Applied Intelligence
https://doi.org/10.1007/s10489-018-1358-0

A self-organizing map based hybrid chemical reaction optimization algorithm for multiobjective optimization

Hongye Li[1] · Lei Wang[1]

Abstract

Multiobjective particle swarm optimisation (MOPSO) is faced with convergence difficulties and diversity deviation, owing to combined learning orientations and premature phenomena. In MOPSO, leader selection is an important factor that can enhance the algorithm convergence rate. Inspired by this case, and aimed at balancing the convergence and diversity during the searching procedure, a self-organising map is used to construct the neighbourhood relationships among current solutions. In order to increase the population diversity, an extended chemical reaction optimisation algorithm is introduced to improve the diversity performance of the proposed algorithm. In view of the above, a self-organising map-based multiobjective hybrid particle swarm and chemical reaction optimisation algorithm (SMHPCRO) is proposed in this paper. Furthermore, the proposed algorithm is applied to 35 multiobjective test problems with all Pareto set shape and compared with 12 other multiobjective evolutionary algorithms to validate its performance. The experimental results indicate its advantages over other approaches.

Keywords Multiobjective optimization · Hybrid chemical reaction optimization · Self-organizing map · Multiobjective particle swarm optimization

1 Introduction

Most real-world optimisation problems involve multiple optimisation criteria, which are often in conflict with one another. For example, improvement in one objective leads to deterioration in another; no single solution can optimise all objectives simultaneously. Consequently, there exists a set of optimal solutions that present a trade-off among different objectives. Therefore, the multiobjective optimisation problem (MOP) yields a set of trade-off solutions (Pareto optimal solutions), known as the Pareto set (*PS*) in the decision space and Pareto front (*PF*) in the objective space [1]. Research on multiobjective optimisation is important, because it offers the ability to solve numerous practical applications, such as software cost estimation design [2], community detection in dynamic networks [3], the semi-supervised clustering technique [4], trust optimisation problems [5], portfolio optimisation problems [6], and the multiobjective vehicle routing problem [7], among others.

Two types of methods exist for dealing with MOPs, namely traditional analytic algorithms and evolutionary algorithms (EAs). The EA optimisation technique is population-based and not problem dependent; it can approximate the *PF* (*PS*) of an MOP in a single run. Thus, the MOEA has become a major approach to the optimisation of MOPs. Under mild conditions, the Pareto front (Pareto set) of an m-objective continuous MOP is constructed by a (m-1)-dimensional piecewise continuous manifold. Zhang et al. [8] used this property to develop model-based offspring reproduction. Zhang et al. [9] also used this property for designing a model-based method that associates each solution with a sub-problem. Zhou et al. [10] proposed a self-organising multiobjective evolutionary algorithm (SMEA) for this property to deal with MOPs.

Several researchers have developed different MOEAs for dealing with different complicated benchmarks. While most existing MOEAs focus on the development of optimisation operators, many researchers have developed different EAs [1], such as the particle swarm optimisation (PSO) algorithm [11], artificial raindrop algorithm [12], chemical reaction optimisation (CRO) [13, 14], grasshopper optimisation algorithm [15], and bat algorithm [3].

PSO is a popular paradigm in current studies, and has also been investigated for tackling MOPs in recent

✉ Hongye Li
lihongye8@163.com

Lei Wang
leiwang@xaut.edu.cn

[1] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

Published online: 11 January 2019

years. Almost all types of PSO approaches have been designed by mimicking the individual cognition and social learning behaviour of bird flocking [11]. When designing a multiobjective PSO (MOPSO) algorithm, two particular difficulties are faced. The first is the selection of the global best (gbest) and personal best (pbest) in the MOPSO algorithm [16–18]. In the MOPSO algorithm, a set of Pareto optimal solutions can be nominated as gbest and pbest. Their selection has a significant impact on the MOPSO algorithm performance. The second is the rapid loss of diversity as a result of its fast convergence speed, as pointed out in [19]. This behaviour may lead to premature convergence or becoming stuck in local optima. To improve this problem, several MOPSO methods have adopted a perturbation operator on each particle [18–20].

Based on the above difficulties, this study designed a hybrid multiobjective optimisation algorithm that combines the PSO algorithm and extends the CRO algorithm to optimise the MOP effectively. The leaders of the MOPSO can guide the population to search for the optimal solution direction. The extended CRO algorithm can enhance the proposed algorithm diversity, which prevents premature population convergence. In this work, we proposed a new selection method for leaders, which easily causes premature population convergence. Therefore, the self-organising map (SOM) method is used to enhance diversity, which involves building a (m-1)-dimensional topological structure of the current population and defining neighbourhood relationships among solutions in the population. Each molecule or particle has a subpopulation, which exhibits a distributed organisation in the decision space. This makes it easy to select neighbourhood leaders. This study adopts the minimax distance between the reference point and current subpopulation solutions as the neighbour leader. The global leader is the same as the neighbour leader, which is the minimax distance between the reference point and the entire set of population solutions. The selection of leaders will cause the algorithm to converge prematurely; thus, extended CRO is applied and the SOM method can enhance the local exploitation ability. A new solution is generated by the self-organising map-based multiobjective hybrid particle swarm and chemical reaction optimisation (SMHPCRO) algorithm, which can use the extended CRO or PSO algorithm by means of probability. When the extended CRO algorithm generates a new solution, it requires another solution that differs from the current solution. The selection of another solution can be made from the neighbourhood and entire population in a probability.

The remainder of this paper is organised as follows. Section 2 presents the related work. Section 3 describes the proposed SMHPCRO. Section 4 describes and discusses the simulation experimental results. Finally, conclusions are presented in section 5.

2 Related work

2.1 CRO algorithm

CRO is a recent variable population-based swarm intelligence meta-heuristics algorithm developed by Lam and Li [13, 14]. The method has been applied to solving MOPs [20–22], which encode the solutions as molecules and mimic the interactions of molecules in chemical reactions by searching for the optimal solutions in a container. A molecule possesses two types of energy, namely potential energy (*PE*) and kinetic energy (*KE*). *PE* can be described in mathematical terms as follows:

$$PE_x = f(x) \tag{1}$$

where f is an objective function and x is the structure of a molecule that represents a point in the decision space. The collisions among molecules in a container are classified into two categories: uni-moliecular and inter-molecular collisions. Uni-molecular collisions can be classified as two types, namely on-wall ineffective collision and decomposition. The inter-molecular collisions include inter-molecular ineffective collision and synthesis. A flowchart of the chemical reaction algorithm is presented in Fig. 1.

1) On-wall ineffective collision

When the molecules collide with the container walls and bounce back, this procedure is the on-wall ineffective collision operator. Following the on-wall ineffective collision, the molecule structures will change. The structure of x becomes x', which is generated from the neighbourhood of x. The reaction process is defined by Eq. (3).

$$x(i) = x(i)' \tag{2}$$

where $x(i)' = Neighbour(x(i))$. If $x(i)$ is replaced by $x(i)'$, the energy conservation condition must be satisfied as follows:

$$PE_x + KE_x \geq PE_{x'} \tag{3}$$

Its *KE* is updated by:

$$KE_{x'} = \left(PE_x - PE_{x'} + KE_{x'}\right) \times \alpha \tag{4}$$

where α is a random number that lies between [*KELossRate*, 1] in which *KELossRate* is a parameter of CRO.

2) Decomposition

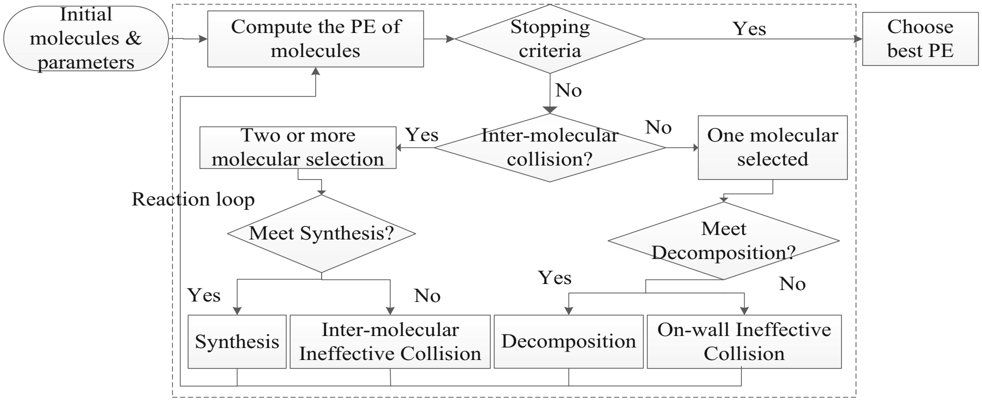

Initial molecules & parameters
Compute the PE of molecules
Stopping criteria
Yes
Choose best PE
No
Two or more molecular selection
Yes
Inter-molecular collision?
No
One molecular selected
Reaction loop
Meet Synthesis?
Meet Decomposition?
Yes
No
Yes
No
Synthesis
Inter-molecular Ineffective Collision
Decomposition
On-wall Ineffective Collision

Fig. 1 The flowchart of CRO

The decomposition operator means that a molecule x hits a container wall and then breaks into two or more molecules. Then, the decomposition criterion is demonstrated as follows:

$$\mathrm{NumHit}_x - \mathrm{MinHit}_x > \alpha \tag{5}$$

The decomposition operator must be satisfied as follows:

$$PE_x + \quad KE_x + \delta_1 \times \delta_2 \times buffer \geq PE_{x_1'} + PE_{x_2'} \tag{6}$$

If (7) holds, the current molecule w is replaced by the two new molecules, $x_1^{'}$ and $x_2^{'}$. Here, δ_1 and δ_2 are two uniformly generated random numbers from the range [0, 1].

3) Inter-molecular ineffective collision

The inter-molecular ineffective collision operator means that two or more molecules collide with one another and then bounce back. The concrete operation involves the two molecules x_1 and x_2 being carried on the neighbourhoods respectively, which can become x_1' and x_2'. The energy conservation condition must be met, as follows:

$$PE_{x_1} + PE_{x_2} + \quad KE_{x_1} + KE_{x_2} \geq PE_{x_1'} + PE_{x_2'} \tag{7}$$

4) Synthesis

The synthesis operator achieves the opposite of decomposition; two or more molecules collide and are then combined to form one new, single molecule.

The condition of synthetic operations occurring is met as follows:

$$KE_{x_1} < \beta \text{ and } KE_{x_2} < \beta \tag{8}$$

The energy conservation condition is demonstrated as follows:

$$PE_{x_1} + PE_{x_2} + \quad KE_{x_1} + KE_{x_2} \geq PE_x \tag{9}$$

If (10) holds, the current molecules x_1'and x_2' are replaced by one new molecule x.

2.1.1 Energy handling

Energy can be transformed from one type to another, but all energy must follow the energy conservation law, which states the energy can be neither created nor destroyed. A generalised form of the elementary reaction can be expressed as follows:

$$x_1 + \ldots + x_k \rightarrow x_1^{'} + \quad \ldots + x_l^{'} \tag{10}$$

where k and l are the numbers of molecules involved before and after the reaction, respectively. For $k = 1$ and $l = 2$, the reaction can be considered as decomposition.

The corresponding energy equation can be expressed as follows:

$$\begin{aligned}(PE_{x_1} + \ldots + PE_{x_k}) + (KE_{x_1} + \ldots + KE_{x_k}) + buffer \\ = \left(PE_{x_1'} + \ldots + PE_{x_l'}\right) + \left(KE_{x_1'} + \ldots + KE_{x_l'}\right) + buffer^{'}\end{aligned}$$

In the above equation, the change in the total energy of the molecules before and after the reaction is represented on the left and right of the equality, respectively.

The general new solution acceptance rule is as follows:

$$\sum_{i=1}^{k} PE_{x_i} + \sum_{i=1}^{k} KE_{x_i} - \sum_{i=1}^{l} PE_{x_i} \geq 0 \tag{11}$$

2.1.2 Sketch of CRO algorithm

As explained above, the step-wise procedure for the CRO implementation can be summarised as follows.

Algorithm 1 CRO( )

```
Begain
    Step 1: Define the optimization problem and initialize the optimization parameters.
    Step 2: Initialize the population of molecules.
    Step 3: while (FES<FESLimit)
        If   (rand > MoleColl) || (the number of population is equal to one)
            go to Step4;
        else
            go to Step5;
        endif
    endwhile
    Step 4: Uni-molecular operator.
    If $x_i$.numHit- $x_i$.minHit>DecThres
        Decomposition operator of CRO ();
    else
        On-wall Ineffective Collision operator of CRO ();
    endif
    Update energy management rules ( );
    Step 5:Inter-molecular operator.
    If($x_i$.KE<SynThres)&&($x_j$.KE <SynThres)
        Synthesis operator of CRO ();
    else
        Inter-molecular Ineffective Collision operator of CRO ();
    endif
    Step 6: Termination criterion. Stop if the maximum generation number is achieved; otherwise repeat from Step 3.
end
```

2.2 The standard PSO algorithm

The PSO [11] algorithm conducts search operation using a population of particles that correspond to molecules in the extended CRO. Each particle represents a potential solution within the searching space in which the particle dimension is assumed as D-dimensional, and a particle i is represented by a vector $x =\ (x_{i1}, \ldots, x_{id}, \ldots, x_{iD})$. The velocity of the particle x_i is represented by another vector $v_i = (v_{i1}, \ldots, v_{id}, \ldots, v_{iD})$. In each generation, the ith particle learns the best position (X_{pbest}) that it has achieved thus far and the best position of all particles (X_{gbest}) in the current generation. The basic PSO algorithm can be described as follows:

$$\text{Velocity equation}: v_{id}(t+1) = wv_{id}(t) + c_1 r_1 \left(x_{pbest_{id}}(t) - x_{id}(t)\right) + c_2 r_2 \left(x_{gbest_{id}}(t) - x_{id}(t)\right) \tag{12}$$

$$\text{Position equation}: \quad x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \tag{13}$$

```
Algorithm 2 PSO( )
Begain
    Step 1: Initialize a population of particles.
    Step 2: Evaluate the objective values of all particles, and defined the initial pbest and gbest.
    Step 3: Update the velocity and position of each particle.
    Step 4: Evaluate the velocity and position of all particles.
    Step 5: update pbest and gbest.
    For $i$=1:$N$
        If $f(x_i)$<pbesti
            pbesti= $f(x_i)$;
        endif
        if $f(x_i)$<gbest
            gbest= $f(x_i)$;
        endif
    endfor
    Step 6: If a predefined stopping criterion is met, then output gbest value; otherwise go to back to Step 3.
end
```

where c_1 is the cognitive weight and c_2 is the social weight; r_1 and r_2 are two random values uniformly distributed in [0, 1]; w is the inertia weight benefit for global searching; $x_{pbestid}(t)$ is the dth dimension of the personal best (*pbest*) of particle i in cycle t; and $x_{gbestid}$ (t) is the dth dimension of the global best (*gbest*) in cycle k.

The main drawback of the PSO algorithm is that it does not guarantee global convergence prone to premature convergence [19].

2.3 SOM

The SOM [10, 23] method uses a neighbourhood function to preserve the topological properties of the data. It produces low-dimensional representations of the training points located in a high-dimensional input space. The topology of a 2D SOM is illustrated in Fig. 2. The SOM procedure includes two stages, namely learning and clustering.

1) Learning stage: The SOM extracts significant features and adjusts the weight value of each neuron for recognising the features in the future.
2) Clustering stage: The SOM classifies the input samples according to weightsand maps them to the neurons.

3 Description of proposed SMHPCRO

In this section, the proposed SMHPCRO algorithm is introduced. The distinct feature of this algorithm is the use of an SOM to discover the population distribution structure as local optimisation leaders for the particles. In order to enhance the PSO diversity, a combined hybrid CRO algorithm with PSO and an SOM method is proposed to prevent premature convergence of the PSO. The algorithmic framework of the SMHPCRO is demonstrated in Algorithm 4, where P and A are the evolution and archive populations, respectively. Following initialisation, two main procedures are iteratively run in the SMHPCRO, namely an SOM to construct the population distribution structure and a hybrid CRO algorithm to generate the new population. Then, the current evolution population is archived through the environmental selection, and the SMHPCRO can obtain a set of optimal Pareto solutions and update the population. The environmental selection is conducted using the procedure outlined in [10].

3.1 Selection of global and local optimal solutions

Prior to introducing the algorithm framework, we present the selection method for $x_{gbest}(t)$ and $x_{lbest}(t)$. The selection of $x_{gbest}(t)$ and $x_{lbest}(t)$ differs from that of other multiobjective PSO algorithms in which the reference points of the current and neighbourhood

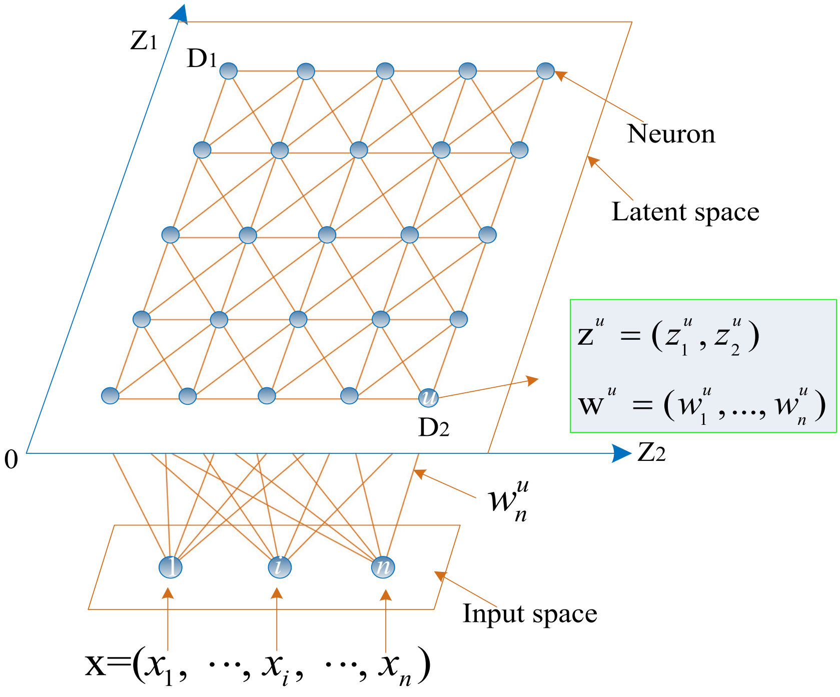


Fig. 2 An illustration of a 2-dimensional SOM [10]

populations are firstly calculated, and then $x_{gbest}(t)$ and $x_{lbest}(t)$ are respectively set by computing the shortest distance from the current solution to the reference point of the current and neighbourhood populations. The setting of reference points involves calculating the minimum values of every objective function in the current population. Taking the calculation of reference points for two-objective optimisation problems as an example, the selection of $x_{gbest}(t)$ and $x_{lbest}(t)$ is described by Fig. 3.

3.2 Framework of SMHPCRO

The above sub-sections explained the procedures for the selection of the global and local optimal solutions. The SMHPCRO procedures are described in the following. In this work, we combine the CRO algorithm and PSO as an optimisation algorithm, HPCRO. And the flow chart of HPCRO algorithm is shown in Fig. 4. In order to enhance the global search ability of the extended CRO, the PSO algorithm is embedded. Moreover, in order to enhance the PSO diversity, the extended CRO algorithm is embedded. Therefore, we propose a multiobjective hybrid CRO algorithm and PSO, which differs from the hybrid algorithm based on particle swarm and chemical reaction optimisation [20]. Combining PSO with the CRO algorithm can effectively solve MOPs and avoid premature convergence.

The proposed algorithm firstly uses the SOM method to divide the population into an organised, low-dimensional feature population in the decision space, where each molecule or particle has a neighbourhood in the decision space. Thereafter, judgement is made for a molecule or particle as to whether the parameter α satisfies the PSO or CRO update. If the PSO update is met, it will continue with formula (14); if it satisfies the CRO update, two situations will exist. When the parameter η is larger than or equal to 0.3 and smaller than 0.6, the algorithm carries out the inter-molecular ineffective collision operation. When the parameter η is greater than or equal to 0.6, the algorithm carries out the synthesis operation. Following the above operation, the polynomial mutation (*PM*) operation is performed, which is described in Algorithm 4 and Figs. 3 and 4.

$$\begin{aligned} v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (x_{lbest_{id}}(t) - x_{id}(t)) \\ + c_2 r_2 \left(x_{gbest_{id}}(t) - x_{id}(t) \right) \end{aligned} \tag{14}$$

where $x_{lbestid}(t)$ is the dth dimension of the neighbourhood best (*lbest*) of particle i in cycle t.

Algorithm 4 SMHPCRO

```
Input: P (a set of solutions (particles/ molecules))
Output: A set of Pareto optima solutions
if α<0.3
    Carry on the PSO () in formula (14) and (13)
else
    if   0.6>η≥0.3
        Inter-molecular Ineffective Collision operator ()
    else
        Synthesis operator ()
    endif
endif
Polynomial Mutation ()
Update ()
```

The pseudo-code for the SMHPCRO framework is presented in Algorithm 6 and Fig. 5. SMHPCRO begins with an initial population P and an archive A by means of environmental selection. In every generational cycle, SMHPCRO uses the SOM method of dividing the input population P into an organised, low-dimensional feature population. Thereafter, the population is evolved with the operator of the hybrid CRO algorithm with PSO.

The initial settings include:

- $N = N_1 \times \cdots \times N_{m-1}$: number of neurons, which is the same as the population size;
- τ_0: initial SOM learning rate;
- $\sigma_0 = \frac{1}{2}\sqrt{\frac{\sum_{i=1}^{m-1} N_i^2}{m-1}}$: initial neighborhood radius for updating neuron weight vectors;
- H: size of neighborhood reaction pools;
- N: population size;
- β: probability of mating restriction within neighborhood;
- T: maximum number of generations.

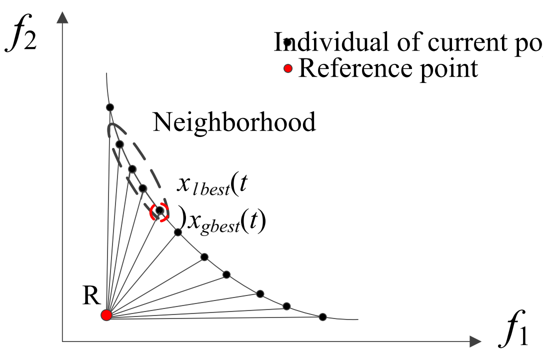
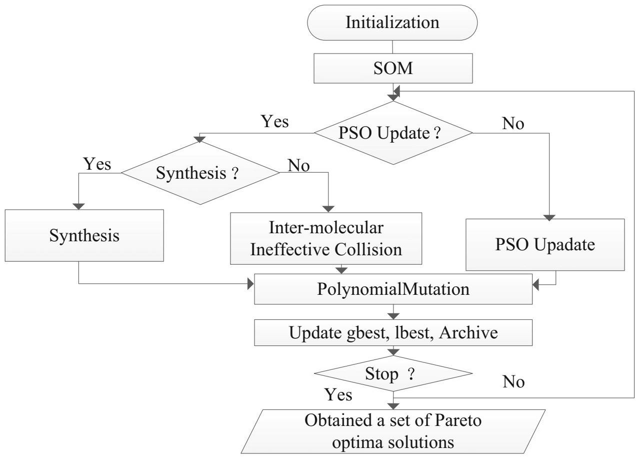


Fig. 3 The selection of $x_{gbest}(t)$ and $x_{lbest}(t)$



Initialization
SOM
PSO Update ?
Yes
No
Synthesis ?
Yes
No
Synthesis
Inter-molecular Ineffective Collision
PSO Upadate
PolynomialMutation
Update gbest, lbest, Archive
Stop ?
Yes
No
Obtained a set of Pareto optima solutions

Fig. 4 The flow chart of the hybrid chemical reaction algorithm with particle swarm optimization algorithm

Algorithm 5 SOM Framework

```
Initialize each neuron weight vector as a training point randomly selected from S.
For $g = 1,\ldots, G$ do
    Adjust the neighborhood radius, $\sigma$, and learning rate, $\tau$:
        $\sigma = \sigma_0 \times (1 - \frac{g}{G}),\ \tau = \tau_0 \times (1 - \frac{g}{G}).$
    Randomly select a training point $x \in S$.
    Find the closest neuron:
        $u' = \arg\min_{1 \le u \le D} \| x\text{-}w^u \|_2$ .
    Locate the neighboring neurons:
        $U = \{u \mid 1 \le u \le D \wedge \left\| z^u - z^{u'} \right\|_2 < \sigma\}$.
    Update all neighboring neurons $(u \in U)$ as
    $w^u = w^u + \tau \cdot \exp\left(-\left\| z^u - z^{u'} \right\|_2\right)\left(x\text{-}w^u\right)$
end
return the neuron weight vectors $w^u$, $u=1, \ldots, D$.
```

Algorithm 6 SMHPCRO Framework

```
Randomly initialize the population $\mathbf{P} = \{x^1, \ldots, x^N\}$. Set the initial training set $\mathbf{S} = \mathbf{P}$, and the set of
neuron weight vector $\{w^1, \ldots, w^N\} = \mathbf{P}$. Let $u^k$ be the index of the $k^{th}$ nearest neuron to neuron $u$ in the
latent space.
for $t = 1,\ldots, T$ do
    for each $X^s \in \mathbf{S}$, s = 1,..., |S| do
        Update the training parameters: $\sigma = \sigma_0 \times \left(1 - \frac{(t-1)N+s}{TN}\right),\ \tau = \tau_0 \times \left(1 - \frac{(t-1)N+S}{TN}\right)$
        Find the closet neuron to $x^s$: $u' = \arg\min_{1 \le u \le N} \|\mathbf{x} \quad - \mathbf{w}^u\|_2$
        Locate the neighboring neurons: $U = \left\{u \mid 1 \le u \le N \wedge \|z^u - z^{u'}\|_2 < \sigma\right\}$
        Update all neighboring neurons $(u \in U)$ as
            $\mathbf{w}^u = \mathbf{w}^u + \tau \cdot \exp\left(-\|\mathbf{z}^u - \mathbf{z}^{u'}\|_2\right)(\mathbf{x} \quad - \mathbf{w}^u)$
    end
    Set $\mathbf{A} = \mathbf{P}$, and $U = \{1,\ldots,N\}$.
    while $A \ne \emptyset$ do
        Randomly choose $x \in \boldsymbol{A}$ & $\boldsymbol{A} \backslash \{x\}$
        Set $x^u = x$ where $u = \arg\min_{u'} \|\mathbf{x} \quad - \mathbf{w}^u\|_2$
        Set $U = U \backslash \{u\}$.
    end
    Set $\mathbf{A} = \mathbf{P}$.
    foreach $u \in \{1, \ldots, N\}$ do
        Set the reaction pool Q for $x^u$ as
            $Q = \begin{cases} \bigcup_{k=1}^{H} \{x^{u^k}\} & \text{if } rand(\ ) < \beta \\ P & \text{otherwise} \end{cases}$
        Generate a new solution y = SMHPCRO ($Q$,$x^u$).
        Update the population $\mathbf{P}$ = Select ($\mathbf{P}$, y).
    end
    Update the set of training data: $\mathbf{S} = \mathbf{P} \backslash \mathbf{A}$.
end
return the population P.
```

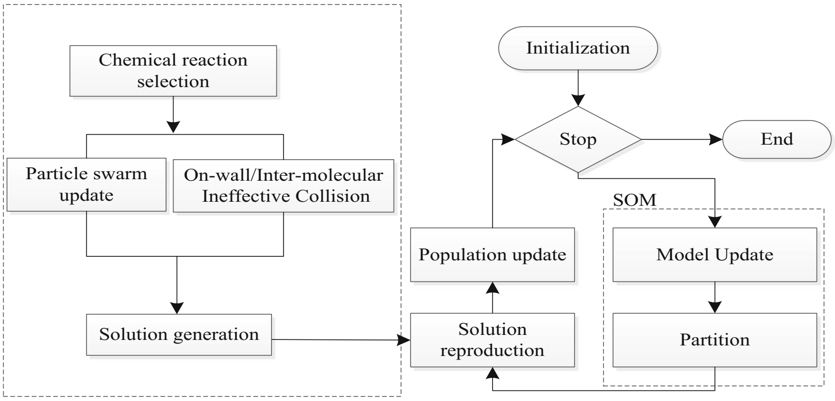


Fig. 5 The flowchart of SMHPCRO

4 Experimental results and discussion

4.1 Experimental platform and termination criterion

In all experiments, 30 independent runs were carried out on the same machine with a Celeron 3.40 GHz CPU, 4 GB memory, and Windows 7 operating system with Matlab 2015, except for NCRO and SMSEMOA in JAVA, and conducted with the maximum number of function evaluations (MaxFES) as the termination criterion. The goal was to ensure a fair comparison and reduce the statistical error.

4.2 Test functions

This study focused on 34 different MOP benchmarks. These include the Zitzler-Deb-Thiele (ZDT) test suite, Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [24], bi-objective walking fish group (WFG) [25], UF1-UF10 [26], and GTL1-GTL6 [23]. Features of the ZDT test suite include variable linkage and multimodality; therefore, it is not difficult to solve the ZDT test suite. The WFG test suite (WFG1-WFG9) [25] was included, which is characterised by various features, such as non-separable, deceptive, degenerate problems, mixed PF shape, variable dependencies, and multi-modality. The benchmarks of UF1-UF10 [26] are variable linkages and complicated PSs. The benchmarks of GTL1-GTL6 [23] are variable linkages and complicated PFs.

These include the initial ZDT-x and DTLZ-x benchmarks [24] with a simple *PS* shape and independent variable attributes. Owing to its lack of features, such as variable linkages and objective function multimodality, this test suite is not particularly challenging. Moreover, the bi-objective WFG [25], MOPs with variable linkages, UF1-UF10 [26], and MOPs with complicated PF and PS shapes, GTL1-GTL6 [23], are included.

4.3 Performance metric

Quality indicators are required to evaluate the performance of the concerned algorithms. The inverted generational distance (*IGD*) [9] is a widely used indicator in the evolutionary multiobjective optimization (EMO) literature, which represents the convergence and diversity of a solution set. The calculation formula for the *IGD* is as follows.

$$IGD\left(P^{*},P\right)=\frac{\sum_{x^{*}\in P^{*}}d\left(x^{*},P\right)}{\left|P^{*}\right|} \tag{15}$$

where $\boldsymbol{P}^*$ is a set of uniformly distributed Pareto optimal points in the *PF*, and $\boldsymbol{P}$ is a non-dominated front. Moreover, $d(x^*, P)$ is the minimal distance between x* and any point in $\boldsymbol{P}$, while $|\boldsymbol{P}^*|$ is the cardinality of $\boldsymbol{P}^*$. The *PF* must be known when using the *IGD* metric.

4.4 Parameter settings for SMHPCRO algorithm and its competitors

In this section, we validate the performance of the SMECRO algorithm by comparing it with different algorithms using the *IGD* metric.

The comparative algorithms include different fitness assignment schemes, as follows.

1) NSGA-II [27] used the Pareto-based dominance method and elitism approach.
2) MOEA/D-DE [9] which decomposed the MOP into *N* sub-problems and used DE algorithm to optimize those sub-problems.
3) MOECRO/D [28] used CRO to optimize by applying the MOEA/D framework.
4) NCRO [29] was based on the quasi-linear average time complexity quick non-dominated sorting algorithm, which incurred a low computational cost.
5) MOGOA [15] used the grasshopper optimization algorithm (GOA) with the archive and target selection

technique integrated into the algorithm and estimated the Pareto optimal front.

6) IM-MOEA [8] which used the inverse modelling for generating offspring by building a map set from the objective space to the decision space.
7) SMEA [10] used an SOM to construct an organization population and applies the EA for optimisation.
8) SMSEMOA [30] used the S-metric selection based evolutionary multiobjective algorithm.
9) OMOPSO [16] used Pareto dominance and crowding-distance information to identify the list of leader solutions.
10) DMOPSO [17] was fully dependent on the decomposition approach for solving MOPs.
11) MMOPSO [19] was introduced in [18] to use two search strategies for velocity update, promoting the convergence speed and maintaining the population diversity.
12) AgMOPSO [18] was an external archive-guided MOPSO algorithm, where the leaders for the velocity update were all selected from the external archive.

The parameter settings of the 12 compared algorithms and SMHPCRO are further summarized in Table 1. Moreover, the 12 comparative algorithms combine a PSO algorithm, CRO and other MOEAs. The population size is set as $N = 100$ and 105 for the two- and three- objective MOPs, respectively, in the comparative algorithms. The number of decision variables is set as $n = 30$ for ZDT-x, GLT1-GLT6, CEC09 and WFG1-WFG9, and $n = 10$ for DTL1 and DTLZ3. The termination condition for each algorithm is set to a maximum number of fitness evaluations (FEs) and *Max_FES* is set to 30,000 for a test instance with *n* dimensions. For the experiments, 1000 evenly distributed points are selected from the PF to form ***P**** in the IGD metric. A smaller IGD value results in a superior ***P***. The small IGD value ***P*** must be close to the true *PF* and ***P*** must be the diversity.

4.5 Comparison of the non-parametric Wilcoxon rank sum test results

The mean and standard deviation of the IGD metric values of 30 final populations obtained using the seven algorithms for each instance are displayed in Tables 2, 3, 4 and 5. The

Table 1 The parameter settings of 10 comparative algorithms

Algorithm	Parameter
SMEA	SOM structures: 1-dimensional structure 1×100 for bi-objective, and 2-dimensional structure 7×15 for tri-objective MOPs;
	Initial learning rate: τ_0=0.7; Size of neighborhood mating pools: $H = 5$; Control parameters for DE operator: F = 0.9, CR = 1;
	Control parameter for polynomial mutation operator: $p_m = 1/n$, and $\eta_m = 20$
MOEA/D-DE	Neighborhood size: NS = 0.1 N; $n_r = 0.01$ N; Probability of selecting parents from neighborhood: β=0.9; $F = 0.5$, $CR = 1.0$;
	Control parameters for polynomial mutation operator: $p_m = 1/n$, and $\eta_m = 20$,
IM-MOEA	The number of reference vectors K: K = 10; the model group size L:$L = 3$
NSGA-II	Control parameters for SBX $p_c = 1.0$, and $\eta_c = 20$; Control parameters for polynomial mutation operator: $p_m = 1/n$, and $\eta_m = 20$
MOECRO/D	Neighborhood size: NS = 5; Probability of selecting parents from neighborhood:$\beta = 0.9$; $F = 0.9$, $CR = 0.8$;
	Control parameters for polynomial mutation operator: $p_m = 1/n$, and $\eta_m = 20$;
	IniKE = 1000, *iniBuffer* = 1000, *lossRate* = 0.1, *decThres* = 15,000
NCRO	crossover probability: 0.9; mutation probability: 1/*n*;
	simulated binary crossover (SBX) distribution index:20; polynomial mutation index:20
	InitialKE: 10000; *KELoss Rate*: 0.6; *MoleColl*: 0.7; *DecThres*: 15; *SynThres*: 10
SMSEMOA	Control parameters for DE operator: $F = 0.5$, $CR = 1$;
	Control parameters for polynomial mutation operator: $p_m = 1/n$, and $\eta_m = 20$
SMHPCRO	SOM structures:1-dimensional structure 1×100 for bi-objective, and 2-dimensional structure 7×15 for tri-objective MOPs;
	Initial learning rate: τ_0=0.7; Size of neighborhood mating pools: $H = 5$;*IniKE* = 1000, *iniBuffer* = 1000, *lossRate* = 0.1, $\beta = 0.9$, *decThres* = 15,000
OMOPSO	$0.1 < w < 0.5$, $1.5 < C1 < 2.0$,$1.5 < C2 < 2.0$,$r1$、 $r2$∈[0,1]
DMOPSO	$p_c = 0.9$, $p_m = 1/n$, $\eta_m = 20$,$\eta_c = 20$, w∈[0.1, 0.5], c_1∈[1.5, 2.0], c_2∈[1.5, 2.0]
MMOPSO	$p_c = 0.9$, $p_m = 1/n$, $\eta_m = 20$,$\eta_c = 20$, w∈[0.1, 0.5], c_1∈[1.5, 2.0], c_2∈[1.5, 2.0], $\delta = 0.9$
AgMOPSO	$p_c = 0.9$, $p_m = 1/n$, $\eta_m = 20$,$\eta_c = 20$, w∈[0.1, 0.5], $F_2 = 0.5$, $T = 20$

Table 2 The experimental results of OMOPSO, DMOPSO, SMOPSO, MMOPSO, AgMOPSO, MOECRO/D, NCRO and SMHPCRO over 30 independent runs on 19 MOPs of ZDTx, DTLZx and WFGx

Fun	OMOPSO	DMOPSO	MMOPSO	AgMOPSO	MOECRO/D	NCRO	SMHPCRO
ZDT1	5.712E-03+	3.867E-03+	3.688E-03+	1.847E-03+	4.269E-03+	5.240E-03+	**1.007E-03**
	(7.572E-03)	(2.347E-04)	(2.482E-05)	(1.509E-03)	(6.780E-06)	(2.822E-03)	(1.337E-05)
Rank_zdt1	7	4	3	2	5	6	1
ZDT2	7.857E-03+	6.438E-02+	3.793E-03+	1.974E-03+	1.805E-03+	4.911E-03+	**1.321E-03**
	(1.514E-04)	(1.877E-03)	(2.791E-05)	(1.506E-03)	(4.143E-06)	(1.003E-03)	(5.065E-05)
Rank_zdt2	7	6	4	3	2	5	1
ZDT3	3.508E-02+	1.065E-02+	4.304E-03+	**2.225E-03**-	1.063E-02+	1.044E-02+	3.182E-03
	(7.572E-03)	(4.224E-04)	(4.426E-02)	(1.706E-03)	(1.274E-05)	(1.671E-03)	(2.051E-04)
Rank_zdt3	7	6	3	1	5	4	2
ZDT4	8.383E+00+	4.480E-01+	3.666E-03+	2.038E-03+	4.068E-03+	3.654E-01+	**1.069E-03**
	(3.786E-02)	(0.938E-02)	(1.328E-01)	(1.559E-03)	(6.661E-04)	(2.679E-02)	(1.272E-03)
Rank_zdt4	7	6	3	2	4	5	1
ZDT6	4.472E-02+	3.537E-02+	2.379E-03+	1.487E-03+	1.278E-02+	3.227E-02+	**1.193E-03**
	(3.028E-04)	(2.816E-03)	(2.656E-04)	(1.137E-03)	(1.047E-04)	(9.298E-03)	(5.774E-06)
Rank_zdt1	7	6	3	2	4	5	1
DTLZ1	9.999E+00+	2.178E-01+	7.107E+00+	2.064E-02+	1.618E-02+	1.760E-02+	**1.308E-02**
	(6.057E-04)	(1.408E-03)	(1.770E-04)	(4.337E-02)	(1.318E-05)	(3.020E-03)	(7.937E-05)
Rank_dtlz1	7	5	6	4	2	3	1
DTLZ2	3.613E-01+	5.359E-01+	4.000E-02+	2.740E-02+	4.153E-02+	3.865E-02+	**2.228E-02**
	(5.300E-02)	(4.693E-03)	(2.213E-01)	(2.092E-02)	(3.378E-04)	(7.157E-04)	(8.950E-04)
Rank_dtlz2	6	7	4	2	5	3	1
DTLZ3	1.004E+00+	4.305E+00+	3.596E+00+	4.913E+00+	4.354E-02+	6.243E-01+	**3.959E-02**
	(2.271E-03)	(4.691E-04)	(3.541E-02)	(5.562E+00)	(1.980E-03)	(4.754E-01)	(3.866E-04)
Rank_dtlz3	6	7	4	5	2	3	1
DTLZ4	7.542E-02+	1.819E-01+	4.194E-02+	2.856E-02+	2.267E-02≈	4.048E-02+	**2.239E-02**
	(4.543E-02)	(3.285E-03)	(3.098E-03)	(2.181E-02)	(3.417E-04)	(3.754E-04)	(5.772E-04)
Rank_dtlz4	6	7	5	3	2	4	1
DTLZ6	8.172E-02+	8.275E-02+	7.352E-02+	3.952E-02+	6.175E-02+	6.428E-02+	**3.028E-02**
	(6.814E-03)	(3.755E-01)	(3.984E-03)	(3.034E-02)	(1.525E-03)	(9.017E-01)	(5.953E-04)
Rank_dtlz6	6	7	5	2	3	4	1
WFG1	1.696E+00+	2.675E+00+	1.461E-01+	1.333E-01+	1.402E+00+	2.342E+00+	**1.085E-01**
	(3.110E-01)	(3.732E-03)	(6.205E-02)	(7.132E-02)	(3.704E-04)	(4.321E-02)	(2.797E-02)
Rank_wfg1	5	7	3	2	4	6	1
WFG2	5.451E-02+	4.281E-02+	3.782E-02+	4.974E-02+	2.863E-02+	9.881E-02+	**1.666E-02**
	(5.598E-04)	(1.948E-03)	(3.102E-02)	(3.815E-02)	(4.023E-04)	(2.548E-02)	(5.253E-05)
Rank_wfg2	6	5	3	4	2	7	1
WFG3	3.297E-01+	1.736E-01+	1.125E-02≈	5.662E-02+	2.016E-02+	8.537E-01+	**1.108E-02**
	(2.488E-03)	(2.728E-02)	(1.551E-04)	(4.324E-02)	(2.990E-04)	(3.785E-01)	(5.564E-05)
Rank_wfg3	5	6	2	4	3	7	1
WFG4	4.089E-02+	2.472E-02+	1.051E-02≈	5.355E-02+	3.878E-02+	5.965E-02+	**1.032E-02**
	(1.244E-01)	(1.559E-02)	(3.878E-02)	(4.091E-03)	(5.867E-04)	(3.691E-03)	(2.637E-03)
Rank_wfg4	6	4	2	7	5	8	1
WFG5	7.188E-02+	6.702E-02+	6.159E-02+	3.399E-02+	6.031E-02+	5.477E-01+	**2.766E-02**
	(1.866E-02)	(1.169E-03)	(4.654E-02)	(2.613E-02)	(9.195E-04)	(8.672E-01)	(6.909E-04)
Rank_wfg5	7	6	5	2	4	3	1
WFG6	2.942E-01+	1.850E-01+	1.243E-01+	6.224E-01+	2.227E-01+	3.453E-01+	**1.065E-01**
	(4.976E-01)	(3.507E-02)	(2.327E-01)	(4.762E-03)	(4.096E-04)	(5.664E-01)	(1.375E-02)
Rank_wfg6	5	3	2	7	4	6	1
WFG7	6.034E-02+	5.938E-02+	1.189E-02+	5.956E-02+	5.310E-02+	3.561E-02+	**1.133E-02**
	(4.354E-02)	(2.338E-03)	(7.57E-04)	(4.545E-03)	(2.505E-04)	(5.787E-02)	(7.986E-05)
Rank_wfg7	7	6	2	5	4	3	1
WFG8	1.486E-01+	8.624E-02+	9.448E-02+	6.785E-02+	2.339E-02+	1.130E-01+	**2.144E-02**
	(6.221–03)	(3.971E-03)	(5.430E-03)	(6.068E-02)	(9158E-04)	(4.557E-02)	(4.337E-04)
Rank_wfg8	7	4	5	3	2	6	2
WFG9	5.675E-01+	5.598E-01+	1.191E-01+	6.646E-01+	3.190E-01+	4.731E-01+	**1.036E-01**
	(3.732E-03)	(7.95E-04)	(6.981E-03)	(5.146E-03)	(8.762E-04)	(6.381E-01)	(3.326E-05)
Rank_wfg9	5	6	2	7	3	4	1
Sum_ Rank	119	108	66	67	65	92	20
Rank	7	6	3	2	4	5	1
±/≈	19/0/0	19/0/0	17/0/2	18/1/0	18/0/1	19/0/0	

The bold data in the table are the best mean metric values yielded by the algorithms for each instance

Table 3 The experimental results of NSGA-II, SMSEMOA, MOEA/D-DE, IM-MOEA, MOGOA, SMEA and SMHPCRO over 30 independent runs on nineteen MOPs of ZDTx, DTLZx and WFGx

Fun	NSGA-II	SMSEMOA	MOEA/D-DE	IM-MOEA	MOGOA	SMEA	SMHPCRO
ZDT1	4.696E-03+ (1.435E-04)	3.814E-03+ (2.041E-05)	3.706E-03+ (6.355E-06)	3.330E-03+ (4.483E-04)	4.600E-03+ (2.470E-02)	3.617E-03+ (2.082E-05)	**1.007E-03** (1.337E-05)
Rank_zdt1	7	5	4	2	6	3	1
ZDT2	4.724E-03+ (1.390E-04)	4.389E-03+ (2.865E-05)	3.806E-03+ (4.679E-06)	2.560E-03+ (1.344E-04)	4.900E-03+ (9.000E-03)	4.357E-03+ (8.145E-05)	**1.321E-03** (5.065E-05)
Rank_zdt2	6	5	3	2	7	4	1
ZDT3	4.724E-02+ (1.788E-04)	4.683E-03+ (2.224E-04)	4.623E-02+ (9.740E-06)	3.880E-03+ (1.861E-04)	3.230E-02+ (3.400E-03)	4.441E-03+ (8.544E-05)	**3.182E-03** (2.051E-04)
Rank_zdt3	7	6	5	3	2	4	1
ZDT4	4.880E-01+ (7.713E-04)	8.261E-01+ (9.947E-03)	3.752E-03+ (1.101E-04)	2.833E-03+ (6.598E-05)	5.036E-02+ (4.728E-03)	6.623E-03+ (2.169E-02)	**1.069E-03** (1.272E-03)
Rank_zdt4	4	7	3	2	5	6	1
ZDT6	4.261E-02+ (6.549E-05)	4.402E-02+ (6.526E-04)	2.739E-03+ (4.436E-06)	2.731E-03+ (1.194E-04)	1.576E-02+ (1.260E-03)	2.247E-03+ (1.002E-04)	**1.193E-03** (5.774E-06)
Rank_zdt1	6	7	5	4	2	3	1
DTLZ1	3.982E-02+ (1.121E-03)	1.918E-02+ (3.025E-03)	1.658E-02+ (3.302E-04)	3.876E-02+ (2.465E-01)	7.880E-02+ (1.260E-03)	1.319E-02≈ (9.001E-05)	**1.308E-02** (7.937E-05)
Rank_dtlz1	6	4	3	5	7	2	1
DTLZ2	7.696E-02+ (1.435E-03)	6.022E-01+ (2.571E-02)	4.280E-02+ (8.601E-04)	3.876E-02+ (2.465E-01)	4.728E-02+ (6.304 E-02)	5.301E-02+ (1.464E-04)	**2.228E-02** (8.950E-04)
Rank_dtlz2	7	6	3	2	4	5	1
DTLZ3	8.741E-02+ (5.430E-02)	7.143E-02+ (3.216E-03)	4.426E-02+ (5.166E-04)	**3.876E-02-** (2.465E-01)	6.304E-02+ (1.418E-01)	4.991E-02+ (4.600E-04)	3.959E-02 (3.866E-04)
Rank_dtlz3	7	6	3	1	5	4	2
DTLZ4	5.951E-02+ (1.365E-03)	7.232E-02+ (3.368E-02)	4.340E-02+ (2.935E-04)	3.876E-02+ (2.465E-01)	4.571E-02+ (1.785E-01)	3.934E-02+ (2.357E-04)	**2.239E-02** (5.772E-04)
Rank_dtlz4	6	7	4	2	5	3	1
DTLZ6	4.156E-01+ (1.483E-03)	3.678E+00+ (2.056E-02)	4.323E-01+ (1.772E-03)	3.876E-01+ (2.465E-01)	3.214E-01+ (1.785E-02)	4.712E-02+ (6.004E-04)	**3.028E-02** (5.953E-04)
Rank_dtlz6	5	3	6	4	2	7	1
WFG1	1.516E+00+ (6.329E-02)	1.523E+00+ (6.413E-02)	1.418E+00+ (6.522E-02)	1.421E+00+ (4.384E-04)	1.671E+00+ (2.499E-01)	1.521E+00+ (9.483E-02)	**1.085E-01** (2.797E-02)
Rank_wfg1	5	6	2	3	7	4	1
WFG2	2.342E-01+ (7.035E-04)	1.465E-01+ (6.942E-04)	3.676E-02+ (1.136E-04)	1.491E-01+ (1.233E-04)	1.938E-01+ (1.661E-02)	2.333E-02+ (2.424E-03)	**1.666E-02** (5.253E-05)
Rank_wfg2	7	4	3	5	6	2	1
WFG3	1.392E-01+ (7.297E-04)	1.406E-01+ (2.192E-04)	1.271E-02≈ (4.973E-05)	1.929E-02+ (1.996E-04)	1.071E-01+ (7.139E-03)	1.416E-02+ (5.141E-04)	**1.108E-02** (5.564E-05)
Rank_wfg3	6	7	2	4	5	3	1
WFG4	5.834E-01+ (5.286E-03)	4.651E-01+ (8.776E-05)	2.264E-01+ (3.301E-03)	2.753E-01+ (6.783E-05)	3.569E-01+ (2.379E-02)	9.865E-02+ (3.934E-03)	**1.032E-02** (2.637E-03)
Rank_wfg4	7	6	4	3	5	2	1
WFG5	6.8581E-01+ (3.619E-04)	6.713E-01+ (1.695E-02)	1.708E-01+ (5.972E-05)	7.976E-02+ (1.258E-04)	5.949E-01+ (1.071E-01)	6.665E-02+ (2.048E-04)	**2.766E-02** (6.909E-04)
Rank_wfg5	7	6	4	3	5	2	1
WFG6	3.423E-01+ (1.155E-02)	3.376E-01+ (2.790E-04)	3.475E-01+ (3.391E-02)	1.513E-01+ (2.465E-01)	2.379E-01+ (1.918E-02)	3.456E-01+ (1.774E-02)	**1.065E-01** (1.375E-02)
Rank_wfg6	5	4	7	2	3	6	1
WFG7	4.931E-01+ (1.389E-03)	1.253E-02≈ (2.790E-04)	1.633E-02+ (4.193E-05)	4.088E-02+ (2.668E-03)	1.189E-01+ (3.837E-02)	1.369E-02+ (5.854E-04)	**1.133E-02** (7.986E-05)
Rank_wfg7	7	2	4	5	6	3	1
WFG8	1.104E-01+ (6.841E-03)	5.618E-02+ (6.438E-03)	3.309E-02+ (1.448E-02)	2.943E-02+ (8.327E-02)	9.592E-02+ (1.376E-02)	3.689E-02+ (6.502E-03)	**2.144E-02** (4.337E-04)
Rank_wfg8	7	5	3	2	6	4	1
WFG9	2.782E-01+ (1.057E-02)	2.879E-01+ (1.618E-02)	2.028E-01+ (2.616E-04)	2.521E-01+ (7.168E-03)	2.871E-01+ (3.931E-02)	2.302E-01+ (4.479E-05)	**1.036E-01** (3.326E-05)
Rank_wfg9	5	7	2	4	6	3	1
Sum_ Rank	117	103	70	58	94	70	20
Rank	7	6	4	2	5	3	1
±/≈	19/0/0	18/0/1	18/0/1	18/1/0	19/0/0	18/0/1	

The bold data in the table are the best mean metric values yielded by the algorithms for each instance

Table 4 The experimental results of OMOPSO, DMOPSO, SMOPSO, MMOPSO, AgMOPSO, MOECRO/D, NCRO and SMHPCRO over 30 independent runs on UFx and GLTx on sixteen MOPs

Fun	OMOPSO	DMOPSO	MMOPSO	AgMOPSO	MOECRO/D	NCRO	SMHPCRO
UF1	1.298E-01+ (3.934E-03)	8.415E-02+ (6.124E-02)	8.961E0–02+ (1.050E-02)	3.952E-02+ (3.034E-02)	6.912E-02+ (7.235E-01)	1.166E-01+ (3.501E-03)	**4.398E-03** (3.007E-04)
Rank_uf1	7	4	5	2	3	6	1
UF2	7.876E-02+ (6.557E-02)	2.984E-02+ (1.531E-02)	3.435E-02+ (1.399E-03)	2.001E-02+ (1.559E-02)	6.912E-02+ (7.235E-01)	2.412E-02+ (2.904E-03)	**1.105–02** (1.159E-03)
Rank_uf2	7	4	5	2	6	3	1
UF3	3.965E-01+ (4.590E-01)	3.099E-01+ (2.296E-02)	2.859E-01+ (3.149E-02)	1.244E-01+ (9.695E-02)	2.912E-01+ (7.235E-01)	2.582E-01+ (6.870E-02)	**8.582E-02** (5.957E-03)
Rank_uf3	7	6	4	2	5	3	1
UF4	1.723E-01+ (2.623E-02)	2.619E-01+ (0.765E-01)	8.094E-02+ (2.449E-01)	1.326E-02+ (1.015E-02)	6.912E-02+ (7.235E-01)	4.378E-02+ (1.939E-02)	**1.084E-02** (6.864E-03)
Rank_uf4	6	7	5	2	4	3	1
UF5	1.604E+00+ (5.901E-01)	1.385E+00+ (4.593E-01)	6.392E-01+ (1.749E-01)	4.244E-01+ (3.708E-01)	6.912E-01+ (7.235E-01)	5.331E-01+ (1.871E-01)	**3.283E-01** (1.278E-01)
Rank_uf5	7	6	4	2	5	3	1
UF6	5.662E-01+ (3.278E-01)	5.633E-01+ (7.655E-02)	1.200E-01≈ (7.000E-04)	1.903E-01+ (1.929E-01)	6.912E-01+ (7.235E-01)	1.777E-01+ (8.327E-02)	**1.052E-01** (5.347E-02)
Rank_uf6	6	5	2	4	8	3	1
UF7	6.087E-01+ (5.245E-02)	3.918E-01+ (6.889E-02)	4.080E-02+ (2.799E-02)	9.227E-02+ (1.407E-01)	2.912E-01+ (7.235E-01)	3.562E-01+ (1.816E-02)	**5.176E-03** (4.829E-04)
Rank_uf7	7	6	2	3	4	5	1
UF8	4.217E-01+ (1.9672E-01)	3.901E-01+ (3.827E-02)	1.389E-01+ (2.099E-01)	**1.169E-01-** (9.324E-02)	2.912E-01+ (7.235E-01)	3.769E-01+ (6.778E-02)	1.234E-01 (1.284E-01)
Rank_uf8	7	6	3	1	4	5	2
UF9	1.140E-01+ (6.557E-01)	2.103E-01+ (3.062E-02)	1.202E-01+ (3.499E-01)	1.804E-01+ (1.453E-01)	1.212E-01+ (7.235E-01)	1.615E-01+ (8.357E-02)	**3.638E-02** (5.958E-02)
Rank_uf9	3	7	2	6	4	5	1
UF10	4.930E-01+ (1.311E-02)	6.502E-01+ (5.358E-02)	3.166E-01+ (3.501E-03)	4.801E-01+ (5.749E-01)	3.017E-01+ (5.201E-01)	4.912E-01+ (7.235E-01)	**1.753E-01** (3.222E-02)
Rank_uf10	6	7	3	4	2	5	1
GLT1	1.768E-01+ (1.867E-02)	4.480E-01+ (4.815E-03)	1.306E-01+ (3.922E-01)	9.189E-02+ (7.017E-02)	3.214E-03+ (6.780E-03)	6.118E-03+ (3.278E-03)	**1.051E-03** (2.732E-03)
Rank_gtl1	6	7	5	4	2	3	1
GLT2	1.453E+00+ (1.245E-02)	4.298E-01+ (5.778E-02)	1.326E+00+ (2.615E-03)	1.928E-01+ (1.472E+00)	3.505E-02+ (4.387E-03)	4.901E-02+ (1.253E-03)	**1.347E-02** (9.443E-04)
Rank_gtl2	7	5	6	4	2	3	1
GLT3	5.745E-02+ (3.734E-02)	3.694E+00+ (2.889E-02)	5.811E-02+ (6.538E-03)	4.746E-02+ (4.118E-02)	8.163E-03+ (1.325E-03)	6.143E-03+ (1.684E-03)	**4.391E-03** (2.609E-03)
Rank_gtl3	6	7	5	4	3	2	1
GLT4	5.167E-01+ (6.225E-02)	4.781E-01+ (3.852E-02)	4.639E-01+ (1.307E-02)	2.497E-01+ (1.904E-01)	4.108E-02+ (6.661E-03)	6.018E-03+ (2.529E-02)	**5.206E-03** (9.101E-03)
Rank_gtl4	7	6	5	4	3	2	1
GLT5	1.184E-01+ (3.112E-03)	3.229E-01+ (1.926E-02)	9.944E-02+ (1.961E-03)	6.028E-02+ (4.603E-02)	3.386E-02≈ (9.047E-04)	3.907E-02+ (9.128E-03)	**3.089E-02** (3.804E-04)
Rank_gtl5	6	7	5	4	2	3	1
GLT6	3.454E-01+ (2.489E-02)	3.394E-01+ (9.631E-04)	4.959E-02+ (3.268E-02)	3.174E-02+ (2.427E-02)	5.618E-02+ (1.318E-05)	6.713E-02+ (3.124E-03)	**2.257E-02** (4.232E-04)
Rank_gtl6	7	6	3	2	4	5	1
Sum_ Rank	102	96	64	50	61	59	20
Rank	7	6	5	2	4	3	1
±/≈	16/0/0	16/0/0	15/0/1	15/0/1	15/0/1	16/0/1	

The bold data in the table are the best mean metric values yielded by the algorithms for each instance

parameter setting of the compared algorithms are presented in Table 1. The Wilcoxon rank sum test at a 5% significance level was conducted to compare the significance of difference between the mean metric value yielded by a comparison algorithm and SMHPCRO. The symbols '+', '-', '≈' in Tables 2, 3, 4 and 5 represent the mean and standard deviation of the IGD values of the 12 compared MOEAs, which indicate that the performance of the SMHPCRO is better than, worse than and similar to that of the comparison algorithm, respectively, according to the Wilcoxon rank sum test. The bold data with a

Table 5 The experimental results of NSGA-II, SMSEMOA, MOEA/D-DE, IM-MOEA, MOGOA, SMEA and SMHPCRO over 30 independent runs on UFx and GLTx on sixteen MOPs

Fun	NSGA-II	SMSEMOA	MOEA/D-DE	IM-MOEA	MOGOA	SMEA	SMHPCRO
UF1	9.473E-02+	1.195E-01+	4.433E-03+	4.346E-02+	1.892E-01+	4.524E-03≈	**4.398E-03**
	(3.249E-03)	(3.029E-02)	(2.745E-04)	(1.365E-01)	(2.501E-02)	(4.819E-04)	(3.007E-04)
Rank_uf1	5	6	4	3	7	2	1
UF2	3.5071E-02+	3.704E-02+	**7.788E-03-**	1.477E-02+	4.940E-02+	1.043E-02-	1.105–02
	(1.479E-03)	(1.192E-02)	(2.755E-03)	(5.252E-03)	(3.860E-02)	(1.197E-03)	(1.159E-03)
Rank_uf2	5	6	1	4	7	2	3
UF3	9.082E-02+	2.563E-01+	1.077E-02+	6.138E-02+	2.166E-01+	9.662E-03+	**1.582E-03**
	(1.682E-02)	(4.008E-02)	(8.849E-03)	(1.283E-02)	(6.620E-02)	(4.803E-03)	(5.957E-03)
Rank_uf3	5	7	3	4	6	2	1
UF4	8.074E-02+	7.146E-02+	6.224E-02+	6.196E-02+	7.960E-02+	6.197E-02+	**5.484E-02**
	(2.809E-03)	(5.760E-04)	(4.522E-03)	(2.126E-03)	(4.800E-03)	(5.403E-03)	(6.864E-03)
Rank_uf4	7	5	4	2	6	3	1
UF5	5.201E-01+	5.684E-01+	2.930E-01≈	**2.234E-01-**	1.147E+00+	4.146E-01+	3.283E-01
	(5.162E-02)	(1.749E-01)	(8.714E-02)	(5.160E-02)	(1.661E-01)	(7.817E-02)	(1.278E-01)
Rank_uf5	5	6	2	1	7	4	3
UF6	8.073E-01+	3.275E-01+	2.738E-01+	1.836E-01+	7.345E-01+	1.964E-01+	**1.052E-01**
	(6.460E-02)	(7.831E-02)	(2.313E-01)	(6.163E-02)	(2.769E-01)	(1.981E-01)	(5.347E-02)
Rank_uf6	7	5	4	2	6	3	1
UF7	1.898E-01+	1.896E-01+	5.487E-03+	1.412E-02+	1.567E-01+	5.218E-03≈	**5.176E-03**
	(1.959E-03)	(1.626E-01)	(4.163E-04)	(2.641E-03)	(6.331E-02)	(9.349E-04)	(4.829E-04)
Rank_uf7	6	7	3	4	5	2	1
UF8	4.132E-01+	3.917E-01+	3.208E-01+	2.439E-01-	2.497E-01+	**1.993E-01-**	2.234E-01
	(2.742E-03)	(1.122E-01)	(1.013E-02)	(1.564E-02)	(7.490E-02)	(6.615E-02)	(1.284E-01)
Rank_uf8	7	6	5	3	4	1	2
UF9	3.070E-01+	2.263E-01+	7.935E-02+	9.605E-02+	3.145E-01+	1.519E-01+	**3.638E-02**
	(6.810E-04)	(6.590E-02)	(5.291E-02)	(3.624E-02)	(1.445E-01)	(9.544E-02)	(5.958E-02)
Rank_uf9	6	5	2	3	7	4	1
UF10	5.599E-01+	4.570E-01+	4.597E-01+	3.407E-01+	4.645E-01	2.025E-01≈	**1.753E-01**
	(1.254E-02)	(9.995E-02)	(7.055E-02)	(6.147E-02)	(1.445E-01)	(1.560E-01)	(3.222E-02)
Rank_uf10	7	5	4	3	6	2	1
GLT1	5.317E-02+	4.324E-02+	4.285E-03+	3.028E-02+	3.014E-02+	3.004E-03+	**1.051E-03**
	(3.368E-02)	(7.536E-03)	(3.609E-02)	(1.550E-03)	(2.261E-03)	(3.515E-03)	(2.732E-03)
Rank_gtl1	7	6	3	5	4	2	1
GLT2	8.006E-02+	9.402E-02+	4.042E-02+	1.697E-02+	8.007E-02+	3.543E-02-	**1.347E-02**
	(3.082E-02)	(2.783E-03)	(1.548E-03)	(4.386E-02)	(6.783E-02)	(1.304E-03)	(9.443E-04)
Rank_gtl2	6	7	4	3	5	2	1
GLT3	3.562E-02+	3.186E-02+	2.795E-02+	2.187E-02+	5.276E-02+	6.233E-03+	**4.391E-03**
	(1.009E-02)	(8.956E-03)	(6.666E-03)	(5.67E-03)	(7.537E-02)	(4.561E-03)	(2.609E-03)
Rank_gtl3	6	5	4	3	7	2	1
GLT4	4.521E-02+	4.001E-02+	2.066E-02+	1.676E-01+	1.071E-02+	5.794E-03+	**1.206E-03**
	(1.806E-01)	(6.192E-03)	(1.272E-02)	(8.734E-03)	(5.951E-03)	(1.324E-04)	(9.101E-03)
Rank_gtl4	6	5	4	7	3	2	1
GLT5	6.181E-02+	5.271E-02+	8.521E-02+	3.843E-02+	7.118E-02+	**3.009E-02≈**	3.089E-02
	(3.246E-03)	(4.584E-04)	(2.236E-03)	(2.886E-02)	(3.571E-03)	(3.528E-04)	(3.804E-04)
Rank_gtl5	7	5	8	3	6	1	2
GLT6	5.705E-02+	5.863E-02+	5.428E-02+	5.215E-02+	8.532E-02+	3.672E-02+	**2.257E-02**
	(3.641E-03)	(5.289E-04)	(1.550E-03)	(5.621E-03)	(1.066E-03)	(3.235E-03)	(4.232E-04)
Rank_gtl6	5	6	4	3	7	2	1
Sum_Rank	90	92	59	57	93	36	22
Rank	5	6	4	3	7	2	1
±/≈	16/0/0	16/0/0	14/1/1	14/2/0	16/0/0	9/3/4	

The bold data in the table are the best mean metric values yielded by the algorithms for each instance

grey background in the table are the best mean metric values yielded by the algorithms for each instance.

It can be observed that the proposed SMHPCRO outperforms the 12 compared multiobjective MOEAs, including the multiobjective PSO algorithms, multiobjective CRO algorithms, dominance-based method, decomposition-based method, and indicator-based MOEAs. The SMEA is a model-based method that uses the SOM method to construct the population. The MOEA/D-DE is a decomposition-based method that adopts the Chebyshev method to decompose the MOPs into a scalar number of single objective sub-problems. The IM-MOEA is a model-based method that can alleviate the solution diversity requirement. The main concept of the IM-MOEA is to construct Gaussian process-based inverse models that map all identified non-dominated solutions from the objective space to the decision space.

The multiobjective PSO algorithms and multiobjective CRO algorithms are compared to 19 MOPs without variable linkages of ZDTx, DTLZx and WFGx.

Table 2 indicates that the SMHPCRO algorithms perform significantly better than their corresponding multiobjective PSO MOEAs and multiobjective CRO MOEAs on ZDTx, DTLZx, and WFGx. For all 35 benchmarks under consideration, the SMHPCRO achieved statistically superior *IGD* values on 19 benchmarks better than OMOPSO, which was ranked seventh. The SMHPCRO had 19 benchmarks better than DMOPSO, which was ranked sixth. The SMHPCRO had 17 benchmarks better than MMOPSO, which was ranked third, and on WFG3 and WFG4, the MMOPSO was similar to SMHPCRO. The SMHPCRO had 18 benchmarks better than AgMOPSO, which was ranked second, and on the ZDT3 the AgMOPSO was better than SMHPCRO. The SMHPCRO had 18 benchmarks better than MOECRO/D, which was ranked fourth, and on the DTLZ4, the MOECRO/D was similar to SMHPCRO. The SMHPCRO had 19 benchmarks better than NCRO, which was ranked fifth.

It can be observed that the proposed SMHPCRO is suited to solving test problems with multiple local fronts. On the ZDT4, DTLZ1 and DTLZ3, the proposed SMHPCRO could obtain a set of non-dominated solutions with effective convergence and diversity. This can also be observed from Fig. 6, which plots the Pareto front with the best *IGD* value among 30 runs for the SMHPCRO and the seven compared multiobjective PSO and multiobjective CRO algorithms on the three-objective DTLZ1 in the objective space, respectively. The main reason for this phenomenon is that the proposed algorithm uses the SOM method to decompose the decision space into several subspaces. Thereafter, the HPCRO algorithm optimisation can be used to obtain effective

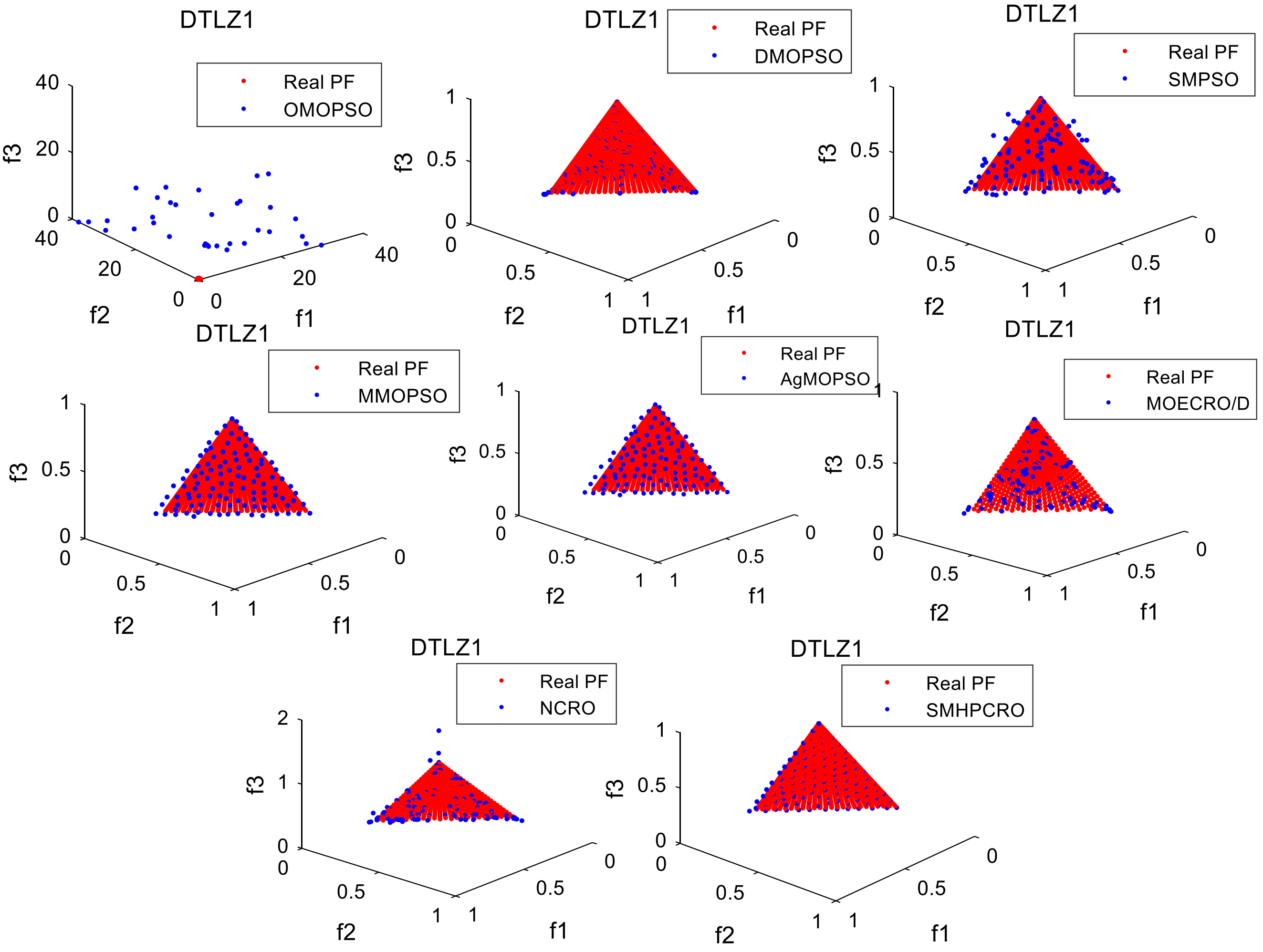


Fig. 6 The set of nondominated associated with the best IGD value among 30 runs obtained by SMHPCRO and seven compared multiobjective PSO algorithms and multiobjective CRO algorithms on DTLZ1

performance on multiple local fronts. In Fig. 6, it is observed that the SMHPCRO can achieve effective performance in terms of diversity and convergence on the DTLZ1.

Table 3 indicates that the SMHPCRO algorithms perform significantly better than the corresponding existing dominance-based method, decomposition-based method and indicator-based MOEAs on ZDTx, DTLZx and WFGx. SMHPCRO had 19 benchmarks better than NSGA-II, which was ranked seventh. The SMHPCRO had 18 benchmarks better than SMSEMOA, which was ranked sixth, and on the WFG7, SMSEMOA was similar to the SMHPCRO. The SMHPCRO had 18 benchmarks better than MOEA/D-DE, which was ranked fourth, and on the WFG3, the MOEA/D-DE was similar to the SMHPCRO. The SMHPCRO had 18 benchmarks better than IM-MOEA, which was ranked second, and on the DTLZ3, the IM-MOEA was better than the SMHPCRO. The SMHPCRO had 19 benchmarks better than the MOGOA, which was ranked fifth. The SMHPCRO had 18 benchmarks better than the SMEA, which was ranked third, and on the DTLZ1, the SMEA was similar to the SMHPCRO.

Table 4 indicates that the SMHPCRO algorithms perform significantly better than the corresponding multiobjective PSO and multiobjective CRO algorithms on 19 MOPs, which include UFx with complicated *PS*s and GLTx with complicated *PF*s. The SMHPCRO had 16 benchmarks better than the OMOPSO, which was ranked eighth. The SMHPCRO had 16 benchmarks better than the DMOPSO, which was ranked seventh. The SMHPCRO had 15 benchmarks better than the SMOPSO, which was ranked sixth, and on the UF10, the SMOPSO was similar to the SMHPCRO. The SMHPCRO had 15 benchmarks better than the MMOPSO, which was ranked fifth, and on the UF6, the MMOPSO was similar to the SMHPCRO. The SMHPCRO had 15 benchmarks better than the AgMOPSO, the RM-MEDA was ranked second, and on the UF8, the MMOPSO was better than the SMHPCRO. The SMHPCRO had 16 benchmarks better than the NCRO, and the MOECRO/D was ranked third. In Figs. 7 and 8, the results of the compared algorithms on the UF1 benchmark demonstrate that the SMHPCRO can obtain effective convergence diversity.

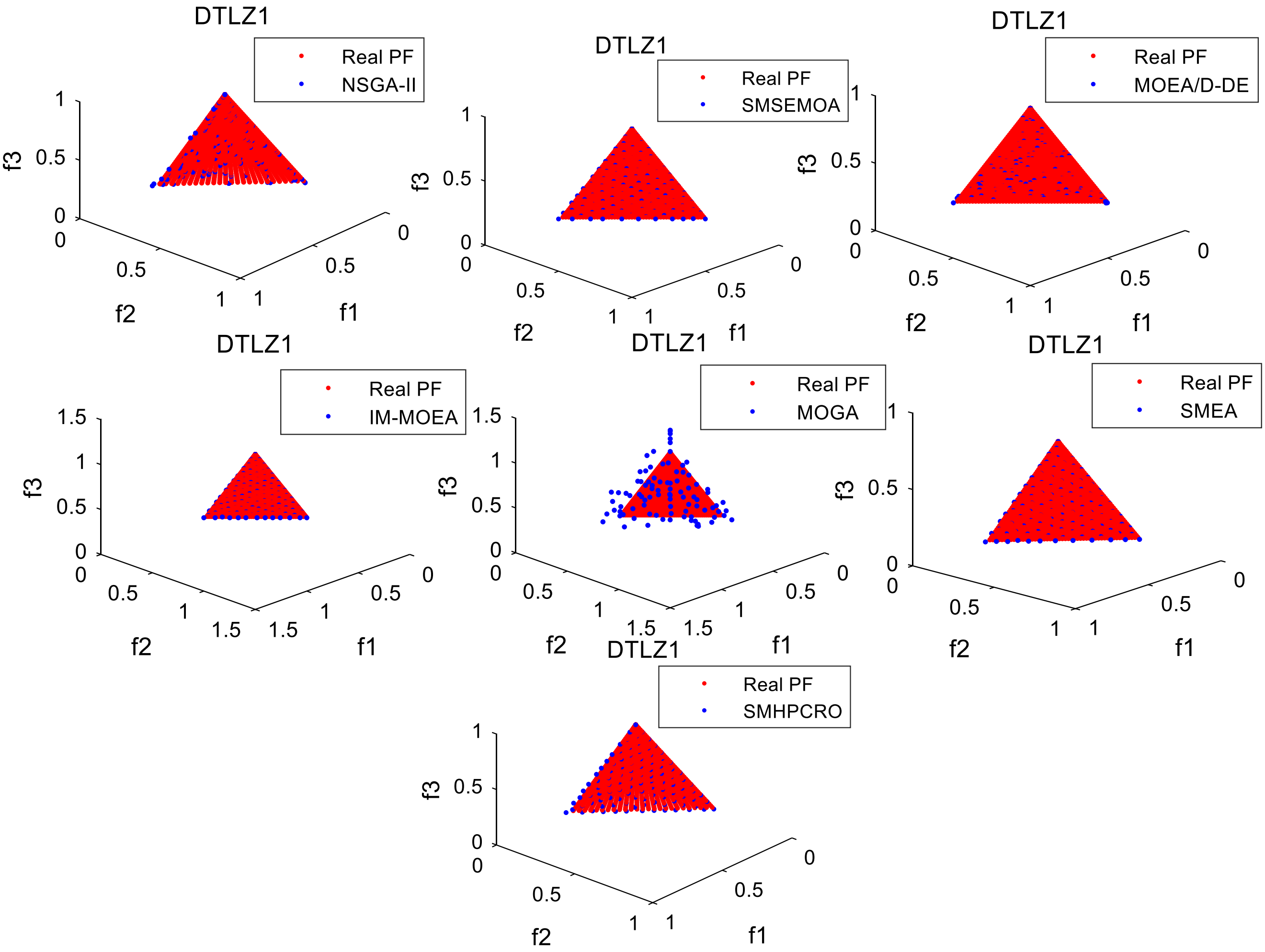


Fig. 7 The set of nondominated associated with the best IGD value among 30 runs obtained by SMHPCRO and seven compared dominance, decomposition and indicator based MOEAs on nineteen MOPs multiobjective on DTLZ1

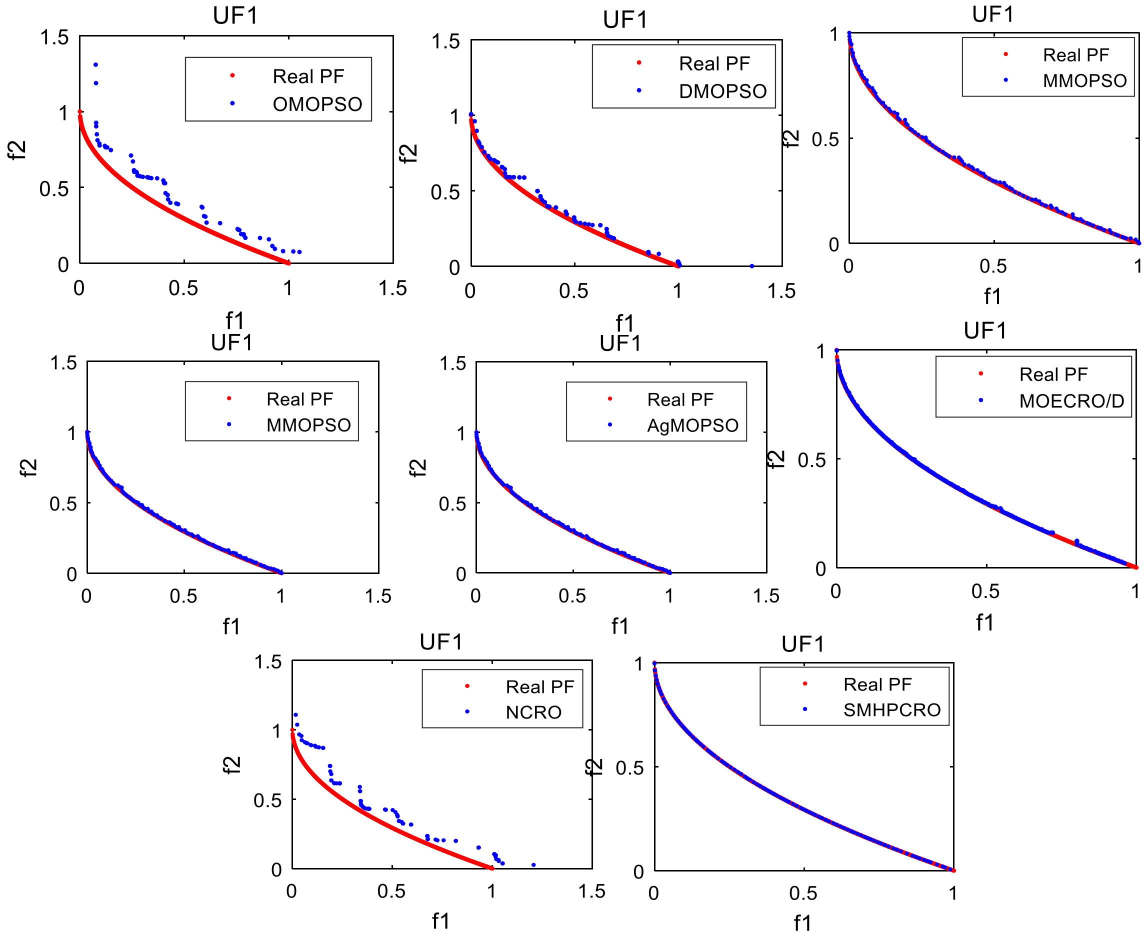


Fig. 8 The set of nondominated associated with the best IGD value among 30 runs obtained by SMHPCRO and seven compared multiobjective PSO algorithms and multiobjective CRO algorithms on UF1

A comparison of existing dominance, decomposition, and indicator-based MOEAs was conducted on 19 MOPs without variable linkages of ZDTx, DTLZx, and WFGx.

From Table 5, it can be observed that the SMHPCRO algorithms perform significantly better than the corresponding existing dominance-based method, decomposition-based method, and indicator-based MOEAs on ZDTx, DTLZx, and WFGx. SMHPCRO had 16 benchmarks better than NSGA-II, which was ranked fifth. The SMHPCRO had 16 benchmarks better than SMSEMOA, which was ranked sixth. The SMHPCRO had 14 benchmarks better than MOEA/D-DE, which was ranked fourth, and on the UF2, the MOEA/D-DE was better than the SMHPCRO, whereas on the UF5, the MOEA/D-DE was similar to the SMHPCRO. SMHPCRO had 14 benchmarks better than the IM-MOEA, which was ranked third, and on the UF5 and UF8, the IM-MOEA was better than the SMHPCRO.

A comparison of multiobjective PSO and multiobjective CRO algorithms was conducted on 19 MOPs using variable linkages of UFx with complicated *PS*s and GLTx with complicated *PF*s.

The SMHPCRO had 14 benchmarks better than MOGOA, which was ranked seventh. The SMHPCRO has nine benchmarks better than SMEA, which was ranked second, and on the UF1, UF7, UF10 and GLT5, the SMEA was similar to the SMHPCRO, while on the UF2, UF8 and GLT2, the SMEA was better than the SMHPCRO. In Fig. 9, the results of the compared algorithms on the UF1 benchmark demonstrate that the SMHPCRO can achieve effective convergence diversity. Moreover, Fig. 9 illustrates that the SMSEMOA can achieve effective convergence, featuring a selection operator based on the hyper-volume measure combined with non-dominated sorting, but exhibits poor diversity. The UF1 possesses the characteristics of variable linkages and complicated *PS*s, whereas the SMSEMOA exhibits poor decoupling ability on complicated benchmarks. The MOEA/D-DE uses the Tchebycheff method to decompose the MOPs into a scalar number of single-objective sub-problems, each of which has a direction vector that can maintain the population diversity. The IM-MOEA exhibits effective diversity, the MOGA offers

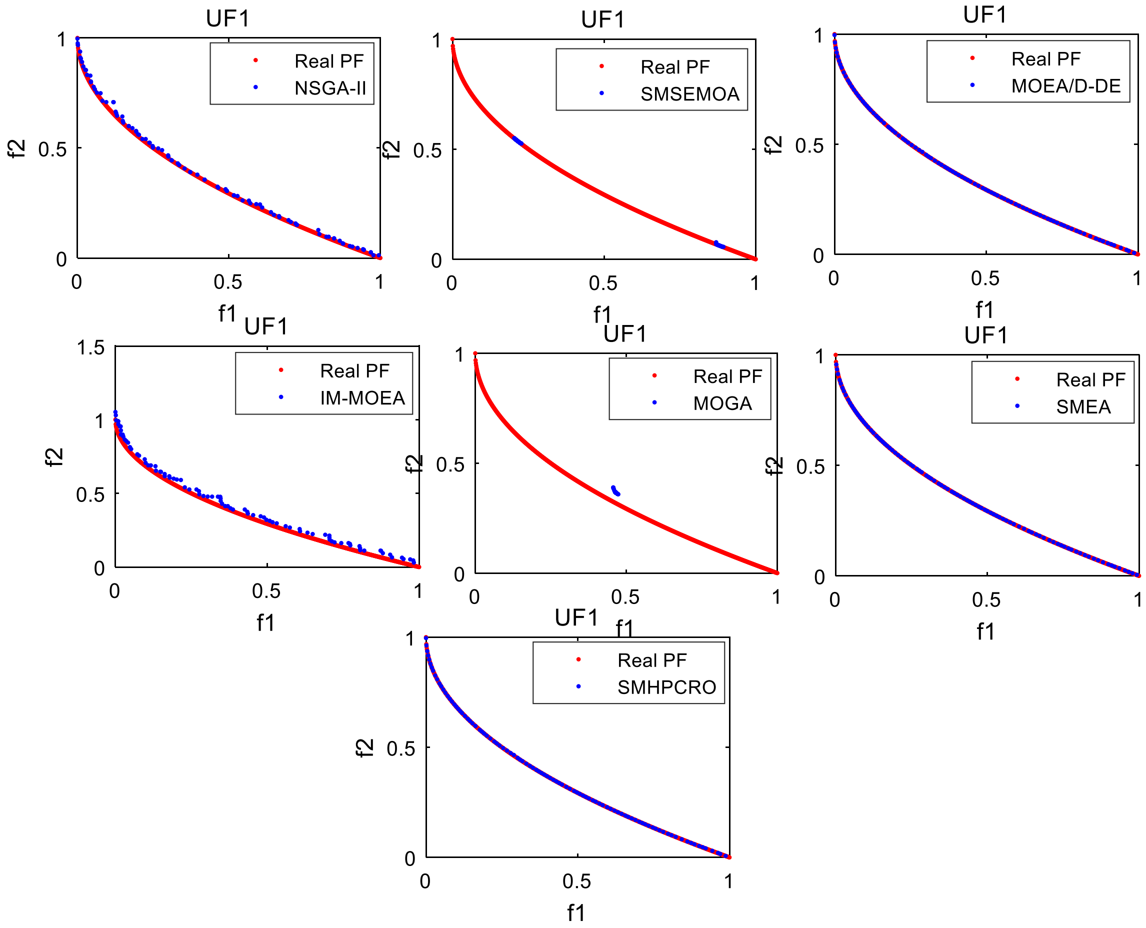


Fig. 9 The set of nondominated associated with the best IGD value among 30 runs obtained by SMHPCRO and seven compared dominance, decomposition and indicator based MOEAs on nineteen MOPs multiobjective on UF1

poor performance on the complicated benchmark UF1 and the SMEA exhibits effective performance on the UF1.

A comparison with the existing dominance-based method, decomposition-based method, and indicator-based MOEAs was carried out on 16 MOPs using variable linkages of UFx with complicated *PS*s and GLTx with complicated *PF*s.

Table 6 The experimental results of HPCRO and SMHPCRO over 30 independent runs on six MOPs

Fun	MOPSO	HPCRO	SMHPCRO
GLT1	3.964E-02 (3.425E-02)	1.671E-02 (1.941E-03)	**1.051E-03** (2.732E-03)
GLT2	8.245E-02 (7.366E-03)	1.579E-02 (2.912E-03)	**1.347E-02** (9.443E-04)
GLT3	2.519E-02 (5.667E-02)	1.392E-02 (4.854E-02)	**4.391E-03** (2.609E-03)
GLT4	3.964E-02 (5.914E-03)	2.785E-02 (1.456E-02)	**1.206E-02** (9.101E-03)
GLT5	9.687E-02 (7.661E-03)	5.570E-02 (2.426E-03)	**3.089E-02** (3.804E-04)
GLT6	1.446E-01 (3.921E-03)	1.114E-01 (9.708E-03)	**2.257E-02** (4.232E-04)

4.6 Impact of SOM on SMHPCRO

In this study, a hybrid multiobjective optimisation algorithm combining the PSO and extend CRO algorithms is designed, using the SOM method to partition the population into subpopulations. Each molecule or particle has a subpopulation, with a distributed organisation in the decision space. This makes it easy to select neighbourhood leaders. This study adopts the minimax distance between the reference point and current subpopulation solutions as the neighbour leader. The global leader is the same as the neighbour leader, which is the minimax distance between the reference point and the entire population solutions. The selection of leaders will result in premature convergence of the algorithm; thus, the extended CRO is used and the SOM method can enhance the local exploitation ability. A new solution is generated by the SMHPCRO algorithm, which can use the extended CRO or

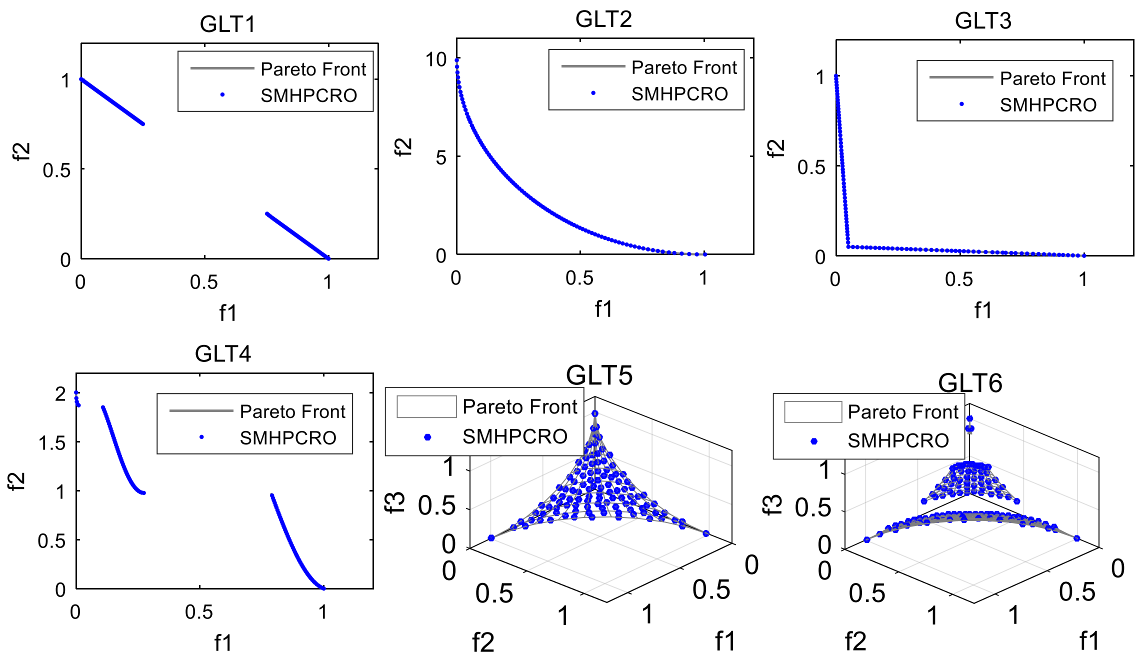


Fig. 10 The set of nondominated associated with the best Pareto front obtained by SMHPCRO on GLT-x

PSO algorithm by means of a certain probability. When the extended CRO algorithm generates a new solution, it requires another solution that differs from the current solution. The selection of another solution can be conducted from the neighbourhood and the entire population according to a probability. In this manner, the exploration and exploitation can be balanced in the SMHPCRO.

In order to explain the role of SOM in the SMHPCRO more clearly, a combination of the PSO and extended CRO algorithms without using the SOM method (HPCRO) is investigated. The following presents a comparison between the SMHPCRO and HPCRO algorithms on the benchmarks GLT-x with complicated *PF*s.

Table 6 displays the means and standard deviations of the *IGD* metric values of 30 final populations obtained by means of the SMHPCRO and HPCRO. It can be observed that the SMHPCRO provides superior mean *IGD* metric values on GLT-x to HPCRO. From Fig. 10, it can be observed that the SMHPCRO exhibits effective diversity and convergence on GLT-x. In order to analyse the difference between the SMHPCRO and HPCRO more carefully, Fig. 11 illustrates the convergence among 30 runs obtained by GLT1-GLT4. It

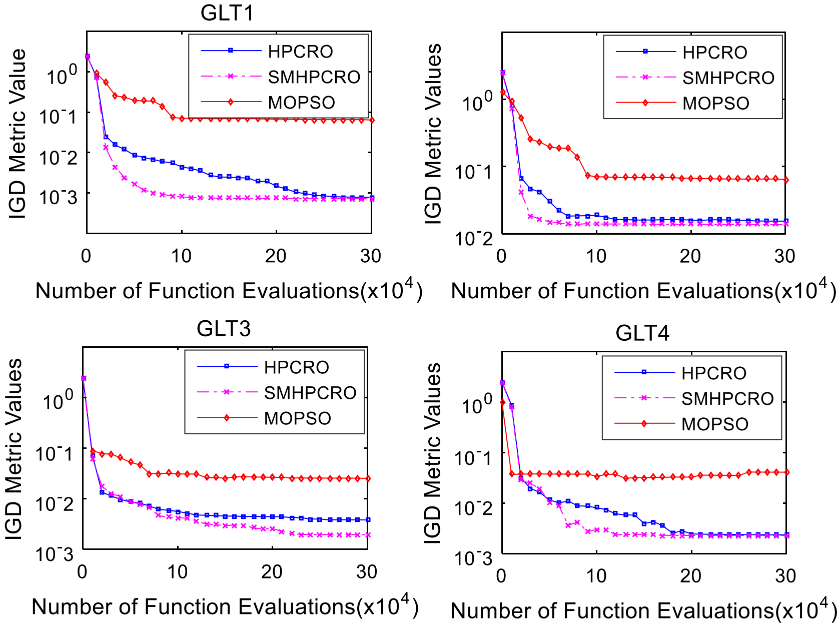


Fig. 11 The set of nondominated associated with the mean convergence among 30 runs obtained by SMHPCRO and HPCRO on GLT1-GLT4

can be observed that the HPCRO exhibits poor convergence during the early stage compared to SMHPCRO on GLT1. In the later stage, HPCRO can achieve effective convergence. The SMHPCRO uses the SOM method, which can enhance diversity, thereby preventing the algorithm from falling into premature convergence. The HPCRO exhibits effective convergence during the end stage, because it uses the synthesis operator of two molecules, which can promote the algorithm jumping out of the local optimum. The benchmark of GLT2 is similar to GLT1, but the SMHPCRO achieves better convergence than the HPCRO. On the GLT3, the SMHPCRO can achieve superior convergence to the HPCRO. This demonstrates that the SMHCRO offers an effective ability to balance the exploration and exploitation. On the GLT4, the SMHCRO exhibits similar performance convergence during the early stage, and effective convergence in the middle stage. During the later stage, the SMHCRO and HPCRO exhibit similar convergence performance. Furthermore, the red diamond in Fig. 11 represents the convergence curve of the MOPSO algorithm on different test functions. It can be observed from Fig. 11 that the MOPSO algorithm appears to converge prematurely. The experimental results demonstrate that the CRO algorithm combined with PSO can enhance the diversity of the MOPSO. Moreover, it can prevent the algorithm from falling into the local optimal solution set. Therefore, the hybrid CRO and PSO algorithm can effectively solve MOPs.

5 Conclusions

According to the SOM method, we clustered the population space, which can enhance the population diversity and make it convenient to select the neighbour leader. In order to enhance the global search ability, the extended CRO algorithm was used to prevent premature convergence of the developed PSO. Furthermore, the SMHPCRO was investigated on a large number of benchmarks with either complicated *PF* shapes or complicated *PS* shapes, and its performance was compared with 12 other MOPs. The experimental results demonstrate that the proposed approach is effective for most benchmark problems, particularly on multimodal and difficult benchmarks, which suggests that it can be considered as a very promising MOEA in the field of MOPs.

Further work will include research into the multi-swarm and cooperation mechanism to make the algorithm more efficient. Moreover, the algorithm may be applied to the constrained, dynamic, and noisy multiobjective optimisation domain. It is expected that the SMHPCRO will be used in real-world optimisation problems.

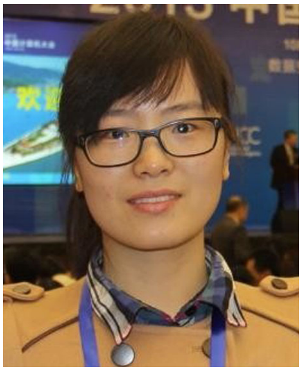
Acknowledgments The authors would like to thank them for sharing their source codes and providing guidelines to tune the control parameters. This work is partly supported by the National Natural Science Foundation of China under Grant 61272283, 61073091, 61100173, 61403304 and 14JK1511. The statements made herein are solely the responsibility of the author[s].

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

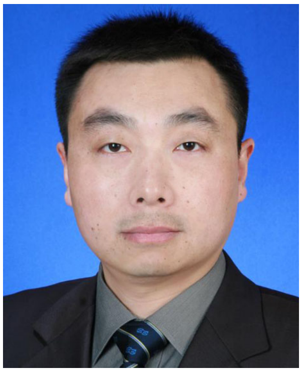
References

1. Zhou A, Qu BY, Li H et al (2011) Multiobjective evolutionary algorithms: a survey of the state of the art [J]. Swarm Evol Comput 1(1):32–49
2. Girão-Silva R, Craveirinha J, Gomes T et al (2017) A network-wide exact optimization approach for multiobjective routing with path protection in multiservice multiprotocol label switching networks [J]. Eng Optim 49(7):1226–1246
3. Wang G, Chen J, Cai T et al (2013) Decomposition-based multi-objective differential evolution particle swarm optimization for the design of a tubular permanent magnet linear synchronous motor [J]. Eng Optim 45(9):1107–1127
4. Li X, Deb K, Fang Y (2017) A derived heuristics based multi-objective optimization procedure for micro-grid scheduling [J]. Eng Optim 49(6):1078–1096
5. Tejani GG, Pholdee N, Bureerat S et al (2018) Multiobjective adaptive symbiotic organisms search for truss optimization problems [J]. Knowl-Based Syst 161(1):398–414
6. Liagkouras K (2018) A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem[J]. Knowl-Based Syst
7. Li H, Wang L, Hei X et al (2018) A decomposition-based chemical reaction optimization for multi-objective vehicle routing problem for simultaneous delivery and pickup with time windows[J]. Memetic Computing 10(1):103–120
8. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm [J]. IEEE Trans Evol Comput 12(1):41–63
9. Zhang Q, Li H (2007) MOEA/D: a multi-objective evolutionary algorithm based on decomposition [J]. IEEE Trans Evol Comput 11(6):712–731
10. Zhang H, Zhou A, Zhang Q et al (2016) A self-organizing multiobjective evolutionary algorithm [J]. IEEE Trans Evol Comput 20(5):792–806
11. Bergh FVD (2002) An analysis of particle swarm optimizers. University of Pretoria, Pretoria
12. Jiang Q, Wang L, Hei X et al (2016) The performance comparison of a new version of artificial raindrop algorithm on global numerical optimization [J]. Neurocomputing 179:1–25
13. Lam AYS, Li VOK (2012) Chemical reaction optimization: a tutorial [J]. Memet Comput 4(1):3–17
14. Lam AYS, Li VOK, Yu JJQ (2012) Real-coded chemical reaction optimization [J]. IEEETrans Evol Comput 16(3):339–353
15. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems [J]. Appl Intell 48(4):805–820
16. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization [J]. IEEE Trans Evol Comput 8(3):256–279
17. Zapotecas Martínez S, Coello Coello CA (2011) A multi-objective particle swarm optimizer based on decomposition [C]. Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM 69–76

18. Zhu Q, Lin Q, Chen W et al (2017) An external archive-guided multi-objective particle swarm optimization algorithm [J]. IEEE Trans Cybern 47(9):2794–2808
19. Lin Q, Li J, Du Z et al (2015) A novel multi-objective particle swarm optimization with multiple search strategies [J]. Eur J Oper Res 247(3):732–744
20. Li Z, Nguyen TT, Chen SM et al (2015) A hybrid algorithm based on particle swarm and chemical reaction optimization for multi-object problems [J]. Appl Soft Comput 35:525–540
21. Duan H, Gan L (2014) Orthogonal multi-objective chemical reaction optimization approach for the brushless DC motor design [J]. IEEE Trans Magn 51(1):1–7
22. Bechikh S, Said LB (2014) An indicator-based chemical reaction optimization algorithm for multi-objective search [C]. GECCO (Companion):85–86
23. Cheng R, Jin Y, Narukawa K et al (2015) A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling [J]. IEEE Trans Evol Comput 19(6):838–856
24. Deb K, Thiele L, Laumanns M, Zitzle E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (eds) Evolutionary multiobjective optimization. Theoretical advances and applications. Springer, USA, pp 105–145
25. Huband S, Barone L, While RL et al (2005) A scalable multi-objective test problem toolkit [C]. EMO 3410:280–295
26. Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S, (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition, Technical Report CES-487, The School of Computer Science and Electronic Engineering, University of Essex, Tech Rep, Colchester
27. Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Trans Evol Comput 6(2):182–197
28. Li H, Wang L, Hei X (2016) Decomposition-based chemical reaction optimization (CRO) and an extended CRO algorithm for multiobjective optimization [J]. J Comput Sci 17:174–204
29. Bechikh S, Chaabani A, Ben Said L (2014) An efficient chemical reaction optimization algorithm for multi-objective optimization [J]. IEEE Trans Cybern 45(10):1–1
30. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume [J]. Eur J Oper Res 181(3):1653–1669



Hongye Li received the B.S. degree in electrical and information engineering from Shaanxi University of Technology, Shaanxi, China, in 2006, the M.S. degree in pattern recognition and intelligent systems from Xi'an University of Technology, Xi'an, China, in 2010. She is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at the School of Computer Science and Engineering, Xian University of Technology, Xi'an, China. Her current research interests include metaheuristics, evolutionary computation, and multi-objective optimization.



Lei Wang received his B.S. degree and M.S. degree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 1994 and 1997, respectively, and his Ph.D. degree from in electronic science and technology from Xidian University, Xi'an, China, in 2001. He is currently a professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include evolutionary algorithms, neural networks, and data mining.