Applied Soft Computing 145 (2023) 110538

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A universal large-scale many-objective optimization framework based on cultural learning

Xia Wang [a], Hongwei Ge [a,b,*], Naiqiang Zhang [a], Yaqing Hou [a], Liang Sun [a]

[a] School of Computer Science and Technology, Dalian University of Technology, Dalian 116023, China
[b] Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA

ARTICLE INFO

Article history:
Received 21 November 2022
Received in revised form 10 May 2023
Accepted 10 June 2023
Available online 23 June 2023

Keywords:
Many-objective evolutionary algorithm
Cultural algorithm
Large-scale
Diversity

ABSTRACT

When solving large-scale many-objective optimization problems (LMaOPs), due to the large number of variables and objectives involved, the algorithm is faced with a very high-dimensional and complex search space, which is difficult to be explored with limited resources. To address these issues, this paper proposes a universal large-scale many-objective optimization framework based on cultural learning (UCLMO). First, a universal framework is proposed, and multi-objective optimizers can be embedded into the framework to accelerate the convergence. Moreover, inspired by cultural learning, an individual selection strategy based on historical knowledge is proposed to promote the diversity of the population, and an assisted evolution strategy based on normative knowledge is presented to accelerate the convergence of the algorithm. Experiments have been conducted on multi-objective knapsack problems and LMaOPs with decision variables ranging from 500 to 1500, and the number of objectives ranging from 5 to 15. The experimental results verify the superiority and competitiveness of the proposed UCLMO framework in solving LMaOPs compared with state-of-the-art algorithms.

1. Introduction

With the development of cyber–physical systems, more and more practical engineering optimization problems involve more than three optimization objectives, such as water resource allocation problem [1,2], knapsack problem [3,4], software module computing [5,6], autonomous robotics services [7,8], etc. The optimization problems with more than three objectives are called many-objective optimization problems (MaOPs). In recent years, many types of many-objective evolutionary algorithms (MaOEAs) have been proposed to solve MaOPs. The aim of MaOEAs is to generate a set of optimal solutions as well as maintain convergence and diversity simultaneously. The convergence makes the objective values of the solutions close to the Pareto optimal front (PF), and the diversity makes the solutions well represent the distribution of PF. However, the current optimization algorithms are mainly used to solve MaOPs with low-dimensional decision variables.

When dealing with MaOPs with high-dimensional decision variables, the performance of MaOEAs tends to decline rapidly as the number of decision variables increases. Therefore, the optimization of MaOPs with large-scale decision variables is a challenge. In order to address the challenge, existing researchers mainly conduct some efficient algorithms from the two aspects of decomposition strategy [9–11] and large-scale exploration strategy [12–14].

The algorithms belonging to the decomposition strategy are performed by decomposing MaOPs into multiple sub-problems for simultaneous optimization. It has been shown that the decomposition strategy based on the objective space has advantages in solving MaOPs [15]. The main reason is that the decomposition strategy based on the analysis of decision variables can effectively deal with the problems caused by the many-objective space. The algorithms based on large-scale exploration strategy are performed by using various operations to impose selection pressures, which promote the convergence of the algorithm [16]. The performance of algorithms for solving large-scale MaOPs remains unsatisfactory due to the difficulty in achieving a balance between diversity and convergence of the population. Therefore, in order to promote diversity and accelerate convergence in the optimization process, this paper proposes a universal large-scale many-objective optimization framework based on cultural learning (UCLMO).

The main contributions of this paper are as follows:

(1) A universal cultural learning framework for large-scale many-objective optimization (UCLMO) is proposed. Multi-objective optimizers can be embedded into UCLMO, and the integrated framework can accelerate the convergence of the algorithm.

* Corresponding author at: School of Computer Science and Technology, Dalian University of Technology, Dalian 116023, China.
E-mail address: hwge@dlut.edu.cn (H. Ge).

https://doi.org/10.1016/j.asoc.2023.110538
1568-4946/

(2) Inspired by cultural learning, this paper proposes an individual selection strategy based on historical knowledge to promote the diversity of the population. Besides, an assisted evolution strategy based on normative knowledge is proposed to promote the efficiency of evolution.
(3) Four different optimizers are embedded in UCLMO to verify the effectiveness. Extensive experiments are conducted on LSMOPs, DTLZs, and multi-objective knapsack problems, the UCLMO achieves state-of-the-art performance.

The rest of this paper is organized as follows: Section 2 gives the related work, including basic conception, related work about large-scale MaOPs, and cultural algorithm. Section 3 elaborates on the proposed UCLMO. Section 4 gives the experiment results and analysis. Section 5 concludes this work and gives future work.

2. Related work

In this section, the definition of the large-scale many-objective optimization problem is first explained, and then this section reviews the existing multi-objective optimization algorithms for solving large-scale MaOPs. Besides, the research on the cultural algorithm is reviewed. Finally, the motivation of this paper is explained.

2.1. Large-scale many-objective optimization problems

Multi-objective optimization problems (MOPs) mainly involve optimization problems containing multiple conflicting objectives, and can be defined as follows:

$$\begin{aligned} &\min \quad f(x) = (f_1(x), f_2(x), \ldots, f_M(x)) \\ &\text{subject to } x \in \Omega \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_D) \in \Omega$ represents the decision vector, and $\Omega \in R^D$ represents the feasible decision space. M and D denote the number of objectives and the number of decision variables, respectively.

The large-scale many-objective optimization problems are with 4 or more objectives (i.e. $M \geq 4$), and with hundreds or even thousands of decision variables (i.e. $D \geq 100$) [17,18].

2.2. Existing MOEAs for large-scale MaOPs

According to the idea and characteristics of the algorithms, the existing MOEAs for solving large-scale MaOPs can be generally divided into two categories: the method based on decomposition strategy and the method based on large-scale exploration strategy.

The algorithms belonging to decomposition adopt a divide-and-conquer strategy to decompose MaOP into multiple subproblems for simultaneous optimization. MOEA based on Decomposition (MOEA/D) [15] is the first multi-objective optimization algorithm based on decomposition strategy. It transforms the multi-objective optimization problem into multiple single-objective subproblems through a set of weight vectors, and solves them cooperatively. MOEA/D finally outputs a set of Pareto optimal solutions at one time. Antonio et al. propose a cooperative co-evolutionary GDE3 (CCGDE3) [9] for solving large-scale MOPs, which is an effective co-evolutionary algorithm combined with GDE3, through the cooperation of multiple sub-populations to form a feasible solution. Ma et al. [19] propose an MOEA based on decision variable analyses (MOEA/DVA), which divides decision variables into three groups before applying the idea of cooperative coevolution, namely, position variables, distance variables, and mixed variables. The decision variables of different groups can be optimized by different optimization strategies. Song et al. [20] embed a random-based dynamic grouping strategy into MOEA/D and propose MOEA/D-RDG to improve the optimization efficiency of the algorithm. Zhang et al. [10] propose an evolutionary algorithm for large-scale MaOPs (LMEA) based on the decision variable clustering method, which is used to identify the contribution of decision variables, and uses the polling optimization of diversity- and convergence-related variables to maintain the diversity of the population. The experimental results show that it achieves better results in higher dimensions than MOEA/DVA. Chen et al. [18] present a scalable small subpopulations-based covariance matrix adaptation evolution strategy (S^3-CMA-ES), which better balances the diversity and convergence of the population. Zhang et al. [21] present a differential evolution algorithm based on multiple populations (LSMaODE), which can solve large-scale MaOPs efficiently and effectively. The population is divided into two subpopulations, which are optimized using a randomized coordinate descent strategy and a non-dominated guided interpolation strategy, respectively. Liu et al. [22] propose a new MOEA/D algorithm, which divides the whole PF into a group of segments, and each segment can be regarded as a subproblem.

The large-scale exploration strategies use various techniques to impose selection pressures that promote the convergence of the algorithm. The cross-mutation mechanism in MOEAs slightly changes decision variables to improve the generation quality of better solutions. However, this approach can cause the algorithm to fall into local optima and slow down the convergence speed when dealing with large-scale MaOPs. To overcome this challenge, some algorithms make significant changes to the decision variables in an effective way to achieve rapid optimization. Some large-scale exploration strategies focus on the optimization mechanism. For instance, the linear combination-based search algorithm (LCSA) [13] assigns a weight value to each solution in the current population, weights the weight value to generate a new solution, and then optimizes these weight values for large-scale exploration. Similarly, the weighted optimization framework (WOF) [12] divides decision variables into several groups and optimizes the weight variable for each group to generate new better individuals. The large-scale multi-objective optimization framework (LSMOF) [12] uses two reference points to guide the change of decision variables and reconstructs LSMOP into a low-dimensional single-objective optimization problem. The MOEA with dual local search (DLS-MOEA) [23] improves the diversification capabilities of large-scale multi-objective algorithms by designing a dual local search strategy. Some large-scale exploration strategies focus on heuristic design. The grouped and linked polynomial mutation operator (GLMO) [24] proposes a mutation mechanism based on linking and grouping to achieve large-scale exploration by assuming that related decision variables should change simultaneously. The differential grouping evolution algorithm (DGEA) [25] uses non-dominated individuals to guide and generate new solutions through hashing with Gaussian distribution. The large-scale multi-objective competitive swarm optimizer algorithm (LMOCSO) [26] increases the effective exploration degree in the optimization process through a winner-take-all mechanism to avoid falling into the local optimum. Finally, the Multi-objective orthogonal opposition-based crow search algorithm (M2O-CSA) [27] uses a parallel orthogonal opposition strategy and the dominance concept to update the crow's memory.

While the aforementioned algorithms have demonstrated promising results in addressing large-scale multi-objective optimization problems, they still exhibit some limitations. On the one hand, when dealing with large-scale MaOPs, MaOEA based on objective space decomposition is difficult to balance diversity and convergence to obtain a satisfactory solution within a reasonable cost of function evaluations as the decision variables grow significantly. On the other hand, although MaOEA based on the large-scale exploration strategy can accelerate the optimization process through the designs of optimization mechanisms and heuristic operators, this kind of MaOEA is prone to fall into local optimum. To tackle these issues, this work proposes a universal cultural learning-inspired framework for large-scale many-objective optimization which includes an individual selection strategy based on historical knowledge and an assisted evolution strategy based on normative knowledge. The proposed framework can promote the diversity of the population while accelerating the convergence of the algorithm.

2.3. Cultural algorithm

The inheritance of culture has enabled human beings to pass on the experience and knowledge gained in a long historical career, making children a few years old know far more knowledge than the ancients. The cultural algorithm (CA) is produced by simulating this kind of phenomenon. It extracts the knowledge from the population, and then reacts to the population to accelerate the optimization. In CA, evolution space consists of belief space and population space. Belief space refers to the space where the knowledge is located, and population space refers to the space where the individual is located. Generally speaking, knowledge can be divided into five categories, including normative knowledge, situational knowledge, domain knowledge, historical knowledge, and topographical knowledge. Normative knowledge reflects the search upper and lower bounds of high-quality individuals. Situational knowledge provides a set of typical cases, which can guide individuals to move toward exemplary individuals. Domain knowledge records the domain information of the problem. Historical knowledge reflects important events in the process of evolution or representative individuals in history. Topographical knowledge reflects the fitness distribution of individuals in the search space.

The cultural algorithm only provides a reference framework based on cultural evolution [28]. Its key lies in what knowledge to extract, how to extract and how to use knowledge to promote the evolution process. Ref. [29] designs a grid diagram by extracting the normative knowledge of the population in the objective space, and selecting individuals through the grid diagram to maintain the diversity and convergence of the population. Coello et al. extract the normative knowledge of the parameters of differential evolution in the population, and use the normative knowledge to design corresponding rules to achieve the purpose of dynamic parameter adjustment [30]. Liu et al. extract situational knowledge of the population in decision space to guide individuals to perform local search strategies, and improve the diversity and convergence of the population by defining a non-dominated solution set as historical knowledge [31].

2.4. Motivation

On the one hand, MaOEA based on objective space decomposition is an effective kind of method for solving MaOPs because MaOEA is not affected by the ineffectiveness of Pareto dominance. However, when dealing with large-scale MaOPs, it is difficult to balance diversity and convergence to obtain a satisfactory solution within a reasonable cost of function evaluations as the decision variables grow significantly. On the other hand, although MaOEA based on the large-scale exploration strategy can accelerate the optimization process through the designs of optimization mechanisms and heuristic operators, this kind of MaOEA is prone to fall into local optimum. A cultural algorithm (CA) can effectively improve the optimization efficiency of the problem by explicitly extracting knowledge of the population in the optimization process. The CA framework can promote the diversity of the population through historical knowledge, and can accelerate the convergence through the guidance of normative knowledge for offspring generation. Therefore, this paper proposes a universal large-scale many-objective optimization framework based on cultural learning (UCLMO).

3. Proposed UCLMO

3.1. Universal framework of the proposed UCLMO algorithm

Cultural algorithms provide a basic framework based on cultural evolution. Inspired by this, this paper proposes a universal cultural learning framework for large-scale many-objective optimization (UCLMO). The overall framework of the algorithm is shown in Fig. 1.

As it can be seen from Fig. 1, UCLMO is composed of two spaces: population space and belief space. The population space contains the embedded optimizer (such as MOEA) and the proposed normative knowledge-assisted evolutionary algorithm. The belief space is composed of historical knowledge and normative knowledge. Among them, historical knowledge is used as a knowledge set for generating inherited populations, and also as a knowledge set for generating normative knowledge by a proposed individual selection strategy. The inherited population and normative knowledge are used to perform normative knowledge-assisted evolution. In this paper, the embedded optimizer is executed alternately with the normative knowledge-assisted evolutionary algorithm.

Historical knowledge was originally proposed for dynamic optimization problems. As an evolutionary time series, when the algorithm is found to fall into a local optimum or the environment changes, it can return to the previously recorded feasible solution space. In this paper, all offspring individuals generated by the embedded optimizer through a fixed number of iterations are recorded as historical knowledge. Historical knowledge can retain all kinds of individuals to prevent the loss of knowledge caused by the balance of diversity and convergence in the evolutionary process. Normative knowledge is a feasible search space that describes the optimal solution. This paper designs a normative knowledge for each reference vector to maintain the diversity of the population.

In this framework, the proposed individual selection strategy based on historical knowledge and the assisted evolution strategy based on normative knowledge play an important role in promoting diversity and convergence, respectively. They are described in detail next.

3.2. Individual selection strategy based on historical knowledge

The specific process of normative knowledge generation and individual selection strategy based on historical knowledge is shown in Algorithm 1.

Firstly, a systematic sampling method is used to generate a uniformly distributed reference vector in the objective space.

Secondly, a number of solutions are associated with each reference vector according to the angle metric. More specifically, the association method first generates an M-dimensional unit matrix V to represent a set of reference vectors, and then the individual solutions are normalized:

$$f_i'(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{\min}}{z_i^{\max} - z_i^{\min}} \quad (2)$$

where $\mathbf{x} \in Data$ represents the individual in the data set, $z_i^{\max}$ and $z_i^{\min}$ represent the maximal and minimal of the ith objective

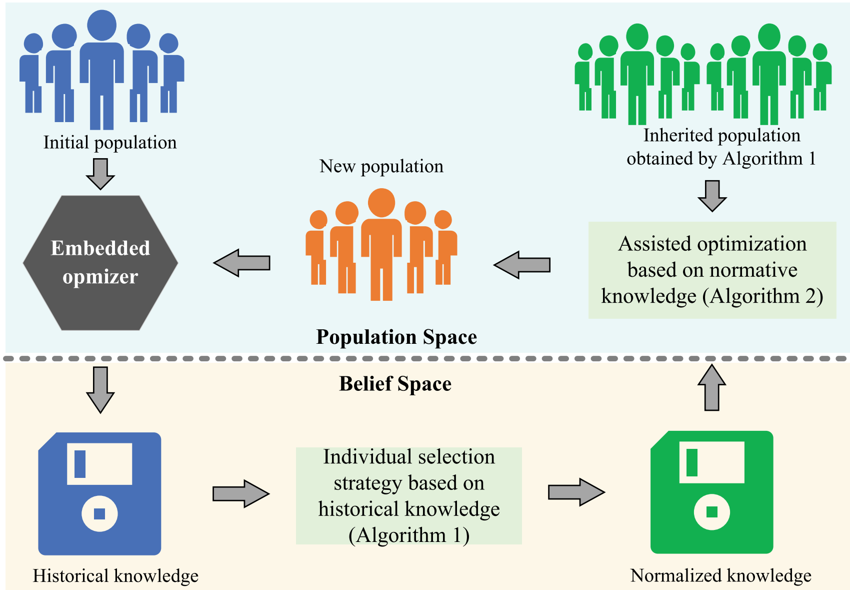


Fig. 1. Overall framework of UCLMO.

Algorithm 1: Individual selection strategy based on historical knowledge

---

Input: Historical knowledge *Data*, the number of associated solutions for each reference vector T, acceptance rate P_{accept}

Output: Inherited population P, normative knowledge C

Associate T solutions for each reference line based on the angle metric from *Data*;
for $i = 1 : |W|$ **do**
  Sort the solutions related to the i-th reference vector by PBI index;
  Select the first T solutions as statistical solutions;
  Perform normative knowledge statistics according to Eq. (7);
  Collect individuals with the smallest PBI value as the optimal solution in this direction;
end

---

values. The angle between the ith normalized individual $f'\left(\mathbf{x}^i\right)$ and the jth reference vector V^j is calculated as by:

$$\text{angle}\left(\mathbf{x}^i, \mathbf{v}^j\right) = \arccos \frac{\left|f'\left(\mathbf{x}^i\right)\cdot \mathbf{V}^j\right|}{\left\|f'\left(\mathbf{x}^i\right)\right\| \times \left\|\mathbf{V}^j\right\|} \tag{3}$$

Hereafter, each individual is attached to the vector that has a minimal angle with it. This strategy ensures that each reference vector can be associated with the solution.

Then, the penalty-based boundary intersection (PBI) value is used to select N_{accept} better individuals for each reference line to generate normative knowledge. The PBI method evolves along the direction of weight vector. For a weight vector λ^i, the projection distance $d_1\left(\mathbf{x} \mid \lambda^i, \mathbf{z}^*\right)$ and vertical distance $d_2\left(\mathbf{x} \mid \lambda^i, \mathbf{z}^*\right)$ from the current solution to the weight vector are calculated respectively. The calculation process can be formulated as follows:

$$\begin{aligned} &\min\ g^{pbi}\left(\mathbf{x}|\boldsymbol{\lambda}^{\mathbf{i}}, \mathbf{z}^*\right) = d_1\left(\mathbf{x}|\boldsymbol{\lambda}^{\mathbf{i}}, \mathbf{z}^*\right) + \theta * d_2\left(\mathbf{x}|\boldsymbol{\lambda}^{\mathbf{i}}, \mathbf{z}^*\right) \\ &\textit{subject to } \mathbf{x} \in \Omega \end{aligned} \tag{4}$$

$$d_1\left(\mathbf{x}|\boldsymbol{\lambda}^{\mathbf{i}}, \mathbf{z}^*\right) = \frac{\left\|\left(F(\mathbf{x}) - \mathbf{z}^*\right)^T \boldsymbol{\lambda}^{\mathbf{i}}\right\|}{\left\|\boldsymbol{\lambda}^{\mathbf{i}}\right\|} \tag{5}$$

$$d_2\left(\mathbf{x}|\boldsymbol{\lambda}^{\mathbf{i}}, \mathbf{z}^*\right) = \left\|F(\mathbf{x}) - \left(\mathbf{z}^* + d_1 \frac{\boldsymbol{\lambda}^{\mathbf{i}}}{\left\|\boldsymbol{\lambda}^{\mathbf{i}}\right\|}\right)\right\| \tag{6}$$

where $\mathbf{z}^*$ represents the vector composed of the minimum value of each objective in the historical population. The parameter θ is used for weighting, and is chosen to be 5 as it is in penalty-based boundary intersection (PBI) approach [32].

Finally, the normative knowledge is calculated for each reference vector. Among them, the normative knowledge on the k-th reference vector is calculated as follows:

$$\begin{cases} min\_cul_i^k = \min\left(\left\{x_i^k(j) \,\middle|\, j = 1, 2, \ldots, N_{accept}\right\}\right) \\ max\_cul_i^k = \max\left(\left\{x_i^k(j) \,\middle|\, j = 1, 2, \ldots, N_{accept}\right\}\right) \end{cases} \tag{7}$$

where, $x_i^k(j)$ represents the value of the i-th decision variable of the j-th individual associated on the k-th reference vector.

The normative knowledge of the entire population can be expressed as:

$$\begin{aligned} &\left(min\_cul^k; max\_cul^k\right) \\ &= \left(min\_cul_1^k, min\_cul_2^k, \ldots, min\_cul_D^k; max\_cul_1^k, max\_cul_2^k, \ldots, max\_cul_D^k\right) \end{aligned} \tag{8}$$

where $k = 1, 2, \ldots, NW$. NW represents the number of reference vectors, which is taken as the population size N_p.

While calculating the normative knowledge, the individuals with the smallest PBI value are selected into the inherited population. In this way, the individual selection strategy based on historical knowledge can promote the diversity of the population, thus enhancing the algorithm's exploration ability.

3.3. Assisted evolution strategy based on normative knowledge

The existing offspring solution generation strategy is mainly based on the mating between individuals in the current population or the mating between individuals in the neighborhood to produce offspring. However, as the number of decision variables and objectives increases, it is a challenge to effectively improve the efficiency of optimization. Therefore, this paper proposes an assisted evolution strategy to improve the efficiency of optimization and accelerate convergence. The strategy takes normative knowledge as the upper and lower bounds of generated individuals, and randomly generates new individuals through uniform distribution to assist the evolution of individuals in the population. Then, the individuals generated by normative knowledge are mated with the individuals from the inherited population to generate new offspring. This way can improve the exploitation efficiency of the solutions and improve the overall convergence speed. The process of the assisted evolutionary algorithm based on normative knowledge is shown in Algorithm 2.

Algorithm 2: Normative knowledge-assisted evolutionary algorithm

```
Input: Inherited population P, nadir points Z_min, normative
       knowledge C, reference vector W, Maximum
       number of evaluations used for optimization t_2
Output: Population P, nadir points Z_min
while The current number of evaluations is less than t_2 do
    for i = 1 : |W| do
        Generate individuals based on normative
         knowledge;
        The generated individuals mate with P_i to produce
         offspring;
        Update Z_min;
        Update P_i by PBI indicator;
    end
end
```

The embedded MOEA is executed alternately with the proposed assisted evolutionary algorithm based on normative knowledge. After the assisted evolutionary algorithm based on normative knowledge is executed, the existing historical knowledge and normative knowledge are discarded. When the embedded MOEA starts to perform evolution again, all new individuals generated during the optimization process are recorded as new historical knowledge. Then the normative knowledge is updated through historical knowledge. The pseudo-code of the universal large-scale many-objective optimization framework based on cultural learning is shown in Algorithm 3.

Algorithm 3: Framework of our proposed UCLMO

```
Input: The maximum number of evaluations t_1 embedded
       in MOEA, the maximum number of evaluations t_2
       used for normative knowledge-assisted
       optimization, and the maximum number of
       evaluations for population evolution MaxFEs
Output: Final population P
Initialize population P, reference vector W, denote the
 current number of evaluations as FEs;
while FEs ≤ MaxFEs do
    Use the embedded MOEA for t_1 evaluations;
    Construct normative knowledge based on reference
     vectors (Algorithm 1);
    Perform assisted optimization based on normative
     knowledge for t_2 evaluations (Algorithm 2);
end
```

3.4. Time complexity analysis

The UCLMO framework comprises various components, such as an embedded MOEA, the generation of historical knowledge, an individual selection strategy based on historical knowledge, the extraction of normative knowledge, and an assisted evolution strategy based on normative knowledge. Assume that the time complexity of each generation of the embedded MOEA is O_{MOEA}, the time complexity of the individual selection strategy based on historical knowledge is $O(1)$, and the time complexity of normative knowledge for each generation is $O(MN^2)$, where N represents the population size. The assisted evolution strategy based on normative knowledge involves generating individuals randomly through normative knowledge for mating, without additional operations. Therefore, the time complexity of this strategy is $O(N)$. As a result, the time complexity of UCLMO can be expressed as $\max O_{MOEA}, O(MN^2)$.

4. Result analysis and discussion

4.1. Benchmark test suites and performance metric

Benchmark test functions: LSMOPs (LSMOP1-LSMOP9) [17] and DTLZs (DTLZ1-DTLZ5) [33] are used to test the performance of UCLMO. Table 1 shows the property of the nine LSMOPs and five DTLZs functions used in this paper.

Performance metric: In this paper, three performance indicators are used to evaluate optimization algorithms: the inverted generational distance (IGD) [34,35], generational distance (GD) [36], and average Hausdorff distance (Δ_P) [37]. The IGD metric is calculated for a given reference set, which is generated by systematic sampling using uniformly distributed sampling points on the hyperplane of unit intercept. The calculation formula of IGD is shown as Eq. (9) [36].

$$IGD\left(P^{*}, P\right) = \frac{\sum_{v \in P^{*}} d\left(v, P\right)}{\left|P^{*}\right|} \tag{9}$$

where P^{*} is a set of reference points uniformly sampled on PF, and P is the approximate value of PF obtained, and $d\left(v, P\right)$ represents the minimum Euclidean distance between v and all points in P. The smaller the value of IGD, the better the optimization performance of the algorithm.

The GD measures the degree of approximation of the solution set to the true Pareto front, with smaller values indicating better convergence performance. Δ_P can be considered as the average Hausdorff distance between the Pareto-approximated solution set and the PF, and is able to measure both convergence and diversity of solution sets. The smaller the value of Δ_P, the better the performance of the algorithm.

In order to verify the significant difference between the proposed algorithm and the comparison algorithm, the Wilcoxon rank-sum test method and the Friedman test are selected as the statistical test method [41]. They are both non-parametric statistical test methods, and the significance level is usually set to 0.05. Among them, "+/-/≈" shows that the comparison algorithm is significantly better than UCLMO, significantly worse than UCLMO, and similar to UCLMO.

Experimental environment: The experimental results of this paper are all obtained by running on the PlatEMO platform [42] on Matlab.

4.2. Peer algorithms and parameter settings

In order to verify the optimization performance of the proposed UCLMO,[1] this paper selects ten state-of-the-art algorithms as comparison algorithms.

(1) CCGDE3 [9]: CCGDE3 is a large-scale MaOEA based on co-evolution strategy. The algorithm uses a random grouping strategy to randomly divide the decision variables into multiple sub-populations. It uses the fast non-dominated

[1] https://github.com/dut-ww/UCLMO.

Table 1
Benchmark functions from LSMOP and DTLZ test suites.

Test suite	Benchmark problem	Features of Pareto front	Features of landscape	Separability of objective functions
LSMOPs (2017)	LSMOP1	Linear	Unimodal	Separable
	LSMOP2	Linear	Mixed	Partially separable
	LSMOP3	Linear	Multimodal	Mixed
	LSMOP4	Linear	Mixed	Mixed
	LSMOP5	Concave	Unimodal	Separable
	LSMOP6	Concave	Mixed	Partially separable
	LSMOP7	Concave	Multimodal	Mixed
	LSMOP8	Concave	Mixed	Mixed
	LSMOP9	Disconnected	Mixed	Separable
DTLZs (2005)	DTLZ1	Linear	Multimodal	Separable
	DTLZ2	Concave	Unimodal	Separable
	DTLZ3	Concave	Multimodal	Separable
	DTLZ4	Concave	Unimodal, biased	Separable
	DTLZ5	Concave, degenerate	Unimodal	Separable

Table 2
Parameter settings of all algorithms.

Algorithm	Parameters	Values
CCGDE3 [9]	The number of species	2
LMOCSO [26]	Penalty parameter	2
LSMaODE [21]	Regulable parameter to set the proportion of subpopulation	0.1
CVEA3 [38]	The probability in selecting partner	0.7
AMPDEA	Number of perturbations on each solution for decision variable clustering	50
BCE-IBEA [39]	Scaling factor	0.05
GLMO [24]	The number of groups	4
DGEA [25]	The number of direction vectors	10
WOF [12]	The number of groups	4
	The number of evaluations for the original problem	1000
	The number of evaluations for transformed problem	500
	The fraction of function evaluation to use for the alternating weight-optimization phase	0.5
	The number of chosen solutions to do weight optimization	M+1
FDV [40]	Fuzzy evolution rate	0.8
	Step acceleration	0.4
	MaxFEs embedded in MOEA	500
UCLMO	MaxFEs of culture-assisted evolutionary algorithms	500
	Individual correlation ratio of each reference line	0.06
	The proportion of high-quality individuals selected by each reference line	0.35

sorting method to obtain the optimal non-dominated solution for each sub-population, thereby obtaining the final solution set.

(2) LMOCSO [26]: LMOCSO is an efficient search method for solving large-scale MOPs. The algorithm adopts a new particle update strategy and a two-stage position update strategy to improve search efficiency.
(3) LSMaODE [21]: LSMaODE is a differential evolution algorithm based on multiple populations. The population is divided into two subpopulations, which are optimized using randomized coordinate descent strategy and non-dominated guided interpolation strategy, respectively.
(4) CVEA3, AMPDEA, BCE-IBEA [38,39]: CVEA3, AMPDEA, and BCE-IBEA are the winners of the 2018 IEEE CEC competition on many-objective optimization.[2] The codes for these methods are available at https://github.com/ranchengcn/IEEE-CEC-MaOO-Competition/tree/master/2018.
(5) GLMO [24]: GLMO studies the influence of variable grouping within mutation operators on large-scale MOPs. The algorithm introduces three new mutation operators on the basis of polynomial mutation.
(6) DGEA [25]: DGEA is an adaptive offspring generation method for large-scale MOPs. The algorithm uses two directional vectors to adaptively generate progeny solutions, and can be embedded into existing MOEAs for large-scale MOPs.
(7) WOF [12]: WOF implements the transformation of the original optimization problem (problem transformation) through variable grouping and weighting. It transforms the optimization of large-scale decision variables into the optimization of lower dimensional weight vectors, and realizes the dimensionality reduction of search space.
(8) FDV [40]: FDV is a fuzzy decision variable framework for large-scale MOPs. It divides the evolution process into two stages: fuzzy and precise evolution, which makes the population converge quickly and improves diversity.

The parameters of this paper are set as follows:

Parameter settings of test instances: To ensure fairness, the parameters of test instances are set as follows: The number of objectives (M) is set to 5, 8, 10, 15, the corresponding number of decision variables (D) is set to $M \times 100$, and the maximum number of function evaluations (*MaxFEs*) is set to 100 000. The population size (N) is set as follows: N is set to 126 for 5-objectives, 128 for 8-objectives, 230 for 10-objectives, 240 for 15-objectives. Each algorithm was run 30 times independently on each test instance to arrive at a statistically meaningful observation.

Parameter settings of all algorithms: For the parameters in the comparison algorithms, the parameters recommended in the corresponding literature are used to achieve their optimal performance. The parameter settings of all algorithms are shown in Table 2.

2 https://github.com/ranchengcn/IEEE-CEC-MaOO-Competition/tree/master/2018

Table 3
The mean and standard deviation of the IGD values solved by the three algorithms and embedding algorithms on LSMOP and DTLZ test suites.

Problem	M	NSGA-III	UC-NSGA-III	SMPSO	UC-SMPSO	RVEA	UC-RVEA	LMOCSO	UC-LMOCSO
LSMOP1	5	3.1716e+0 (3.54e−1) −	5.1466e−1 (3.40e−2)	5.7001e+0 (1.14e+0) −	8.5976e−1 (6.01e−2)	7.3293e−1 (8.77e−2) −	4.5859e−1 (3.62e−2)	1.1490e+0 (2.13e−1) −	8.4205e−1 (6.17e−2)
	8	4.7104e+0 (1.06e+0) −	4.8085e−1 (2.85e−2)	8.3971e+0 (2.42e+0) −	7.6872e−1 (8.54e−2)	1.0348e+0 (2.36e−1) −	3.8122e−1 (1.50e−2)	8.7846e−1 (1.99e−1) −	6.1407e−1 (7.51e−2)
	10	6.8716e+0 (8.80e−1) −	3.7839e−1 (1.01e−2)	1.0314e+1 (1.78e+0) −	6.7518e−1 (4.58e−2)	1.3138e+0 (1.33e−1) −	4.0206e−1 (9.42e−3)	9.6196e−1 (1.44e−1) −	5.7843e−1 (4.02e−2)
	15	6.8983e+0 (1.04e+0) −	4.0678e−1 (1.32e−2)	1.1941e+1 (1.93e+0) −	7.3834e−1 (6.51e−2)	1.6617e+0 (4.14e−1) −	4.9779e−1 (2.08e−2)	1.0202e+0 (2.14e−1) −	6.6019e−1 (4.82e−2)
LSMOP2	5	1.7496e−1 (4.85e−4) −	1.6888e−1 (5.20e−4)	1.8828e−1 (6.02e−3) −	1.6697e−1 (4.17e−4)	1.7443e−1 (9.97e−4) −	1.5270e−1 (1.43e−3)	1.7332e−1 (1.37e−3) −	1.6544e−1 (5.76e−4)
	8	3.1072e−1 (1.91e−3) −	2.8495e−1 (1.38e−3)	3.0968e−1 (8.72e−3) −	2.6937e−1 (2.06e−3)	2.9099e−1 (4.80e−3) −	1.9911e−1 (2.25e−3)	5.6534e−1 (2.47e−1) −	2.7662e−1 (4.83e−3)
	10	3.4168e−1 (1.70e−3) −	2.9566e−1 (4.32e−3)	3.1434e−1 (1.05e−2) −	2.6279e−1 (2.54e−3)	3.1815e−1 (7.13e−3) −	2.1815e−1 (4.17e−3)	6.3464e−1 (2.19e−1) −	2.5887e−1 (1.28e−2)
	15	2.9777e−1 (3.29e−2) −	2.6895e−1 (1.17e−3)	3.7804e−1 (1.13e−2) −	2.6446e−1 (1.28e−3)	2.9824e−1 (6.32e−2) =	2.8396e−1 (7.07e−3)	4.6666e−1 (1.14e−1) −	2.5883e−1 (2.45e−3)
LSMOP3	5	1.0040e+1 (2.08e+0) −	1.7525e+0 (3.35e−1)	2.0334e+1 (2.06e+0) −	9.0624e+0 (1.01e+0)	4.8663e+0 (1.67e+0) −	2.7466e+0 (1.14e+0)	1.1577e+1 (5.05e+0) =	1.1714e+1 (2.47e+0)
	8	4.3773e+1 (9.26e+1) −	2.2597e+0 (6.47e−1)	1.7299e+1 (4.00e+0) −	5.4925e+0 (1.88e+0)	9.0774e+0 (2.84e+0) −	2.9809e+0 (3.41e−1)	7.2008e+0 (7.48e+0) +	1.0230e+1 (3.23e+0)
	10	3.3075e+1 (3.76e+1) −	2.5988e+0 (3.26e−1)	2.1529e+1 (2.73e+0) −	7.8427e+0 (1.82e+0)	7.9964e+0 (2.66e+0) −	4.0118e+0 (3.31e−1)	8.7782e+0 (8.86e+0) =	1.0290e+1 (2.25e+0)
	15	2.6422e+1 (2.57e+0) −	3.4351e+0 (2.60e−1)	2.4606e+1 (3.92e+0) −	8.6199e+0 (9.13e−1)	1.4528e+1 (4.13e+0) −	1.3873e+1 (1.75e+1)	2.4558e+1 (6.15e+0) −	1.0877e+1 (1.88e+0)
LSMOP4	5	3.0846e−1 (4.97e−3) −	2.8688e−1 (1.50e−3)	3.3150e−1 (1.52e−2) −	2.8488e−1 (1.34e−3)	2.9664e−1 (1.14e−2) −	2.5924e−1 (5.93e−3)	2.9089e−1 (3.40e−3) −	2.8164e−1 (2.78e−3)
	8	3.6223e−1 (1.69e−3) −	3.2594e−1 (2.54e−3)	3.6819e−1 (1.59e−2) −	3.0489e−1 (2.79e−3)	3.4887e−1 (9.84e−3) −	2.2872e−1 (5.25e−3)	4.7422e−1 (1.49e−1) −	3.1990e−1 (4.45e−3)
	10	3.8199e−1 (2.25e−3) −	3.1256e−1 (4.47e−3)	3.6205e−1 (1.31e−2) −	2.8226e−1 (3.13e−3)	3.5577e−1 (1.15e−2) −	2.3185e−1 (4.11e−3)	5.3421e−1 (1.97e−1) −	2.7569e−1 (9.57e−3)
	15	3.3599e−1 (3.71e−2) −	2.7627e−1 (2.36e−3)	4.0388e−1 (1.48e−2) −	2.7607e−1 (2.03e−3)	3.1039e−1 (2.50e−2) −	2.9200e−1 (1.22e−2)	6.7432e−1 (2.40e−1) −	2.6884e−1 (2.88e−3)
LSMOP5	5	7.1850e+0 (8.28e−1) −	8.6487e−1 (2.83e−2)	1.8902e+1 (4.82e+0) −	8.1227e−1 (5.11e−2)	1.1061e+0 (6.80e−3) −	6.8728e−1 (1.37e−2)	1.9243e+0 (3.71e−1) −	8.8193e−1 (3.36e−2)
	8	1.4476e+1 (4.54e+0) −	8.6706e−1 (1.30e−2)	2.3395e+1 (4.94e+0) −	9.9407e−1 (3.43e−2)	1.1891e+0 (4.73e−2) −	9.8836e−1 (4.00e−2)	1.6798e+0 (4.31e−1) −	9.6419e−1 (2.34e−2)
	10	1.7814e+1 (4.24e+0) −	9.3563e−1 (1.43e−2)	2.0990e+1 (4.33e+0) −	1.0388e+0 (2.23e−2)	1.3057e+0 (2.66e−1) −	1.0271e+0 (1.34e−2)	2.1837e+0 (5.18e−1) −	1.0121e+0 (2.08e−2)
	15	3.4458e+1 (1.51e+1) −	9.9579e−1 (1.18e−2)	1.7657e+1 (2.35e+0) −	1.0793e+0 (1.60e−2)	1.3837e+0 (1.75e−1) −	1.1273e+0 (1.78e−2)	1.9267e+0 (4.02e−1) −	1.1251e+0 (1.98e−2)
LSMOP6	5	7.5065e+1 (1.75e+2) −	1.3280e+0 (9.69e−2)	8.2077e+4 (2.48e+4) −	1.3449e+0 (1.12e−1)	9.9254e+0 (5.69e+0) −	1.4642e+0 (4.41e−2)	1.8866e+1 (2.27e+1) −	1.2467e+0 (1.04e−1)
	8	2.8072e+3 (3.56e+3) −	1.2776e+0 (3.22e−2)	2.0594e+0 (6.50e−2) −	1.3564e+0 (9.13e−2)	1.2291e+0 (5.61e−2) +	1.5232e+0 (4.30e−2)	1.7385e+0 (3.90e−2) −	1.4171e+0 (7.10e−2)
	10	2.0420e+4 (1.90e+4) −	1.1782e+0 (1.39e−2)	1.5651e+0 (6.28e−3) −	1.3537e+0 (5.16e−2)	1.2196e+0 (6.28e−2) +	1.3475e+0 (4.99e−2)	1.4780e+0 (2.18e−2) −	1.2457e+0 (2.04e−2)
	15	1.4513e+5 (6.24e+4) −	1.9720e+0 (1.55e−1)	3.6873e+4 (2.04e+4) −	3.0766e+0 (6.13e−1)	3.1875e+1 (3.03e+1) −	2.1602e+0 (1.21e−1)	1.3593e+2 (1.15e+2) −	4.0714e+0 (7.42e−1)
LSMOP7	5	1.4020e+3 (2.00e+3) −	1.3581e+0 (1.66e−1)	3.3148e+0 (1.32e−1) −	9.1231e−1 (2.08e−1)	1.0278e+0 (6.19e−2) +	1.9897e+0 (3.03e−1)	2.6476e+0 (2.57e−1) −	8.9563e−1 (1.36e−1)
	8	2.3814e+3 (3.61e+3) −	1.6927e+0 (1.37e−1)	8.5588e+4 (2.94e+4) −	1.9236e+0 (3.78e−1)	7.3386e+0 (4.58e+0) −	1.7602e+0 (2.59e−2)	4.2711e+1 (5.48e+1) −	1.8432e+0 (4.45e−1)
	10	1.2044e+4 (8.30e+3) −	1.9707e+0 (1.94e−1)	6.3842e+4 (3.11e+4) −	3.5616e+0 (6.53e−1)	4.1933e+1 (2.79e+1) −	2.0363e+0 (1.92e−1)	1.6149e+2 (1.58e+2) −	3.3562e+0 (4.76e−1)
	15	1.7468e+4 (2.49e+4) −	1.4712e+0 (5.46e−2)	1.8715e+0 (2.66e−2) −	1.4452e+0 (3.83e−2)	1.3687e+0 (4.32e−2) +	1.6150e+0 (6.00e−2)	1.7830e+0 (1.33e−2) −	1.5769e+0 (5.74e−2)
LSMOP8	5	1.1605e+0 (7.39e−3) −	9.4163e−1 (1.97e−2)	1.2163e+0 (9.07e−3) −	9.1228e−1 (1.89e−2)	1.1024e+0 (2.61e−2) −	8.2411e−1 (2.97e−2)	9.1439e−1 (4.25e−2) +	9.4098e−1 (1.77e−2)
	8	4.8923e+0 (1.76e+0) −	7.9252e−1 (7.95e−3)	1.9206e+1 (5.11e+0) −	8.3372e−1 (2.11e−2)	9.5746e−1 (6.03e−2) =	9.6437e−1 (3.64e−2)	1.3326e+0 (3.16e−1) −	8.2833e−1 (1.29e−2)
	10	8.0999e+0 (1.62e+0) −	8.1279e−1 (7.62e−3)	1.6039e+1 (3.36e+0) −	8.8660e−1 (1.61e−2)	1.2188e+0 (1.09e−1) −	9.8756e−1 (3.43e−2)	1.9429e+0 (3.67e−1) −	8.7949e−1 (1.54e−2)
	15	9.7282e+0 (6.10e+0) −	1.0512e+0 (1.45e−2)	1.3272e+0 (1.13e−2) −	1.0028e+0 (1.58e−2)	1.1632e+0 (7.31e−2) =	1.1378e+0 (1.45e−2)	1.2092e+0 (2.49e−2) −	1.0070e+0 (4.08e−2)
LSMOP9	5	1.7065e+1 (2.39e+0) −	1.0815e+0 (3.97e−1)	5.6095e+1 (4.21e+0) −	1.0541e+0 (4.04e−2)	1.0552e+2 (6.33e+1) −	1.5334e+0 (5.46e−1)	1.6679e+0 (6.54e−1) −	1.1778e+0 (1.53e−1)
	8	1.2944e+2 (1.39e+1) −	1.1627e+1 (1.87e+0)	8.9970e+1 (6.96e+0) −	1.9941e+1 (2.44e+0)	4.4005e+2 (9.70e+1) −	1.1956e+1 (1.47e+0)	8.9316e+0 (2.96e+0) +	1.4706e+1 (2.72e+0)
	10	3.2505e+2 (2.48e+1) −	3.2538e+1 (3.60e+0)	1.3701e+2 (1.71e+1) −	5.6380e+1 (5.81e+0)	7.6145e+2 (1.31e+2) −	3.3534e+1 (2.50e+0)	1.4778e+1 (4.06e+0) +	5.5902e+1 (9.05e+0)
	15	1.7218e+3 (2.58e+2) −	2.1087e+2 (1.22e+1)	4.3327e+2 (4.29e+1) −	2.9985e+2 (1.54e+1)	2.1713e+3 (1.80e+2) −	2.0933e+2 (1.42e+1)	5.7676e+1 (8.54e+0) +	3.7025e+2 (8.41e+1)
DTLZ1	5	5.2318e+3 (4.11e+2) +	6.3974e+3 (2.57e+2)	3.6247e+3 (2.18e+3) −	9.2711e+2 (9.34e+2)	2.4830e+3 (2.90e+2) +	9.7504e+3 (2.72e+2)	1.4750e+3 (2.81e+2) +	1.7629e+3 (2.56e+2)
	8	1.4188e+4 (5.63e+2) +	1.4804e+4 (3.76e+2)	9.3525e+3 (1.85e+3) −	2.5424e+3 (1.22e+3)	4.8598e+3 (5.47e+2) +	1.0995e+4 (2.32e+3)	2.4073e+3 (7.75e+2) +	2.7551e+3 (4.14e+2)
	10	2.4796e+4 (2.97e+2) −	2.1677e+4 (7.67e+2)	1.5452e+4 (1.26e+3) −	6.8404e+3 (2.22e+3)	1.4466e+4 (1.44e+3) −	1.0006e+4 (1.31e+3)	3.0474e+3 (7.24e+2) +	3.8695e+3 (6.39e+2)
	15	2.7087e+4 (2.30e+3) −	2.2045e+4 (1.50e+3)	2.2258e+4 (1.12e+3) −	8.9926e+3 (2.06e+3)	1.5884e+4 (1.73e+3) −	1.1920e+4 (6.10e+2)	4.3502e+3 (1.36e+3) +	5.1473e+3 (8.97e+2)
DTLZ2	5	1.8515e+0 (1.55e−1) −	3.0509e−1 (7.89e−3)	1.3944e+1 (4.06e+0) −	1.1002e+0 (1.07e−1)	1.2550e+0 (8.55e−2) −	4.2218e−1 (2.21e−2)	2.1487e+0 (3.33e−1) −	8.1060e−1 (7.32e−2)
	8	1.5357e+1 (3.25e+0) −	1.1134e+0 (4.63e−2)	5.7054e+1 (1.66e+1) −	2.5700e+0 (2.08e−1)	6.2463e+0 (5.12e−1) −	1.7228e+0 (1.87e−1)	3.4335e+0 (1.52e+0) −	2.3487e+0 (2.97e−1)
	10	4.6170e+1 (2.22e+0) −	2.5841e+0 (1.78e−1)	9.4840e+1 (2.04e+1) −	2.6474e+0 (2.19e−1)	1.4684e+1 (5.54e−1) −	2.1996e+0 (1.68e−1)	4.3644e+0 (9.58e−1) −	2.3315e+0 (2.49e−1)
	15	9.9778e+1 (4.30e+0) −	2.2487e+0 (2.07e−1)	1.2995e+2 (2.81e+1) −	4.4657e+0 (3.75e−1)	4.8697e+1 (2.06e+0) −	4.1122e+0 (4.97e−1)	6.6090e+0 (2.75e+0) −	4.9890e+0 (5.70e−1)
DTLZ3	5	2.1398e+4 (1.22e+3) +	2.5628e+4 (7.95e+2)	6.9407e+3 (4.27e+3) −	1.1134e+3 (7.62e+2)	1.4389e+4 (1.43e+3) +	4.0358e+4 (6.27e+2)	5.4622e+3 (9.72e+2) +	6.7557e+3 (5.27e+2)
	8	4.9499e+4 (4.46e+3) =	5.0832e+4 (1.01e+3)	2.1928e+4 (1.46e+3) −	7.3332e+3 (7.00e+3)	3.2543e+4 (2.12e+3) +	6.7336e+4 (1.12e+3)	8.9005e+3 (1.46e+3) +	1.0010e+4 (1.53e+3)
	10	8.3721e+4 (1.34e+3) −	8.0272e+4 (9.45e+2)	3.2945e+4 (4.79e+3) =	2.8274e+4 (1.12e+4)	7.2902e+4 (1.19e+3) +	8.5532e+4 (2.70e+3)	1.1452e+4 (1.77e+3) +	1.2871e+4 (1.44e+3)
	15	1.4031e+5 (3.62e+3) −	1.2928e+5 (7.81e+2)	7.0111e+4 (1.47e+4) −	3.1405e+4 (1.52e+4)	1.2048e+5 (2.66e+3) −	1.1364e+5 (1.36e+4)	1.6268e+4 (2.75e+3) =	1.6869e+4 (1.92e+3)
DTLZ4	5	2.7074e+0 (6.33e−1) −	3.4005e−1 (1.81e−2)	7.1483e+0 (2.05e+0) −	1.9404e+0 (2.88e−1)	2.2139e+0 (6.07e−1) −	4.9268e−1 (6.62e−2)	3.1144e+0 (7.54e−1) −	2.6333e+0 (6.03e−1)
	8	1.5336e+1 (7.12e+0) −	1.0880e+0 (8.85e−2)	5.3211e+1 (1.32e+1) −	2.9224e+0 (3.53e−1)	7.7586e+0 (6.27e−1) −	1.0837e+0 (1.39e−1)	6.1322e+0 (1.12e+0) −	4.2157e+0 (8.44e−1)
	10	3.8959e+1 (4.52e+0) −	3.5156e+0 (1.86e−1)	6.7761e+1 (6.09e+0) −	2.8394e+0 (2.98e−1)	1.7780e+1 (7.50e−1) −	1.2652e+0 (8.59e−2)	6.2307e+0 (7.87e−1) −	3.9178e+0 (4.64e−1)
	15	8.2297e+1 (5.88e+0) −	3.1585e+0 (2.24e−1)	9.7379e+1 (9.89e+0) −	3.9346e+0 (3.42e−1)	5.5569e+1 (1.61e+0) −	1.6921e+0 (1.12e−1)	9.3086e+0 (1.07e+0) −	4.4540e+0 (6.26e−1)
DTLZ5	5	2.7705e+0 (2.04e−1) −	1.5628e−1 (5.90e−3)	2.2153e+1 (6.41e+0) −	9.2815e−1 (1.06e−1)	1.0134e+0 (9.27e−2) −	1.4929e−1 (1.12e−2)	1.9836e+0 (4.55e−1) −	6.6876e−1 (6.67e−2)
	8	2.1670e+1 (9.74e+0) −	4.7670e−1 (5.22e−2)	4.7562e+1 (1.42e+1) −	2.1576e+0 (2.64e−1)	5.1718e+0 (4.93e−1) −	9.7944e−1 (2.87e−1)	2.5578e+0 (1.40e+0) =	1.9829e+0 (2.09e−1)
	10	6.0988e+1 (8.47e+0) −	1.0531e+0 (1.64e−1)	6.3597e+1 (2.00e+1) −	2.0542e+0 (1.57e−1)	1.2684e+1 (6.02e−1) −	1.3954e+0 (3.38e−1)	4.1656e+0 (1.19e+0) −	1.9647e+0 (2.33e−1)
	15	1.1053e+2 (3.81e+0) −	1.5642e+0 (1.57e−1)	1.0515e+2 (3.30e+1) −	4.0168e+0 (4.37e−1)	4.6521e+1 (2.73e+0) −	3.4871e+0 (4.79e−1)	5.5178e+0 (2.91e+0) −	4.6881e+0 (5.87e−1)
+/−/≈		3/52/1		0/55/1		9/44/3		12/40/4	

4.3. Improvements to the original algorithms

In order to verify the effectiveness of the UCLMO framework, this section selects four benchmark optimizers NSGA-III [43], RVEA [44], SMPSO [45] and LMOCSO [26] for embedding. After the optimizer is embedded into our framework UCLMO, the final algorithms are denoted as UC-NSGA-III, UC-RVEA, UC-SMPSO, and UC-LMOCSO, respectively. The comparison results are shown in Table 3 (the optimal results are shaded in gray).

It can be seen from Table 3 that for most of the test problems, UCLMO can better improve the performance of the original algorithm. Compared with the four original algorithms, UC-NSGA-III, UC-RVEA, UC-SMPSO, and UC-LMOCSO achieve significantly better optimization performance on 52, 55, 44, and 40 test problems over all 56 test cases, respectively. For some test problems, there are still phenomena that are inferior to the original algorithm. For example, on LSMOP7, DTLZ1, and DTLZ3 test problems, the performance of UC-NSGA-III and UC-RVEA is slightly lower than that of the original algorithm, while UC-SMPSO is better than the original algorithm. The main reason may be that the fitness landscape of these problems is too complicated, and the individuals generated by NSGA-III and RVEA have insufficient diversity in the decision-making space, so cultural assistance does not have a positive effect on the evolution. SMPSO uses particle swarm optimization to increase the diversity of the population in the decision space, which makes the positive guidance effect by cultural assistance.

Fig. 2 shows the convergence curves of the original algorithms and the corresponding algorithms embedded into our UCLMO on some LSMOPs and DTLZ4 test problems with $M = 8$, in which the abscissa represents the number of function evolutions (FEs), and the ordinate represents the IGD value. As can be seen from Fig. 2, UCLMO can significantly improve the performance of the original algorithm, and the search efficiency is also improved.

4.4. Comparisons with state-of-the-art algorithms on LSMOP and DTLZ

In this section, to validate the performance of the proposed UCLMO, CCGDE3, LMOCSO, LSMaODE, CVEA3, AMPDEA, BCE-IBEA, GLMO, DGEA, WOF, and FDV are selected for comparison. Among them, the first five algorithms are state-of-the-art representative algorithms, and the last five algorithms are state-of-the-art frameworks that can be integrated with different MOEAs. CVEA3, AMPDEA, and BCE-IBEA are the winners of the 2018 IEEE CEC competition on many-objective optimization. NSGA-III is chosen as the benchmark optimizer, and embedded into our UCLMO. For a fair comparison, NSGA-III is also integrated with GLMO, DGEA, WOF, and FDV. The experimental results are shown in Table 4 (the optimal results are shaded).

It can be seen from Table 4 that among the 56 test cases, UCLMO outperforms CCGDE3, LMOCSO, LSMaODE, CVEA3, AMPDEA, BCE-IBEA, GLMO, DGEA, WOF, and FDV in 53, 42, 43, 44, 56, 43, 37, 55, 43 and 47 test cases, respectively. In the test of LSMOP1-9 and DTLZ1-5, UCLMO is slightly better than GLMO and

Table 4
Comparisons of IGD results obtained by UCLMO and 10 state-of-the-art algorithms on LSMOPs and DTLZs.

Problem	M	CCGDE3	LMOCSO	LSMaODE	CVEA3	AMPDEA	BCE-IBEA	GLMO	DGEA	WOF	FDV	UCLMO
LSMOP1	5	1.1941e+1 (1.17e+0) −	1.1490e+0 (2.13e−1) −	4.3088e−1 (5.89e−2) +	9.0180e−1 (7.46e−2) −	2.5349e+1 (1.86e+1) −	2.1164e+0 (4.31e−1) −	7.6894e−1 (9.21e−2) −	1.0523e+1 (8.04e−1) −	8.6782e−1 (2.13e−2) −	2.5130e+0 (4.10e−1) −	5.1466e−1 (3.40e−2)
	8	1.2326e+1 (1.27e+0) −	8.7846e−1 (1.99e−1) −	1.1684e+0 (3.48e−1) −	1.2863e+0 (1.94e−1) −	3.6123e+1 (1.95e+1) −	3.0358e+0 (6.69e−1) −	1.0168e+0 (1.95e−1) −	1.0428e+1 (9.78e−1) −	1.1248e+0 (4.22e−1) −	3.6361e+0 (8.11e−1) −	4.8085e−1 (2.85e−2)
	10	1.2109e+1 (1.23e+0) −	9.6196e−1 (1.44e−1) −	3.2574e+0 (3.80e−1) −	1.8402e+0 (3.38e−1) −	4.2199e+1 (2.85e+1) −	3.7613e+0 (9.53e−1) −	1.1231e+0 (1.97e−1) −	9.7661e+0 (8.69e−1) −	2.1035e+0 (1.28e+0) −	5.0978e+0 (7.87e−1) −	3.7839e−1 (1.01e−2)
	15	1.2117e+1 (1.03e+0) −	1.0202e+0 (2.14e−1) −	5.4200e+0 (6.17e−1) −	2.1608e+0 (9.19e−1) −	4.7308e+1 (3.04e+1) −	4.8851e+0 (6.29e−1) −	1.1101e+0 (2.73e−1) −	1.1157e+1 (9.57e−1) −	5.8756e+0 (2.18e+0) −	5.3859e+0 (6.08e−1) −	4.0678e−1 (1.32e−2)
LSMOP2	5	1.8731e−1 (6.20e−3) −	1.7332e−1 (1.37e−3) −	1.5441e−1 (2.57e−3) +	1.7294e−1 (4.39e−3) −	7.2625e−1 (1.81e−1) −	1.5031e−1 (4.26e−3) +	1.7213e−1 (1.28e−3) −	2.0868e−1 (1.36e−2) −	1.7756e−1 (1.71e−3) −	1.7312e−1 (5.76e−4) −	1.6888e−1 (5.20e−4)
	8	3.0318e−1 (1.14e−2) −	5.6534e−1 (2.47e−1) −	3.5672e−1 (2.93e−2) −	2.9156e−1 (6.28e−3) −	1.2580e+0 (6.37e−1) −	3.4681e−1 (1.07e−1) −	3.0419e−1 (2.07e−3) −	4.2955e−1 (3.05e−2) −	3.1309e−1 (2.25e−3) −	2.9980e−1 (2.00e−3) −	2.8495e−1 (1.38e−3)
	10	3.0512e−1 (1.04e−2) −	6.3464e−1 (2.19e−1) −	4.0695e−1 (1.82e−2) −	2.8261e−1 (1.09e−2) +	9.7036e−1 (3.05e−1) −	2.5905e−1 (2.75e−2) +	3.3610e−1 (2.12e−3) −	4.4813e−1 (2.11e−2) −	3.4118e−1 (5.65e−3) −	3.3640e−1 (3.20e−3) −	2.9566e−1 (4.32e−3)
	15	3.6695e−1 (2.69e−2) −	4.6666e−1 (1.14e−1) −	4.9730e−1 (2.27e−2) −	4.1263e−1 (2.16e−2) −	8.6548e−1 (1.71e−1) −	3.3808e−1 (5.14e−2) −	2.9092e−1 (2.05e−2) −	5.2832e−1 (2.10e−2) −	2.8999e−1 (1.28e−2) −	2.8771e−1 (4.86e−3) −	2.6895e−1 (1.17e−3)
LSMOP3	5	2.7790e+1 (4.28e+0) −	1.1577e+1 (5.05e+0) −	1.3260e+1 (9.32e+0) −	3.7121e+0 (5.08e−1) −	4.9818e+4 (3.17e+4) −	8.5523e+0 (2.83e+0) −	8.2923e+0 (6.87e−1) −	1.4123e+3 (1.13e+3) −	1.8060e+0 (1.45e+0) −	8.7732e+0 (8.04e−1) −	1.7525e+0 (3.35e−1)
	8	3.2092e+1 (3.72e+0) −	7.2008e+0 (7.48e+0) =	3.8586e+2 (5.24e+2) −	3.9042e+0 (1.50e+0) −	2.6714e+5 (2.93e+5) −	2.6233e+1 (3.81e+0) −	8.9582e+0 (5.93e+0) −	8.1264e+3 (4.27e+3) −	4.5282e+0 (3.04e+0) −	1.9225e+1 (2.41e+1) −	2.2597e+0 (6.47e−1)
	10	3.4273e+1 (4.23e+0) −	8.7782e+0 (8.86e+0) =	2.3731e+3 (1.81e+3) −	1.2545e+1 (4.91e+0) −	3.8446e+5 (4.39e+5) −	2.3933e+1 (4.25e+0) −	1.1268e+1 (4.15e+0) −	6.1081e+3 (3.05e+3) −	1.4251e+1 (6.49e+0) −	1.5517e+1 (1.57e+0) −	2.5988e+0 (3.26e−1)
	15	3.8808e+1 (6.06e+0) −	2.4558e+1 (6.15e+0) −	1.8732e+3 (1.14e+3) −	2.2584e+1 (1.03e+1) −	5.4939e+4 (4.09e+4) −	3.0350e+1 (9.58e+0) −	8.9836e+0 (1.75e+0) −	2.3436e+3 (1.80e+3) −	2.0453e+1 (6.46e+0) −	2.0995e+1 (2.75e+0) −	3.4351e+0 (2.60e−1)
LSMOP4	5	3.5326e−1 (1.64e−2) −	2.9089e−1 (3.40e−3) −	2.8689e−1 (1.22e−2) =	2.3600e−1 (6.73e−3) +	1.8200e+0 (9.71e−1) −	2.2819e−1 (5.92e−3) +	2.9495e−1 (6.81e−3) −	3.9433e−1 (2.83e−2) −	2.9718e−1 (6.39e−3) −	2.9754e−1 (5.33e−3) −	2.8688e−1 (1.50e−3)
	8	3.5369e−1 (1.33e−2) −	4.7422e−1 (1.49e−1) −	3.5682e−1 (1.84e−2) −	3.2219e−1 (5.57e−3) +	9.2802e−1 (1.35e−1) −	2.8475e−1 (5.35e−2) +	3.4664e−1 (4.14e−3) −	4.3322e−1 (2.17e−2) −	3.6895e−1 (3.42e−3) −	3.4775e−1 (3.55e−3) −	3.2594e−1 (2.54e−3)
	10	3.4562e−1 (1.66e−2) −	5.3421e−1 (1.97e−1) −	3.9880e−1 (1.64e−2) −	2.9924e−1 (8.45e−3) +	8.4789e−1 (1.62e−1) −	2.5650e−1 (9.20e−3) +	3.7664e−1 (7.16e−3) −	4.3604e−1 (1.70e−2) −	3.7162e−1 (1.12e−2) −	3.6934e−1 (4.73e−3) −	3.1256e−1 (4.47e−3)
	15	3.8632e−1 (1.78e−2) −	6.7432e−1 (2.40e−1) −	5.0877e−1 (2.61e−2) −	4.1847e−1 (1.82e−2) −	1.1163e+0 (3.58e−1) −	3.6249e−1 (6.72e−2) −	3.3443e−1 (3.15e−2) −	5.3621e−1 (2.21e−2) −	3.3503e−1 (3.05e−2) −	3.1540e−1 (7.04e−3) −	2.7627e−1 (2.36e−3)
LSMOP5	5	1.7822e+1 (3.23e+0) −	1.9243e+0 (3.71e−1) −	7.7899e−1 (9.45e−2) +	1.5558e+0 (2.36e−1) −	5.5022e+1 (4.38e+1) −	5.7111e+0 (1.43e+0) −	4.3637e−1 (5.77e−2) +	1.4777e+1 (1.33e+0) −	4.2202e−1 (1.96e−2) +	6.1392e+0 (1.14e+0) −	8.6487e−1 (2.83e−2)
	8	1.9015e+1 (2.58e+0) −	1.6798e+0 (4.31e−1) −	1.9570e+0 (4.88e−1) −	2.2778e+0 (5.49e−1) −	3.4957e+1 (2.28e+1) −	1.1099e+1 (2.95e+0) −	9.1043e−1 (7.40e−1) −	1.4818e+1 (9.88e−1) −	5.5972e+0 (4.35e+0) −	1.1838e+1 (1.16e+0) −	8.6706e−1 (1.30e−2)
	10	1.9056e+1 (1.82e+0) −	2.1837e+0 (5.18e−1) −	4.5849e+0 (8.62e−1) −	3.5223e+0 (1.15e+0) −	4.8831e+1 (3.59e+1) −	8.7662e+0 (2.35e+0) −	2.3523e+0 (1.07e+0) −	1.5388e+1 (6.35e−1) −	7.7369e+0 (5.67e+0) −	1.5879e+1 (1.05e+0) −	9.3563e−1 (1.43e−2)
	15	1.7368e+1 (1.61e+0) −	1.9267e+0 (4.02e−1) −	6.3505e+0 (9.01e−1) −	3.0924e+0 (1.53e+0) −	3.6826e+1 (2.52e+1) −	1.3237e+1 (3.90e+0) −	5.0206e+0 (1.77e+0) −	1.4215e+1 (5.93e−1) −	1.5090e+1 (6.13e+0) −	1.0175e+1 (3.17e+0) −	9.9579e−1 (1.18e−2)
LSMOP6	5	4.7940e+4 (1.65e+4) −	1.8866e+1 (2.27e+1) −	2.3843e+1 (1.48e+1) −	2.9232e+1 (2.29e+1) −	2.6789e+5 (2.87e+5) −	2.6960e+2 (1.42e+3) −	8.1632e+0 (5.65e+0) −	1.7701e+4 (4.98e+3) −	1.1780e+0 (6.96e−2) +	2.1528e+1 (1.47e+1) −	1.3280e+0 (9.69e−2)
	8	2.0115e+0 (8.35e−2) −	1.7385e+0 (3.90e−2) −	2.1468e+3 (4.02e+3) −	5.0304e+1 (1.19e+2) −	1.0759e+5 (1.42e+5) −	3.4031e+2 (1.57e+3) −	1.3438e+0 (1.39e−1) −	1.5826e+3 (1.48e+3) −	1.3314e+1 (4.20e+1) −	1.7391e+0 (4.09e−2) −	1.2776e+0 (3.22e−2)
	10	1.5652e+0 (6.48e−3) −	1.4780e+0 (2.18e−2) −	2.0544e+3 (1.57e+3) −	7.2389e+1 (1.70e+2) −	1.2127e+5 (2.15e+5) −	8.2924e+3 (2.05e+4) −	1.8299e+0 (2.59e−1) −	1.1750e+3 (1.22e+3) −	6.7898e+3 (2.41e+4) −	2.9512e+2 (1.19e+3) −	1.1782e+0 (1.39e−2)
	15	2.3258e+4 (6.48e+3) −	1.3593e+2 (1.15e+2) −	3.9316e+3 (1.53e+3) −	3.2164e+3 (4.68e+3) −	6.2934e+4 (4.41e+4) −	3.3742e+4 (1.63e+4) −	2.3110e+1 (3.69e+1) −	1.6510e+4 (2.04e+3) −	1.3377e+4 (1.20e+4) −	1.9414e+3 (2.97e+3) −	1.9720e+0 (1.55e−1)
LSMOP7	5	1.2277e+2 (6.55e+2) −	2.6476e+0 (2.57e−1) −	1.0813e+2 (1.93e+2) −	1.9479e+0 (3.74e−1) −	3.3157e+5 (3.39e+5) −	6.4046e+1 (2.14e+2) −	1.1831e+0 (2.98e−2) +	1.8483e+3 (1.79e+3) −	1.2483e+0 (1.92e−1) +	2.8307e+0 (1.37e−1) −	1.3581e+0 (1.66e−1)
	8	3.3436e+4 (1.67e+4) −	4.2711e+1 (5.48e+1) −	3.3716e+2 (2.31e+2) −	3.3193e+2 (3.94e+2) −	1.3873e+5 (2.37e+5) −	1.8498e+4 (2.23e+4) −	1.7752e+0 (1.20e+0) −	1.7941e+4 (4.19e+3) −	7.0090e+2 (1.39e+3) −	2.5682e+2 (4.34e+2) −	1.6927e+0 (1.37e−1)
	10	3.3519e+4 (1.53e+4) −	1.6149e+2 (1.58e+2) −	2.2752e+3 (7.92e+2) −	2.1119e+3 (2.00e+3) −	1.6068e+5 (2.05e+5) −	5.1471e+4 (2.26e+4) −	2.3510e+1 (3.90e+1) =	1.8379e+4 (2.60e+3) −	9.1020e+3 (1.45e+4) −	2.1623e+3 (1.89e+3) −	1.9707e+0 (1.94e−1)
	15	1.2267e+3 (3.05e+3) −	1.7830e+0 (1.33e−2) −	1.4791e+3 (1.01e+3) −	8.6859e+1 (2.21e+2) −	5.4260e+4 (9.11e+4) −	1.4509e+4 (1.47e+4) −	4.0459e+0 (7.60e+0) −	1.0540e+3 (1.02e+3) −	7.4817e+2 (2.60e+3) −	5.2193e+1 (2.76e+2) −	1.4712e+0 (5.46e−2)
LSMOP8	5	1.2061e+0 (2.07e−2) −	9.1439e−1 (4.25e−2) +	9.9081e−1 (1.73e−1) =	9.5982e−1 (7.11e−2) −	2.5603e+1 (2.28e+1) −	1.2385e+0 (2.47e−1) −	3.3217e−1 (6.57e−3) +	1.0428e+0 (5.23e−2) −	3.5269e−1 (4.92e−2) +	1.1625e+0 (1.37e−2) −	9.4163e−1 (1.97e−2)
	8	1.3897e+1 (2.66e+0) −	1.3326e+0 (3.16e−1) −	1.4640e+0 (1.82e−1) −	1.4353e+0 (2.19e−1) −	2.4545e+1 (1.60e+1) −	6.7559e+0 (2.00e+0) −	7.4797e−1 (4.03e−1) +	9.2939e+0 (8.60e−1) −	1.7009e+0 (1.09e+0) −	3.4025e+0 (1.29e+0) −	7.9252e−1 (7.95e−3)
	10	1.2896e+1 (2.12e+0) −	1.9429e+0 (3.67e−1) −	3.1406e+0 (3.41e−1) −	1.9621e+0 (6.06e−1) −	1.9842e+1 (1.03e+1) −	5.6228e+0 (2.68e+0) −	1.5970e+0 (7.61e−1) −	9.8789e+0 (5.56e−1) −	3.0928e+0 (2.81e+0) −	6.4348e+0 (8.93e−1) −	8.1279e−1 (7.62e−3)
	15	1.3225e+0 (9.36e−3) −	1.2092e+0 (2.49e−2) −	1.3849e+0 (1.74e−1) −	1.2812e+0 (5.51e−2) −	1.5235e+1 (1.32e+1) −	1.7331e+0 (9.34e−1) −	2.3705e+0 (8.33e−1) −	1.2638e+0 (1.94e−2) −	2.1719e+0 (2.37e+0) −	1.4064e+0 (4.56e−1) −	1.0512e+0 (1.45e−2)
LSMOP9	5	1.7649e+2 (2.35e+1) −	1.6679e+0 (6.54e−1) −	2.7957e+0 (1.59e+0) −	7.8672e+0 (6.39e−1) −	5.0678e+2 (2.24e+2) −	7.6054e+0 (3.22e−1) −	2.3304e+0 (5.74e−1) −	3.2848e+2 (1.17e+1) −	1.9192e+0 (6.85e−2) −	1.0191e+1 (1.29e+0) −	1.0815e+0 (3.97e−1)
	8	5.1525e+2 (6.83e+1) −	8.9316e+0 (2.96e+0) +	4.0857e+1 (1.20e+1) −	6.2790e+1 (1.34e+1) −	1.3835e+3 (5.54e+2) −	7.9386e+1 (6.92e+0) −	5.0343e+0 (4.53e−1) +	7.7409e+2 (2.32e+1) −	4.4962e+0 (8.89e−1) +	5.7999e+1 (1.01e+1) −	1.1627e+1 (1.87e+0)
	10	9.1507e+2 (1.21e+2) −	1.4778e+1 (4.06e+0) +	2.8433e+2 (2.44e+1) −	3.8491e+2 (2.87e+1) −	2.0685e+3 (1.04e+3) −	1.8772e+2 (2.53e+1) −	6.4942e+0 (5.94e−2) +	1.1985e+3 (4.32e+1) −	2.6173e+1 (1.47e+1) +	1.2672e+2 (1.04e+1) −	3.2538e+1 (3.60e+0)
	15	2.4237e+3 (1.58e+2) −	5.7676e+1 (8.54e+0) +	1.1454e+3 (1.35e+2) −	1.2167e+3 (1.22e+2) −	5.3590e+3 (2.52e+3) −	6.7049e+2 (4.66e+1) −	1.2050e+1 (8.00e−2) +	2.9311e+3 (7.12e+1) −	8.9768e+2 (4.90e+2) −	5.6692e+2 (4.98e+1) −	2.1087e+2 (1.22e+1)
DTLZ1	5	9.9069e+3 (1.28e+3) −	1.4750e+3 (2.81e+2) +	9.7464e+2 (5.58e+1) +	3.1324e+3 (1.94e+2) +	1.7529e+4 (3.19e+3) −	1.2319e+3 (5.59e+1) +	5.6472e+3 (3.15e+2) +	1.2468e+4 (4.72e+2) −	2.8556e+3 (6.30e+2) +	5.2246e+1 (5.60e+0) +	6.3974e+3 (2.57e+2)
	8	1.4699e+4 (1.60e+3) =	2.4073e+3 (7.75e+2) +	2.3046e+3 (2.02e+2) +	6.3248e+3 (7.51e+2) +	2.9169e+4 (5.60e+3) −	2.7898e+3 (1.43e+2) +	1.2248e+4 (5.69e+2) +	1.7993e+4 (8.66e+2) −	9.9933e+3 (3.35e+3) +	2.5206e+2 (1.63e+1) +	1.4804e+4 (3.76e+2)
	10	1.7056e+4 (1.76e+3) +	3.0474e+3 (7.24e+2) +	5.4002e+3 (4.21e+2) +	1.0722e+4 (1.31e+3) +	3.5475e+4 (7.38e+3) −	9.1997e+3 (6.83e+2) +	1.7569e+4 (5.07e+2) +	2.1952e+4 (1.19e+3) =	1.9567e+4 (4.36e+3) +	7.5954e+2 (4.67e+1) +	2.1677e+4 (7.67e+2)
	15	2.3717e+4 (2.46e+3) −	4.3502e+3 (1.36e+3) +	1.2193e+4 (7.49e+2) +	1.9971e+4 (2.05e+3) +	5.3103e+4 (1.06e+4) −	1.8840e+4 (1.44e+3) +	2.1632e+4 (2.43e+3) =	3.2791e+4 (1.32e+3) −	2.6086e+4 (2.36e+3) −	1.5423e+3 (1.02e+2) +	2.2045e+4 (1.50e+3)
DTLZ2	5	2.9400e+1 (5.74e+0) −	2.1487e+0 (3.33e−1) −	3.5655e−1 (1.81e−2) −	2.2567e+0 (2.11e−1) −	3.8649e+1 (1.06e+0) −	4.2299e−1 (3.15e−2) −	2.3063e−1 (4.32e−3) +	3.7431e+1 (4.73e−1) −	4.0869e+0 (6.91e−1) −	1.1599e+0 (9.94e−2) −	3.0509e−1 (7.89e−3)
	8	5.9695e+1 (6.99e+0) −	3.4335e+0 (1.52e+0) −	1.3509e+0 (8.17e−2) −	7.1452e+0 (5.91e−1) −	6.3112e+1 (8.49e−1) −	4.1201e+0 (4.81e−1) −	1.4166e+0 (4.10e−1) −	6.1426e+1 (8.85e−1) −	5.0063e+1 (7.64e+0) −	6.5432e+0 (5.70e−1) −	1.1134e+0 (4.63e−2)
	10	8.0576e+1 (6.71e+0) −	4.3644e+0 (9.58e−1) −	9.1148e+0 (1.50e+0) −	1.8832e+1 (1.62e+0) −	7.8712e+1 (1.15e+0) −	6.8128e+0 (4.02e−1) −	1.5916e+1 (5.68e+0) −	7.6771e+1 (7.55e−1) −	1.0704e+2 (1.27e+1) −	1.6908e+1 (1.86e+0) −	2.5841e+0 (1.78e−1)
	15	1.2281e+2 (5.91e+0) −	6.6090e+0 (2.75e+0) −	2.6005e+1 (3.44e+0) −	4.1498e+1 (3.48e+0) −	1.1893e+2 (1.59e+0) −	1.9696e+1 (1.15e+0) −	5.5047e+1 (2.78e+1) −	1.1671e+2 (7.18e−1) −	1.7120e+2 (1.50e+1) −	5.4008e+1 (2.92e+0) −	2.2487e+0 (2.07e−1)
DTLZ3	5	3.3233e+4 (4.61e+3) −	5.4622e+3 (9.72e+2) +	3.1474e+3 (7.84e+1) +	8.7225e+3 (6.73e+2) +	5.0718e+4 (7.93e+2) −	4.9178e+3 (2.00e+2) +	2.1624e+4 (1.56e+3) +	4.9762e+4 (8.67e+2) −	1.4382e+4 (4.38e+3) +	2.1460e+2 (1.83e+1) +	2.5628e+4 (7.95e+2)
	8	5.7433e+4 (6.44e+3) −	8.9005e+3 (1.46e+3) +	7.8672e+3 (2.05e+2) +	2.1369e+4 (1.17e+3) +	8.2325e+4 (8.24e+2) −	1.1680e+4 (4.01e+2) +	4.1421e+4 (2.12e+3) +	8.0650e+4 (8.03e+2) −	4.1222e+4 (1.33e+4) +	9.0803e+2 (2.08e+2) +	5.0832e+4 (1.01e+3)
	10	8.0977e+4 (6.13e+3) −	1.1452e+4 (1.77e+3) +	2.0822e+4 (4.25e+2) +	3.7847e+4 (2.33e+3) +	1.0337e+5 (1.22e+3) −	3.3839e+4 (2.78e+3) +	5.9647e+4 (2.07e+3) +	1.0118e+5 (7.24e+2) −	7.0074e+4 (1.31e+4) +	2.4455e+3 (2.38e+2) +	8.0272e+4 (9.45e+2)
	15	1.2223e+5 (8.57e+3) +	1.6268e+4 (2.75e+3) +	4.8842e+4 (7.66e+2) +	7.1689e+4 (4.19e+3) +	1.5549e+5 (1.67e+3) −	8.2317e+4 (2.98e+3) +	9.8023e+4 (5.68e+3) +	1.5361e+5 (9.29e+2) −	1.2320e+5 (1.78e+4) =	6.6186e+3 (2.46e+2) +	1.2928e+5 (7.81e+2)
DTLZ4	5	2.0287e+1 (6.09e+0) −	3.1144e+0 (7.54e−1) −	6.4625e−1 (7.91e−2) −	1.9866e+0 (4.38e−1) −	3.9025e+1 (8.23e−1) −	5.7899e−1 (5.07e−2) −	3.8180e−1 (1.81e−1) −	3.7951e+1 (7.22e−1) −	5.5462e+0 (1.06e+0) −	2.2433e−1 (1.53e−2) +	3.4005e−1 (1.81e−2)
	8	4.9138e+1 (8.52e+0) −	6.1322e+0 (1.12e+0) −	1.7524e+0 (8.15e−2) −	8.1309e+0 (8.67e−1) −	6.2932e+1 (1.53e+0) −	4.5765e+0 (4.13e−1) −	9.3057e−1 (7.81e−1) +	6.1520e+1 (9.51e−1) −	3.7189e+1 (4.76e+0) −	6.0410e+0 (6.34e−1) −	1.0880e+0 (8.85e−2)
	10	7.1498e+1 (7.80e+0) −	6.2307e+0 (7.87e−1) −	1.2155e+1 (1.79e+0) −	2.5602e+1 (2.07e+0) −	7.8867e+1 (1.38e+0) −	7.3047e+0 (7.24e−1) −	2.7881e+0 (8.25e−1) +	7.6951e+1 (9.52e−1) −	9.8401e+1 (1.46e+1) −	1.8555e+1 (2.82e+0) −	3.5156e+0 (1.86e−1)
	15	1.1976e+2 (5.08e+0) −	9.3086e+0 (1.07e+0) −	3.0040e+1 (4.18e+0) −	5.8193e+1 (2.20e+0) −	1.1922e+2 (1.37e+0) −	1.8513e+1 (1.01e+0) −	3.4153e+0 (1.77e+0) −	1.1689e+2 (1.16e+0) −	1.5887e+2 (2.12e+1) −	7.3948e+1 (1.71e+0) −	3.1585e+0 (2.24e−1)
DTLZ5	5	2.9221e+1 (6.44e+0) −	1.9836e+0 (4.55e−1) −	2.9867e−1 (1.32e−1) −	2.4352e+0 (2.57e−1) −	3.8530e+1 (9.02e−1) −	7.8644e−1 (1.67e−1) −	5.6840e−1 (1.05e−1) −	3.7265e+1 (7.68e−1) −	6.6706e+0 (1.39e+0) −	2.1405e+0 (1.64e−1) −	1.5628e−1 (5.90e−3)
	8	5.6352e+1 (7.64e+0) −	2.5578e+0 (1.40e+0) −	2.3857e+0 (8.62e−1) −	7.8150e+0 (5.90e−1) −	6.2187e+1 (1.32e+0) −	4.6098e+0 (5.37e−1) −	3.7067e+0 (1.79e+0) −	6.0992e+1 (7.73e−1) −	3.8488e+1 (6.88e+0) −	9.5933e+0 (4.39e+0) −	4.7670e−1 (5.22e−2)
	10	7.6493e+1 (9.81e+0) −	4.1656e+0 (1.19e+0) −	1.1489e+1 (2.39e+0) −	1.7093e+1 (9.01e−1) −	7.8675e+1 (9.22e−1) −	7.1966e+0 (4.73e−1) −	2.2330e+1 (1.00e+1) −	7.6079e+1 (1.09e+0) −	5.2638e+1 (1.14e+1) −	2.1601e+1 (3.84e+0) −	1.0531e+0 (1.64e−1)
	15	1.1996e+2 (1.07e+1) −	5.5178e+0 (2.91e+0) −	2.8171e+1 (5.19e+0) −	3.9039e+1 (4.29e+0) −	1.1869e+2 (1.95e+0) −	2.0423e+1 (9.55e−1) −	7.6279e+1 (2.33e+1) −	1.1597e+2 (1.32e+0) −	1.1471e+2 (3.43e+1) −	6.5441e+1 (2.60e+0) −	1.5642e+0 (1.57e−1)
+/−/=		2/53/1	12/42/2	11/43/2	12/44/0	0/56/0	13/43/0	17/37/2	0/55/1	12/43/1	9/47/0	

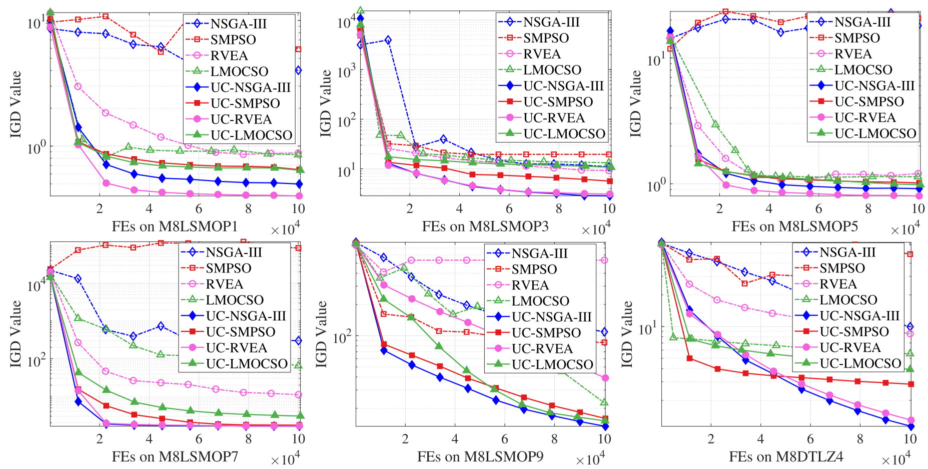
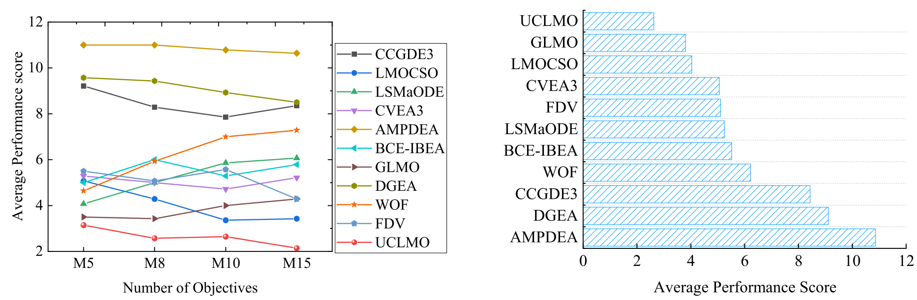


Fig. 2. Convergence curves of original algorithms and embedding algorithms on some LSMOPs and DTLZ4 test suites with $M = 8$. Dashed lines represent the original algorithms, while the embedded algorithms are depicted by solid lines of the same color as their corresponding original algorithms.



(a) Point-line chart on different objectives (b) Bar chart on all test suits

Fig. 3. Point-line and bar charts of the ranking statistics of all algorithms.

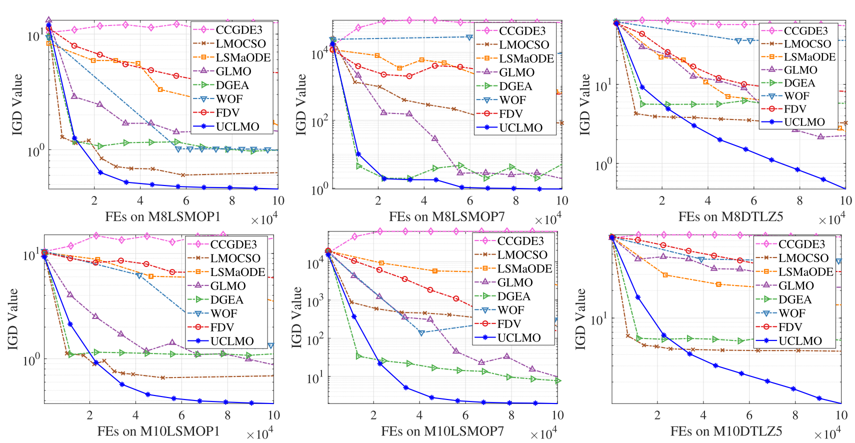


Fig. 4. Convergence curves of UCLMO and compared algorithms on LSMOP1, LSMOP7 and DTLZ5 test suites with $M = 8$ and $M = 10$.

FDV, and far better than CCGDE3, CVEA3, AMPDEA, BCE-IBEA, DGEA, LMOCSO, WOF, and LSMaODE.

4.4.1. Statistical analysis of rankings

To intuitively compare the performance of the algorithms, based on the analysis of the ranking statistics of all algorithms on MaOPs with 5, 8, 10, 15 objectives and all test suites, the point-line and bar charts are plotted, as shown in Fig. 3. In Fig. 3(a), the horizontal coordinate indicates the number of different objectives and the vertical coordinate indicates the average performance score of the algorithm. In Fig. 3(b), the horizontal coordinate indicates the average performance score of the algorithm, and the vertical coordinate indicates the algorithm. The smaller the average performance score, the better performance.

As shown in Fig. 3(a), with the increase in the number of objectives, UCLMO can still obtain the optimal ranking, indicating that UCLMO possesses a strong optimization ability for MaOPs with different-level objectives. It can be seen from Fig. 3(b) that UCLMO obtains the optimal performance on all test suits.

4.4.2. Convergence analysis

To verify the convergence pattern of UCLMO, Fig. 4 shows the IGD curves of UCLMO and other algorithms on LSMOP1, LSMOP7, and DTLZ5 with 8-, 10-objectives. It can be seen from Fig. 4 that

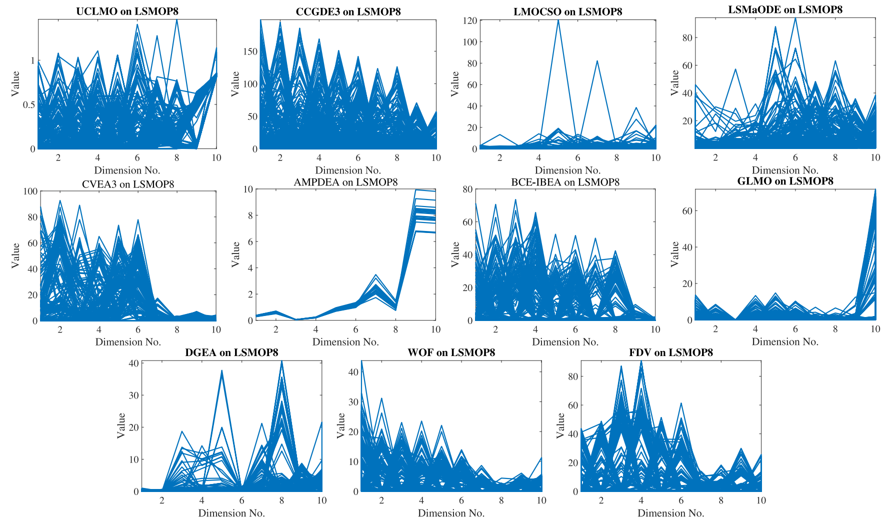


Fig. 5. Final solution set obtained by different algorithms on LSMOP8 with $M = 10$, shown by parallel coordinates.

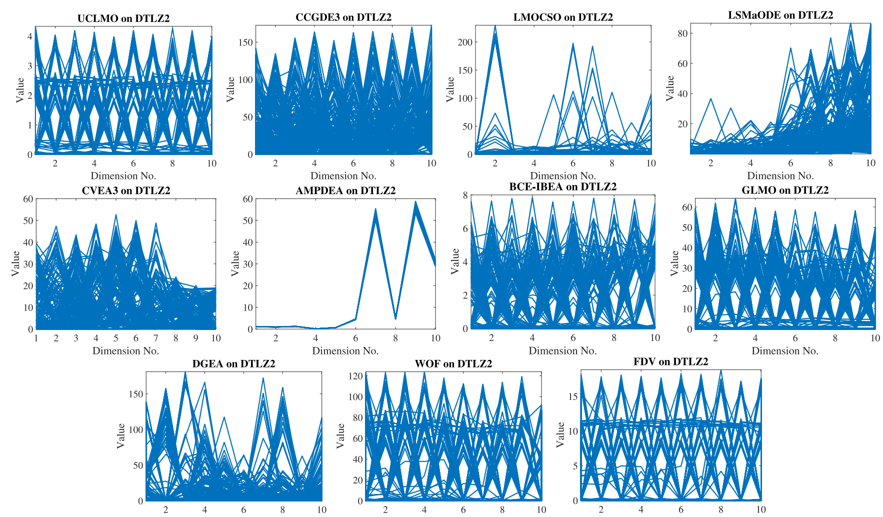


Fig. 6. Final solution set obtained by different algorithms on DTLZ2 with $M = 10$, shown by parallel coordinates.

UCLMO can converge to the optimal solution quickly compared with other algorithms, which further verifies the effectiveness of the proposed algorithm.

4.4.3. Statistical analysis of the final solution set

To better verify the performance of the algorithm, Fig. 5 shows the final solution set obtained by each algorithm after the same function evaluation on 10-objective LSMOP8 as a representative problem. The abscissa represents the dimension of the objective, and the ordinate represents the value of the objective. From Fig. 5, compared with the comparison algorithms, the solution set obtained by the proposed UCLMO has the best convergence and diversity.

Similarly, DTLZ2 is also selected from the DTLZ test suites. Fig. 6 shows the final solution set obtained by each algorithm on 10-objective DTLZ2. It can be concluded that the solution set obtained by the proposed UCLMO also has optimal convergence and diversity.

4.4.4. Non-parametric statistical analysis of Friedman test

To perform statistical tests, this section applies the Friedman test, which is a non-parametric statistical test. Table 5 provides a comprehensive overview of the results obtained from the Friedman test conducted for UCLMO and 10 other algorithms, based on the IGD, GD, and Δ_P indicators. Notably, smaller values of the IGD, GD, and Δ_P indicators indicate better performance. As shown in Table 5, UCLMO attains the best overall ranking in terms of the IGD, GD, and Δ_P indicators. Notably, the p-values of the Friedman test are all significantly lower than 0.05, providing sufficient evidence of a statistically significant difference between UCLMO and the other algorithms.

4.5. Analysis of ablation experiments

The effectiveness of UCLMO mainly comes from the guidance of historical knowledge and normative knowledge for evolution. Historical knowledge is used as a knowledge set to generate normative knowledge and new populations. Normative knowledge is used to assist the evolution of new populations. Therefore, this section conducts ablation experiments to analyze the effectiveness of these two strategies.

(1) This section verifies the effectiveness of the individual selection strategy based on historical knowledge. Based on the guidance of historical knowledge, the strategy selects individuals with the smallest PBI value. Thus, this section compares the proposed

Table 5
Results of Friedman test of UCLMO and other algorithms.

	CCGDE3	LMOCSO	LSMaODE	CVEA3	AMPDEA	BCE-IBEA
IGD Ranking	8.43	4.03	5.24	5.07	10.86	5.52
GD Ranking	9.81	5.78	6.83	5.50	9.97	6.72
Δ_P Ranking	9.83	4.33	7.36	6.33	9.03	7.31
	GLMO	DGEA	WOF	FDV	UCLMO	*p*-value
IGD Ranking	3.81	9.11	6.21	5.09	**2.63**	1.4797E−61
GD Ranking	3.40	5.65	4.81	6.06	**1.47**	3.7148E−37
Δ_P Ranking	3.28	5.25	5.19	6.53	**1.56**	4.5053E−36

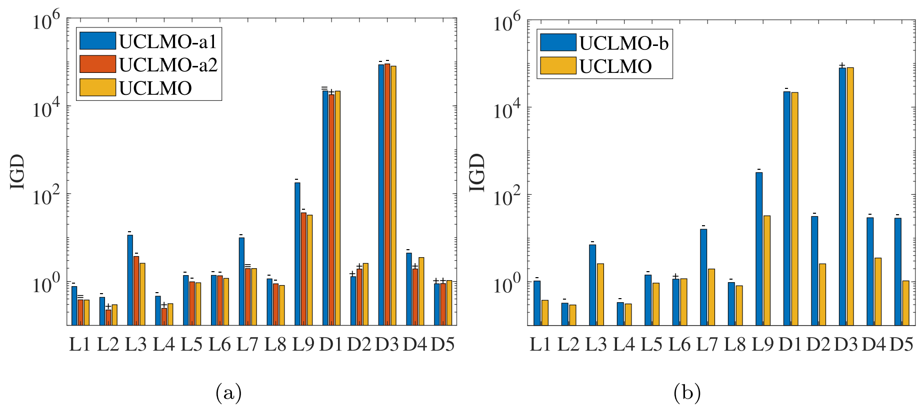


Fig. 7. Illustrative diagram of effectiveness. (a) shows the effectiveness of the individual selection strategy, and (b) shows the effectiveness of the assisted evolutionary strategy. The horizontal coordinates indicate the function names of LSMOPs and DTLZs, for example, L1 and D1 indicate the abbreviations of LSMOP1 and DTLZ1, respectively.

UCLMO with the algorithm without using the guidance of historical knowledge and PBI approach (UCLMO-a1), and the algorithm without the guidance of historical knowledge (UCLMO-a2).

Fig. 7(a) shows the IGD performance of UCLMO and variant algorithms on all test suites with 8-objectives. Each bar graph is signed with the Wilcoxon rank sum test for statistical comparison. As it can be seen from Fig. 7(a), the proposed UCLMO using PBI strategy and historical knowledge can achieve the best performance. The reason attributes to the balance between diversity and convergence. In the process of evolution, PBI pays more attention to the convergence of the population, and historical knowledge promotes the diversity of the population.

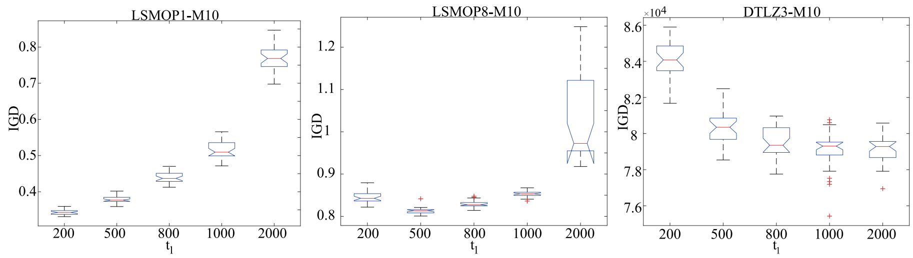
(2) This section verifies the effectiveness of the normative knowledge-assisted evolutionary strategy. The strategy uses normative knowledge to generate new individuals to assist optimization. Thus, this section compares the proposed UCLMO with the algorithm using neighborhood individuals for mating (UCLMO-b). Fig. 7(b) shows the IGD performance of UCLMO and the variant algorithm UCLMO-b on all test suites with 10-objectives. Each bar graph is signed with the Wilcoxon rank sum test for statistical comparison. As it can be seen from Fig. 7(b), in most test problems, UCLMO obtains the best performance. It indicates that the individuals generated based on normative knowledge are better than the individuals generated by neighborhood, which improves the evolution efficiency.

4.6. Analysis of parametric experiments

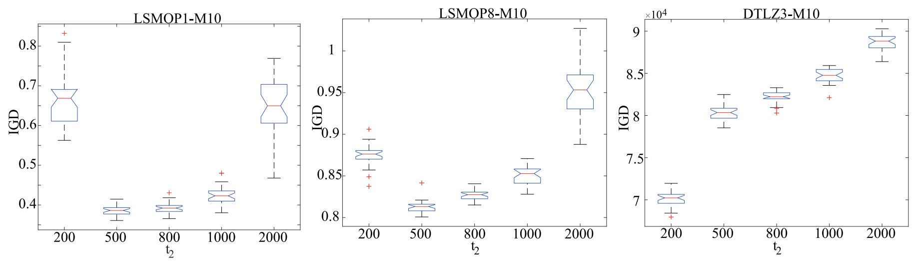
This section analyzes the effect of the parameters on the performance of UCLMO, including the maximum number of function evaluation t_1 embedded in MOEA, the maximum number of function evaluation t_2 of culture-assisted evolutionary algorithm, and the individual association ratio α of each reference line. Among them, t_1 determines the scale of historical knowledge, t_2 determines the number of effective iterations of normative knowledge-assisted evolution, and α determines the balance of diversity and convergence of the population. Therefore, these parameters are analyzed. The range of t_1 is set to 200, 500, 800, 1000, 2000. The range of t_2 is set to 200, 500, 800, 1000, 2000. The range of α is set to 0.04, 0.06, 0.08, 0.1, 0.3.

(1) Set t_2 =500, α=0.06, change t_1. Fig. 8(a) shows the box plot of the impact of t_1 on the IGD metric on LSMOP1, LSMOP8, and DTLZ3 test suites with $M = 10$. When $t_1 = 200$, 500, 1000, the optimal IGD values are obtained on LSMOP1, LSMOP8, and DTLZ3 respectively. Obviously, when $t_1 = 200$, although the optimal IGD value is achieved on LSMOP1, the opposite performance is achieved on DTLZ3. When $t_1 = 1000$, the algorithm obtains non-optimal IGD values on LSMOP1 and LSMOP8. On LSMOP8, the IGD value increases as t_1 increases, while on DTLZ3, as t_1 increases, the IGD value first decreases and then increases. The main reason may lie in that the amount of cultural information has different smoothing effects on the fitness landscape. Less cultural information may smooth the fitness landscape, resulting in faster optimization, but it will also bring the risk that it is difficult to identify the fitness landscape. More cultural information is helpful to identify the fitness landscape, but at the same time, it increases the risk of the algorithm falling into local optimum. Therefore, the compromised value 500 is selected as the value of t_1.

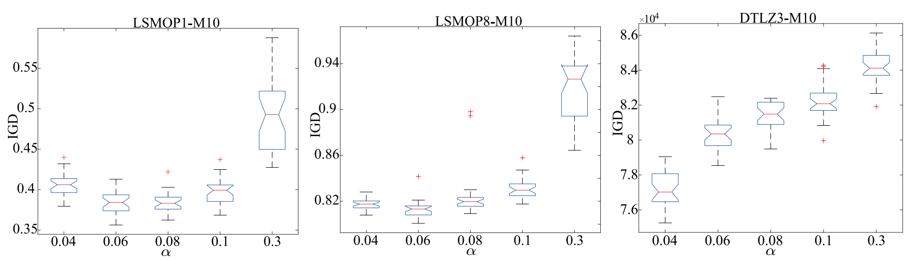
(2) Set t_1=500, α=0.06, change t_2. Fig. 8(b) shows the box plot of the impact of t_2 on the IGD metric on LSMOP1, LSMOP8, and DTLZ3 test suites with $M = 10$. When $t_2 = 500$, the optimal IGD values are obtained on LSMOP1 and LSMOP8. Obviously, when $t_2 = 200$, although the optimal IGD value is achieved on DTLZ3, the opposite performance is achieved on LSMOP1. On LSMOP1 and LSMOP8, as t_2 increases, the IGD value first decreases and then increases, while on DTLZ3, the IGD value continues to increase. The main reason may lie in that UCLMO can explore the normative knowledge for a special problem through the evolution of appropriate iterations. So it means that the performance of



(a)



(b)



(c)

Fig. 8. Boxplot of IGD metric for different t_1, t_2 and α on LSMOP1, LSMOP8 and DTLZ3 with M = 10.

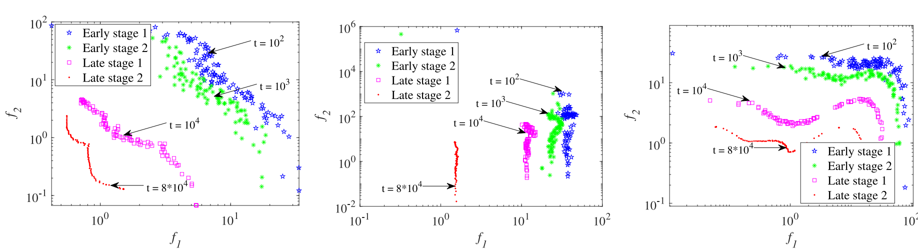


Fig. 9. Demonstration of visualization of the population of the early and late stages in UCLMO on LSMOP1 (left), LSMOP3 (middle), and LSMOP8 (right).

the algorithm does not always improve with the increase of t_2. Therefore, the compromise value 500 is selected as the value of t_2.

(3) Set t_1=500, t_2=500, change α. Fig. 8(c) shows the box plot of the impact of α on the IGD metric on LSMOP1, LSMOP8, and DTLZ3 test suites with $M = 10$. When $\alpha = 0.06$, the optimal IGD values are obtained on LSMOP1 and LSMOP8. When $\alpha = 0.04$, although the optimal IGD value is achieved on DTLZ3, the optimal performance cannot be obtained on LSMOP1 and LSMOP8. On LSMOP1 and LSMOP8, as α increases, the IGD value first decreases and then increases, while on DTLZ3, the IGD value increases. The main reason is that the larger the parameter α is, the more likely it is to include a solution that is far away in the objective space. This may prevent normative knowledge from playing a positive role in convergence. When α is too small, the learned normative knowledge is insufficient, which is not conducive to optimization. Therefore, 0.06 is selected as the value of α in this paper.

4.7. Analysis of exploration and exploitation behavior

To better illustrate the exploration and exploitation behavior, Fig. 9 shows the distribution diagrams of individuals in the early and late stages of the population on LSMOP1, LSMOP3, and LSMOP8 with two objectives. In Fig. 9, t represents the number of iterations, blue and green represent the distribution of individuals in the population at different iterations in the early stages, while red and purple represent the distribution of individuals in the late stages. In the early stages of an evolutionary algorithm, the population's distribution is usually scattered because the algorithm is still exploring the

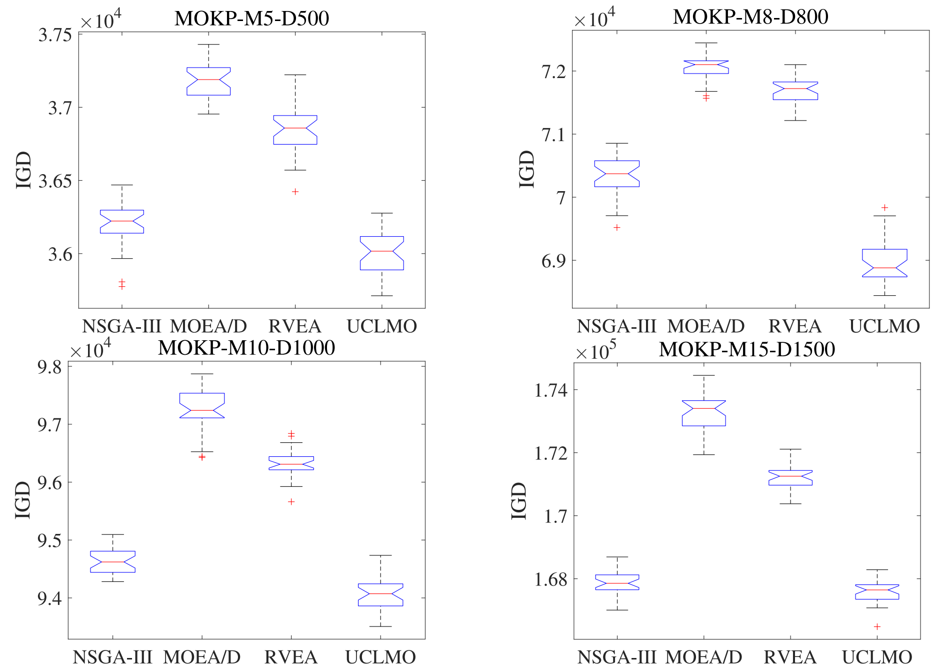


Fig. 10. Boxplot of IGD results for the four algorithms on MOKP.

search space and trying to discover as many solutions as possible. Therefore, diversity behavior is more important, i.e., maintaining population diversity and avoiding premature convergence to local optima. As the search process proceeds, the population gradually converges, and the algorithm focuses more on convergence behavior, i.e., gradually approaching the optimal solution set. Therefore, the algorithm will gradually reduce exploration behavior and focus more on discovering better solutions in the local search space. It can be seen from the figure that the algorithm has achieved different degrees of convergence during the evolution process, mainly relying on the historical knowledge-based individual selection strategy in the early stage and the assisted evolution strategy in the later stage to explore and exploit the algorithm.

4.8. Comparisons on the multi-objective knapsack problem (MOKP)

In order to further verify the optimization performance of UCLMO, this section chooses the classical multi-objective knapsack problem (MOKP) as a real-world problem.

4.8.1. Problem definition and parameter settings

Problem definition: For an MOKP problem with n items and m backpacks, it can be mathematically expressed as:

$$
\begin{aligned}
&x = (x_1, x_2, \ldots, x_m) \in \{0, 1\}^m \\
&\sum_{j=1}^{m} w_{i,j} \cdot x_j \le c_i \quad \forall i \in \{1, 2, \ldots, n\} \\
&\mathit{maximize} \quad f_i(x) = \sum_{j=1}^{m} p_{i,j} \cdot x_j
\end{aligned}
\tag{10}
$$

where $p_{i,j}$ is the profit of item j in backpack i, $w_{i,j}$ is weight of item j in backpack i, c_i is capacity of knapsack i. The objective of the task is to put these j items into i backpacks to maximize the profits, and the items placed in each backpack cannot exceed the capacity of the backpack [3].

Parameter settings: This section compares UCLMO with three comparison algorithms, including NSGA-III [43], MOEA/D [15], and RVEA [44]. The number of backpacks M is 5, 8, 10, and 15, respectively, and the number of items D is 500, 800, 1000, and 1500, respectively. The population size and the number of maximum function evaluations of the MOKP problem are consistent with those of benchmark problems. The experiments are carried out 30 times and the average value is taken.

4.8.2. Experimental results

Table 6 shows the mean and standard deviation of IGD results of the four algorithms on the MOKP problem, and the bold font indicates that the algorithm obtains the best performance. As it can be seen from Table 6, the UCLMO exhibits better performance than the other three comparison algorithms. In terms of the Wilcoxon rank sum test, the UCLMO outperforms NSGA-III, MOEA/D, and RVEA on 12, 16, and 16 test instances, respectively. This further validates that UCLMO has good performance in solving MOKP problems.

Fig. 10 shows the box plots of the four algorithms on the 5, 8, 10, and 15-objective MOKP problems. From Fig. 10, with the increase in the number of objectives and the number of decision variables, the proposed UCLMO shows better performance in handling MOKP problems.

5. Conclusions and future work

This paper proposes a universal cultural learning framework for large-scale many-objective optimization (UCLMO). The knowledge explored by cultural learning from the evolutionary process is used to assist the evolution to improve optimization efficiency. Especially, an individual selection strategy based on historical knowledge is proposed to promote diversity, and an assisted evolution strategy based on normative knowledge is proposed to accelerate the convergence. The experimental results show that the UCLMO has competitive performance on diverse large-scale MaOPs with different levels of objective dimension and decision variable dimension, and the effectiveness of its components is thoroughly examined. Moreover, the experiments validate that the existing optimizers for MaOPs can be embedded into the UCLMO framework to improve optimizer performance.

Table 6
The comparison results of IGD on the MOKP problem.

Problem	M	D	NSGA-III	MOEA/D	RVEA	UCLMO
MOKP	5	500	3.6191e+4 (1.62e+2) −	3.7183e+4 (1.24e+2) −	3.6853e+4 (1.67e+2) −	3.6000e+4 (1.56e+2)
MOKP	5	800	5.7698e+4 (1.81e+2) −	5.9149e+4 (1.90e+2) −	5.8640e+4 (1.28e+2) −	5.6198e+4 (3.24e+2)
MOKP	5	1000	7.1558e+4 (1.99e+2) −	7.3739e+4 (1.67e+2) −	7.2848e+4 (1.73e+2) −	6.9504e+4 (4.23e+2)
MOKP	5	1500	1.0835e+5 (3.55e+2) −	1.1135e+5 (2.76e+2) −	1.1018e+5 (2.58e+2) −	1.0397e+5 (5.55e+2)
MOKP	8	500	4.3726e+4 (1.34e+2) −	4.4731e+4 (1.06e+2) −	4.4412e+4 (9.66e+1) −	4.3314e+4 (2.32e+2)
MOKP	8	800	7.0326e+4 (3.23e+2) −	7.2051e+4 (2.19e+2) −	7.1705e+4 (2.17e+2) −	6.8972e+4 (3.44e+2)
MOKP	8	1000	8.8144e+4 (2.76e+2) −	8.9988e+4 (3.10e+2) −	8.9588e+4 (1.83e+2) −	8.6056e+4 (3.71e+2)
MOKP	8	1500	1.3119e+5 (3.77e+2) −	1.3374e+5 (4.37e+2) −	1.3323e+5 (2.97e+2) −	1.2735e+5 (5.29e+2)
MOKP	10	500	4.7391e+4 (1.70e+2) +	4.8989e+4 (1.63e+2) −	4.8286e+4 (1.27e+2) −	4.7505e+4 (1.88e+2)
MOKP	10	800	7.7019e+4 (2.69e+2) −	7.9559e+4 (2.44e+2) −	7.8434e+4 (1.80e+2) −	7.6329e+4 (2.06e+2)
MOKP	10	1000	9.4638e+4 (2.48e+2) −	9.7251e+4 (3.64e+2) −	9.6313e+4 (2.48e+2) −	9.4083e+4 (3.01e+2)
MOKP	10	1500	1.4207e+5 (4.11e+2) −	1.4662e+5 (2.57e+2) −	1.4448e+5 (3.92e+2) −	1.4037e+5 (4.98e+2)
MOKP	15	500	5.6387e+4 (2.94e+2) +	5.9121e+4 (1.88e+2) −	5.7994e+4 (1.84e+2) −	5.7388e+4 (2.39e+2)
MOKP	15	800	8.9925e+4 (3.23e+2) +	9.3471e+4 (4.11e+2) −	9.1976e+4 (2.55e+2) −	9.0581e+4 (3.71e+2)
MOKP	15	1000	1.1265e+5 (4.09e+2) =	1.1685e+5 (3.23e+2) −	1.1513e+5 (2.51e+2) −	1.1286e+5 (3.50e+2)
MOKP	15	1500	1.6788e+5 (4.31e+2) −	1.7326e+5 (5.96e+2) −	1.7121e+5 (4.25e+2) −	1.6759e+5 (3.69e+2)
+/−/=			3/12/1	0/16/0	0/16/0	

However, the two proposed strategies are carried out in a fixed and alternant manner in the present study. How to adaptively perform individual selection strategy and assisted evolution strategy based on the optimization status, still needs to be further studied, hence balancing the exploration and exploitation abilities more effectively. In future work, we attempt to apply the UCLMO framework to real-world applications, and the main consideration is how to effectively integrate prior knowledge of the problem into the UCLMO. We also attempt to apply the UCLMO framework to solve constrained optimization problems, and the main consideration is how to introduce effective constraint handling techniques such as penalty function methods or constraint optimization techniques.

CRediT authorship contribution statement

Xia Wang: Investigation, Methodology, Writing – original draft, Validation. **Hongwei Ge:** Conceptualization, Methodology, Writing – original draft, Editing, Supervision. **Naiqiang Zhang:** Resources, Data curation. **Yaqing Hou:** Project administration. **Liang Sun:** Investigation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61976034, U1808206), the Dalian Science and Technology Innovation Fund (2022JJ12GX013), the Natural Science Foundation of Liaoning Province, China (2022YGJC20), and the Fundamental Research Funds for the Central Universities, China (DUT21YG106).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2023.110538.

References

[1] J.F. Qiao, Y. Hou, H.G. Han, Optimal control for wastewater treatment process based on an adaptive multi-objective differential evolution algorithm, Neural Comput. Appl. 31 (2019) 2537–2550.
[2] A.A. Jalili, M. Najarchi, S. Shabanlou, R. Jafarinia, Multi-objective optimization of water resources in real time based on integration of NSGA-II and support vector machines, Environ. Sci. Pollut. Res. (2022) 1–12.
[3] Y. Zhang, G.-G. Wang, K. Li, W.-C. Yeh, M. Jian, J. Dong, Enhancing MOEA/D with information feedback models for large-scale many-objective optimization, Inform. Sci. 522 (2020) 1–16.
[4] X. Pan, L. Wang, M. Zhang, Q. Qiu, A dynamic resource allocation strategy for collaborative constrained multi-objective optimization algorithm, Appl. Intell. (2022) 1–26.
[5] P. Amarjeet, Software module clustering using grid-based large-scale many-objective particle swarm optimization, Soft Comput. 26 (2022) 8709–8730.
[6] Amarjeet, J.K. Chhabra, Many-objective artificial bee colony algorithm for large-scale software module clustering problem, Soft Comput. 22 (2018) 6341–6361.
[7] A. Tolba, Z. Al-Makhadmeh, Modular interactive computation scheme for the internet of things assisted robotic services, Swarm Evol. Comput. 70 (2022) 101043.
[8] F. Wang, Y. Laili, L. Zhang, Multi-granularity service composition in industrial cloud robotics, Robot. Comput.-Integr. Manuf. 78 (2022) 102414.
[9] L.M. Antonio, C.A.C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2758–2765.
[10] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization, IEEE Trans. Evol. Comput. 22 (1) (2018) 97–112.
[11] M. Wu, K. Li, S. Kwong, Q. Zhang, Evolutionary many-objective optimization based on adversarial decomposition, IEEE Trans. Cybern. 50 (2) (2020) 753–764.
[12] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A framework for large-scale multiobjective optimization based on problem transformation, IEEE Trans. Evol. Comput. 22 (2) (2018) 260–275.
[13] H. Zille, S. Mostaghim, Linear search mechanism for multi- and many-objective optimisation, in: K. Deb, E. Goodman, C.A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, P. Reed (Eds.), Evolutionary Multi-Criterion Optimization, Springer International Publishing, Cham, 2019, pp. 399–410.
[14] S. Qi, J. Zou, S. Yang, J. Zheng, A level-based multi-strategy learning swarm optimizer for large-scale multi-objective optimization, Swarm Evol. Comput. 73 (2022) 101100.
[15] M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimization, IEEE Trans. Evol. Comput. 19 (3) (2015) 445–460.
[16] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Trans. Evol. Comput. 23 (6) (2019) 949–961.
[17] R. Cheng, Y. Jin, M. Olhofer, B. sendhoff, Test problems for large-scale multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (12) (2017) 4108–4121.

[18] H. Chen, R. Cheng, J. Wen, H. Li, J. Weng, Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations, Inform. Sci. 509 (2020) 457–469.

[19] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables, IEEE Trans. Evol. Comput. 20 (2) (2016) 275–298.

[20] A. Song, Q. Yang, W.-N. Chen, J. Zhang, A random-based dynamic grouping strategy for large scale multi-objective optimization, in: 2016 IEEE Congress on Evolutionary Computation, CEC, 2016, pp. 468–475.

[21] K. Zhang, C. Shen, G.G. Yen, Multipopulation-based differential evolution for large-scale many-objective optimization, IEEE Trans. Cybern. (2022) 1–13.

[22] H.-L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Trans. Evol. Comput. 18 (3) (2014) 450–455.

[23] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, X. Yao, A scalable indicator-based evolutionary algorithm for large-scale multiobjective optimization, IEEE Trans. Evol. Comput. 23 (3) (2019) 525–537.

[24] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, Mutation operators based on variable grouping for multi-objective large-scale optimization, in: 2016 IEEE Symposium Series on Computational Intelligence, SSCI, 2016, pp. 1–8.

[25] C. He, R. Cheng, D. Yazdani, Adaptive offspring generation for evolutionary large-scale multiobjective optimization, IEEE Trans. Syst. Man Cybern. (2020) 1–13.

[26] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Trans. Cybern. 50 (8) (2020) 3696–3708.

[27] R.M. Rizk-Allah, A.E. Hassanien, A. Slowik, Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization, Neural Comput. Appl. 32 (17) (2020) 13715–13746.

[28] R.S. Al-Gharaibeh, M.Z. Ali, M.I. Daoud, R. Alazrai, H. Abdel-Nabi, S. Hriez, P.N. Suganthan, Real-parameter constrained optimization using enhanced quality-based cultural algorithm with novel influence and selection schemes, Inform. Sci. 576 (2021) 242–273.

[29] T. Liu, L. Jiao, W. Ma, J. Ma, R. Shang, A new quantum-behaved particle swarm optimization based on cultural evolution mechanism for multiobjective problems, Knowl.-Based Syst. 101 (2016) 90–99.

[30] C. Coello, R. Becerra, Evolutionary multiobjective optimization using a cultural algorithm, in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), 2003, pp. 6–13.

[31] Y. Liu, H. Qin, L. Mo, Y. Wang, D. Chen, S. Pang, X. Yin, Hierarchical flood operation rules optimization using multi-objective cultured evolutionary algorithm based on decomposition, Water Res. Manag. 33 (1) (2019) 337–354.

[32] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, C.L.P. Chen, A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning, IEEE Trans. Evol. Comput. 23 (4) (2019) 572–586.

[33] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, Evolut. Multiobject. Optimi. (2005) 105–145.

[34] C.A.C. Coello, C.N. Cruz, Solving multiobjective optimization problems using an artificial immune system, Genetic Program. Evolv. Mach. 6 (2005) 163–190.

[35] X. Cai, Y. Xiao, M. Li, H. Hu, H. Ishibuchi, X. Li, A grid-based inverted generational distance for multi/many-objective optimization, IEEE Trans. Evol. Comput. 25 (1) (2021) 21–34.

[36] X. Wang, Y. Jin, S. Schmitt, M. Olhofer, An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization, Inform. Sci. 519 (2020) 317–331.

[37] O. Schutze, X. Esquivel, A. Lara, C.A.C. Coello, Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 16 (4) (2012) 504–522.

[38] J. Yuan, H.-L. Liu, F. Gu, A cost value based evolutionary many-objective optimization algorithm with neighbor selection strategy, in: 2018 IEEE Congress on Evolutionary Computation, CEC, 2018, pp. 1–8.

[39] M. Li, S. Yang, X. Liu, Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 645–665.

[40] X. Yang, J. Zou, S. Yang, J. Zheng, Y. Liu, A fuzzy decision variables framework for large-scale multiobjective optimization, IEEE Trans. Evol. Comput. 23 (6) (2021) 1–15.

[41] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

[42] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Comput. Intell. Mag. 12 (4) (2017) 73–87.

[43] K. Deb, 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. 18 (4) (2014) 577–601.

[44] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.

[45] A. Nebro, J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, E. Alba, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, in: 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM, 2009, pp. 66–73.