than those of FDEA-II, which outperforms this paper's algorithm only for three test functions, and the IGD values of RSEA (0.001) are better than those of this paper's algorithm (0.0037) for MaF2 with M=5, which proves the practicality of the fuzzy clustering-particle swarm combination algorithm based on fuzzy clustering-particle swarm combination algorithm for the solution strategy in the multi-objective optimization problem reference value.

Index Terms fuzzy clustering, particle swarm, multi-objective optimization, solution strategy

I. Introduction

As the human society enters a new data-intensive era, the scale of corresponding optimization problems gradually expands, such as large-scale network community detection, large-scale path planning, large-scale signal analysis, etc [1]. As a result, research oriented towards multi-objective optimization problems in large-scale computing environments has received extensive attention from both industry and academia [2], [3]. In order to promote the development and communication of large-scale computational optimization, the Conference on Evolutionary Computation (CEC) has set up a large-scale optimization competition unit since 2008 and continuously updated the set of test functions [4]. In 2015, Goh, S. K et al [5] set up an Evolutionary Large-Scale Optimization Competition unit aiming to apply evolutionary computation to deal with real-world large-scale data optimization problems. For large-scale optimization problems, the search space grows exponentially with the increase in the dimensionality of the decision variables, and it is difficult for traditional evolutionary algorithms to traverse the entire search space in a limited time frame [6]. The interrelationships between variables become more and more complex with the increase in the dimensionality of decision variables, which leads to the traditional optimization algorithms easily falling into local optimality [7], [8]. In addition, various complex large-scale optimization problems further exacerbate the solution difficulty, such as large-scale computationally expensive problems and large-scale multi-objective problems [9]. Therefore, how to efficiently solve complex large-scale optimization problems within a limited time frame is the focus and difficulty of current research.

Unlike large-scale single-objective optimization, large-scale multi-objective optimization (LSMOPs) problems are extremely challenging, and this research area is still in the early stages of exploration [10]. Multi-objective optimization algorithms based on Pareto domination are a classical class of algorithms for efficiently solving multi-objective optimization problems [11]. However, when the number of objective functions increases, the selection pressure in the environment selection is too small, and the comparability between individuals becomes poor, making it difficult to select the surviving high-quality individuals [12]. Currently, the methods for solving LSMOPs can be categorized into three main categories classes, namely, decision variable decomposition methods, decision space and objective space approximate reduction methods, and enhanced search strategies [13].

Decision-variable decomposition is based on the “divide-and-conquer” framework, where the decision variables are randomly or heuristically divided into multiple subproblems, and each subproblem is independently evolved for optimization [14]. The original decision variable grouping strategy was borrowed from large-scale single-objective optimization problems such as stochastic grouping and difference grouping. This decomposition method focuses on dividing the decision variables in the decision space and ignores the population diversity in the objective space [15], [16]. Later, the control characteristics of decision variables were analyzed and classified into positional variables, distance variables, and mixture variables based on the relationship between the decision variables and the objective function [17]. In the real world, few decision variables can satisfy the strict conditions of location and distance variables because the objective function is too complex [18]. Therefore, Zhang, X et al [19] used a decision variable clustering approach to classify decision variables more generally into convergence-related and diversity-related variables. Ma, L et al [20] proposed an adaptive local decision variable analysis method for the LSMOPs problem, which combines a reference vector bootstrap and an adaptive decision variable optimization strategy. He, C et al [21] proposed an efficient decision variable analysis method based on reformulation, and based on this, proposed a large-scale multi-objective evolutionary algorithm which showed good performance on LSMOPs problems with up to 2000 decision variables. However, the decision variable grouping method needs to consume a large number of evaluation models and increase time consumption [22]. In addition, some LSMOPs problems are inherently undecomposable, and incorrect decomposition can lead to failure to find an optimal solution [23].

Approximate reduction methods can be categorized into decision space approximation and target space approximation [24]. Decision space approximate reduction can be performed by shortening the decision vectors of the parent individuals and generating the child individuals, and then restoring the decision vectors of the child individuals to the original decision space for function evaluation [25]. Therefore, the algorithm only needs to find the optimal value of a short vector without searching in a high-dimensional decision space. Yao, X et al [26] proposed a knowledge-guided evolutionary algorithm based on data degradation and solution to solve the LSMOPs problem, which approximately reduces the target space by a clustering integration method. Zhang, K et al [27] proposed a new multiple swarm based differential evolution (LSMaODE) algorithm for solving the LSMOPs problem and demonstrated the advantages of this algorithm on benchmarking problems. Cartis, C et al [28] proposed a stochastic subspace algorithmic framework X-REGO for lipschitz-continuous objective global optimization and analyzed its convergence using quadratic integral geometry tools. Although spatial approximation can reduce the challenges posed by large-scale and multi-objective, there is an inevitable loss of diversity in the approximation process, and a certain degree of limitation in the flexibility of the method.

Enhanced search strategy to directly solve LSMOPs by proposing new child generation strategies in the original decision space, including new recombination operators and probabilistic models [29]. Zhang, Y et al [30] proposed a multi-objective cooperative particle swarm optimization (MOCPSO) algorithm with a dual search strategy and outperformed the existing state-of-the-art large-scale multi-objective evolutionary algorithms in most of the tested instances. Kropp, I et al [31] proposed a new set of evolutionary operators, including vssp, S-SBX and S-PM, and combined them with the NSGA-II algorithm to effectively solve the sparsity problem of LSMOPs. Although the enhanced search strategy solves some limitations of the decision variable decomposition and approximate reduction methods, the enhanced search strategy consumes a large amount of computational resources during operation. Meanwhile, the real-life large-scale optimization problems tend to be more and more complex, so the establishment of large-scale optimization algorithms for efficiently solving the large-scale optimization problems with different characteristics is one of the main research objectives of this paper.

In this paper, based on the clustering matrix, the initial clustering center is selected, and then the distance from each sample to each clustering center is calculated, and each sample is dispatched to the cluster where its nearest clustering center is located, and the process has been repeated until the condition is satisfied, which in turn results in the particle with the highest level of selected nondominance in each class. Incorporate this particle into the parameter updating formula of the multi-objective particle swarm algorithm, and finally design a multi-objective optimization problem solving strategy based on the fuzzy clustering-particle swarm combination algorithm, and verify the validity of this paper's research scheme from the aspects of the test function, test function.

II. Multi-objective optimization problem solving based on fuzzy clustering algorithm

II. A. Multi-objective optimization

II. A. 1) Concepts related to multi-objective optimization

Multi-objective optimization problems (MOPs) are more complex than single-objective extremum problems and also more widespread in the real world [32], [33]. In the vast majority of cases, improving one sub-objective might lead to a reduction in the performance of other objectives. It can be seen that it is basically impossible to make all the objectives reach the best expected value at the same time, so it is necessary to make a balanced performance as

well as a compromise between the objectives to make each single objective reach the best expected value as much as possible, on the other hand, the number of optimal solutions composed of the optimal solution is relatively large, or even an infinite number of Pareto-optimal solutions composed of the optimal solution, the said Pareto-optimal solution refers to all the schemes in the The Pareto optimal solution refers to the best one among all the solutions. Therefore, the essence of optimization is to seek an optimal solution so that the desired indicators of each objective can reach the optimum. For all the research objectives, the performance of each individual in the set of Pareto optimal solutions cannot be compared with its good or bad performance. The decision variables in MOP realize the existence of constraints among the objectives, and optimization of one objective will sacrifice the performance of other objectives, besides, the generality of each computational unit for the objective is different, so it is difficult to evaluate the goodness and characteristics of the solutions. Up to now, many optimization algorithms have been presented to solve multi-objective optimization problems, which are basically dealt with in a static mode, and the typical approach takes the steps of constraints, weighting, very large and very small, and goal planning to solve the optimal value. Due to the ease of understanding, implementation and high efficiency of fuzzy clustering algorithms, they have gained popularity among experts and scholars in recent years, and it is of great practical value to use fuzzy clustering algorithms as a method for solving multi-objective optimization problems.

II. A. 2) Mathematical model of a multi-objective optimization problem

In the vast majority of cases, improving one sub-goal may result in a reduction in the performance of other goals. It can be seen that it is basically impossible to make all the objectives reach the best expectation at the same time, so it is necessary to do a balancing of their performance as well as a trade-off between the objectives, so that each single objective can reach the best expectation as much as possible. Therefore, most multi-objective optimization problems are NP difficult problems, and linear programming and other single-objective optimization problems there are differences between the complex relationship between the various optimization objectives, usually if you get a vector of optimal values of the various sub-objective functions composed of the ideal solution is impossible, only to seek the relative optimal solution. However, the relative optimal solution is often not unique, in order to standardize the selection of the relative optimal solution, it is necessary to first give a general description of the mathematical model of MOP. Let $x \in R^n$, $f(x)$, $g_i(x) \le 0(\mathrm{i}=1,2,\cdots,p)$, $h_i(x)=0(i=p+1,\cdots,q)$.

For the given n metafunction, then in the usual case, the multi-objective optimization problem poses the constraint of finding the vector x such that the function $f(x)$ takes, within $g_i(x) \le 0(i=1,2,\cdots,p)$ and $h_i(x)=0,(i=p+1,\cdots,q)$, the extremely small or extremely large values. Here $f(x)$ is called the objective function, $g_i(x) \le 0$ is called the inequality constraint, $h_i=0$ is called the inequality constraint, and $x=\left(x_1,x_2,x_3\cdots,x_n\right)^T$ is called the design or decision variable. The optimization problem is abbreviated as:

$$\begin{cases} \min f(x) \\ s.tg_i(x) \le 0, i=1,2,\cdots,p \\ h_i(x)=0, i=p+1,p+2,\cdots,q \end{cases} \tag{1}$$

Solving an optimization problem is also known as solving the global optimal solution. That is, under a set of constraints $g_i(x) \le 0\left(i=1,2,\cdots,p\right)$, $h_i(x)=0\left(i=p+1,\cdots,q\right)$ so that the objective function $f(x)$ achieves a great value, or sometimes a very small value. where $x=\left(x_1,x_2,x_3\cdots,x_n\right)^T$ is a n-dimensional decision vector, where $f(x)$ is called the objective function, $g_i(x) \le 0$ is called the inequality constraint, and $h_i=0$ the inequality constraints.

A few important definitions are given below.

Definition 1 (feasible solution): for some $x \in X$, if x satisfies the constraints $g_i(x) \le 0\left(i=1,2,\cdots,p\right)$ in Eq. (1) and $h_i(x)=0\left(i=p+1,\cdots,q\right)$, then x is said to be a feasible solution.

Definition 2 (Set of feasible solutions): the set consisting of all valid solutions in x is labeled as the set of feasible solutions, labeled X_f, and $X_f \in \mathrm{X}$.

Definition 3 (Pareto): Assume that $x_1,x_2 \in X_f$ is a feasible solution to the above MOP problem, and say that x_1 Pareto dominates x_2 (denoted as $x_1 < x_2$) if and only if $\forall i=1,2,\cdots,m$: $f_i\left(x_1\right) \le f_i\left(x_2\right) \wedge \exists j=1,2,\cdots,m$: $f_i\left(x_1\right) < f_i\left(x_2\right)$ holds.

Definition 4 (Pareto optimal solution): Assume that $x^* \in X_f$ and no other solution $\overline{x^*} \in X_f$ exists such that $f_i\left(\overline{x^*}\right) \le f_i\left(x^*\right)\left(i=1,2,\cdots m\right)$ holds and there is at least one strict inequality, then it becomes that x^* is a Pareto optimal solution to (1).

Definition 5 (Pareto Optimal Solution Set): The Pareto Optimal Solution Set (PS) is the set of all Pareto optimal solutions, i.e., $PS=\left\{x^*\right\}=\left\{x \in X_f \mid \neg\exists x' \in X_f : f_i\left(x'\right) \le f_i x^*, i=1,2,\cdots,m\right\}$.

Definition 6 (Pareto Frontier): the Pareto frontier is the set of value Pareto optimal solutions mapped into the target space, i.e., $PF=\left\{F(x) \mid x \in PS\right\}$.

Definition 7 (Non-inferior solution set of the current population NDS): let $Pop(t)$ be the t th generation population of the MOEA algorithm, and the individual $x^* \in Pop(t)$ be the non-inferior solution of the population, if and only if $\neg\exists x' \in Pop(t) : x' < x^*$. The set of all non-inferior solutions x^* in the current population of $Pop(t)$ is said to be the set of non-inferior solutions NDS, i.e., $NDS=\left\{x^*\right\}=\left\{x \mid x \in Pop(t)\right.$ and $\left.\neg\exists x' \in \mathrm{Pop}(t)\right.$ such that $\left.x' < x\right\}$.

In the above equation (1), if the objective function $F(x)=\left(f_1(x), f_2(x), \cdots, f_m(x)\right)^T$, $m \ge 2$, that is, $F(x)$ is the objective vector function of x, is shown in the following equation (2):

$$\begin{cases} \min F(x)=\min\left(f_1(x), f_2(x), \cdots, f_m(x)\right) \\ s.\mathrm{tg}_i(x) \le 0, i=1,2,\cdots,p \\ h_i(x)=0, i=p+1, p+2, \cdots, q \end{cases} \tag{2}$$

Then Eq. (3) is called a multi-objective optimization problem.

Generally, in order to make the realization of multi-objective optimization problem more convenient, each sub-objective function optimal solution will be uniformly turned into the minimum value of the optimal or the maximum value of the optimal. For example, the conversion of maximization to minimization can be used as shown in Eq. (3), and the conversion of minimization to maximization can be deduced in the same way. For:

$$\max(f(x))=-\min\left(f_i(X)\right) \tag{3}$$

Similarly, for the constraints of a multi-objective optimization problem, a similar idea can be used to convert them if they need to be unified, taking Eq. (1) as an example, they can be converted as shown in Eq. (4) below:

$$-g(X) \le 0\left(i=1,2,3\cdots,k\right) \tag{4}$$

By the above method, the constraints and sub-objective functions of the multi-objective optimization problem can be converted into a unified one, which is conducive to the simplicity of the process of solving the multi-objective optimization problem to find the relative optimal solution.

II. A. 3) Key theories of multi-objective optimization

For explicit multi-objective optimization algorithms, the composition of the non-dominated solution set requires scaling the size of the undistributed solution set in a certain way [34]. However, the maintenance of the distributivity of the undominated solution set is determined by the properties of the multi-objective optimization algorithm. Next, one of the construction methods of Pareto optimal solution sets is briefly described. That is, the construction of Pareto optimal solution sets.

The Pareto solution set construction method based on dominance ordering sets up a construction set p' when the population to be evolved is p. At the initial run of the program, the first individual within the population is added to the construction set p', and then each individual p in the population p to be evolved is taken out in turn and added to the construction set p', and at the same time, p is compared with all the individuals in p' to delete p' with all individuals dominated by p in p', and removing p from p' if the individual p is dominated by any of the individuals in p'. Until all individuals in p are put into p', at which point p' is the non-dominated set of p.

II. A. 4) Test functions

The superiority of the fuzzy clustering algorithm is confirmed through the comparison test of each algorithm on the test function, in order to ensure the effectiveness of the algorithm and no loss of generality, the number of iterations in the execution of the algorithm is selected according to the complexity of the function, and based on this, several tests are carried out in order to ensure that the results of the test are not accidental. Researchers and scholars

generally compare and evaluate the performance of various intelligent optimization algorithms, because the optimization experiments on real problems may not be conditionally permissible, so they use a specific test function instead of the fuzzy clustering algorithm in the study of the test function used is oriented to multi-objective optimization. The Pareto front of MOP contains continuous or dispersed, concave and convex, uniformly distributed or unevenly hooked and other characteristics.

II. B.Application of Fuzzy Clustering Algorithm in Multiobjective Optimization

II. B. 1) Fuzzy Clustering Algorithm

Cluster analysis is a mathematical and scientific method to study the “similarity” between objects. The principle of cluster analysis is to use mathematical methods to quantitatively describe the closeness of the relationship between multiple objects, according to the size of the relationship between the closeness of the type of division, so as to achieve the effect of clustering. The method of cluster analysis is a multivariate analysis method based on mathematical and statistical theory.

There are many algorithms for clustering, and in various clustering algorithms, the size of the difference between the characteristics of the data objects is usually calibrated by the distance as a criterion for division, which is based on the distance from the object to a center in space. Cluster analysis official application data mining, pattern recognition field, etc..

The traditional cluster analysis gives a very strong feeling, just according to the qualities of each object tough division, strictly in accordance with the object of each study of the qualities (as an indicator), if the object under study has the qualities, that is, the division of the class, such as no, is discarded, or either one or the other, the division of boundaries in accordance with the indicators are clearly distinguishable. However, in reality, in many practical situations, almost all objects do not have absolutely strict attributes, or most of the distinction between the object attributes is not so strict extreme, there are generally intermediate states, not suitable for hard division, but suitable for soft division.

Soft division, that is, to deal with clustering problems with fuzzy clustering criteria, that is, fuzzy clustering method. Fuzzy cluster analysis method, analyze the degree of uncertainty of each sample for each type, fully consider the multi-attribute intermediate state of each sample, establish a descriptive model of uncertainty of each sample for each category $u(n,m)$, n is the number of individual samples, m is the number of models.

The dataset vector $X=[x_1,x_2,\cdots,x_n]$, n is the number of samples, aggregated into m categories. The fuzzy clustering matrix $U=[u_{jf}]_{m\times n}$, u_{jf} represents the degree of agreement of the altered samples f on j^{th}. Usually, the highest degree of agreement determines which category the sample should belong to. To wit:

$$C_f = j,\ where\ \max\{u_{jf}\}; j=1,2,\cdots m,\ for\ each\ f \tag{5}$$

Different fuzzy clustering algorithms will behave differently, and these distance matrix-based clustering algorithms are the most familiar. For example, fuzzy C-means (FCM). These algorithms use an optimization criterion J_b that divides individuals based on similarity. For:

$$J_b(U,v)=\sum_{f=1}^{n}\sum_{j=1}^{m}(u_{jf})^b(d_{jf})^2 \tag{6}$$

d_{jf} is a distance function that estimates the similarity property, measured by the individual-to-cluster center vector $v=[v_1,v_2,\cdots,v_n]$:

$$d_{jf}=(x_f-v_j)^T H_j(x_f-v_j) \tag{7}$$

In the FCM clustering method, $H_j=1$, and in the GKM clustering method, H_j is defined by the following equation:

$$H_j=\left[\delta_j \det(F_j)\right]^{\frac{1}{n}}(F_j)^{-1} \tag{8}$$

The FCM and GKM clustering algorithms are iterative processes that cluster n individuals into $m(1<m<n)$ classes, and m is chosen by based on user needs. The classification prototypes are defined randomly and are changed during the iterative process. Finally, the fuzzy dataset reaches equilibrium, i.e., the end of the iteration, which is:

$$\left\|U_t - U_{t-1}\right\| \le \varepsilon, \varepsilon \text{ is the termination tolerance} \tag{9}$$

The fuzzy equivalence matrix, $\overline{R}$, can be programmed by Matlab or C language tools to implement the computation of the passing closure to obtain the clustering result of Δ. First follow the absolute value subtraction method:

$$r_{i,j} = \begin{cases} 1 & (i=j) \\ 1 - p\sum_{k=1}^{m}\left|\Delta_{ik} - \Delta_{jk}\right| & (i \ne j) \end{cases} \left(i, j = 1, 2, \cdots, n\right) \tag{10}$$

A suitable value of p should be chosen such that $0 \le r_{kl} \le 1$, to establish a similarity matrix between Δ_k and Δ_l and then to find the fuzzy similarity matrix R:

$$R = (r_{kl})_{n \times n} \tag{11}$$

The squared self-synthesis method of the similarity matrix R can be solved to pass the closure $t(R)$ to obtain the equivalence matrix $\overline{R} = t(R)$. Given the confidence level value $\lambda \in [0,1]$, truncating the matrix by λ completes the dynamic clustering.

II. B. 2) Class global optimal particles based on fuzzy clustering

Fuzzy clustering algorithm, is a simple to operate distance-based clustering algorithm [35], [36]. Distance is used as an evaluation index of similarity, i.e., it is considered that the closer the distance between two objects, the more similar they are. The clusters in this algorithm are composed of samples that are close to each other, as the ultimate goal of fuzzy clustering is to obtain compact and independent clusters. Fuzzy clustering algorithm is a cluster analysis algorithm with iterative solution. The steps are to first select the initial cluster centers based on the clustering matrix, then calculate the distance from each sample to each cluster center and assign each sample to the cluster where its nearest cluster center is located. The cluster centers and the samples assigned to them represent a class. For each assigned individual, the cluster center of the class is recalculated based on the existing samples in the class. This process is repeated until some termination condition is met, at which point the samples cannot (or minimally) be reassigned to different clusters, the cluster centers cannot (or minimally amount) be changed, and the sum of squares of the errors is locally minimized. After clustering is complete, the particle with the highest nondominance rank in each class is selected as the class globally optimal particle.

II. B. 3) Particle update method based on class global optimal solution

In this section, the class global optimal solution obtained based on the fuzzy clustering algorithm is applied to the velocity update formula of the multi-objective particle swarm algorithm to improve the particle swarm algorithm, and the specific implementation process and steps are as follows:

Step 1: Use the fuzzy clustering algorithm to cluster and group the particles in the population, and non-dominated sort the particles in each class.

Step 2: A particle with the highest non-dominated rank in each class is randomly selected as the guiding individual to guide the update of all other individuals in the class, and the improved particle velocity and position update formula is as follows:

$$v_j^i(t+1) = \mu \cdot v_j^i(t) + \tau_1 r_1 (pb_j(t) - p_j^i(t)) + \tau_2 r_2 (pC^i(t) - p_j^i(t)) \tag{12}$$

$$p_j^i(t+1) = p_j^i(t) + v_j^i(t+1) \tag{13}$$

where $pb_j(t)$ is the personal history optimal solution, $pC^i(t)$ is the class global optimal solution, μ is the inertia weight, and τ_1, τ_2 is the acceleration coefficient.

Step 3: If the number of iterations is greater than the maximum number of iterations, output the optimal solution. Otherwise, go to step 2.

II. B. 4) Parameter adaptive tuning

The update of velocity in a multi-objective particle swarm is affected by three components, the current velocity, the individual optimal position and the global optimal position. Each of the three components has corresponding coefficients μ, $\tau_1 r_1$, $\tau_2 r_2$, where μ is the inertia weight, r_1, r_2 are two random numbers in the range of $[0,1]$ and τ_1, τ_2 are the acceleration coefficients.

(1) Inertia weights adaptive adjustment

The inertia weight controls the moving step of the particle, the larger the moving step is, the faster the particle moves, in the iterative process of particle evolution, the particle is constantly close to the Pareto optimal solution, so the inertia weight should be constantly reduced, in this paper, we make the inertia weight linearly decreasing from 0.9 to 0.1.

Let t be the number of iterations, $\max t$ be the maximum number of iterations, then the inertia weight decreases linearly from 0.9 to 0.1 can be expressed as the following formula:

$$\mu = -\frac{4}{5} \cdot \max t \cdot t + 0.9 \tag{14}$$

(2) Adaptive adjustment of acceleration coefficient

Acceleration coefficient in the multi-objective particle swarm algorithm controls the step size of the particle to move to the direction of the individual's historical optimal position, the global optimal position direction. Based on the improved multi-objective particle swarm algorithm in this paper, the parameter τ_1 controls the “individual cognition”, which promotes the particle to obtain the best position in the history, and is conducive to the development of the best solution in the localization. The parameter τ_2 controls the “social cognition”, which promotes the particle to converge to the class of globally optimal solutions, accelerates the speed of convergence of the particle, and balances the diversity of solutions and convergence.

In this paper, the value of τ_1 will be controlled according to the advantages and disadvantages of individual historical optimal position and global optimal position, and the value of τ_2 will be controlled according to the advantages and disadvantages of the class global optimal position compared with the individual historical optimal position, as well as the advantages and disadvantages of the class global optimal position compared with the global optimal position of the neighboring classes.

II. B. 5) Overall framework

Clustering Global Particle Selection and Parameter Adaptation, after the improvement of the above two sections, this section proposes the fuzzy clustering-multiobjective particle swarm algorithm called large-scale multiobjective particle swarm algorithm based on fuzzy clustering algorithm, denoted as SAMOPSO.

First, we use the fuzzy clustering algorithm in Section 2.2.1 to classify all the particles and select a class global bootstrap particle for each class by non-dominated sorting method, and then use the parameter adaptive method in Section 2.2.2 to select different parameters for different particles' performance. Finally, the updated formula is utilized to guide the particles to approach the Pareto front continuously. The specific steps of the algorithm are as follows:

Step 1: Initialize the population, including the velocity and position of the population particles.

Step 2: Cluster the particles using the fuzzy clustering algorithm proposed in Section 2.2.1 and perform a non-dominated sorting to select the class global optimal solution to generate the velocity update formulas and position update formulas.

Step 3: Adjust the inertia weights and acceleration coefficients using the parameter adaptive method proposed in Section 2.2.4.

Step 4: Update the velocity and position of the particle using the formulas.

Step 5: If the termination condition is satisfied, output the optimal solution. Otherwise, go to step 2.

III. Multi-objective optimization problem solving analysis

III. A. Test function analysis

III. A. 1) Parameter setting

To have evaluated the performance of the algorithm in this paper, I compared the algorithm in this paper with five other algorithms on two different sets of test functions (DTLZ and MaF). All experiments were conducted based on the PlatEMO platform. In the experiments, the number of targets M is set to 3, 5, 10, and 15. The corresponding population sizes N are set to 100, 150, 200, 250, and 300. The maximum number of evaluations (maxFEs) of the functions are set to 30,000, 50,000, 100,000, 150,000, and 200,000, respectively. Each algorithm is run for 30 times on each test instance. The final results are averaged. The parameters of all the above comparison algorithms were set to default parameters.

III. A. 2) Comparison Algorithms and Evaluation Indicators

The performance metrics chosen in this paper are Inverse Generation Distance (IGD) and Hyper Volume (HV).Smaller value of IGD means better convergence and diversity of the algorithm. Higher value of HV means better overall performance of the algorithm.HV indicates the distribution and convergence of the solution while HV

strictly follows the Pareto dominance relation. To verify the performance of the algorithm in this paper, it is compared with five other algorithms. The compared algorithms are listed below:

(1) RSEA: an evolutionary algorithm based on radial space partitioning.

(2) FDEA-II: a high-dimensional multi-objective evolutionary algorithm based on fractional dominance relations and improved goal space decomposition strategies.

(3) KnEA: an inflection point-based evolutionary algorithm.

(4) TS-NSGA-II: a two-stage evolutionary algorithm based on NSGA-II.

(5) Two_Arch2: an improved double archiving algorithm.

III. A. 3) Analysis of experimental results

In order to evaluate the performance of various algorithms of this paper on the test set, two widely accepted performance metrics IGD and HV are used. Table 1 lists the HV values of six multi-objective optimization solving algorithms on the DTLZ test set. From the results, RSEA performs better than MTaEA on five test functions but MTaEA outperforms RSEA on 21 test functions.FDEA-II, KnEA and Two_Arch2 outperform this paper's algorithm on only two test functions. The algorithm of this paper outperforms them on 32, 29 and 26 test functions respectively. TS-NSGA-II obtains better results than this paper's algorithm on DTLZ2 and DTLZ7 at M=10 and 15.

Table 1: The HV value on the DTLZ test set

Problem	M	RSEA	FDEA-II	KnEA	TS-NSGA-II	Two_Arch2	Ours
DTLZ1	3	0.591	0.538	0.325	0.446	0.051	0.856
	5	0.259	0.626	0.647	0.706	0.085	0.676
	10	0.618	0.025	0.343	0.111	0.231	0.871
	15	0.714	0.321	0.357	0.428	0.12	0.953
DTLZ2	3	0.056	0.25	0.244	0.553	0.273	0.728
	5	0.524	0.447	0.205	0.449	0.44	0.791
	10	0.341	0.248	0.58	0.175	0.716	0.706
	15	0.245	0.232	0.667	0.098	0.848	0.716
DTLZ3	3	0.468	0.229	0.657	0.645	0.663	0.649
	5	0.322	0.615	0.503	0.328	0.224	0.717
	10	0.314	0.536	0.288	0.688	0.616	0.926
	15	0.631	0.235	0.552	0.65	0.304	0.686
DTLZ4	3	0.195	0.481	0.275	0.54	0.419	0.839
	5	0.104	0.266	0.64	0.475	0.54	0.979
	10	0.434	0.472	0.147	0.709	0.558	0.975
	15	0.281	0.602	0.073	0.684	0.642	0.602
DTLZ5	3	0.385	0.534	0.692	0.115	0.714	0.706
	5	0.647	0.096	0.138	0.414	0.261	0.761
	10	0.065	0.054	0.709	0.689	0.707	0.735
	15	0.297	0.465	0.096	0.677	0.428	0.977
DTLZ6	3	0.374	0.304	0.438	0.623	0.408	0.629
	5	0.254	0.325	0.114	0.646	0.194	0.694
	10	0.148	0.099	0.307	0.42	0.486	0.966
	15	0.171	0.619	0.272	0.174	0.18	0.991
DTLZ7	3	0.164	0.285	0.066	0.591	0.278	0.791
	5	2.182	2.644	0.583	1.771	0.624	0.961
	10	7.635	3.411	2.531	6.433	9.496	8.742
	15	0.151	8.282	3.543	0.111	9.022	8.651

Table 2 gives the IGD values obtained by the six multi-objective optimization solution algorithms on the MaF test set. It can be seen that RSEA performs better than this paper's algorithm for MaF2 with M = 5. For the remaining 36 test functions, this paper's algorithm is not worse than RSEA. This paper's algorithm outperforms FDEA-II on 42 test functions, and FDEA-II outperforms this paper's algorithm only on three test functions. Compared with KnEA and TS-NSGA-II, this paper's algorithm still obtains better results on most problems.Two_Arch2 is not worse than this paper's algorithm on MaF2 and MaF5, but this paper's algorithm outperforms Two_Arch2 on 22 test functions.For MaF1, MaF4, and MaF7 for M=15 and 20, both FDEA-II and this paper's algorithm Both achieve good

performance. KnEA outperforms this paper's algorithm on MaF6, MaF8 and MaF9.TS-NSGA-II is not worse than this paper's algorithm on MaF4 and MaF5. For problems with a large number of targets, Two_Arch2 cannot obtain better results than the algorithm in this paper.

Table 2: The IGD value on the DTLZ test set

Problem	M	RSEA	FDEA-II	KnEA	TS-NSGA-II	Two_Arch2	Ours
MaF1	3	0.554	0.532	0.281	0.45	0.063	0.0029
	5	0.262	0.604	0.674	0.678	0.094	0.0059
	10	0.615	0.024	0.299	0.153	0.259	0.0052
	15	0.703	0.324	0.363	0.372	0.104	0.0017
MaF2	3	0.075	0.238	0.245	0.543	0.249	0.0021
	5	0.001	0.448	0.17	0.468	0.451	0.0037
	10	0.339	0.237	0.536	0.174	0.763	0.0024
	15	0.238	0.23	0.607	0.109	0.437	0.0031
MaF3	3	0.47	0.214	0.697	0.616	0.626	0.0014
	5	0.345	0.6	0.561	0.315	0.232	0.0011
	10	0.346	0.512	0.343	0.684	0.627	0.0013
	15	0.639	0.185	0.559	0.613	0.311	0.0028
MaF4	3	0.18	0.443	0.284	0.529	0.41	0.0046
	5	0.118	0.23	0.675	0.421	0.554	0.0057
	10	0.415	0.428	0.097	0.678	0.565	0.0055
	15	0.319	0.595	0.058	0.698	0.639	0.0058
MaF5	3	0.37	0.48	0.719	0.118	0.736	0.006
	5	0.709	0.08	0.153	0.366	0.236	0.0023
	10	0.021	0.025	0.706	0.705	0.742	0.0017
	15	0.342	0.48	0.057	0.643	0.399	0.0041
MaF6	3	0.374	0.316	0.435	0.633	0.428	0.0054
	5	0.27	0.327	0.115	0.669	0.214	0.0035
	10	0.194	0.126	0.296	0.451	0.493	0.0014
	15	0.133	0.62	0.283	0.18	0.204	0.0057
MaF7	3	0.191	0.317	0.082	0.581	0.259	0.0054
	5	2.224	2.635	0.578	1.819	0.658	0.0016
	10	0.508	1.212	0	0.148	0.021	0.004
	15	0.124	0.092	0.023	0.118	0.001	0.0049

Figure 1 shows the distribution of solutions obtained by the six algorithms on MaF8 (M=3), where (a) to (f) denote RSEA, FDEA-III, KnEA, TS-NSGA-11, Two_Arch2, and the algorithm of this paper, respectively. It can be seen that the solution distributions of FDEA-III, KnEA, TS-NSGA-11 and Two_Arch2 are very poor, but RSEA and the algorithm of this paper achieve better results.

III. B. Analysis of the solution strategy

III. B. 1) Comparison of the obtained solution set distributions

In order to visually analyze the performance of various strategies, this paper selects five relatively representative multi-objective optimization problems, FDA1, DMOP2, F5, F6 and F9, and plots the distribution of the solution sets obtained by the three strategies at different moments of solving different test problems, as shown in Fig. 2~Fig. 6, in which (a)~(c) are the algorithms of RSEA, TS-NSGA-11, and the algorithm of this paper, and the horizontal and vertical coordinates represent the values of the obtained solution sets on each objective, respectively. and vertical coordinates represent the values of the obtained solution sets on each objective, the magenta dots are the obtained solution sets, and the magenta curve is the Pareto optimal frontier Through the comparison, it is not difficult to draw the same conclusion as above, for the several changes of the large-scale computing environment in the beginning, the convergence and distribution of this paper's method are much better than that of RSEA and TS-NSGA-11, which indicates that this paper's method can respond to the changes of the large-scale computing environment faster and more accurately, while this paper's method can respond to the changes of the large-scale computing environment faster and more accurately. In the late stage of the algorithm, for the latter changes of the large-scale computing environment, the convergence and distribution of this paper's method are also better than those of RSEA and TS-NSGA-11, which indicates that this paper's method is faster and more accurate than RSEA and TS-NSGA-11, and

that the method is able to respond more accurately to the changes of the large-scale computing environment, and that the convergence and distribution of this paper's method for the latter changes of the large-scale computing environment in the late stage of the algorithm are also better than those of RSEA and TS-NSGA-11. The advantage of this paper's algorithm is even more intuitive in terms of its ability to deal with complex problems F9, as neither RSEA nor TS-NSGA-11 can obtain better convergence and distribution, while this paper's algorithm can track down new optimal solutions more accurately on certain large-scale changes in the computing environment, and obtains the set of Pareto optimal solutions that have better convergence and uniform distribution.

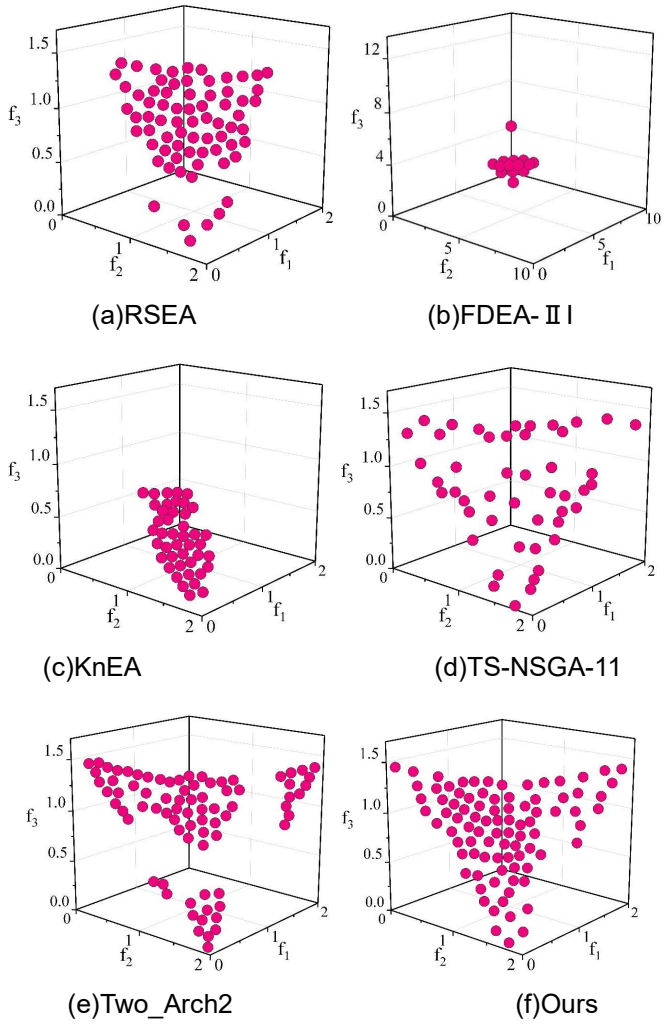
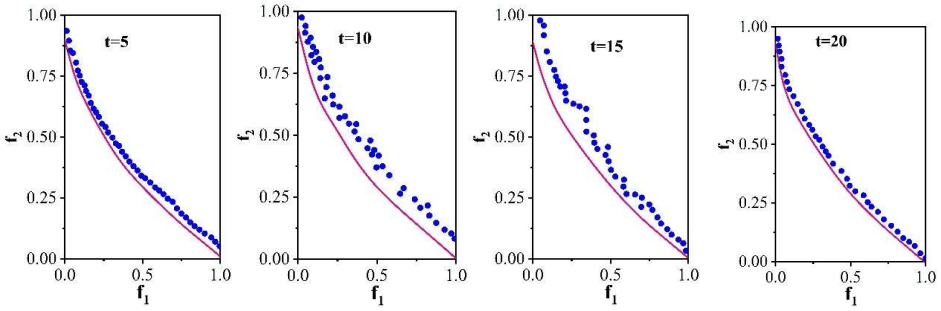
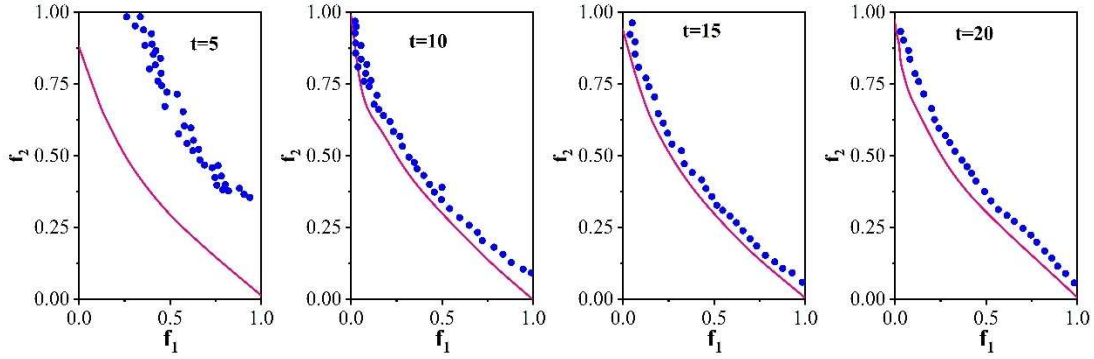


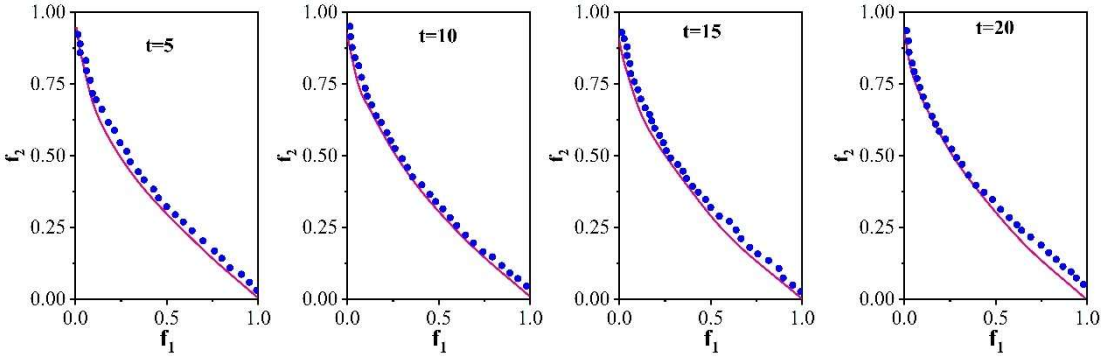
Figure 1: The distribution of solutions on the MaF8 function with three objectives



(a)RSEA

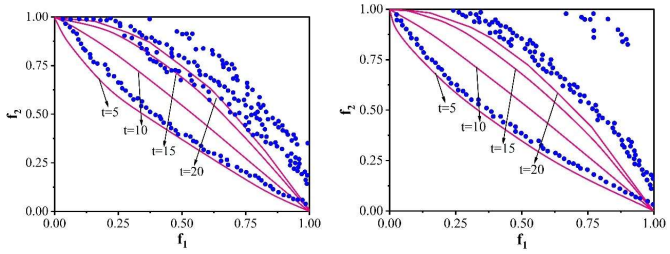


(b)TS-NSGA-11

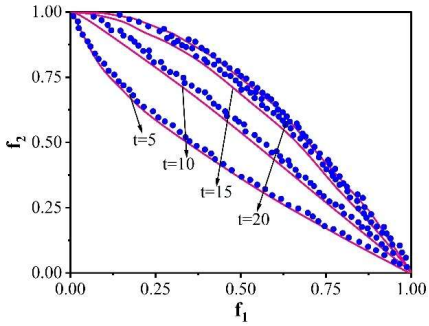


(c)Ours

Figure 2: Three strategies are used to solve the solution set of FDA1

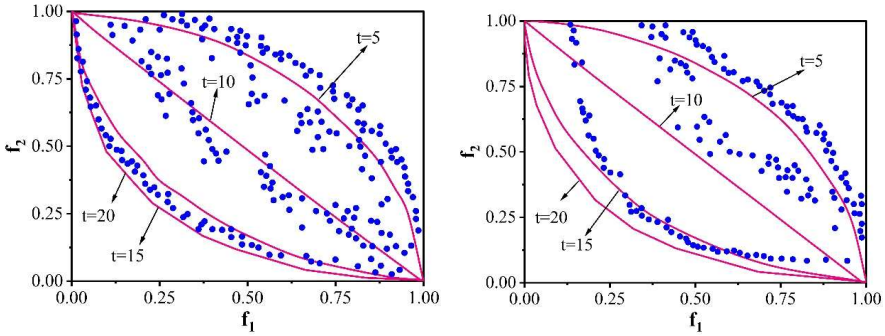


(a)RSEA (b)TS-NSGA-11

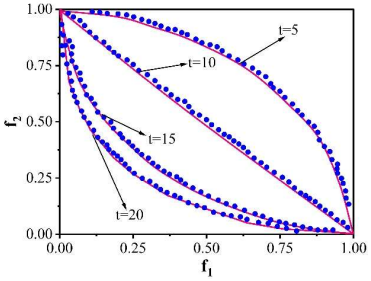


(c)Ours

Figure 3: The solution set obtained in the process of solving DMOP2

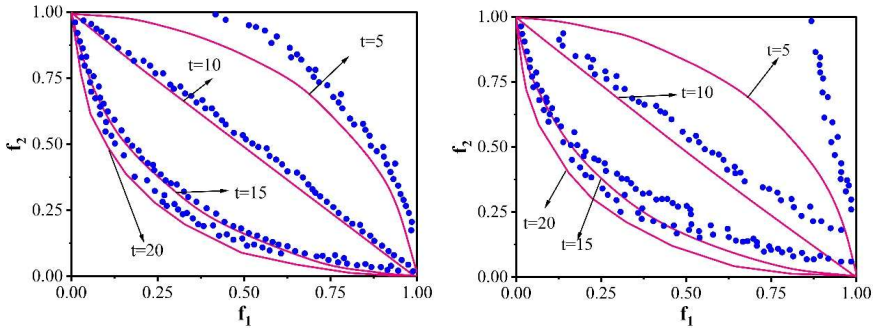


(a)RSEA (b)TS-NSGA-11

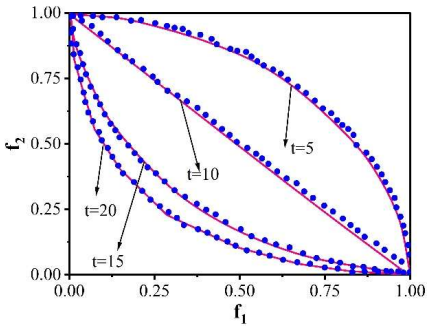


(c)Ours

Figure 4: The solution set obtained in the process of solving F5

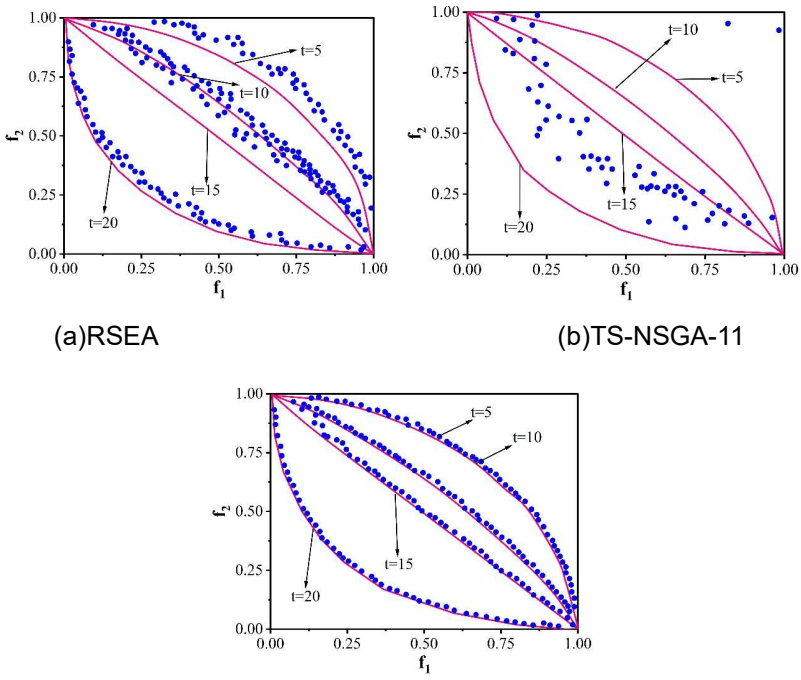


(a)RSEA (b)TS-NSGA-11



(c)Ours

Figure 5: The solution set obtained in the process of solving F6



(a)RSEA (b)TS-NSGA-11

(c)Ours

Figure 6: The solution set obtained in the process of solving F9

III. B. 2) Runtime comparison

In this paper, four more representative multi-objective optimization problems, FDA1, DMOP2, F5, F6 and F9, are selected to track 100 large-scale computing environment changes. The experimental hardware is IntelCorei5 3. 40GHz CPU, 16GB RAM computer, and the software is Visual Studio 2015.Table 3 lists the average computation time consumed by the three strategies for solving multi-objective optimization problems in different large-scale computing environments. Table 3: RSEA and the algorithms in this paper consume less computation time for all problems and the difference is not significant, reflecting the fact that RSEA and the algorithms in this paper have the the fastest running speed. TS-NSGA-11 took the most time (20.67a~21.44s) for multi-objective solving. This may be related to the fact that TS-NSGA-11 has to predict both population centroid position and population shape. In addition. The time of this paper's algorithm is slightly more than RSEA. This is mainly caused by the fact that the fuzzy clustering classification in this paper's algorithm leads to be more time-consuming than RSEA.

Table 3: Running time comparison

Problem	RSEA	TS-NSGA-11	Ours
FDA1	7.76	8.07	7.92
FDA4	20.67	21.44	20.86
F5	8.59	8.46	8.66
F6	8.68	9.18	8.76
F9	8.65	8.91	8.51

IV. Conclusion

In this paper, we propose a research scheme of multi-objective optimization based on fuzzy clustering-particle swarm combination algorithm in a large-scale computing environment, and simulate and analyze the research scheme of this paper from the aspects of testing method and solving strategy. The research results of this paper are shown as follows:

(1) The IGD value of the test function of this paper's algorithm is better than that of RSEA, FDEA-II, KnEA, TS-NSGA-II, and Two_Arch2, and the same phenomenon exists in the HV evaluation indexes, which illustrates the superiority of this paper's algorithm in the solution of multi-objective optimization problems.

(2) Whether it is better convergence, uniform distribution, Pareto optimal solution set, computation time, this paper's algorithm is superior to RSEA and TS-NSGA-11, which verifies the effectiveness of the multi-objective optimization problem solving strategy based on the fuzzy clustering-particle swarm combination algorithm.

References

[1] Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., & Nolan, G. P. (2010). Computational solutions to large-scale data management and analysis. Nature reviews genetics, 11(9), 647-657.

[2] Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. (2012). Surface codes: Towards practical large-scale quantum computation. Physical Review A—Atomic, Molecular, and Optical Physics, 86(3), 032324.

[3] Mayer, R., Graser, L., Gupta, H., Saurez, E., & Ramachandran, U. (2017, October). Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In 2017 IEEE Fog World Congress (FWC) (pp. 1-6). IEEE.

[4] Bujok, P., & Kolenovsky, P. (2022, July). Eigen crossover in cooperative model of evolutionary algorithms applied to CEC 2022 single objective numerical optimisation. In 2022 IEEE congress on evolutionary computation (CEC) (pp. 1-8). IEEE.

[5] Goh, S. K., Tan, K. C., Al-Mamun, A., & Abbass, H. A. (2015, May). Evolutionary big optimization (BigOpt) of signals. In 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 3332-3339). IEEE.

[6] Tian, Y., Zheng, X., Zhang, X., & Jin, Y. (2019). Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. IEEE Transactions on Cybernetics, 50(8), 3696-3708.

[7] Hsieh, S. T., Sun, T. Y., Liu, C. C., & Tsai, S. J. (2008, June). Solving large scale global optimization using improved particle swarm optimizer. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) (pp. 1777-1784). IEEE.

[8] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., & Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. Advances in neural information processing systems, 32.

[9] Mairal, J. (2013). Stochastic majorization-minimization algorithms for large-scale optimization. Advances in Neural Information Processing Systems, 26.

[10] Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K. C., & Jin, Y. (2021). Evolutionary large-scale multi-objective optimization: A survey. ACM Computing Surveys (CSUR), 54(8), 1-34.

[11] Cui, M., Li, L., Zhu, S., & Zhou, M. (2021, December). An improved competitive swarm optimizer based on generalized Pareto dominance for large-scale multi-objective and many-objective problems. In 2021 IEEE International Conference on Networking, Sensing and Control (ICNSC) (Vol. 1, pp. 1-6). IEEE.

[12] Li, Y., Li, L., Lin, Q., Wong, K. C., Ming, Z., & Coello, C. A. C. (2022). A self‐organizing weighted optimization based framework for large‐scale multi‐objective optimization. Swarm and Evolutionary Computation, 72, 101084.

[13] Lu, Y., Li, B., Liu, S., & Zhou, A. (2023). A Population Cooperation based Particle Swarm Optimization algorithm for large-scale multi-objective optimization. Swarm and Evolutionary Computation, 83, 101377.

[14] Liu, J., & Liu, R. (2024). Objective contribution decomposition method and multi-population optimization strategy for large-scale multi-objective optimization problems. Information Sciences, 678, 120950.

[15] Chen, L., Zhang, J., Wu, L., Cai, X., & Xu, Y. (2024). Large-ScaleMulti-Objective Optimization Algorithm Based on Weighted Overlapping Grouping of Decision Variables. CMES-Computer Modeling in Engineering & Sciences, 140(1).

[16] Kiani-Moghaddam, M., Shivaie, M., Weinsier, P. D., Kiani-Moghaddam, M., Shivaie, M., & Weinsier, P. D. (2019). Introduction to multi-objective optimization and decision-making analysis. Modern Music-Inspired Optimization Algorithms for Electric Power Systems: Modeling, Analysis and Practice, 21-45.

[17] Du, W., Zhong, W., Tang, Y., Du, W., & Jin, Y. (2018). High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach. IEEE transactions on industrial informatics, 15(1), 293-304.

[18] Ma, X., Liu, F., Qi, Y., Wang, X., Li, L., Jiao, L., ... & Gong, M. (2015). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. IEEE Transactions on Evolutionary Computation, 20(2), 275-298.

[19] Zhang, X., Tian, Y., Cheng, R., & Jin, Y. (2016). A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. IEEE Transactions on evolutionary Computation, 22(1), 97-112.

[20] Ma, L., Huang, M., Yang, S., Wang, R., & Wang, X. (2021). An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. IEEE Transactions on Cybernetics, 52(7), 6684-6696.

[21] He, C., Cheng, R., Li, L., Tan, K. C., & Jin, Y. (2022). Large-scale multiobjective optimization via reformulated decision variable analysis. IEEE transactions on evolutionary computation, 28(1), 47-61.

[22] Li, Y., Feng, X., & Yu, H. (2024). Solving high-dimensional expensive Multiobjective optimization problems by adaptive decision Variable Grouping. IEEE Transactions on Evolutionary Computation.

[23] Xu, B., Gong, D., Zhang, Y., Yang, S., Wang, L., Fan, Z., & Zhang, Y. (2022). Cooperative co-evolutionary algorithm for multi-objective optimization problems with changing decision variables. Information Sciences, 607, 278-296.

[24] Wang, H., Jiao, L., Shang, R., He, S., & Liu, F. (2015). A memetic optimization strategy based on dimension reduction in decision space. Evolutionary computation, 23(1), 69-100.

[25] Zhang, X., & Miao, D. (2014). Reduction target structure-based hierarchical attribute reduction for two-category decision-theoretic rough sets. Information Sciences, 277, 755-776.

[26] Yao, X., Zhao, Q., Gong, D., & Zhu, S. (2021). Solution of large-scale many-objective optimization problems based on dimension reduction and solving knowledge-guided evolutionary algorithm. IEEE Transactions on Evolutionary Computation, 27(3), 416-429.

[27] Zhang, K., Shen, C., & Yen, G. G. (2022). Multipopulation-based differential evolution for large-scale many-objective optimization. IEEE Transactions on Cybernetics, 53(12), 7596-7608.

[28] Cartis, C., Massart, E., & Otemissov, A. (2023). Global optimization using random embeddings. Mathematical Programming, 200(2), 781-829.

[29] Wang, Z., Pei, Y., & Li, J. (2023). A survey on search strategy of evolutionary multi-objective optimization algorithms. Applied Sciences, 13(7), 4643.

[30] Zhang, Y., Li, B., Hong, W., & Zhou, A. (2023). MOCPSO: A multi-objective cooperative particle swarm optimization algorithm with dual search strategies. Neurocomputing, 562, 126892.
[31] Kropp, I., Nejadhashemi, A. P., & Deb, K. (2023). Improved evolutionary operators for sparse large-scale multiobjective optimization problems. IEEE Transactions on Evolutionary Computation, 28(2), 460-473.
[32] Ziyu Hu,Xuetao Nie,Hao Sun,Lixin Wei,Jinlu Zhang & Cong Wang. (2025). Sparse large-scale multi-objective optimization algorithm based on impact factor assistance. Engineering Applications of Artificial Intelligence,151,110615-110615.
[33] Team DeFI, Inria Saclay Île-de-France,ArianeGroup & Team DeFI, Inria Saclay Île-de-France. (2019). Surrogate-assisted Bounding-Box approach for optimization problems with tunable objectives fidelity. Journal of Global Optimization,75(4),1079-1109.
[34] Tian Ye,Lu Chang,Zhang Xingyi,Cheng Fan & Jin Yaochu. (2020). A Pattern Mining-Based Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems. IEEE transactions on cybernetics,PP,
[35] Ganbin Xu & Jianwei Lin. (2025). Automatic motion recognition technology based on fuzzy clustering algorithm and VR video image. Alexandria Engineering Journal,124,462-469.
[36] Yi Chen,Jingsong Sun,Ziyue Xu,Genglong Zhang,Naibin Qi & Yuchen Song. (2023). A Study of Digital Museum Collection Recommendation Algorithm Based on Improved Fuzzy Clustering Algorithm. International Journal of Computational Intelligence and Applications,23(01),