Swarm and Evolutionary Computation 54 (2020) 100651

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems

Zhun Fan [a,b,c,d,*], Zhaojun Wang [a], Wenji Li [a], Yutong Yuan [a], Yugen You [a], Zhi Yang [a], Fuzan Sun [a], Jie Ruan [a]

[a] Department of Electronic Engineering, Shantou University, Guangdong, China
[b] Key Lab of Digital Signal and Image Processing of Guangdong Province, Shantou University, Guangdong, China
[c] Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, 515063, Guangdong, China
[d] State Key Lab of Digital Manufacturing Equipment & Technology, Huazhong University of Science and Technology, 43003, Wuhan, China

ARTICLE INFO

Keywords:
Push and pull search
Constraint-handling mechanisms
Constrained multi-objective evolutionary algorithms
NSGA-II
Multi-objective to multi-objective (M2M) decomposition

ABSTRACT

In dealing with constrained multi-objective optimization problems (CMOPs), a key issue of multi-objective evolutionary algorithms (MOEAs) is to balance the convergence and diversity of working populations. However, most state-of-the-art MOEAs show poor performance in balancing them, and can cause the working populations to concentrate on part of the Pareto fronts, leading to serious imbalanced searching between preserving diversity and achieving convergence. This paper proposes a method which combines a multi-objective to multi-objective (M2M) decomposition approach with the push and pull search (PPS) framework, namely PPS-M2M. To be more specific, the proposed algorithm decomposes a CMOP into a set of simple CMOPs. Each simple CMOP corresponds to a sub-population and is solved in a collaborative manner. When dealing with constraints, each sub-population follows a procedure of "ignore the constraints in the push stage and consider the constraints in the pull stage", which helps each working sub-population get across infeasible regions. In order to evaluate the performance of the proposed PPS-M2M, it is compared with the other nine algorithms, including CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE on a set of benchmark CMOPs. The experimental results show that the proposed PPS-M2M is significantly better than the other nine algorithms. In addition, a set of constrained and imbalanced multi-objective optimization problems (CIMOPs) are suggested to compare PPS-M2M and PPS-MOEA/D. The experimental results show that the proposed PPS-M2M outperforms PPS-MOEA/D on CIMOPs.

1. Introduction

Many real-world optimization problems can be formulated as constrained multi-objective optimization problems (CMOPs), which can be defined as follows [1]:

$$\begin{cases} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, i = 1, \ldots, q \\ & h_j(\mathbf{x}) = 0, j = 1, \ldots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{cases} \tag{1}$$

where $F(\mathbf{x})$ is an m-dimensional objective vector, and $F(\mathbf{x}) \in \mathbb{R}^m$. $g_i(\mathbf{x}) \geq 0$ is an inequality constraint, and q is the number of inequality constraints. $h_j(\mathbf{x}) = 0$ is an equality constraint, and p represents the number of equality constraints. $\mathbf{x} \in \mathbb{R}^n$ is an n-dimensional decision vector.

In order to solve CMOPs with equality constraints, the equality constrains are often transformed into inequality constraints by using an extremely small positive number as follows:

$$h_j(\mathbf{x})' \equiv \delta - |h_j(\mathbf{x})| \geq 0 \tag{2}$$

To deal with a set of constraints in CMOPs, an overall constraint violation is employed as follows:

$$\phi(\mathbf{x}) = \sum_{i=1}^{q} |\min(g_i(\mathbf{x}), 0)| + \sum_{j=1}^{p} |\min(h_j(\mathbf{x})', 0)| \tag{3}$$

* Corresponding author. Department of Electronic Engineering, Shantou University, Guangdong, China.
E-mail address: zfan@stu.edu.cn (Z. Fan).

https://doi.org/10.1016/j.swevo.2020.100651
Received 15 December 2018; Received in revised form 26 January 2020; Accepted 26 January 2020
Available online 31 January 2020

Given a solution $\mathbf{x}^k \in \mathbb{R}^n$, if $\phi(\mathbf{x}^k) = 0$, $\mathbf{x}^k$ is a feasible solution, otherwise it is infeasible. All feasible solutions form a feasible solution set.

To deal with constraints, Deb et al.[2] defined a *constraint-domination* principle (CDP), that is, if a solution x^p is said to constrained-dominate a solution x^q, one of the following conditions must be met:

1. Solution x^p is a feasible solution and solution x^q is an infeasible solution.
2. Both solution x^p and solution x^q are infeasible solutions, and the overall constraint violation of solution x^p is smaller than that of solution x^q.
3. Both solution x^p and solution x^q are feasible solutions, and solution x^p dominates solution x^q in terms of objectives.

A feasible solution set $FS = \{\phi(x) = 0, x \in \mathbb{R}^n\}$ is constituted by all feasible solutions. Given a solution $\mathbf{x}^* \in FS$, if there is no any other solution $\overline{\mathbf{x}}^* \in FS$ satisfying $f_i(\overline{\mathbf{x}}^*) \leq f_i(\mathbf{x}^*)(i \in \{1, \ldots, m\})$, $\mathbf{x}^*$ is called a Pareto optimal solution. All Pareto optimal solutions constitute a Pareto set (PS). The set of the mapping vectors of PS in the objective space is called a Pareto front (PF), which is defined as $PF = \{F(\mathbf{x}) \mid \mathbf{x} \in PS\}$.

Over the past decade, a lot of research has been done in the field of multi-objective evolutionary algorithms (MOEAs) [3,4]. Recently, several constrained multi-objective evolutionary algorithms (CMOEAs) [5–8] have been proposed to solve CMOPs. CMOEAs are particularly suitable for solving CMOPs, because they can find a number of Pareto optimal solutions in a single run, and are not affected by the mathematical properties of the objective functions. Therefore, the use of evolutionary algorithms to solve multi-objective optimization problems [9,10] has become a research hot-spot in recent years.

To better balance minimizing the objectives and satisfying the constraints for CMOPs, many constraint-handling mechanisms have been suggested [11], such as penalty function method [12], constraint dominance principle (CDP) [13], stochastic ranking (SR) [14], multi-objective concepts [15] and an ensemble of constraint handling methods (ECHM) [16]. For example, the penalty function methods transform constrained optimization problems into an unconstrained optimization problem by adding constraints multiplied by penalty factors to the objectives. If penalty factors remain constant throughout the optimization process for a period of time, it is called a static penalty method [17]. If penalty factors are constructed as a function of the number of iterations or the iteration time, it is called a dynamic penalty approach [18]. If penalty factors change according to the feedback information [19] during the search process, it is called an adaptive penalty approach [20].

Deb proposed a constraint-handling method called CDP [13], in which the fitness of a feasible solution is always better than that of an infeasible solution. Subsequently, CDP was extended to differential evolution (DE) by Mezura-Montes et al. [21] to select target vectors and trial vectors. Moreover, CDP was used for designing parameter control in DE for constrained optimization [22].

In order to overcome the weakness of penalty constraint-handling methods, stochastic ranking was proposed [14], which uses a bubble-sort-like process to deal with the constrained optimization problems. Stochastic ranking uses a probability parameter $p_f \in [0, 1]$ to determine whether the comparison is based on the objectives or on the constraints. In the case of $p_f = 0$, stochastic ranking has a behavior similar to the feasibility rules. Furthermore, it can couple with various algorithms. For example, stochastic ranking has been combined with DE [23] and ant colony optimization [24].

Motivated by the no-free-lunch theorem, the MODE-ECHM [16] is developed. The MODE-ECHM combines three constraint-handling techniques, including epsilon constraint-handling (EC) [25], self-adaptive penalty functions (SP) [20] and superiority of feasible solutions (SF) [13]. Each constraint-handling technique is applied to evolve a specific sub-population in MODE-ECHM, and these sub-populations can closely communicate to share all of their offsprings, which also means that an individual discarded by its sub-population may survive in another population [11].

In this paper, we propose a new approach, namely PPS-M2M, to solve CMOPs, which combines a M2M decomposition approach [26] with push and pull search (PPS) framework [27]. Unlike other constraint-handling mechanisms, the PPS-M2M decomposes a CMOP into a number of simple constrained multi-objective optimization subproblems in the initial phase. Each sub-problem corresponds to a sub-population. During the search process, each sub-population evolves in a collaborative manner to ensure the diversity of the population. Inspired by the idea of information feedback model, some information about the constrained landscape is collected to help the parameters setting in the constraint-handling mechanisms.

In addition, the PPS-M2M divides the search process into two different phases. In the first phase, each sub-population approaches as close as possible to the unconstrained PF without considering any constraints. In the second phase, each sub-population is pulled back to the constrained PF using some constraint-handling approaches. The pull search stage is divided into two parts: (1) Only an improved epsilon constraint-handling mechanism is used [6] to optimize each subproblem for the first 90% generations; (2) In the last 10% of the generations, all sub-populations are merged into one population. Then an improved epsilon constraint-handling mechanism [6] is employed to pull the population to the feasible and non-dominated regions. In summary, the proposed PPS-M2M has the following advantages.

1. At the beginning of the search, the method decomposes the population into a set of sub-populations, and each sub-population searches for a different multi-objective sub-problem in a coordinated manner. In other words, the PFs of all these subproblems constitute the PF of a CMOP. So the computational complexity is reduced by limiting the operator to focus on each subpopulation, and the convergence and diversity of the population are effectively ensured.
2. When dealing with constraints, the method follows a procedure of "ignore the constraints first and consider the constraints second", so that infeasible regions encountered a distance before the true PF present literally no extra barriers for the working population.
3. Since the landscape of constraints has been probed in the unconstrained push searching stage, this information can be employed to guide the parameter settings for mechanisms of constraint handling in the pull search stage.

The remainder of this paper is organized as follows. Section 2 introduces the general idea of PPS framework and the M2M decomposition approach. Section 3 gives an instantiation of PPS-M2M. Section 4 designs a set of experiments to compare the proposed PPS-M2M with ten state-of-the-art CMOEAs, including PPS-MOEA/D [27], CM2M [28], MOEA/D-Epsilon [29], MOEA/D-SR [30], MOEA/D-CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], MODE-ECHM [16], CM2M2 [32] and MODE-SaE [33]. Finally, conclusions are drawn in section 5.

2. Related work

In this section, we introduce the general idea of PPS framework and the M2M population decomposition approach, which are used in the rest of this paper.

2.1. The general idea of PPS framework

The push and pull search (PPS) framework was introduced by Fan et al. [27]. Unlike other constraint handling mechanisms, the search process of PPS is divided into two different stages: the push search and the pull search, and follows a procedure of "push first and pull second", by which the working population is pushed toward the unconstrained PF without considering any constraints in the push stage, and a constraint handling mechanism is used to pull the working population to

the constrained PF in the pull stage.

In order to convert from the push search stage to the pull search stage, the following condition is applied

$$r_k \equiv \max\{rz_k, rn_k\} \le \epsilon \tag{4}$$

where ϵ is a threshold, which is defined by users. In Eq. (4), we set $\epsilon = 1e-3$ in this work. During the last l generations, rz_k is the rate of change of the ideal point according to Eq. (5), and rn_k is the rate of change of the nadir point according to Eq. (6).

$$rz_k = \max_{i=1,\dots,m} \{\frac{|z_i^k - z_i^{k-l}|}{\max\{|z_i^{k-l}|, \Delta\}}\} \tag{5}$$

$$rn_k = \max_{i=1,\dots,m} \{\frac{|n_i^k - n_i^{k-l}|}{\max\{|n_i^{k-l}|, \Delta\}}\} \tag{6}$$

where $z^k = (z_1^k, \dots, z_m^k), n^k = (n_1^k, \dots, n_m^k)$ are the ideal and nadir points in the k-th generation. $z^{k-l} = (z_1^{k-l}, \dots, z_m^{k-l})$, $n^{k-l} = (n_1^{k-l}, \dots, n_m^{k-l})$ are the ideal and nadir points in the $(k-l)$-th generation. rz_k and rn_k are two points in the interval $[0, 1]$. Δ is a very small positive number, which is used to make sure that the denominators in Eq. (5) and Eq. (6) are not equal to zero. In this paper, Δ is set to $1e-6$. When r_k is less than ϵ, the push search stage is completed and the pull search stage is ready to start.

The major advantages of the PPS framework include:

1. In the push stage, the working population conveniently gets across infeasible regions without considering any constraints, which voids the impact of infeasible regions encountered a distance before the true PF.
2. Since the landscape of constraints has already been estimated in the push search stage, valuable information can be collected to guide the parameter settings in the pull search stage, which not only facilitates the parameter settings of the algorithm, but also enhances its adaptability when dealing with CMOPs.

2.2. The M2M population decomposition approach

Employing a number of sub-populations to solve problems in a collaborative way [34] is a widely used approach, which can help an algorithm balance its convergence and diversity. One of the most popular methods is the M2M population decomposition approach [26], which decomposes a multi-objective optimization problems into a number of simple multi-objective optimization subproblems in the initialization, then solves these sub-problems simultaneously in a coordinated manner. For this purpose, K unit vectors $v^1, \dots, v^K$ in $\mathbb{R}_+^m$ are chosen in the first octant of the objective space. Then $\mathbb{R}_+^m$ is divided into K subregions $\Omega_1, \dots, \Omega_K$, where $\Omega_k (k = 1, \dots, K)$ is

$$\Omega_k = \{\mathbf{u} \in \mathbf{R}_+^m \mid \langle \mathbf{u}, \mathbf{v}^k \rangle \le \langle \mathbf{u}, \mathbf{v}^j \rangle \text{ for any } j = 1, \dots, K\} \tag{7}$$

where $\langle \mathbf{u}, \mathbf{v}^j \rangle$ is the acute angle between $\mathbf{u}$ and $\mathbf{v}^j$. Therefore, the population is decomposed into K sub-populations, each sub-population searches for a different multi-objective subproblem. Subproblem P_k is defined as:

$$\begin{cases} \text{minimize} & \mathbf{F}(x) = (f_1(x), \dots, f_m(x)) \\ \text{subject to} & x \in \prod_{i=1}^{n} [a_i, b_i] \\ & F(x) \in \Omega_k \end{cases} \tag{8}$$

Altogether there are K subproblems, and each subproblem is solved by employing a sub-population. Moreover, each sub-population has S individuals. In order to keep S individuals for each subproblem, some selection strategies are used. If sub-population M_k has less than S individuals, then $S - |M_k|$ individuals from Q (the entire population) are randomly selected and added to M_k.

A major advantage of M2M is that it can effectively balance diversity and convergence at each generation by decomposing a multi-objective optimization problem into multiple simple multi-objective optimization problems.

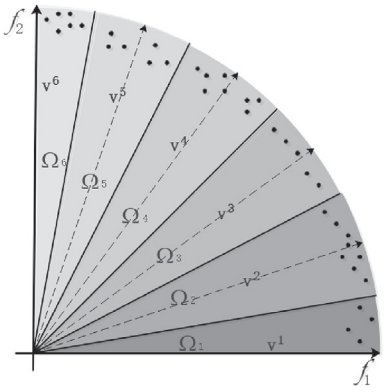


Fig. 1. The population decomposition manner.

3. An instantiation of PPS-M2M

This section describes the details of an instantiation, which combines the M2M decomposition approach with the PPS framework in a non-dominated sorting framework to solve CMOPs.

3.1. The M2M decomposition approach

In the initial stage, PPS-M2M uses a decomposition strategy to decompose a CMOP into a set of sub-problems that are solved in a collaborative manner with each sub-problem corresponding to a sub-population. For the sake of simplicity, we assume that all the objective functions $f_1(x), \dots, f_m(x)$ of a CMOP are non-negative. Otherwise, the objective function $f_i(x)$ is replaced by $f_i(x) - \overline{f}_i$, where $\overline{f}_i$ is the minimum value of the objective function $f_i(x)$ found so far.

Assuming that the objective function space is divided into K subregions, K direction vectors $v^1, \dots, v^k$ are uniformly distributed on the first octant of the unit sphere, where v^k is the center of the kth subregion. Then the objective function space is divided into K non-adjacent sub-regions $\Omega_1, \dots, \Omega_k$, where the kth $(k = 1, \dots, K)$ sub-regions can be obtained by Eq. (8). Through such a decomposition method, the multi-objective optimization problem (Eq. (1)) can be decomposed into K simple CMOPs. The procedure is described in Algorithm 1. As an example when the objective number $m = 2$ and the number of sub-regions $K = 6$, the population decomposition is illustrated in Fig. 1, where $v_1, \dots, v_6$ are six evenly distributed direction vectors.

Algorithm 1 Allocation of Individuals to Sub-populations.

```
Function result = AllocationSubPop(Q,K)
    for i ← 1 to K do
        Initialize M_k as the solutions in Q whose F-values
          are in Ω_k according to Eq. (8);
        // Q: a set of individual solutions and their
          F-values.
    end
    return M_1, ..., M_K;
end
```

3.2. The PPS framework

The search process of the PPS framework is divided into two main search stages: the push search stage and the pull search stage. In the push search stage, each sub-population uses an unconstrained NSGA-II

to search for non-dominated solutions without considering any constraints. When using unconstrained NSGA-II to solve simple multi-objective optimization problems, an individual i is measured by two attributes [2], including the non-domination rank i_{rank} and the crowding distance $i_{distance}$. Since individuals have two attributes, when an individual i is compared to another individual j, if one of the following two requirements is satisfied, then individual i can enter the descendant population.

1. If the individual i dominates the individual j, i.e. $i_{rank} < j_{rank}$;
2. If the individual i and the individual j are non-dominated by each other, then the one with the larger crowding distance is selected, i.e. the condition $i_{rank} = j_{rank}$ is satisfied, and $i_d > j_d$.

In other words, when two individuals belong to different non-domination rank, the individual with a lower (better) rank i_{rank} is selected; when the two individuals have the same rank, the one with less crowding distance is selected.

In the push search stage, a constrained optimization problem is optimized without considering any constraints. The pseudo-code of push search is given in Algorithm 2. In line 2, non-dominated sorting is carried out on the sub-population P_k. In lines 3–6, a number of solutions are selected into P'_k until the number of solutions in P'_k is greater than S_k. Lines 7–9 select $S_k - |P'_k|$ solutions into P'_k from F_i. In line 10, the sub-population P_k is updated by setting $P_k = P'_k$.

Algorithm 2 Push Subproblems.

```
1  Function PushSubproblems(P_k,S_k)
2      F = nondominated-sort(P_k), F = (F_1, F_2, ...);
3      P'_k = ∅ and i = 1;
4      While(|P'_k| + |F_i| ≤ S_k)
5          P'_k = P'_k ⋃ F_i;
6          i = i + 1;
7      calculate crowding-distance in F_i;
8      sort solutions in F_i by crowding-distance in a
         descending order
9      P'_k = P'_k ⋃ F_i[1 : (S_k − |P'_k|)];
10     P_k = P'_k;
11 end
```

In the pull search stage, the constrained optimization problem is optimized by considering constraints, which is able to pull the population to the feasible and non-dominated regions. The pseudocode of push search is given in Algorithm 3.

The mechanism described in Eq. (4) controls the search process to switch from the push to the pull search. At the beginning of the evolutionary process, the value of r_k is initialized to 1.0 in order to ensure a thorough search in the push stage. The value of r_k is updated by Eq. (4). When the value of r_k is less than or equal to the preset threshold ϵ, the search behavior is changed.

In the pull stage, we need to prevent the population from falling into local optimum, and balance evolutionary search between feasible and infeasible regions. To achieve these goals, an improved epsilon constraint-handling mechanism [6] is used to balance minimizing the objectives and satisfying the constraints in the pull search stage, with the detailed definition given as follows.

An improved epsilon constraint-handling mechanism is defined as follows:

$$\epsilon(k) = \begin{cases} \phi_\theta & \text{if } k = 0 \\ (1-\tau)\epsilon(k-1) & \text{if } rf_k < \alpha \\ \epsilon(0)(1-\frac{k}{T_c})^{cp} & \text{if } rf_k \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\epsilon(k)$ is the value of ϵ function, ϕ_θ is the overall constraint violation of the top θ-th individual in the initial population, rf_k is the proportion of feasible solutions in the generation k. τ controls the speed when the relaxation of constraints reduces in the case of $rf_k < \alpha$ ($\tau \in [0,1]$). α controls the searching preference between the feasible and infeasible regions. cp is used to control the reducing interval of relaxation of constraints in the case of $rf_k \geq \alpha$. $\varepsilon(k)$ stops updating until the generation counter k reaches generation T_c. $\varepsilon(0)$ is set to the maximum overall constraint violation when the push search finishes. In the case of $rf_k < \alpha$, Eq. (9) sets $\varepsilon(k)$ with an exponential decreasing speed, which has a potential to find feasible solutions more efficiently than the ε setting in Ref. [35]. In the case of $rf_k \geq \alpha$, Eq. (9) has the same ε setting as adopted in Ref. [35]. In the pull search stage, a new individual is selected using the constraint handling mechanism described by Algorithm 3.

Algorithm 3 Pull Subproblem.

```
1  Function result = PullSubproblems(M_k,gen,T_max)
2      //T_max: the maximum generation.
3      result = false;
4      if gen ≤ 0.9T_max then
5          for k ← 1 to K do
6              An improved epsilon constraint-handling
                 mechanism is used to search for
                 non-dominated and feasible solutions in M_k;
7          end
8      else
9          Merge M_1, M_2, ..., M_K into a single population
             M_s.
10         An improved epsilon constraint-handling
             mechanism is employed on M_s to search
             non-dominated and feasible solutions;
11     end
12     return result;
13 end
```

3.3. M2M embedded in PPS

The pseudo-code of PPS-M2M is introduced in Algorithm 4. The algorithm is initialized at lines 1–3. At line 2, the population of a CMOP is divided into K sub-populations, and the number of individuals for each sub-population is equal to S. At line 3, the maximum rate of change of ideal and nadir points r_k is initialized to 1.0, and the flag of search stage is set to push ($PushStage = true$). The algorithm runs repeatedly from line 4 to 38 until the termination condition is met. Lines 5–13 describe the process of generating new solutions for each sub-population. A number of new solutions are generated at lines 6–10. At lines 12–13, The solution set Q is allocated to each sub-population according to Eq. (7). The max rate of change between the ideal and nadir points during the last l generations r_k is calculated at line 15. The parameter $\epsilon(k)$ is updated at lines 16–25. The updating process for each sub-population is described in lines 26–36. If the size of sub-population M_k is less than S, then $S - |M_k|$ individual solutions are randomly selected from Q and added to M_k. If the size of sub-population M_k is greater than S, then S solutions are selected out by using the PPS framework. More specifically, at the push search stage, S individual solutions are selected out by employing non-dominated sorting method without considering any constraints, as illustrated in line 31. At the pull search stage, S individual solutions are selected out by using an improved epsilon constraint-handling approach, as illustrated in line 33. The generation counter is updated at line 37. At line 39, a set of non-dominated and feasible solutions is selected out.

As an example, the search behavior of PPS-M2M is illustrated in Fig. 2, which can be summarized as follows. At first, five direction vec-

tors $v^1, \ldots, v^5$ are uniformly sampled in the objective space, in which 5 non-adjacent sub-regions $M_1, \ldots, M_5$ are constructed. The working subpopulations have achieved the unconstrained PF without considering any constraints in the push search, as illustrated by Fig. 2(a)–(c). It is notable that in this particular case some solutions located on the unconstrained PF are feasible. In the pull search stage, the infeasible solutions are pulled to the feasible and non-dominant regions, as illustrated by Fig. 2(d)–(f).

Algorithm 4 PPS-M2M.

Input:
K: the number of subproblems;
K unit direction vectors: v^1,...,v^K;
S: the size of each subpopulation;
Q: a set of individual solutions and their F-values;
T_c: the control generation for $\varepsilon(k)$;
T_{max}: the maximum generation.

Output: a set of non-dominated and feasible solutions.

Initialization:
Decompose a population into K sub-populations $(M_1, ..., M_K)$, each sub-population contains S individuals according to $AllocationSubPop(Q, S, K)$;
Set $r_k = 1.0$, $PushStage = true$;
while $gen \leq T_{max}$ **do**
  for $k \leftarrow 1$ **to** K **do**
    foreach $x \in M_k$ **do**
      Randomly choose y from M_k;
      Apply genetic operators on x and y to generate a new solution z;
      $R := R \cup \{z\}$;
    end
    $Q := R \cup (\cup_{k=1}^{K} M_k)$;
    Use Q to set $M_1, ..., M_K$ according to Eq. (7);
  end
  if $gen >= l$ **then**
    Calculate r_k according to Eq. (4) ;
  if $gen < T_c$ **then**
    if $r_k \leq \epsilon$ *and* $PushStage == true$ **then**
      $PushStage = false$;
      $\varepsilon(gen) = \varepsilon(0) = maxViolation$;
    else
      Update $\varepsilon(gen)$ according to Eq. (9);
    end
  else
    $\varepsilon(k) = 0$;
  end
  for $k \leftarrow 1$ **to** K **do**
    if $|M_k| \leq S$ **then**
      randomly select $S - |M_k|$ solutions from Q and add them to M_k.
    else
      if $PushStage == true$ **then**
        result = $PushSubproblems(M_k, S)$;
      else
        result = $PullSubproblems(M_k, gen, T_{max})$;
      end
    end
  end
  $gen = gen + 1$;
end
Output the non-dominated and feasible solutions.

3.4. The differences between PPS-M2M and PPS-MOEA/D

The proposed PPS-M2M and PPS-MOEA/D are two different instantiations of the PPS framework. The main differences between PPS-M2M and PPS-MOEA/D are summarized as follows:

1. The proposed PPS-M2M applies a different decomposition method compared with the PPS-MOEA/D. In PPS-M2M, the M2M decomposition method is applied to decompose a CMOP into a set of simple CMOPs, while a CMOP is decomposed into a number of single objective optimization problems in PPS-MOEA/D.
2. PPS-M2M and PPS-MOEA/D use different selection strategies to select solutions to the next generation. In PPS-M2M, a non-dominated sorting approach and an improved epsilon (IEpsilon) constraint-handling method are combined to select solutions, while a combination of a decomposition and the IEpsilon constraint-handling approach is used to select offsprings in PPS-MOEA/D.
3. Even though both PPS-M2M and PPS-MOEA/D are effective CMOEAs, they are suitable for solving CMOPs with different characteristics. PPS-M2M is more suitable for solving CMOPs with imbalanced objectives and diversity-hard constraint functions, while PPS-MOEA/D is more suitable for solving CMOPs with large infeasible regions. The details can be found in Section 4.4.

4. Experimental study

In this section, the proposed PPS-M2M is compared with the other nine algorithms (CM2M [28], MOEA/D-Epsilon [29], MOEA/D-SR [30], MOEA/D-CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], MODE-ECHM [16], CM2M2 [32] and MODE-SaE [33]) on benchmarking problems. The chosen algorithms have been shown to be effective in solving CMOPs. We select benchmarking problems LIR-CMOP1-14 [6] to test the performance of the selected CMOEAs. The reason is that the feasible regions of these problems are relatively small, which can effectively evaluate the performance of the constraint-handling method in the proposed PPS-M2M.

The test instances CF1-10 [36] are also commonly used CMOPs. However, the feasible regions of them are relatively large, and a CMOEA can approximate their PFs without encountering any infeasible obstacles during the entire evolutionary process, which may not be suitable to evaluate the performance of the constraint-handling approach in the PPS-M2M. In fact, the objective functions of CF1-10 are difficult to be optimized. In the future, we will study CMOEAs which are able to solve CMOPs with difficult objectives, and employ the CF1-10 [36] to help evaluate the performance of these CMOEAs.

Furthermore, in order to illustrate the differences between PPS-M2M and PPS-MOEA/D [27], a set of new constrained and imbalanced multi-objective optimization problems (CIMOPs) are designed according to MOPs [26] and DAS-CMOPs [37]. The objective functions of CIMOPs are the same as those of MOPs [26], which are imbalanced. The constraint functions are set with diversity-hardness, which is inspired by DAS-CMOPs [37]. The suggested CIMOPs have imbalanced objective functions and diversity-hard constraint functions, which can be used to effectively evaluate the performance of PPS-M2M.

4.1. Experimental settings

To evaluate the performance of PPS-M2M, we adopts two test instances, including LIR-CMOP1-14 and CIMOP1-7. For LIR-CMOP1-12 and CIMOP1-5, each of them has two objectives. For LIR-CMOP13-14 and CIMOP6-7, each of them has three objectives. Ten compared algorithms (the proposed PPS-M2M, CM2M [28], MOEA/D-Epsilon [29], MOEA/D-SR [30], MOEA/D-CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], MODE-ECHM [16], CM2M2 [32] and MODE-SaE [33]) are independently run 30 times on LIR-CMOP1-14. The proposed PPS-M2M, PPS-MOEA/D [27], CM2M and CM2M2 are independently run 30 times

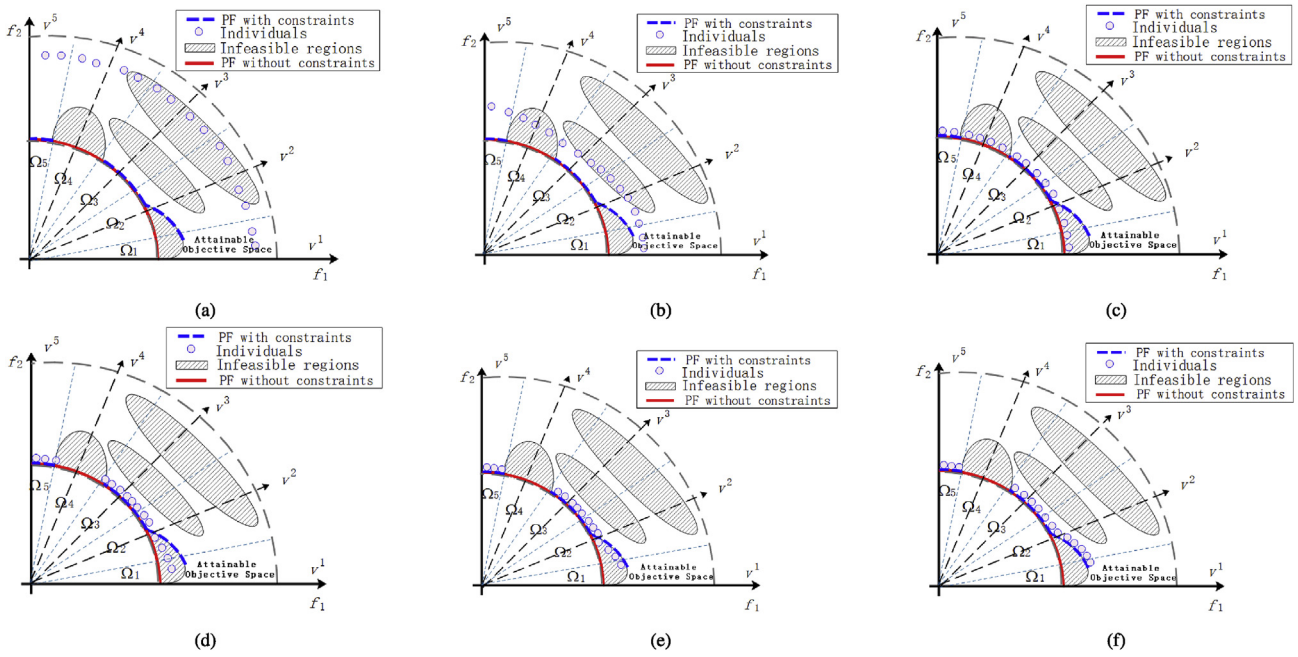


Fig. 2. Infeasible regions make the original unconstrained PF partially feasible. The objective space is divided into 5 sub-regions, and 5 direction vector $v^1, \ldots, v^5$ are uniformly distributed on the first octant of the unit sphere. The objective space is divided into 5 non-adjacent sub-regions $M_1, \ldots, M_5$. (a)–(c) show the push search process, in which the working population of each sub-region crosses infeasible regions without any barriers. The pull search process, in which the infeasible solutions in the working population of each sub-region is pulled to the feasible and non-dominant regions, as shown in (d)–(f).

on CIMOP1-7. The experimental parameters for each algorithm are explained as follows:

(1) Population size and termination condition: The population size N is set to 300 in all the algorithms. Furthermore, the ten algorithms are terminated when 300,000 function evaluations (FEs) are reached.

(2) Parameter settings: These algorithms apply DE operators to generate offsprings. The crossover and mutation operators are set to the same controls parameters, as follows: $F = 0.5$, $CR = 1.0$, where F and CR are the values of scale factor and crossover rate, respectively. Furthermore, we use standard toolkits to implement the peer algorithms, and follow the suggestions in their original study to set algorithmic parameters. Specifically, PPS-M2M and PPS-MOEA/D apply an improved epsilon constraint-handling method and the related parameters are set to those in Ref. [6], and the details are as follows: $\alpha = 0.95$, $\tau = 0.1$, $T_c = 800$ and $\theta = 0.05\,N$. PPS-M2M, CM2M and CM2M2 decompose an objective space into K sub-regions, so their K sub-regions are the same and K is set to $\lfloor\sqrt{N}\rfloor$ according to Ref. [32]. MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP are decomposition-based, and the neighborhood size T is set to 0.1 N in these algorithms. MODE-ECHM and MOEA/D-Epsilon both apply the epsilon constraint handling approach, so the speed of reducing relaxation of constraints (cp), the control generation (T_c) and the top θth individual of the initial population are the same in these algorithms, following the practice in Ref. [29]. According to Ref. [30], the stochastic ranking (S_r) in MOEA/D-SR is set to 0.05 for balancing objective and penalty functions directly. Following the guidelines in Ref. [32], the infeasible weights N_1 is set to 90 and the feasible weights N_2 is set to 210 in CM2M2.

4.2. Performance metric

In order to evaluate the performance of the proposed algorithm (PPS-M2M) and the nine state-of-the-art CMOEAs (CM2M [28], MOEA/D-Epsilon [29], MOEA/D-SR [30], MOEA/D-CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], MODE-ECHM [16], CM2M2 [32] and MODE-SaE [33]), two performance indicators, including the inverted generation distance (IGD) [38] and the hypervolume (HV) [39], are adopted in this paper.

- **Inverted Generational Distance** (IGD):

Inverted Generational Distance(IGD) is an inverse mapping of Generational Distance(GD). It is expressed by the average distance from the individual in Pareto optimal solution set to the non-dominant solution set PF obtained by the algorithm. Therefore, the calculation formula is

$$\begin{cases} IGD(P^*, A) = \dfrac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ d(y^*, A) = \min\limits_{y \in A} \{ \sqrt{\sum_{i=1}^{m} (y_i^* - y_i)^2} \} \end{cases} \tag{10}$$

where P^* is a set of representative solutions in the true PF. $d(y^*, A)$ represents the minimum Euclidean distance from point y_i^* on the Pareto optimal surface to individual y_i in P^*. The smaller the IGD value, the better the performance of the algorithm.

- **Hypervolume** (HV):

HV has become a popular evaluation index, which reflects the closeness of the set of non-dominated solutions achieved by a CMOEA to the true PF. The performance of CMOEA is evaluated by calculating the hypervolume of the space surrounded by the non-dominant solution set and the reference point. The calculation formula is as follows:

$$HV(S) = VOL(\bigcup_{x \in S} [f_1(x), z_1^r] \times \ldots [f_m(x), z_m^r]) \tag{11}$$

where $VOL(\cdot)$ is the Lebesgue measure, m denotes the number of objectives, $\mathbf{z}^r = (z_1^r, \ldots, z_m^r)^T$ is a user-defined reference point in the objective space. The bigger the HV value, the better the performance of the algorithm. The reference point is placed at 1.2 times the distance to the nadir point of the true PF. A larger value of HV indicates better performance regarding diversity and/or convergence.

Table 1
IGD results of PPS-M2M and the other nine CMOEAs on LIR-CMOP1-14. To facilitate the display of this table, Epsilon, CDP, SR, SaE and ECHM in this table are short for MOEA/D-Epsilon [29], MOEA/D-CDP [30], MOEA/D-SR [30], MODE-SaE [33] and MODE-ECHM [16] respectively. Friedman test at a 0.05 significance level is performed between PPS-M2M and each of the other nine CMOEAs, namely CM2M [28], Epsilon [29], SR [30], CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], ECHM [16], CM2M2 [32] and SaE [33]. The best mean among the compared algorithms on the test problem is highlighted in boldface.

Test Instance		PPS-M2M	CM2M [28]	CM2M2 [32]	Epsilon [29]	CDP [30]	SR [30]	C-MOEA/D [31]	NSGA-II-CDP [2]	SaE [33]	ECHM [16]
LIRCMOP1	mean	**1.330E-02**	3.106E-02	1.356E-02	5.74E-02	1.11E-01	1.81E-02	1.26E-01	3.23E-01	2.06E-02	5.429E-01
	std	4.407E-03	9.888E-03	2.993E-03	2.89E-02	5.04E-02	1.66E-02	7.03E-02	7.33E-02	1.15E-02	2.193E-01
LIRCMOP2	mean	1.485E-02	2.742E-02	**1.167E-02**	5.39E-02	1.43E-01	9.63E-03	1.40E-01	3.03E-01	1.51E-02	4.995E-01
	std	5.576E-03	8.596E-03	3.021E-03	2.13E-02	5.55E-02	7.23E-03	5.44E-02	7.24E-02	5.041E-03	2.062E-01
LIRCMOP3	mean	1.987E-02	4.383E-02	1.395E-02	8.81E-02	2.61E-01	1.78E-01	2.80E-01	4.08E-01	**7.44E-03**	7.866E-01
	std	7.371E-03	1.445E-02	5.867E-03	4.36E-02	4.33E-02	7.20E-02	4.21E-02	1.15E-01	1.91E-03	2.603E-01
LIRCMOP4	mean	2.504E-02	4.187E-02	1.196E-02	6.51E-02	2.53E-01	1.95E-01	2.59E-01	3.85E-01	**9.06E-03**	8.266E-01
	std	9.030E-03	2.371E-02	5.604E-03	3.01E-02	4.34E-02	6.40E-02	3.51E-02	1.35E-01	2.89E-03	2.754E-01
LIRCMOP5	mean	**3.922E-03**	2.941E-01	1.181E+00	1.15E+00	1.05E+00	1.04E+00	1.10E+00	5.53E-01	1.07E-01	2.087E+01
	std	1.515E-03	5.013E-01	1.826E-02	1.98E-01	3.63E-01	3.66E-01	2.99E-01	6.88E-01	2.00E-03	2.557E+00
LIRCMOP6	mean	**5.077E-03**	5.356E-01	9.915E-01	1.27E+00	1.09E+00	9.43E-01	1.31E+00	5.74E-01	4.95E-01	2.051E+01
	std	1.117E-02	5.509E-01	5.510E-01	2.95E-01	5.20E-01	5.90E-01	2.08E-01	4.21E-01	1.07E-01	2.994E+00
LIRCMOP7	mean	**3.711E-03**	5.237E-01	7.077E-02	1.51E+00	1.46E+00	1.08E+00	1.56E+00	2.38E-01	7.99E-02	2.824E+00
	std	1.993E-03	7.677E-01	5.172E-02	5.09E-01	5.58E-01	7.58E-01	4.24E-01	4.06E-01	4.07E-02	8.474E-01
LIRCMOP8	mean	**2.949E-03**	7.924E-01	6.582E-02	1.62E+00	1.38E+00	1.01E+00	1.58E+00	6.02E-01	3.31E-01	2.910E+00
	std	9.666E-05	7.455E-01	5.937E-02	3.05E-01	6.15E-01	7.24E-01	3.71E-01	7.39E-01	9.38E-01	8.183E-01
LIRCMOP9	mean	**3.704E-01**	4.599E-01	4.197E-01	4.90E-01	4.81E-01	4.85E-01	4.81E-01	6.44E-01	5.04E-01	1.007E+00
	std	2.529E-02	6.426E-02	8.113E-02	4.22E-02	5.24E-02	4.78E-02	5.24E-02	1.60E-02	4.26E-06	1.685E-01
LIRCMOP10	mean	**1.572E-02**	2.291E-01	6.108E-01	2.13E-01	2.16E-01	1.92E-01	2.13E-01	5.97E-01	4.10E-01	1.258E+00
	std	4.506E-02	8.123E-02	1.375E-01	5.32E-02	6.81E-02	6.81E-02	4.63E-02	3.20E-02	2.86E-05	1.029E-01
LIRCMOP11	mean	**2.770E-02**	4.400E-01	5.135E-01	3.47E-01	3.42E-01	3.16E-01	3.81E-01	4.87E-01	5.33E-01	1.054E+00
	std	4.656E-02	1.038E-01	1.131E-01	9.28E-02	9.22E-02	7.49E-02	8.95E-02	1.05E-02	8.59E-02	1.196E-01
LIRCMOP12	mean	**9.310E-02**	1.488E-01	2.549E-01	2.52E-01	2.69E-01	2.06E-01	2.50E+00	5.80E-01	2.52E-01	1.028E+00
	std	9.310E-02	5.757E-02	1.592E-02	8.98E-02	9.06E-02	5.61E-02	9.63E-02	1.17E-01	7.79E-02	1.303E-01
LIRCMOP13	mean	**1.874E-01**	1.57E+00	1.263E+00	1.20E+00	1.21E+00	8.86E-01	1.18E+00	1.39E+01	8.97E-01	8.530E+00
	std	2.690E-02	1.05E-02	2.254E-01	3.06E-01	3.17E-01	5.76E-01	3.78E-01	2.26E+00	1.34E-01	7.869E-01
LIRCMOP14	mean	**1.746E-01**	2.39E+00	1.220E+00	1.02E+00	1.11E+00	1.03E+00	1.25E+00	1.36E+01	9.45E-01	8.568E+00
	std	2.840E-02	1.63E-02	2.192E-01	4.86E-01	3.98E-01	4.70E-01	5.30E-02	2.17E+00	1.75E-01	0.65187
Friedman test		**1.4286**	4.7857	4.6429	5.8571	6.1786	4.0714	7	7.4286	3.8214	9.7857

Table 2
Adjusted p-values for the Friedman Aligned test in terms of mean metric (IGD). To facilitate the display of this table, Epsilon, CDP, SR, SaE and ECHM in this table are short for MOEA/D-Epsilon [29], MOEA/D-CDP [30], MOEA/D-SR [30], MODE-SaE [33] and MODE-ECHM [16] respectively.

i	algorithm	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	$p_{Holland}$	p_{Rom}	p_{Finner}	p_{Li}
1	ECHM [16]	0	0	0	0	0	0	0	0
2	NSGA-II-CDP [2]	0	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0
3	C-MOEA/D [31]	0.000001	0.000008	0.000008	0.000008	0.000008	0.000007	0.000003	0.000001
4	CDP [30]	0.000033	0.000199	0.000199	0.000199	0.000199	0.000189	0.000075	0.000034
5	Epsilon [29]	0.000109	0.000544	0.000544	0.000544	0.000544	0.000518	0.000196	0.000113
6	CM2M [28]	0.00335	0.013399	0.013399	0.010049	0.013331	0.012776	0.00502	0.003465
7	CM2M2 [32]	0.004972	0.014916	0.014916	0.014916	0.014842	0.014916	0.006388	0.005134
8	SR [30]	0.020916	0.041832	0.036525	0.036525	0.041395	0.036525	0.0235	0.021248
9	SaE [33]	0.036525	0.041832	0.036525	0.036525	0.041395	0.036525	0.036525	0.036525

4.3. Discussion of experiments

Tables 1 and 3 show the IGD and HV values of the ten algorithms on LIR-CMOP1-14. According to the Friedman aligned test, PPS-M2M achieves the highest ranking among the ten CMOEAs. The p-values calculated by the statistics of the Friedman aligned test are close to zero, which reveals the significant differences among the ten algorithms (PPS-M2M, CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE).

Tables 2 and 4 show adjusted p-values of IGD and HV values for the Friedman Aligned test, and PPS-M2M is the control method. To compare the statistical difference among PPS-M2M and the other nine algorithms, we perform a series of post-hoc tests. Each adjusted p value in Tables 2 and 4 is less than the preset significant level 0.05. To control the Family-Wise Error Rate (FWER), a set of post-hoc procedures, including the Holm procedure [40], the Holland procedure [41], the Finner procedure [42], the Hochberg procedure [43], the Hommel procedure [44], the Rom procedure [45] and the Li procedure [46], are used according to Ref. [47]. It is worth noting that some HV values of NSGA-II-CDP and MOED-ECHM are zero in Table 4. The reason is that the achieved solutions by these two algorithms are all dominated by the reference points. From the Friedman test, we can conclude that PPS-M2M is significantly better than the other nine algorithms (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE).

In order to further discuss the advantages of the proposed PPS-M2M in solving CMOPs, we plot non-dominated solutions achieved by each algorithm on LIR-CMOP1, LIR-CMOP7 and LIR-CMOP11 with the median HV values. The feasible and infeasible regions of LIR-CMOP1, LIR-CMOP7 and LIR-CMOP11, corresponding to the three different types of difficulties [37], are plotted in Fig. 3.

Table 3
HV results of PPS-M2M and the other nine CMOEAs on LIR-CMOP1-14. To facilitate the display of this table, Epsilon, CDP, SR, SaE and ECHM in this table are short for MOEA/D-Epsilon [29], MOEA/D-CDP [30], MOEA/D-SR [30], MODE-SaE [33] and MODE-ECHM [16] respectively. Friedman test at a 0.05 significance level is performed between PPS-M2M and each of the other nine CMOEAs, namely CM2M [28], Epsilon [29], SR [30], CDP [30], C-MOEA/D [31], NSGA-II-CDP [2], ECHM [16], CM2M2 [32] and SaE [33]. The best mean among the compared algorithms on the test problem is highlighted in boldface.

Test Instance		PPS-M2M	CM2M [28]	CM2M2 [32]	Epsilon [29]	CDP [30]	SR [30]	C-MOEA/D [31]	NSGA-II-CDP [2]	SaE [33]	ECHM [16]
LIRCMOP1	mean	**1.009E+00**	9.897E-01	1.008E+00	9.590E-01	7.540E-01	9.960E-01	7.410E-01	5.160E-01	2.292E-01	3.718E-01
	std	4.590E-03	1.322E-02	3.563E-03	3.280E-02	8.950E-02	2.910E-02	1.220E-01	5.570E-02	9.270E-03	1.871E-01
LIRCMOP2	mean	1.337E+00	1.321E+00	**1.345E+00**	1.280E+00	1.060E+00	1.340E+00	1.070E+00	8.240E-01	3.587E-01	5.971E-01
	std	7.093E-03	1.124E-02	4.409E-03	2.880E-02	1.080E-01	1.470E-02	9.100E-02	1.150E-01	2.839E-03	2.620E-01
LIRCMOP3	mean	**8.650E-01**	8.407E-01	8.645E-01	7.980E-01	4.860E-01	5.910E-01	4.710E-01	4.080E-01	2.054E-01	1.538E-01
	std	6.010E-03	1.787E-02	5.795E-03	3.930E-02	4.310E-02	1.070E-01	4.090E-02	2.880E-02	9.720E-04	1.793E-01
LIRCMOP4	mean	**1.095E+00**	1.051E+00	1.088E+00	1.020E+00	7.350E-01	8.150E-01	7.310E-01	6.170E-01	3.133E-01	1.707E-01
	std	8.869E-03	3.870E-02	3.633E-03	4.190E-02	5.440E-02	8.700E-02	5.160E-02	1.060E-01	2.900E-03	2.774E-01
LIRCMOP5	mean	**1.455E+00**	1.084E+00	1.513E-01	4.300E-02	1.630E-01	1.820E-01	9.720E-02	9.390E-01	2.603E-01	0.000E+00
	std	6.935E-03	6.097E-01	4.184E-01	2.350E-01	4.430E-01	4.390E-01	3.700E-01	3.210E-01	2.540E-01	0.000E+00
LIRCMOP6	mean	**1.118E+00**	5.337E-01	1.455E-01	5.400E-02	1.880E-01	3.020E-01	2.330E-02	4.130E-01	5.462E-02	0.000E+00
	std	3.888E-02	3.987E-01	3.034E-01	2.210E-01	3.870E-01	4.620E-01	1.280E-01	1.890E-01	1.090E-02	0.000E+00
LIRCMOP7	mean	**3.005E+00**	2.021E+00	2.776E+00	3.030E-01	3.740E-01	9.880E-01	2.040E-01	2.400E+00	2.587E-01	0.000E+00
	std	2.434E-02	1.349E+00	1.706E-01	9.070E-01	9.580E-01	1.270E+00	7.520E-01	6.520E-01	1.020E-02	0.000E+00
LIRCMOP8	mean	**3.016E+00**	1.498E+00	2.769E+00	1.060E-01	5.170E-01	1.100E+00	1.660E-01	1.900E+00	1.044E-01	0.000E+00
	std	2.494E-03	1.277E+00	2.149E-01	5.490E-01	1.050E+00	1.200E+00	6.110E-01	7.560E-01	1.480E-01	0.000E+00
LIRCMOP9	mean	**3.152E+00**	2.826E+00	2.941E+00	2.740E+00	2.770E+00	2.750E+00	2.770E+00	2.060E+00	3.585E-01	1.110E+00
	std	8.752E-02	2.114E-01	1.698E-01	1.480E-01	1.840E-01	1.640E-01	1.840E-01	1.080E-02	9.970E-05	2.421E-01
LIRCMOP10	mean	**3.216E+00**	2.845E+00	1.847E+00	2.890E+00	2.880E+00	2.930E+00	2.890E+00	2.040E+00	4.913E-01	3.834E-01
	std	7.555E-02	1.720E-01	4.934E-01	1.020E-01	1.360E-01	1.350E-01	9.770E-02	4.450E-02	1.070E-04	2.594E-01
LIRCMOP11	mean	**4.343E+00**	3.115E+00	2.738E+00	3.340E+00	3.350E+00	3.380E+00	3.240E+00	3.110E+00	3.083E-01	1.280E+00
	std	1.401E-01	3.003E-01	5.532E-01	2.570E-01	2.570E-01	2.900E-01	2.550E-01	1.540E-02	6.890E-02	2.850E-01
LIRCMOP12	mean	**5.409E+00**	5.221E+00	4.900E+00	4.880E+00	4.830E+00	5.030E+00	4.890E+00	3.280E+00	5.077E-01	1.622E+00
	std	1.717E-01	1.878E-01	1.115E-01	3.170E-01	3.280E-01	1.750E-01	3.450E-01	3.610E-01	2.250E-02	3.114E-01
LIRCMOP13	mean	**4.874E+00**	4.566E-01	3.119E+00	4.550E-01	4.630E-01	1.890E+00	6.290E-01	0.000E+00	9.476E-02	0.000E+00
	std	1.068E-01	2.100E-02	2.746E+00	1.300E+00	1.420E+00	2.570E+00	1.710E+00	0.000E+00	6.010E-02	0.000E+00
LIRCMOP14	mean	**5.449E+00**	4.142E-01	2.313E+00	1.330E+00	8.810E-01	1.270E+00	1.800E-01	0.000E+00	3.178E-02	0.000E+00
	std	1.237E-01	5.110E-03	2.835E+00	2.450E+00	1.970E+00	2.290E+00	2.600E-01	0.000E+00	4.360E-02	0.000E+00
Friedman test		**1.1786**	3.9286	3.5	5.75	5.5357	3.8929	6.3571	6.6429	8.6429	9.5714

Table 4
Adjusted p-values for the Friedman Aligned test in terms of mean metric (HV). To facilitate the display of this table, Epsilon, CDP, SR, SaE and ECHM in this table are short for MOEA/D-Epsilon [29], MOEA/D-CDP [30], MOEA/D-SR [30], MODE-SaE [33] and MODE-ECHM [16] respectively.

i	algorithm	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	$p_{Holland}$	p_{Rom}	p_{Finner}	p_{Li}
1	ECHM [16]	0	0	0	0	0	0	0	0
2	SaE [33]	0	0	0	0	0	0	0	0
3	NSGA-II-CDP [2]	0.000002	0.000013	0.000013	0.000013	0.000013	0.000012	0.000005	0.000002
4	C-MOEA/D [31]	0.000006	0.000036	0.000036	0.000036	0.000036	0.000034	0.000014	0.000006
5	Epsilon [29]	0.000065	0.000324	0.000324	0.000324	0.000324	0.000308	0.000117	0.000068
6	CDP [30]	0.00014	0.000561	0.000561	0.000561	0.000561	0.000535	0.000211	0.000147
7	CM2M [28]	0.016256	0.048767	0.035392	0.032511	0.047979	0.035392	0.020851	0.016694
8	SR [30]	0.017696	0.048767	0.035392	0.035392	0.047979	0.035392	0.020851	0.018146
9	CM2M2 [32]	0.042498	0.048767	0.042498	0.042498	0.047979	0.042498	0.042498	0.042498

As shown in Fig. 3(a), the feasible region of LIR-CMOP1 is very small, which indicates a feasibility-hard problem. Non-dominated solutions achieved by each algorithm on LIR-CMOP1 with the median HV values are plotted in Fig. 4. We can observe that PPS-M2M, CM2M, MOEA/D-SR, MODE-ECHM, CM2M2 and MODE-SaE can converge to the true PF and have good diversity, as shown in Fig. 4(a), (b), (e), (h), (i) and (j). However, according to the HV and IGD metrics, the proposed PPS-M2M is better than the other compared algorithms. The convergence of CM2M is worse than that of PPS-M2M, as shown in Fig. 4(b). A possible reason is that some sub-regions are assigned to infeasible regions in CM2M, and solutions in these sub-regions cannot converge to the true PF. For the rest of four algorithms, their diversity is worse than that of PPS-M2M, as shown in Fig. 4(c), (d), (f) and (g).

On LIR-CMOP7, there are three large non-overlapping infeasible regions in the front of the unconstrained PF, as illustrated in Fig. 3(b). In addition, the unconstrained PF is covered by one of the infeasible regions, which indicates a convergence-hard problem. The results of PPS-M2M and the other nine algorithms (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE) on LIR-CMOP7 are shown in Fig. 5. We can see that only PPS-M2M can get across the infeasible regions to reach the whole true PF, as illustrated in Fig. 5(a). The reason is that PPS-M2M ignore the constraints at the push stage, and can conveniently get across the infeasible regions. However, the other compared algorithms cannot get across the three infeasible regions effectively since two large infeasible regions in front of the PF hinder the way of populations of these nine algorithms to converge to the true PF.

The PF of LIR-CMOP11 is discrete as shown in Fig. 3(c), which suggests a problem with diversity-hardness. There are seven Pareto optimal solutions. Two solutions are located on the unconstrained PF, and five solutions are located on the constrained PF. Fig. 6 shows the results of the nine tested algorithms on LIR-CMOP11. From Fig. 6(a)–(b), we can see that only PPS-M2M and CM2M can find all the Pareto optimal solutions. Furthermore, the PPS-M2M has good convergence com-

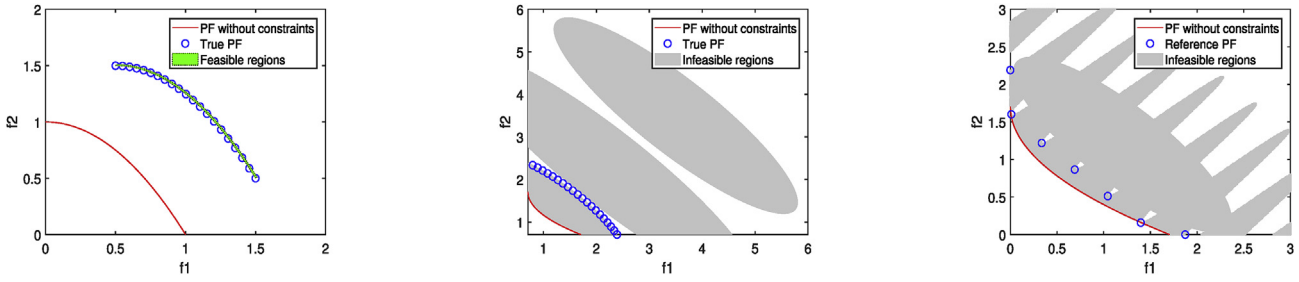


Fig. 3. Illustrations of the feasible and infeasible regions of LIR-CMOP1, LIR-CMOP7 and LIR-CMOP11, corresponding to three different types of difficulties as discussed in Ref. [37].

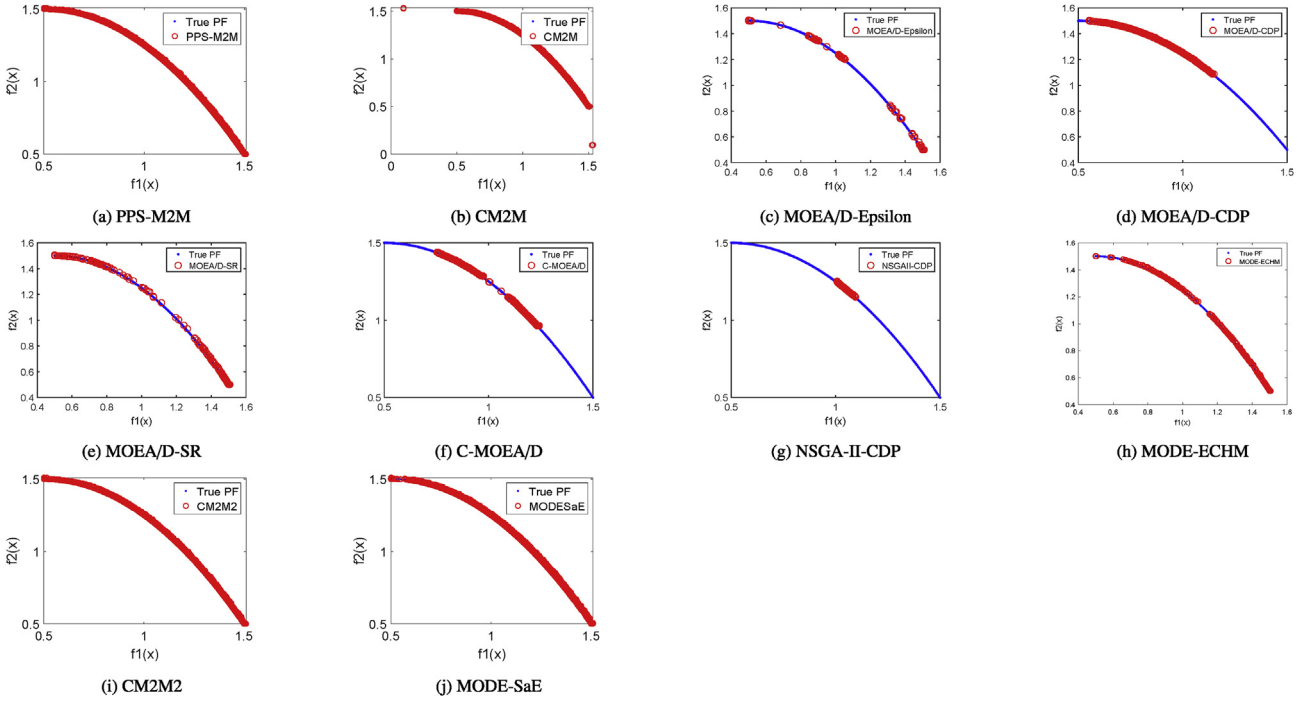


Fig. 4. The non-dominated solutions achieved by each algorithm on LIR-CMOP1 with the median HV values.

pared to the CM2M. A possible reason is that each sub-population is combined into a whole population which is evolved by employing the improved epsilon constraint-handling method at the last ten percentages of the maximum generation. The rest of algorithms (MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP MODE-ECHM, CM2M2 and MODE-SaE) can find only a part of Pareto optimal solutions, because infeasible regions block the way of populations of these algorithms towards the true PF. Furthermore, it is difficult to set parameters of constraint-handling methods properly in these algorithms, because the landscape of constraints is not well explored and estimated during the search.

Based on the above observations and analysis, we can conclude that the proposed PPS-M2M outperforms the other nine CMOEAs (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE) on most of the test instances.

4.4. The comparison between PPS-M2M and PPS-MOEA/D

To compare the two PPS based algorithms, namely PPS-M2M and PPS-MOEA/D, both LIR-CMOPs [6] and CIMOPs are suggested. The detailed definitions of CIMOP1-7 are given in Table 7. The feasible regions and true PFs of CIMOP1-7 are shown in Fig. 7. It is worth noting that while LIR-CMOPs are featured by large infeasible regions, and CIMOPs are featured by imbalanced objective functions and diversity-hard constraint functions.

At first, the proposed PPS-M2M and PPS-MOEA/D are compared on LIR-CMOP1-14. Tables 5 and 6 show the IGD and HV values of PPS-M2M and PPS-MOEA/D on LIR-CMOP1-14. From these tables, we can observe that PPS-MOEA/D outperforms the proposed PPS-M2M on LIR-CMOP1-14. From these tables, we can observe that PPS-MOEA/D outperforms the proposed PPS-M2M on LIR-CMOP1-14, even though PPS-M2M can achieve significantly better results than the other nine algorithms (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE) in this test suite (as shown in Tables 1–4).

Tables 8 and 10 show the IGD and HV values of the four algorithms, including PPS-M2M, PPS-MOEA/D, CM2M and CM2M2, on CIMOP1-7. Since CM2M and CM2M2 adopt M2M decomposition approach, they are included to verify that the performance of PPS-M2M does not only rely on the M2M approach, but also on the constraint handing mechanism in the PPS framework. According to the Friedman aligned test, PPS-M2M achieves the highest ranking among the four CMOEAs. To compare the statistical difference among PPS-M2M and the other three

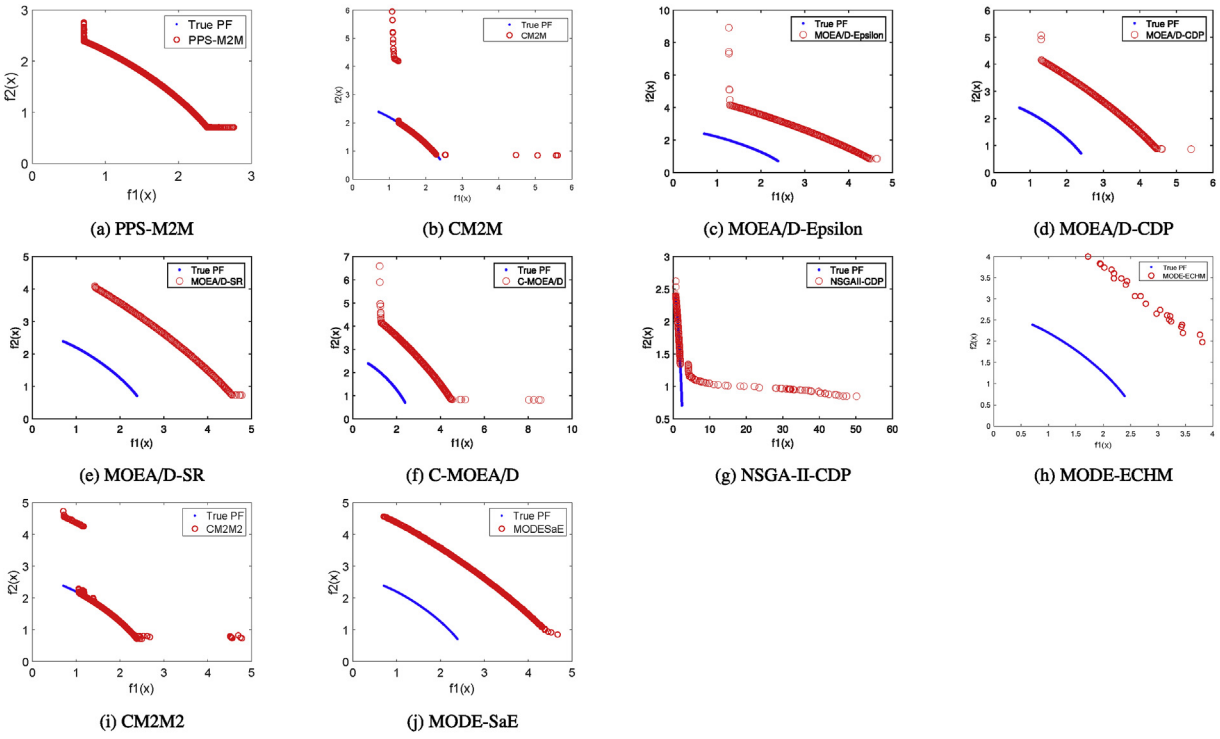


Fig. 5. The non-dominated solutions achieved by each algorithm on LIR-CMOP7 with the median HV values.

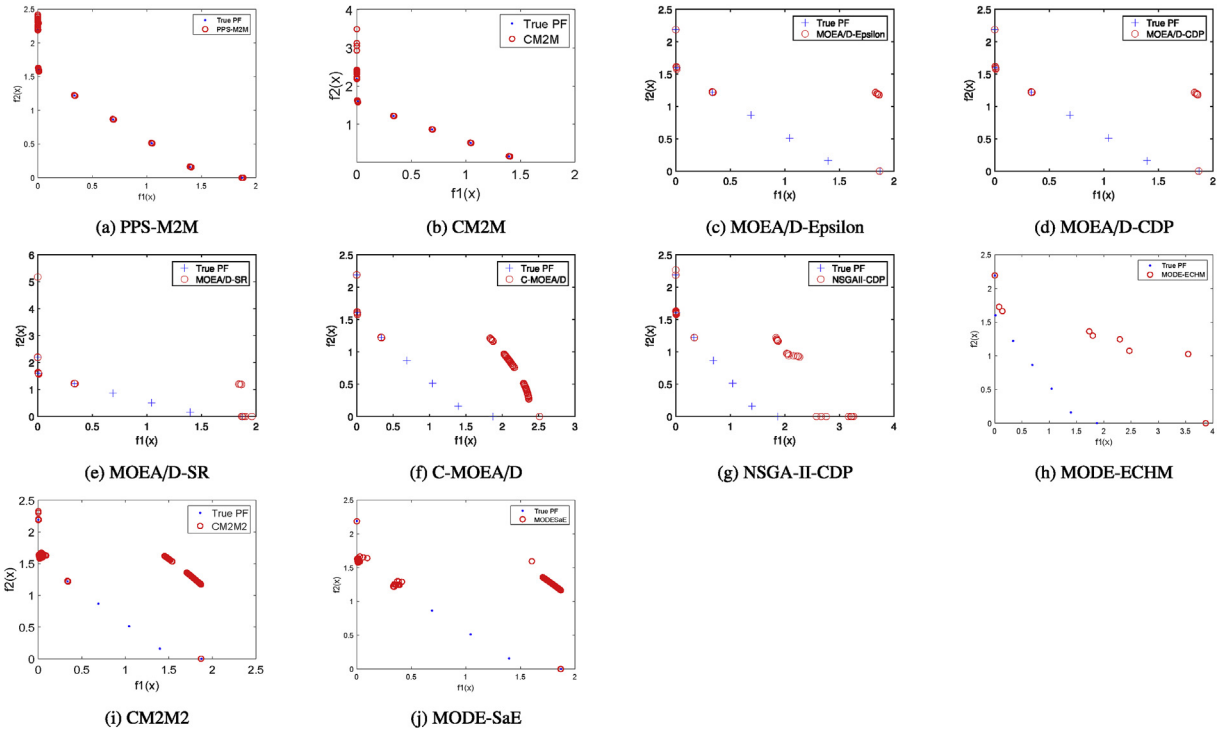


Fig. 6. The non-dominated solutions achieved by each algorithm on LIR-CMOP11 with the median HV values.

algorithms, we perform a series of post-hoc tests, with results shown Tables 9 and 11. Since each adjusted *p*-value in Tables 9 and 11 is less than the preset significant level 0.05, we can conclude that PPS-M2M is significantly better than the other three algorithms (PPS-MOEA/D, CM2M and CM2M2).

In order to illustrate the advantages of the proposed PPS-M2M in solving CIMOPs, we plot non-dominated solutions achieved by each algorithm on CIMOP2 and CIMOP4 with the median HV values, as two examples shown in Figs. 8 and 9. We can observe that PPS-M2M can find most Pareto optimal solutions. However, the other three algorithms can only find a small part of the true PFs. A possible reason is that CIMOP2 and CIMOP4 have diversity-hard characteristics in constraint functions. More specifically, the objective functions of CIMOP2 and CIMOP4 are imbalanced. As a result, and the decompo-

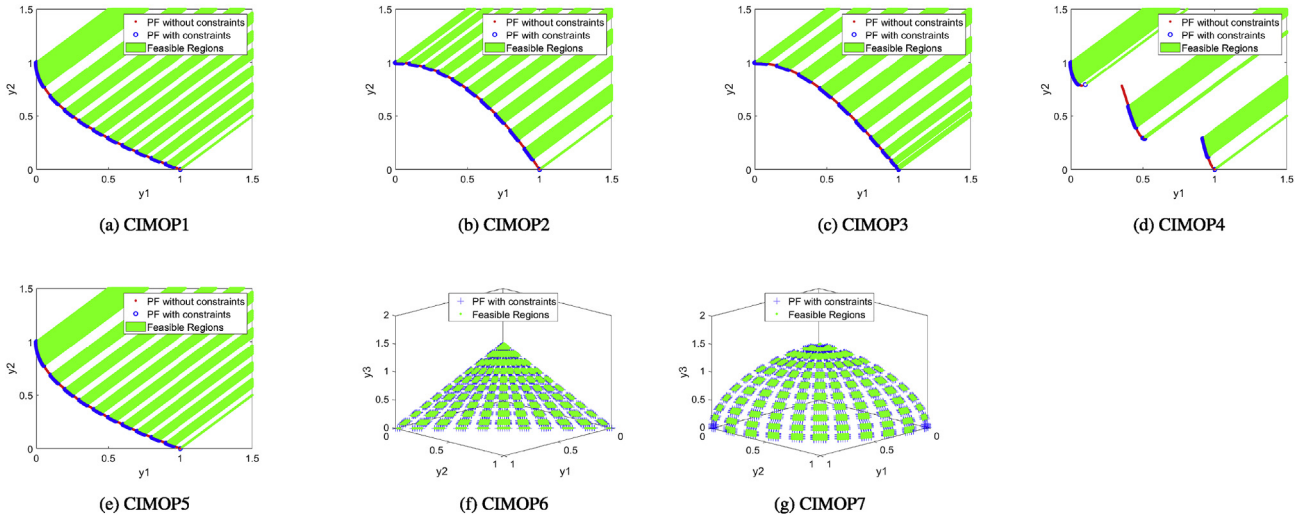


Fig. 7. The feasible regions, *PF*s without constraints and true *PF*s (*PF*s with constraints) of CIMOP1-7 are plotted.

Table 5
IGD and HV results of PPS-M2M and the other three CMOEAs on LIR-CMOP1-14. Friedman test at a 0.05 significance level is performed between PPS-M2M and PPS-MOEA/D [27]. The best mean among the compared algorithms on the test problem is highlighted in boldface.

Test Instances		IGD		HV	
		PPS-M2M	PPS-MOEA/D [27]	PPS-M2M	PPS-MOEA/D [27]
LIRCMOP1	mean	1.330E-02	**6.413E-03**	1.009E+00	**1.016E+00**
	std	4.407E-03	1.938E-03	4.590E-03	1.580E-03
LIRCMOP2	mean	1.485E-02	**4.673E-03**	1.337E+00	**1.349E+00**
	std	5.576E-03	7.844E-04	7.093E-03	1.010E-03
LIRCMOP3	mean	1.987E-02	**8.545E-03**	8.650E-01	**8.703E-01**
	std	7.371E-03	5.184E-03	6.010E-03	2.650E-03
LIRCMOP4	mean	2.504E-02	**4.677E-03**	**1.095E+00**	1.093E+00
	std	9.030E-03	1.116E-03	8.869E-03	2.467E-03
LIRCMOP5	mean	3.922E-03	**1.837E-03**	1.455E+00	**1.462E+00**
	std	1.515E-03	9.263E-05	6.935E-03	2.919E-04
LIRCMOP6	mean	5.077E-03	**2.490E-03**	1.118E+00	**1.129E+00**
	std	1.117E-02	3.399E-04	3.888E-02	1.771E-04
LIRCMOP7	mean	3.711E-03	**2.797E-03**	3.005E+00	**3.015E+00**
	std	1.993E-03	9.854E-05	2.434E-02	2.663E-03
LIRCMOP8	mean	2.949E-03	**2.778E-03**	3.016E+00	**3.017E+00**
	std	9.666E-05	7.558E-05	2.494E-03	1.139E-03
LIRCMOP9	mean	3.704E-01	**9.940E-02**	3.152E+00	**3.570E+00**
	std	2.529E-02	1.519E-01	8.752E-02	2.242E-01
LIRCMOP10	mean	1.572E-02	**2.108E-03**	3.216E+00	**3.241E+00**
	std	4.506E-02	7.754E-05	7.555E-02	3.077E-04
LIRCMOP11	mean	2.770E-02	**2.832E-03**	4.343E+00	**4.390E+00**
	std	4.656E-02	1.359E-03	1.401E-01	2.217E-04
LIRCMOP12	mean	9.310E-02	**2.704E-02**	5.409E+00	**5.614E+00**
	std	5.548E-02	5.002E-02	1.717E-01	1.525E-01
LIRCMOP13	mean	1.874E-01	**6.455E-02**	4.874E+00	**5.710E+00**
	std	2.690E-02	2.177E-03	1.068E-01	1.275E-02
LIRCMOP14	mean	1.746E-01	**6.419E-02**	5.449E+00	**6.193E+00**
	std	2.840E-02	1.690E-03	1.237E-01	1.310E-02
Friedman test		2	1	1.9286	1.0714

sition method of MOEA/D can only find a few parts of the PFs, as discussed in Ref. [26]. In addition, the convergence performance of CM2M and CM2M2 are worse than that of PPS-M2M, since many solutions obtained by CM2M and CM2M2 cannot converge to the true PFs, as shown in Fig. 8(c)–(d) and Fig. 9(c)–(d). One possible reason is that the constrained PFs of CIMOP2 and CIMOP4 are subsets of their unconstrained PFs. Because PPS-M2M first converges to unconstrained PFs without considering any constraints, it can very well deal with problems with this type of constraints using the push and pull mechanism. However, CM2M and CM2M2 adopt a different constraint-handling mechanism that considers the feasibility of the working population all the time, which is not so effective to treat the constraints of CIMOP2 and CIMOP4.

Based on the above observations and analysis, we conclude that PPS-M2M is a powerful algorithm that can very well treat CMOPs with either large infeasible regions or imbalanced objectives and diversity-hard constraints. In comparison with PPS-MOEA/D, another instantiation of PPS framework, PPS-M2M is more suitable for solving CMOPs with imbalanced objectives and diversity-hard constraint functions, while PPS-MOEA/D is more suitable to solve LIR-CMOPs with large infeasible regions.

Table 6
Adjusted p-values for the Friedman Aligned test in terms of mean metric (IGD and HV). PPS-MOEA/D is the control method.

performance indicator	algorithm	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	$p_{Holland}$	p_{Rom}	p_{Finner}	p_{Li}
IGD	PPS-M2M	0.008151	0.008151	0.008151	0.008151	0.008151	0.008151	0.008151	0.008151
HV	PPS-M2M	0.023342	0.023342	0.023342	0.023342	0.023342	0.023342	0.023342	0.023342

Table 7
The objective functions and constraint functions of CIMOP1-7.

Problem	Objectives	Constraints
CIMOP1	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))(1-\sqrt{x_1}) \\ \text{where} & g(\mathbf{x}) = 2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	$\begin{cases} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ a = 20, b = 0 \end{cases}$
CIMOP2	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))(1-x_1^2) \\ \text{where} & g(\mathbf{x}) = 10\sin(\pi x_1)\sum_{i=2}^{n}(\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	They are the same as those of CIMOP1
CIMOP3	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))\cos(\frac{\pi x_1}{2}) \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))\sin(\frac{\pi x_1}{2}) \\ \text{where} & g(\mathbf{x}) = 10\sin(\frac{\pi x_1}{2})\sum_{i=2}^{n}(\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	They are the same as those of CIMOP1
CIMOP4	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))(1-x_1^{0.5}\cos^2(2\pi x_1)) \\ \text{where} & g(\mathbf{x}) = 1+10\sin(\pi x_1)\sum_{i=2}^{n}(\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	They are the same as those of CIMOP1
CIMOP5	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))x_1 \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))(1-\sqrt{x_1}) \\ \text{where} & g(\mathbf{x}) = 2\lvert\cos(\pi x_1)\rvert\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert) \\ & t_i = x_i - \sin(0.5\pi x_1) \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	They are the same as those of CIMOP1
CIMOP6	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))x_1x_2 \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))x_1(1-x_2) \\ \min & f_3(\mathbf{x}) = (1+g(\mathbf{x}))(1-x_1) \\ \text{where} & g(\mathbf{x}) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert) \\ & t_i = x_i - x_1x_2 \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	$\begin{cases} c_1(\mathbf{x}) = \sin(a\pi x_1) - b \geq 0 \\ c_2(\mathbf{x}) = \cos(a\pi x_2) - b \geq 0 \\ a = 20, b = 0 \end{cases}$
CIMOP7	$\begin{cases} \min & f_1(\mathbf{x}) = (1+g(\mathbf{x}))\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \\ \min & f_2(\mathbf{x}) = (1+g(\mathbf{x}))\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2}) \\ \min & f_3(\mathbf{x}) = (1+g(\mathbf{x}))\sin(\frac{x_2\pi}{2}) \\ \text{where} & g(\mathbf{x}) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert) \\ & t_i = x_i - x_1x_2 \\ & n = 10, \mathbf{x} \in [0,1]^n \end{cases}$	They are the same as those of CIMOP6

Table 8
IGD results of PPS-M2M and the other three CMOEAs on CIMOP1-7. Friedman test at a 0.05 significance level is performed between PPS-M2M and each of the other three CMOEAs, namely PPS-MOEA/D [27], CM2M [28] and CM2M2 [32]. The best mean among the compared algorithms on the test problem is highlighted in boldface.

Test Instance		PPS-M2M	PPS-MOEA/D [27]	CM2M [28]	CM2M2 [32]
CIMOP1	mean	**2.05E-02**	2.81E-01	7.96E-01	5.22E-02
	std	1.13E-03	8.44E-02	6.36E-03	1.79E-02
CIMOP2	mean	**4.42E-02**	2.95E-01	6.38E-01	1.81E-01
	std	1.56E-02	1.29E-02	2.01E-02	3.18E-02
CIMOP3	mean	**1.19E-01**	3.51E-01	5.58E-01	2.80E-01
	std	1.13E-02	3.06E-02	1.54E-02	3.66E-02
CIMOP4	mean	**4.72E-03**	2.01E-01	6.47E-01	1.09E-02
	std	1.00E-03	2.92E-02	1.49E-02	3.44E-02
CIMOP5	mean	**4.45E-02**	3.76E-01	6.60E-02	8.26E-02
	std	5.28E-03	4.59E-02	1.25E-02	2.23E-02
CIMOP6	mean	**9.17E-02**	2.13E-01	6.98E-01	2.34E-01
	std	8.67E-03	4.46E-02	2.42E-02	2.46E-02
CIMOP7	mean	**1.64E-01**	3.65E-01	8.00E-01	3.67E-01
	std	1.87E-02	1.64E-02	5.11E-02	4.51E-02
Friedman test		1	2.8571	3.7143	2.4286

Table 9
Adjusted p-values for the Friedman Aligned test in terms of mean metric (IGD).

i	algorithm	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	$p_{Holland}$	p_{Rom}	p_{Finner}	p_{Li}
1	CM2M	0.000084	0.000251	0.000251	0.000251	0.000251	0.000251	0.000251	0.000087
2	PPS-MOEA/D	0.007118	0.014237	0.014237	0.014237	0.014186	0.014237	0.010659	0.007348
3	CM2M2	0.038434	0.038434	0.038434	0.038434	0.038434	0.038434	0.038434	0.038434

Table 10
HV results of PPS-M2M and the other three CMOEAs on CIMOP1-7. Friedman test at a 0.05 significance level is performed between PPS-M2M and each of the other three CMOEAs, namely PPS-MOEA/D [27], CM2M [28] and CM2M2 [32]. The best mean among the compared algorithms on the test problem is highlighted in boldface.

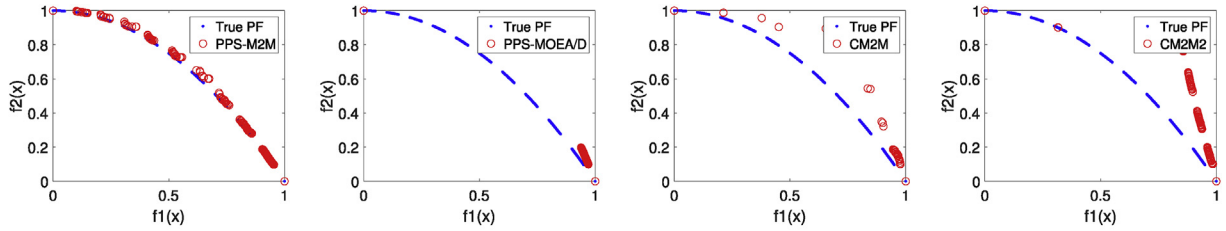
Test Instance		PPS-M2M	PPS-MOEA/D [27]	CM2M [28]	CM2M2 [32]
CIMOP1	mean	**1.065E+00**	6.623E-01	3.833E-01	1.032E+00
	std	1.359E-03	1.388E-01	5.250E-02	2.243E-02
CIMOP2	mean	**7.251E-01**	4.620E-01	7.109E-01	5.237E-01
	std	2.415E-02	1.478E-02	9.110E-03	2.600E-02
CIMOP3	mean	5.834E-01	2.470E-01	**6.630E-01**	3.384E-01
	std	2.507E-02	3.612E-02	2.260E-02	4.312E-01
CIMOP4	mean	**9.288E-01**	5.999E-01	7.856E-01	8.019E-01
	std	2.686E-03	1.747E-02	1.080E-03	7.203E-02
CIMOP5	mean	**1.028E+00**	6.550E-01	8.355E-01	9.617E-01
	std	7.116E-03	2.281E-02	6.810E-03	3.263E-02
CIMOP6	mean	**1.437E+00**	1.326E+00	8.748E-01	9.505E-01
	std	1.227E-02	5.988E-02	1.440E-02	4.994E-02
CIMOP7	mean	**1.041E+00**	9.348E-01	8.319E-01	7.827E-01
	std	3.658E-02	6.182E-03	1.850E-02	3.010E-02
Friedman test		1.1429	3.2857	2.8571	2.7143

Table 11
Adjusted p-values for the Friedman Aligned test in terms of mean metric (HV).

i	algorithm	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	$p_{Holland}$	p_{Rom}	p_{Finner}	p_{Li}
1	PPS-MOEA/D	0.001901	0.005703	0.005703	0.005703	0.005692	0.005703	0.005692	0.001941
2	CM2M	0.012983	0.025966	0.022773	0.022773	0.025797	0.022773	0.019411	0.013111
3	CM2M2	0.022773	0.025966	0.022773	0.022773	0.025797	0.022773	0.022773	0.022773

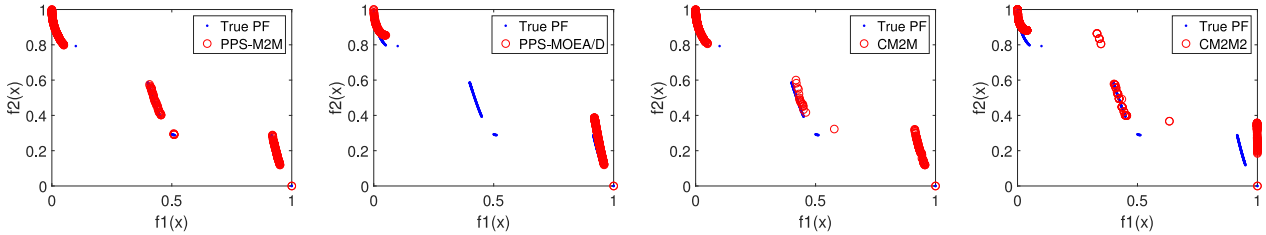
5. Conclusion

This paper proposes a new algorithm, namely PPS-M2M, which combines a multi-objective to multi-objective (M2M) decomposition approach with a push and pull search (PPS) framework to deal with CMOPs. To be more specific, the search process of PPS-M2M is divided into two stages—namely, push and pull search processes. At the push search stage, PPS-M2M uses the M2M decomposition method to decompose a CMOP into a set of simple CMOPs which correspond to a set of sub-populations. Each simple CMOP is solved in a collaborative



(a) PPS-M2M (b) PPS-MOEA/D (c) CM2M (d) CM2M2

Fig. 8. The non-dominated solutions achieved by each algorithm on CIMOP2 with the median HV values.



(a) PPS-M2M (b) PPS-MOEA/D (c) CM2M (d) CM2M2

Fig. 9. The non-dominated solutions achieved by each algorithm on CIMOP4 with the median HV values.

manner without considering any constraints, which can help the sub-populations effortlessly get across infeasible regions.

Furthermore, some constrained landscape information can be estimated during the push search stage, such as the ratio of feasible to infeasible solutions and the maximum overall constraint violation, which can be further employed to guide the parameter settings of constraint-handling mechanisms in the pull search stage. When the max rate of change between ideal and nadir points is less than or equal to a predefined threshold, the search process is switched to the pull search stage. At the beginning of the pull search stage, the infeasible solutions of each sub-population obtained in the push stage are pulled to the feasible and non-dominated regions by adopting the improved epsilon constraint-handling approach. At the last ten percentages of the maximum generation, all sub-populations are merged into a whole population, which is further evolved by employing the improved epsilon constraint-handling method. The comprehensive experimental results demonstrate that the proposed PPS-M2M outperforms the other nine CMOEAs (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE) on most of the LIR-CMOP1-14 significantly. To illustrate the differences between PPS-M2M and PPS-MOEA/D, we compare the performance of PPS-M2M and PPS-MOEA/D on LIR-CMOPs and CIMOPs. The experimental results show that the proposed PPS-M2M achieves significantly better results than PPS-MOEA/D on CIMOPs, while PPS-MOEA/D outperforms PPS-M2M on LIR-CMOPs, even though PPS-M2M can achieve significantly better results than the other nine algorithms (CM2M, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP, C-MOEA/D, NSGA-II-CDP, MODE-ECHM, CM2M2 and MODE-SaE) in comparison in this test suite. From the comprehensive experimental analysis, we can conclude that PPS-M2M is a powerful algorithm that can effectively deal with CMOPs with either large infeasible regions or with imbalanced objectives and diversity-hard constraints. In comparison with PPS-MOEA/D, PPS-M2M is more suitable for solving CMOPs with imbalanced objectives and diversity-hard constraint functions, while PPS-MOEA/D is more suitable to solve LIR-CMOPs with large infeasible regions.

There are many ways to improve the performance of PPS-M2M, including enhancing constraint-handling mechanisms in the pull search stage, and integrating machine learning methods to allocate computational resources dynamically into sub-populations of the PPS-M2M method, and so on. It is worth noting that because PPS is a very powerful framework for solving CMOPs, many new instantiations of the PPS framework can be generated and customized for solving CMOPs with different features, which is also one of our future research directions. In addition, real-world optimization problems will be investigated by using the proposed PPS-M2M.

CRediT authorship contribution statement

Zhun Fan: Conceptualization, Methodology, Writing - review & editing. **Zhaojun Wang:** Writing - original draft, Software. **Wenji Li:** Methodology, Writing - review & editing, Validation. **Yutong Yuan:** Data curation, Visualization. **Yugen You:** Data curation. **Zhi Yang:** Investigation. **Fuzan Sun:** Visualization. **Jie Ruan:** Software.

Acknowledgement

This work was supported by the Key Lab of Digital Signal and Image Processing of Guangdong Province, by the Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, by the Science and Technology Planning Project of Guangdong Province of China under grant 180917144960530 and 2019A050519008, by the Project of Educational Commission of Guangdong Province of China under grant 2017KZDXM032, by the State Key Lab of Digital Manufacturing Equipment & Technology under grant DMETKF2019020, and by the National Defense Technology Innovation Special Zone Projects.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.swevo.2020.100651.

References

[1] D. Kalyanmoy, et al., Multi Objective Optimization Using Evolutionary Algorithms, John Wiley and Sons, 2001.
[2] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[3] X. Cai, Z. Mei, Z. Fan, Q. Zhang, A constrained decomposition approach with grids for evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 22 (4) (2018) 564–577.

[4] X. Cai, Z. Mei, Z. Fan, A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors, IEEE Trans. Cybern. 48 (8) (2018) 2335–2348.

[5] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, E. Goodman, MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems, Appl. Soft Comput. 74 (2019) 621–633.

[6] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, E. Goodman, An improved epsilon constraint-handling method in MOEA/D for cmops with large infeasible regions, Soft Comput. 23 (2019) 1–20, https://doi.org/10.1007/s00500-019-03794-x.

[7] Y. Wang, J. Li, X. Xue, B. Wang, Utilizing the correlation between constraints and objective function for constrained evolutionary optimization, IEEE Trans. Evol. Comput. (2019) 1–15, https://doi.org/10.1109/TEVC.2019.2904900.

[8] Z. Liu, Y. Wang, Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces, IEEE Trans. Evol. Comput. (2019) 1–15, https://doi.org/10.1109/TEVC.2019.2894743.

[9] G. Wang, X. Cai, Z. Cui, G. Min, J. Chen, High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm, IEEE Trans. Emerg. Top. Comput. (2017) 1–12, https://doi.org/10.1109/TETC.2017.2703784.

[10] X. Cai, Y. Li, Z. Fan, Q. Zhang, An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization, IEEE Trans. Evol. Comput. 19 (4) (2015) 508–523.

[11] E. Mezura-Montes, C.A.C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm Evol. Comput. 1 (4) (2011) 173–194.

[12] K. Deb, R. Datta, A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.

[13] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 311–338.

[14] T.P. Runarsson, Xin Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (3) (2000) 284–294.

[15] Y. Wang, Z. Cai, G. Guo, Y. Zhou, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems, IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics) 37 (3) (2007) 560–575.

[16] B.Y. Qu, P.N. Suganthan, Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, Eng. Optim. 43 (4) (2011) 403–416.

[17] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (4) (1994) 242–253.

[18] J.A. Joines, C.R. Houck, On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's, in: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, IEEE, 1994, pp. 579–584.

[19] G. Wang, Y. Tan, Improving metaheuristic algorithms with information feedback models, IEEE Trans. Cybern. 49 (2) (2019) 542–555.

[20] Y.G. Woldesenbet, G.G. Yen, B.G. Tessema, Constraint handling in multiobjective evolutionary optimization, IEEE Trans. Evol. Comput. 13 (3) (2009) 514–525.

[21] E. Mezura-Montes, J. Velázquez-Reyes, C.C. Coello, Modified differential evolution for constrained optimization, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 25–32.

[22] E. Mezura-Montes, A.G. Palomeque-Ortiz, Parameter control in differential evolution for constrained optimization, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 2009, pp. 1375–1382.

[23] Z. Fan, J. Liu, T. Sorensen, P. Wang, Improved differential evolution based on stochastic ranking for robust layout synthesis of mems components, IEEE Trans. Ind. Electron. 56 (4) (2009) 937–948.

[24] G. Leguizamon, C.A. Coello Coello, A boundary search based ACO algorithm coupled with stochastic ranking, in: 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 165–172.

[25] T. Takahama, S. Sakai, Constrained optimization by the constrained differential evolution with an archive and gradient-based mutation, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–9.

[26] H. 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.

[27] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, Swarm Evol. Comput. 44 (2019) 665–679.

[28] H.-L. Liu, C. Peng, F. Gu, J. Wen, A constrained multi-objective evolutionary algorithm based on boundary search and archive, Int. J. Pattern Recogn. Artif. Intell. 30 (1) (2016) 1659002.

[29] Z. Yang, X. Cai, Z. Fan, Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results, in: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO Comp '14, 2014, pp. 1181–1186.

[30] M.A. Jan, R.A. Khanum, A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D, Appl. Soft Comput. 13 (1) (2013) 128–148.

[31] M. Asafuddoula, T. Ray, R. Sarker, K. Alam, An adaptive constraint handling approach embedded MOEA/D, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012, pp. 1–8.

[32] C. Peng, H.-L. Liu, F. Gu, An evolutionary algorithm with directed weights for constrained multi-objective optimization, Appl. Soft Comput. 60 (2017) 613–622.

[33] Y. Yang, J. Liu, S. Tan, H. Wang, A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio, Appl. Soft Comput. 80 (2019) 42–56.

[34] R.M. Rizk-Allah, R.A. El-Sehiemy, G.-G. Wang, A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution, Appl. Soft Comput. 63 (2018) 206–222.

[35] T. Takahama, S. Sakai, Constrained optimization by the constrained differential evolution with gradient-based mutation and feasible elites, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 1–8.

[36] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the Cec 2009 Special Session and Competition, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, 2008. special session on performance assessment of multi-objective optimization algorithms, technical report 264.

[37] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Difficulty Adjustable and Scalable Constrained Multi-Objective Test Problem Toolkit, Evolutionary Computation, 2019, pp. 1–28, https://doi.org/10.1162/evco_a_00259.

[38] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188, https://doi.org/10.1109/TEVC.2003.810761.

[39] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.

[40] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (2) (1979) 65–70.

[41] B.S. Holland, M.D. Copenhaver, An improved sequentially rejective Bonferroni test procedure, Biometrics (1987) 417–423.

[42] H. Finner, On a monotonicity problem in step-down multiple test procedures, Publ. Am. Stat. Assoc. 88 (423) (1993) 920–923.

[43] Y. Hochberg, A sharper bonferroni procedure for multiple tests of significance, Biometrika 75 (4) (1988) 800–802.

[44] G. Hommel, A stage wise rejective multiple test procedure based on a modified bonferroni test, Biometrika 75 (2) (1988) 383–386.

[45] D.M. Rom, A sequentially rejective test procedure based on a modified bonferroni inequality, Biometrika 77 (3) (1990) 663–665.

[46] J. Li, A two-step rejection procedure for testing multiple hypotheses, J. Stat. Plann. Inference 138 (6) (2008) 1521–1527.

[47] 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.