Swarm and Evolutionary Computation 49 (2019) 178–193

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization

Xinye Cai [a,b,c,*], Mi Hu [a,b], Dunwei Gong [d,**], Yi-nan Guo [d], Yong Zhang [d], Zhun Fan [e], Yuhua Huang [a,b]

[a] The College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210016, PR China
[b] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, Jiangsu, 210016, PR China
[c] Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, 300300, PR China
[d] The School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, 221116, PR China
[e] The Department of Electronic Engineering, School of Engineering, Shantou University, Guangdong, 515063, PR China

ARTICLE INFO

Keywords:
Combinatorial multiobjective optimization
Local search
Decomposition
Coevolution

ABSTRACT

Multiobjective evolutionary algorithm based on decomposition (MOEA/D) divides a multiobjective optimization problem into a number of single-objective subproblems and solves them in a collaborative way. MOEA/D can be naturally extended by the common, intensification oriented method of local search for solving combinatorial multiobjective optimization problems (CMOPs). However, the performance of MOEA/D strongly depends on the distribution of direction vectors and the decomposition method it adopts. In this paper, an efficient coevolutionary multiobjective local search based on decomposition (CoMOLS/D) is proposed. In CoMOLS/D, two sets of direction vectors and two populations with different decomposition methods are adopted to coevolve with each other. Among them, one population aims to achieve fast convergence while the other one puts more effort for maintaining the complementarily diverse solutions based on the convergence population. In the experimental studies, CoMOLS/D is compared with four decomposition-based local search heuristics, i.e., MOEA/D-LS (WS, TCH, PBI and iPBI); a dominance-based local search, i.e., ϵ-MOEA-LS; an indicator-based local search, i.e., IBEA-LS; and a state-of-the-art local search with dual populations, i.e., ND/DPP-LS; on two well-known CMOPs. The experimental results show that CoMOLS/D significantly outperforms the compared algorithms on most of the test instances.

1. Introduction

The combinatorial MOPs (CMOPs), the multiobjective traveling salesman problem (TSP) [1–3], multiobjective vehicle routing problem [4,5], multiobjective flowshop scheduling problem [6], multiobjective knapsack problem [7] and multiobjective software next release problem [8,9], have been taking significant interest. In the field of multiobjective optimization, the set of all the Pareto-optimal solutions is usually called the Pareto set (PS) and the image of (PS) on the objective vector space is called the Pareto front (PF). Most CMOPs are $\mathcal{NP}$-hard by nature, which means the exact methods are unable to find the PF of a CMOP within the polynomial time. Under such circumstances, meta/-heuristics, such as iterative local search [3], guided local search [10], tabu search [11], variable neighborhood search [12], ant colony optimization [13] and simulated annealing [14], are widely used to approximate the PF within reasonable execution time.

The decomposition methods have been extensively adopted for CMOPs [15–19]. They divide an MOP into a number of single-objective subproblems using aggregation functions. The derived subproblems can then be solved by the single-objective heuristics in a collaborative way. Multiobjective evolutionary algorithm based on decomposition (MOEA/D) [17] is a well-known example operating through decomposition. In MOEA/D, three decomposition approaches have been used, including Weighted Sum (WS), Tchebycheff (TCH) and Penalty-based Boundary Intersection (PBI) [20].

* Corresponding author. The College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210016, PR China.
** Corresponding author.
E-mail addresses: xinye@nuaa.edu.cn (X. Cai), dwgong@vip.163.com (D. Gong), zfan@stu.edu.cn (Z. Fan).

https://doi.org/10.1016/j.swevo.2019.05.007
Received 7 November 2018; Received in revised form 5 April 2019; Accepted 20 May 2019
Available online 3 June 2019

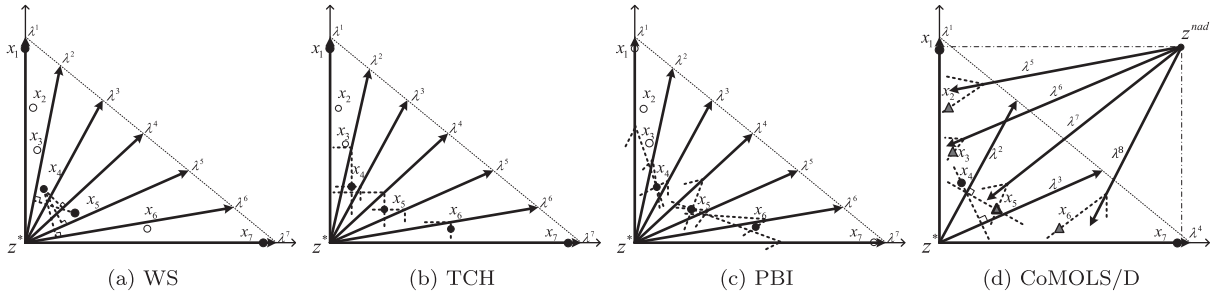


Fig. 1. The selected solutions using different decomposition approaches are given in the figure. In each subfigure, "x_i", denoted by a small circle, indicates one solution and "λ^j", denoted by an arrow, represents a direction vector. Dashed lines represent their contour lines. Solid circles ("•") represent the selected solutions by the first set of direction vectors (λ^1 - λ^4) and solid triangles ("△") represent the selected solution by the second set of direction vectors (λ^5 - λ^8). The unselected solutions are denoted by hollow circles ("∘").

Local search (LS) plays a key role in tackling a wide variety of $\mathcal{NP}$-hard combinatorial optimization problems. LS maintains one candidate solution and iteratively improves it by exploring its neighbors in the search space of solutions. During the optimization process, the current solution is replaced by its better neighbor.

Naturally, LS can be extended to CMOPs. One good example is Pareto LS (PLS) [21]. It works by exploring the neighbors of a non-dominated solution set and update the population by newly generated efficient solutions. Several variants of PLS have been proposed to tackle the different CMOPs [22,23]. A state-of-the-art multiobjective memetic algorithm based on decomposition (MOMAD) [15] is proposed by hybridizing MOEA/D-LS (WS) with PLS. However, both classical PLS and MOMAD store all the nondominated solutions for LS, which may bring unbearable time and space complexity.

However, MOEA/D may fail to obtain a well-distributed PF approximation due to the following two reasons [24]. First, MOEA/D (WS, TCH or PBI) tends to be very sensitive to the shapes of PFs. It can handle PFs with the concave PFs but incompetent to handle PFs with the convex PFs well. Second, in MOEA/D, the same solution is very likely to associate with multiple subproblems, which may lead to the loss of diversity [25]. Fig. 1a–c shows the selected solutions in the MOEA/D framework using WS, TCH or PBI. It can be observed that they are unable to maintain a well-distributed solution set. In addition, when facing the combinatorial MOPs, maintaining a set of diversely populated solutions may become even harder as the discrete search space may lead to very discrete PFs.

To address the aforementioned issues, an inverted PBI (iPBI) has been proposed to tackle the MOPs with extremely convex PFs in Ref. [26]. However, to achieve well-distributed solution set, the use of iPBI still needs to assume the convexity of PFs. Algorithms with dual populations can be adopted to alleviate such an issue. A state-of-the-art algorithm, called NP/DPP [27], was proposed with a dual-population paradigm for multiobjective optimization. Two coevolutionary populations using Pareto- and decomposition-based selection methods have been adopted for complementing each other. More recently, a MOEA/D variant, called MOEA/D-MR [28], with two sets of direction vectors has been proposed for MOPs with both convex and concave PFs. However, it maintains two populations with prefixed direction vectors. The adjustment of direction vectors in MOEA/D is also an alternative for achieving well-distributed solution set. For instance, an algorithm containing two types of adjustments for the direction vectors (MaOEA/D-2ADV) [29] is designed for MOPs with irregular PFs. A preference-inspired co-evolutionary algorithm using weight vectors (PICEA-w) [30] was proposed. It adopts the concept of coevolution between solutions and randomly generated direction vectors, which makes it less sensitive to the geometry of PFs. Another interesting work, called AdaW [31], adapts the weights during the evolutionary process for Pareto fronts with various shapes. Nevertheless, all the aforementioned algorithms are designed to handle the continuous MOPs. One very recent work [32] designed GWS-PLS algorithm based on grid which has better performance on combinatorial many-objective optimization. Nevertheless, very little effort has been made to design algorithms adapted to the geometry of PFs for CMOPs.

In this paper, a coevolutionary multiobjective local search based on decomposition (CoMOLS/D) is proposed to address CMOPs. Different from the previous work, CoMOLS/D coevolves two populations with two sets of direction vectors for balancing convergence and diversity. CoMOLS/D intends to deliver a well-distributed solution set, as shown in Fig. 1d.

The remainder of this paper is organized as follows. To make the paper self-contained, Section 2 gives some preliminary concepts on CMOPs and the background of coevolutionary algorithms. Section 3 details CoMOLS/D. Section 4 describes the test instances used to validate CoMOLS/D. Experimental studies and discussions are presented in Section 5. The sensitivity tests on the parameter θ is conducted; and a variant of CoMOLS/D is presented in this section as well. Section 6 concludes this paper and gives some future research directions.

2. Preliminaries

This section presents the background knowledge on CMOPs as well as four common decomposition methods. The related works on coevolutionary algorithms are also discussed in this section.

2.1. Basic definitions

A *multiobjective optimization problem* (MOP) can be formally stated as follows:

$$\text{minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \quad (1)$$

$$\text{subject to } \mathbf{x} \in \Omega$$

where Ω is the *decision space*, $\mathbf{F}: \Omega \to \mathbb{R}^m$ consists of m real-valued objective functions which conflict with each other. The *attainable objective set* is $\{\mathbf{F}(\boldsymbol{x}) \mid \mathbf{x} \in \Omega\}$. In the case when Ω is a finite set, (1) is called a combinatorial MOP (CMOP).

Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, $\mathbf{u}$ is said to *dominate* $\mathbf{v}$, denoted by $\mathbf{u} \prec \mathbf{v}$, if and only if $u_i \leq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$.[1] Given a set S in $\mathbb{R}^m$, a solution in it is called non-dominated in S if no other solution in S can dominate it. A solution $\mathbf{x}^* \in \Omega$ is *Pareto-optimal* if $\mathbf{F}(\mathbf{x}^*)$ is non-dominated in the attainable objective set. $\mathbf{F}(\mathbf{x}^*)$ is then called a *Pareto-optimal (objective) vector*. In other words, any improvement in one objective of a Pareto

[1] In the case of maximization, the inequality signs should be reversed.

Algorithm 1 Main framework.

Input:

- An MOP;
- A stoping criterion;
- N: The total population size.
- $W_1 : \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, ..., \boldsymbol{\lambda}^n\}$, the first direction vector set;

Output: PF approximation;

Initialize $P : \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n\}$.
Update the ideal point $\mathbf{z}^*$ based on Eq. (2) with P;
Set $Q := \varnothing$;
while *termination criterion is not fulfilled* **do**
  $P \leftarrow$ LocalSearch(P, W_1, $\mathbf{z}^*$, 'WS');
  Update $\mathbf{z}^*$ based on Eq. (2) and $\mathbf{z}^{nad}$ based on Eq. (3) along with $P \bigcup Q$;
  $W_2 \leftarrow$ CoDVs($\mathbf{z}^{nad}$, P, Q, N);
  $Q \leftarrow$ UpdatePops(W_2, $P \bigcup Q$, $\mathbf{z}^{nad}$, 'iPBI');
  $Q \leftarrow$ LocalSearch(Q, W_2, $\mathbf{z}^{nad}$, 'iPBI');
  $P \leftarrow$ UpdatePops(W_1, $P \bigcup Q$, $\mathbf{z}^*$, 'WS');
end
return *The nondominated solutions in* $P \bigcup Q$;

Algorithm 2 Local search (LocalSearch).

Input:

- TP: The population;
- $W : \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, ...\}$, the direction vector set;
- $\mathbf{z}$: The ideal or nadir point;
- $flag$: The flag for selecting one aggregation function;

Output: Updated population TP;

foreach $\boldsymbol{x} \in TP \wedge isSearched(\boldsymbol{x}) == False$ **do**
  $isSearched(\mathbf{x}) := True$;
  foreach $\boldsymbol{y} \in N(\boldsymbol{x})$ **do**
    // $N(\mathrm{x})$: neighborhood exploration
    for $i := 1$ *to* $|TP|$ **do**
      // y^i is the i-*th* individual and $\boldsymbol{\lambda}^i$ is the i-*th* direction vector in TP.
      if $g^*(\boldsymbol{y}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ *is better than* $g^*(\boldsymbol{y}^i|\boldsymbol{\lambda}^i, \boldsymbol{z})$ **then**
        /* The aggregation furcation g^* could be WS or $iPBI$ according to '$flag$'. */
        $\mathbf{y}^i := \mathbf{y}$;
        $isSearched(\mathbf{y}^i) := False$;
      end
    end
  end
end
return TP;

optimal solution must lead to deterioration to at least one another objective.

The *ideal* and *nadir point (objective vectors)* can be used to define the ranges of PFs. The ideal objective vector $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)^T$ can be calculated by

$$z_j^* = \min_{\mathbf{x}\in\Omega} f_j(\mathbf{x}),\ j \in \{1, \ldots, m\}. \tag{2}$$

The nadir objective vector $\mathbf{z}^{nad} = (z_1^{nad}, \ldots, z_m^{nad})^T$ can be calculated by

$$z_j^{nad} = \max_{\mathbf{x}\in PS} f_j(\mathbf{x}),\ j \in \{1, \ldots, m\}. \tag{3}$$

2.2. Decomposition approaches in MOEA/D

Four common decomposition methods [20], such as Weighted Sum, Tchebycheff, Penalty-based Boundary Intersection and inverted Penalty-based Boundary Intersection (iPBI), can be defined as follows.

Let $\lambda^i = (\lambda_1^i, \ldots, \lambda_m^i)^T$ be a direction vector for ith subproblem, where $\lambda_j^i \geq 0, j \in 1, \ldots, m$ and $\sum_{j=1}^m \lambda_j^i = 1$.

Algorithm 3 Coevolving direction vectors (CoDVs).

Input:

- $\mathbf{z}^{nad}$: The nadir point;
- The population P;
- The population Q;
- N: The population size;

Output: Direction vector set W_2;

$L := N - |P|$;
$\mathcal{C} \leftarrow$ UnionFind(P);/* Applying the union find algorithm to find the clusters $\mathcal{C}$ in P (assuming there are k clusters in $\mathcal{C}$). */
$\{\mathbf{u}^1, ..., \mathbf{u}^k\} \leftarrow centroid(\mathcal{C})$;
while $\binom{2}{k} \leq L$ **do**
  // If there are too few clusters.
  Randomly choose one solution $\mathbf{x}$ from P;
  $\{\mathbf{u}^1, ..., \mathbf{u}^{k+1}\} := \{\mathbf{u}^1, ..., \mathbf{u}^k\} \bigcup \{\mathbf{x}\}$;
  $k := k + 1$;
end
All point pairs got from set $\{\mathbf{u}^1, ..., \mathbf{u}^k\}$ and their midpoint coordinates and euclidean distances are stored in array $V : [\mathbf{v}^1, ..., \mathbf{v}^{\binom{2}{k}}]$ and array $D : [d_1, ..., d_{\binom{2}{k}}]$;
$[D,\ I] \leftarrow sort(D)$; // I stores the index of direction vectors after sorting D.
$VL := V[\ I[1 : L]\ ]$;
$W_2 := \varnothing$;
foreach $v \in VL$ **do**
  $\boldsymbol{\lambda} := \mathbf{z}^{nad} - \mathbf{v}$;
  $W_2 := W_2 \bigcup \{\boldsymbol{\lambda}\}$;
end
return W_2;

1. **Weighted Sum (WS) Approach:** The ith subproblem is defined as

$$\begin{aligned} &\text{minimize} \quad g^{ws}(\mathbf{x} \mid \lambda^i) = \sum_{j=1}^{m} \lambda_j^i f_j(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \Omega\,. \end{aligned} \tag{4}$$

2. **Tchebycheff (TCH) Approach:** The ith subproblem is defined as

$$\begin{aligned} &\text{minimize} \quad g^{tch}(\mathbf{x} \mid \lambda^i, \mathbf{z}^*) = \max_{1 \leq j \leq m} \{|f_j(\mathbf{x}) - z_j^*| \cdot \lambda_j^i\} \\ &\text{subject to} \quad \mathbf{x} \in \Omega\,, \end{aligned} \tag{5}$$

where Ω is the feasible region, but $\lambda_j^i = 0$ is replaced by $\lambda_j^i = 10^{-6}$ to avoid the numerical error on a digital computer in Eq. (5).

3. **Penalty-based Boundary Intersection (PBI) Approach:** This approach is a variant of Normal-Boundary Intersection approach [33]. The ith subproblem is defined as

$$\begin{aligned} &\text{minimize} \quad g^{pbi}(\mathbf{x} \mid \lambda^i, \mathbf{z}^*) = d_1^i + \theta d_2^i\,, \\ &\qquad\quad d_1^i = (\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T \cdot \lambda^i / \|\lambda^i\|\,, \\ &\qquad\quad d_2^i = \|\mathbf{F}(\mathbf{x}) - \mathbf{z}^* - (\lambda^i / \|\lambda^i\|) \cdot d_1^i\| \\ &\text{subject to} \quad \mathbf{x} \in \Omega\,, \end{aligned} \tag{6}$$

where $\|\bullet\|$ denotes L_2-norm and θ is the penalty parameter.

4. **Inverted Penalty-based Boundary Intersection (iPBI) Approach:** This approach is a variant of PBI [26]. The ith subproblem is defined as

$$\begin{aligned} &\text{maximize} \quad g^{ipbi}(\mathbf{x} \mid \lambda^i, \mathbf{z}^{nad}) = d_1^i - \theta d_2^i\,, \\ &\qquad\quad d_1^i = (\mathbf{z}^{nad} - \mathbf{F}(\mathbf{x}))^T \cdot \lambda^i / \|\lambda^i\|\,, \\ &\qquad\quad d_2^i = \|\mathbf{z}^{nad} - \mathbf{F}(\mathbf{x}) - (\lambda^i / \|\lambda^i\|) \cdot d_1^i\| \\ &\text{subject to} \quad \mathbf{x} \in \Omega\,. \end{aligned} \tag{7}$$

2.3. Coevolutionary algorithms

Coevolution is a reciprocal evolutionary exchange between species that have interaction with each other. As a natural extension of the traditional EAs, coevolutionary algorithms (CAs) arise from the biological observations that coevolving several species is more in line with the nature than evolving a group of individuals within a single species [34–36].

Over the past two decades, there has been an increasing amount of interest of adopting the coevolution techniques to address MOPs [37,38]. The typical CA involves the use of multiple groups of species (whose individuals are spatially or globally distributed) for a MOP. As species can either compete or cooperate during the search process, CAs can be classified into two types - competitive or cooperative. An example of competitive coevolutionary framework is the predator-pray model [39,40], in which a predator hunts a prey and the weakest prey is eliminated by the predator. On the other hand, in the framework of cooperative coevolution, individuals are rewarded when they have good performance working with other individuals and vice versa. Under this model, each subpopulation represents a piece of the original problem, and all these subpopulations coevolve for better performance gradually. In Ref. [41], each objective to be optimized has one different subpopulation; and in Ref. [30], a weight set and a solution set are coevolving with each other. These are typical examples for cooperative MOEAs based on decomposition in the objective space. In addition, some work [42,43] has also been conducted for the coevolution by the decomposition in the decision space, particularly for dealing with large-scale optimization problems. More recently, a bi-criterion evolution framework [44] has been proposed. It contains one population selected with Pareto criterion and the other selected with Non-Pareto criterion for coevolution.

Algorithm 4 Updating populations (UpdatePops).

Input:

- The direction vector set W : $\{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, ...\}$;
- The population TP;
- $\mathbf{z}$: The ideal or nadir point;
- $flag$: The indicator for the aggregate function;

Output: A new population TQ;

1 Associate each direction vector in W with one random solution from TP to form a new population TQ.
// It aims to assign the best solution to each direction vector.
2 **foreach** $\boldsymbol{x} \in TP$ **do**
3 **for** $i := 1$ ***to*** $|TQ|$ **do**
 // y^i and $\boldsymbol{\lambda}^i$ are the i-*th* individual and direction vector in TQ.
4 **if** $g^*(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$ *is better than* $g^*(\boldsymbol{y}^i|\boldsymbol{\lambda}^i, \boldsymbol{z})$ **then**
 // The aggregate approach in g^* could be ws or $ipbi$ according to '$flag$'.
5 $\mathbf{y}^i := \mathbf{x}$;
6 **end**
7 **end**
8 **end**
9 **return** *Population* TQ;

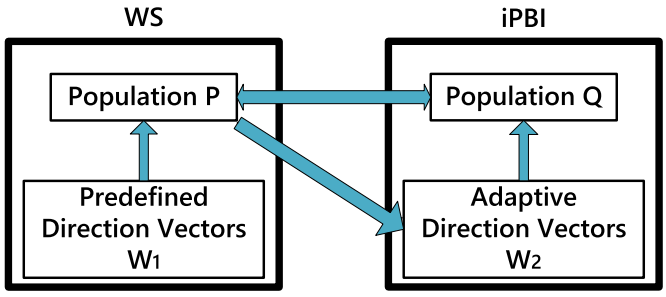


Fig. 2. The coevolutionary framework of two populations in CoMOLS/D.

3. The framework of CoMOLS/D

In this section, the pseudo code of our proposed algorithm, *coevolutionary multiobjective local search based on decomposition* (CoMOLS/D) is given in Algorithm 1. It maintains two populations: the first population P and the secondary population Q.

The first population P adopts WS with a set of uniformly predefined direction vectors W_1, while the secondary population Q adopts iPBI, whose direction vectors W_2 coevolve with P. In addition, two populations coevolve by updating each other. In CoMOLS/D, the goal of the first population P using WS with the predefined direction vectors aims to achieve fast convergence; while the secondary population Q using iPBI with the dynamic direction vectors aims for the complementarily diverse solutions based on the first population P. The coevolution of two populations in CoMOLS/D is illustrated in Fig. 2.

CoMOLS/D contains four major steps: 1) local search on P, 2) coevolution of direction vectors of Q with P, 3) local search on Q and 4) update population P and Q. Each step is detailed as follows.

3.1. Local search

The local search procedure in CoMOLS/D is presented in Algorithm 2. Both P and Q use the same local search heuristic, which is presented as follows. For each "unsearched" solution $\mathbf{x}$ (i.e., "isSearched" = = False) in P or Q, its neighborhood $N(\mathbf{x})$ is explored; then each solution $\mathbf{y} \in N(\mathbf{x})$ is used to update the population based on either WS or iPBI. Once a solution in the population is updated, it is marked as "un-searched"(i.e., "isSearched" = = False). After all the individuals are explored (i.e., "isSearched" = = True), the local search procedure is terminated.

3.2. Coevolving the direction vectors

The primary motivation of the Algorithm 3 is to insert new direction vectors W_2 of the secondary population Q in the sparse region based on the distribution of the first population P.

As the size of $P \bigcup Q$ is N, the number of the second direction vector set W_2 is $L = N - |P|$. Each solution in P is associated with its nearest neighbor based on Euclidean distance. All the individuals are divided into a set of clusters (denoted by $\boldsymbol{C}$), using clustering methods, such as union-find algorithm [45] (line 2). After that, the centroids of all the clusters, denoted by $\{\mathbf{u1}, \ldots, \mathbf{u}^k\}$, could be computed (line 3). If the number of clusters is insufficient (i.e., $\binom{k}{2} \leq L$), which indicates that the number of all the direction vectors inserted between every two centroids is less then L, one individual randomly selected from P is considered the centroid of a new cluster. This process is repeated until the number of clusters becomes sufficient (lines 4–8).

Note that the indexes of all the centroid pairs are denoted by

$$pair\_index = \{(1,2), \ldots, (i,j), \ldots, (k-1,k)\}. \tag{8}$$

The midpoints of all the centroid pairs are denoted by V and the corresponding Euclidean distances between every two of them are denoted by D (line 9). Then, L midpoint in V that has the least distance in D is selected. Each of them and the Nadir point will form L direction vectors (i.e.,W_2) (lines 10–16). For each direction vector in W_2, the solution in $P \bigcup Q$ that has the best aggregation function value, is associated with it, based on Algorithm 4. All the solutions associated with W_2 form a new population Q for local search.

3.3. Updating populations

Algorithm 4 presents the updating of population P with its direction vectors W_1 or Q with its direction vectors W_2. It aims to choose the best solution for each direction vector according to its aggregate

Table 1
Mean and standard deviation of normalized HV obtained by CoMOLS/D, MOEA/D-LS (WS, TCH, PBI and iPBI), ϵ-MOEA-LS, IBEA-LS and ND/DPP-LS.

Instance	CoMOLS/D	MOEA/D-LS [17]				ϵ-MOEA-LS [57]	IBEA-LS [58]	ND/DPP-LS [27]
		WS	TCH	PBI	iPBI			
ClusterAB100	9.49E-1 (1.4E-3)	9.41E-1 (1.8E-3)$^{-}$	9.47E-1 (1.7E-3)$^{\approx}$	9.25E-1 (3.1E-3)$^{-}$	9.41E-1 (4.3E-3)$^{-}$	9.14E-1 (1.1E-2)$^{-}$	8.58E-1 (1.7E-2)$^{-}$	9.46E-1 (1.4E-3)$^{\approx}$
ClusterAB300	9.75E-1 (1.9E-3)	9.64E-1 (2.1E-3)$^{-}$	9.58E-1 (2.5E-3)$^{-}$	9.41E-1 (3.0E-3)$^{-}$	9.58E-1 (2.1E-3)$^{-}$	8.00E-1 (1.7E-2)$^{-}$	6.92E-1 (3.3E-1)$^{-}$	9.63E-1 (1.7E-3)$^{-}$
euclidAB100	7.97E-1 (1.5E-3)	7.76E-1 (3.3E-3)$^{-}$	7.94E-1 (1.7E-3)$^{-}$	7.86E-1 (3.1E-3)$^{-}$	7.90E-1 (2.8E-3)$^{-}$	7.66E-1 (8.1E-3)$^{-}$	6.94E-1 (8.3E-3)$^{-}$	7.95E-1 (2.1E-3)$^{\approx}$
euclidAB300	9.13E-1 (1.6E-3)	9.01E-1 (2.0E-3)$^{-}$	8.95E-1 (1.3E-3)$^{-}$	8.85E-1 (2.4E-3)$^{-}$	8.93E-1 (2.6E-3)$^{-}$	6.94E-1 (2.7E-2)$^{-}$	7.81E-1 (9.9E-3)$^{-}$	9.03E-1 (1.5E-3)$^{-}$
kroAB100	8.70E-1 (2.3E-3)	8.57E-1 (2.8E-3)$^{-}$	8.68E-1 (1.7E-3)$^{-}$	8.53E-1 (3.4E-3)$^{-}$	8.63E-1 (4.1E-3)$^{-}$	8.29E-1 (1.1E-2)$^{-}$	7.69E-1 (1.0E-2)$^{-}$	8.69E-1 (2.7E-3)$^{\approx}$
kroAB200	9.74E-1 (1.7E-3)	9.65E-1 (2.1E-3)$^{-}$	9.60E-1 (1.9E-3)$^{-}$	9.52E-1 (4.0E-3)$^{-}$	9.61E-1 (3.8E-3)$^{-}$	8.68E-1 (1.8E-2)$^{-}$	7.90E-1 (2.1E-1)$^{-}$	9.68E-1 (2.1E-3)$^{-}$
kroAB300	9.48E-1 (1.6E-3)	9.39E-1 (1.7E-3)$^{-}$	9.30E-1 (1.2E-3)$^{-}$	9.17E-1 (1.6E-3)$^{-}$	9.28E-1 (2.4E-3)$^{-}$	7.52E-1 (2.5E-2)$^{-}$	6.75E-1 (3.2E-1)$^{-}$	9.36E-1 (1.6E-3)$^{-}$
euclidABC100	6.68E-1 (1.8E-3)	6.64E-1 (1.3E-3)$^{-}$	6.50E-1 (1.6E-3)$^{-}$	6.15E-1 (3.0E-3)$^{-}$	6.32E-1 (2.8E-3)$^{-}$	4.14E-1 (1.9E-2)$^{-}$	4.73E-1 (5.4E-3)$^{-}$	6.60E-1 (1.5E-3)$^{-}$
kroABC100	7.67E-1 (1.4E-3)	7.65E-1 (1.5E-3)$^{-}$	7.47E-1 (1.4E-3)$^{-}$	7.06E-1 (3.7E-3)$^{-}$	7.34E-1 (2.6E-3)$^{-}$	4.82E-1 (1.4E-2)$^{-}$	5.53E-1 (9.2E-3)$^{-}$	7.44E-1 (2.3E-3)$^{-}$
mQAP_50_2(-0.3)	5.71E-1 (7.0E-3)	5.59E-1 (5.6E-3)$^{-}$	5.70E-1 (6.6E-3)$^{\approx}$	5.54E-1 (6.2E-3)$^{-}$	5.70E-1 (6.9E-3)$^{\approx}$	5.67E-1 (4.8E-3)$^{-}$	5.50E-1 (7.2E-3)$^{-}$	5.70E-1 (6.4E-3)$^{\approx}$
mQAP_50_2(0.0)	5.43E-1 (5.4E-3)	5.32E-1 (6.2E-3)$^{-}$	5.45E-1 (6.6E-3)$^{\approx}$	5.28E-1 (8.0E-3)$^{-}$	5.42E-1 (7.1E-3)$^{\approx}$	5.38E-1 (4.5E-3)$^{-}$	5.27E-1 (5.1E-3)$^{-}$	5.43E-1 (6.4E-3)$^{\approx}$
mQAP_50_2(0.3)	5.62E-1 (8.8E-3)	5.52E-1 (7.6E-3)$^{-}$	5.65E-1 (7.2E-3)$^{\approx}$	5.51E-1 (9.1E-3)$^{-}$	5.64E-1 (6.1E-3)$^{\approx}$	5.64E-1 (6.9E-3)$^{\approx}$	5.38E-1 (9.4E-3)$^{-}$	5.66E-1 (6.3E-3)$^{\approx}$
mQAP_60_2(-0.3)	5.37E-1 (6.3E-3)	5.29E-1 (5.4E-3)$^{-}$	5.37E-1 (5.8E-3)$^{\approx}$	5.23E-1 (7.6E-3)$^{-}$	5.39E-1 (6.4E-3)$^{\approx}$	5.33E-1 (5.0E-3)$^{\approx}$	5.21E-1 (4.0E-3)$^{-}$	5.40E-1 (5.7E-3)$^{\approx}$
mQAP_60_2(0.0)	5.10E-1 (5.9E-3)	5.00E-1 (5.5E-3)$^{-}$	5.09E-1 (5.5E-3)$^{\approx}$	4.98E-1 (7.8E-3)$^{-}$	5.07E-1 (5.7E-3)$^{\approx}$	5.01E-1 (4.4E-3)$^{-}$	4.96E-1 (7.0E-3)$^{-}$	5.10E-1 (5.8E-3)$^{\approx}$
mQAP_60_2(0.3)	5.41E-1 (5.0E-3)	5.35E-1 (3.9E-3)$^{-}$	5.40E-1 (5.1E-3)$^{\approx}$	5.25E-1 (5.8E-3)$^{-}$	5.42E-1 (5.5E-3)$^{\approx}$	5.36E-1 (3.9E-3)$^{-}$	5.26E-1 (4.9E-3)$^{-}$	5.39E-1 (4.5E-3)$^{\approx}$
mQAP_30_3(-0.3)	4.19E-1 (5.7E-3)	4.06E-1 (6.4E-3)$^{-}$	4.14E-1 (6.8E-3)$^{-}$	4.09E-1 (8.5E-3)$^{-}$	4.12E-1 (7.1E-3)$^{-}$	3.56E-1 (9.0E-3)$^{-}$	3.70E-1 (1.4E-2)$^{-}$	4.10E-1 (5.9E-3)$^{-}$
mQAP_30_3(0.0)	2.95E-1 (3.6E-3)	2.84E-1 (5.0E-3)$^{-}$	2.92E-1 (4.3E-3)$^{-}$	2.85E-1 (4.8E-3)$^{-}$	2.92E-1 (4.8E-3)$^{-}$	2.53E-1 (6.2E-3)$^{-}$	2.54E-1 (5.8E-3)$^{-}$	2.90E-1 (3.3E-3)$^{-}$
mQAP_30_3(0.3)	5.90E-1 (9.0E-3)	5.82E-1 (8.1E-3)$^{-}$	5.89E-1 (9.5E-3)$^{\approx}$	5.76E-1 (9.8E-3)$^{-}$	5.99E-1 (8.8E-3)$^{+}$	5.44E-1 (1.1E-2)$^{-}$	5.43E-1 (1.3E-2)$^{-}$	5.91E-1 (8.9E-3)$^{\approx}$
mQAP_40_3(-0.3)	2.78E-1 (4.6E-3)	2.68E-1 (3.7E-3)$^{-}$	2.72E-1 (5.1E-3)$^{-}$	2.67E-1 (4.9E-3)$^{-}$	2.69E-1 (4.4E-3)$^{-}$	2.04E-1 (8.6E-3)$^{-}$	2.40E-1 (5.3E-3)$^{-}$	2.80E-1 (3.2E-3)$^{\approx}$
mQAP_40_3(0.0)	2.86E-1 (4.8E-3)	2.79E-1 (5.2E-3)$^{-}$	2.83E-1 (4.8E-3)$^{-}$	2.76E-1 (4.1E-3)$^{-}$	2.78E-1 (4.2E-3)$^{-}$	2.16E-1 (9.8E-3)$^{-}$	2.43E-1 (6.0E-3)$^{-}$	2.82E-1 (5.2E-3)$^{-}$
mQAP_40_3(0.3)	2.91E-1 (4.0E-3)	2.84E-1 (6.2E-3)$^{-}$	2.87E-1 (4.7E-3)$^{-}$	2.81E-1 (6.0E-3)$^{-}$	2.79E-1 (5.9E-3)$^{-}$	2.16E-1 (6.8E-3)$^{-}$	2.50E-1 (9.4E-3)$^{-}$	2.84E-1 (3.6E-3)$^{-}$

The best HV value is highlighted in gray.
'+', '−' and '≈' indicate that the result is significantly better, significantly worse and statistically similar to that of CoMOLS/D on this test instance, respectively (wilcoxon's rank sum test at the 0.05 significance level).

Table 2
Mean and standard deviation of IGD obtained by CoMOLS/D, MOEA/D-LS (WS, TCH, PBI and iPBI), ϵ-MOEA-LS, IBEA-LS and ND/DPP-LS.

Instance	CoMOLS/D	MOEA/D-LS [17]				ϵ-MOEA-LS [57]	IBEA-LS [58]	ND/DPP-LS [27]
		WS	TCH	PBI	iPBI			
ClusterAB100	8.3E+2 (1.2E+2)	3.7E+3 (4.1E+2)$^-$	9.3E+2 (1.9E+2)$^\approx$	5.9E+3 (8.8E+2)$^-$	2.4E+3 (8.3E+2)$^-$	5.7E+3 (1.4E+3)$^-$	1.5E+4 (2.1E+3)$^-$	2.4E+3 (3.2E+2)$^-$
ClusterAB300	3.4E+3 (4.6E+2)	1.0E+4 (8.3E+2)$^-$	7.1E+3 (7.0E+2)$^-$	2.6E+4 (1.7E+3)$^-$	1.6E+4 (1.4E+3)$^-$	6.6E+4 (5.0E+3)$^-$	1.3E+5 (1.6E+5)$^-$	9.3E+3 (8.4E+2)$^-$
euclidAB100	8.0E+2 (1.3E+2)	4.0E+3 (5.2E+2)$^-$	9.2E+2 (1.3E+2)$^-$	2.6E+3 (3.9E+2)$^-$	2.0E+3 (4.7E+2)$^-$	4.6E+3 (1.3E+3)$^-$	1.4E+4 (1.7E+3)$^-$	2.5E+3 (3.4E+2)$^-$
euclidAB300	3.3E+3 (4.4E+2)	9.6E+3 (9.9E+2)$^-$	7.3E+3 (5.1E+2)$^-$	1.9E+4 (1.2E+3)$^-$	1.6E+4 (1.5E+3)$^-$	7.5E+4 (6.7E+3)$^-$	5.9E+4 (4.2E+3)$^-$	9.2E+3 (8.0E+2)$^-$
kroAB100	9.1E+2 (2.1E+2)	3.5E+3 (4.3E+2)$^-$	9.6E+2 (1.5E+2)$^-$	4.3E+3 (6.2E+2)$^-$	2.4E+3 (8.2E+2)$^-$	6.5E+3 (1.9E+3)$^-$	1.4E+4 (1.5E+3)$^-$	2.3E+3 (3.1E+2)$^-$
kroAB200	2.2E+3 (3.2E+2)	7.0E+3 (8.4E+2)$^-$	4.5E+3 (3.9E+2)$^-$	1.2E+4 (1.7E+3)$^-$	8.6E+3 (2.4E+3)$^-$	3.1E+4 (5.9E+3)$^-$	6.0E+4 (6.6E+4)$^-$	6.2E+3 (6.6E+2)$^-$
kroAB300	2.8E+3 (3.9E+2)	8.7E+3 (1.1E+3)$^-$	6.6E+3 (4.0E+2)$^-$	2.1E+4 (1.1E+3)$^-$	1.6E+4 (1.5E+3)$^-$	7.2E+4 (9.7E+3)$^-$	1.3E+5 (1.5E+5)$^-$	8.3E+3 (7.7E+2)$^-$
euclidABC100	5.9E+3 (2.4E+2)	7.3E+3 (1.7E+2)$^-$	8.1E+3 (1.7E+2)$^-$	1.2E+4 (3.8E+2)$^-$	8.8E+3 (3.1E+2)$^-$	4.3E+4 (3.1E+3)$^-$	3.4E+4 (1.4E+3)$^-$	8.5E+3 (1.9E+2)$^-$
kroABC100	6.3E+3 (2.3E+2)	7.5E+3 (2.3E+2)$^-$	8.3E+3 (2.2E+2)$^-$	1.4E+4 (4.3E+2)$^-$	9.1E+3 (2.7E+2)$^-$	4.5E+4 (2.4E+3)$^-$	3.7E+4 (2.4E+3)$^-$	8.7E+3 (2.5E+2)$^-$
mQAP_50_2(-0.3)	1.4E+7 (2.7E+5)	2.7E+7 (1.1E+6)$^-$	1.4E+7 (1.5E+5)$^\approx$	2.2E+7 (2.3E+5)$^-$	1.5E+7 (1.6E+5)$^\approx$	1.9E+7 (2.9E+5)$^-$	2.2E+7 (2.0E+5)$^-$	1.5E+7 (6.2E+5)$^\approx$
mQAP_50_2(0.0)	1.8E+7 (3.6E+5)	3.3E+7 (8.3E+5)$^-$	1.6E+7 (2.5E+5)$^+$	2.3E+7 (3.4E+5)$^-$	1.7E+7 (2.8E+5)$^\approx$	2.3E+7 (2.3E+5)$^-$	2.6E+7 (3.1E+5)$^-$	1.8E+7 (5.7E+5)$^\approx$
mQAP_50_2(0.3)	1.8E+7 (4.4E+5)	3.0E+7 (1.3E+6)$^-$	1.6E+7 (3.0E+5)$^\approx$	2.0E+7 (3.0E+5)$^-$	1.5E+7 (2.8E+5)$^+$	1.8E+7 (3.7E+5)$^-$	3.0E+7 (4.7E+5)$^-$	1.8E+7 (8.3E+5)$^\approx$
mQAP_60_2(-0.3)	2.0E+7 (3.4E+5)	3.7E+7 (1.2E+6)$^-$	1.8E+7 (2.5E+5)$^+$	2.9E+7 (4.1E+5)$^-$	2.0E+7 (2.3E+5)$^\approx$	2.6E+7 (3.1E+5)$^-$	3.1E+7 (2.5E+5)$^-$	2.1E+7 (7.4E+5)$^\approx$
mQAP_60_2(0.0)	2.2E+7 (4.1E+5)	4.3E+7 (1.8E+6)$^-$	2.1E+7 (3.1E+5)$^\approx$	3.0E+7 (3.4E+5)$^-$	2.2E+7 (3.7E+5)$^\approx$	3.3E+7 (3.8E+5)$^-$	3.1E+7 (3.6E+5)$^-$	2.4E+7 (1.1E+6)$^-$
mQAP_60_2(0.3)	2.0E+7 (4.0E+5)	4.0E+7 (1.0E+6)$^-$	2.0E+7 (3.5E+5)$^\approx$	3.6E+7 (3.9E+5)$^-$	2.2E+7 (2.5E+5)$^-$	2.8E+7 (3.2E+5)$^-$	3.2E+7 (4.7E+5)$^-$	2.2E+7 (7.1E+5)$^-$
mQAP_30_3(-0.3)	9.0E+6 (4.6E+4)	1.5E+7 (1.5E+5)$^-$	9.3E+6 (7.3E+4)$^\approx$	1.1E+7 (8.7E+4)$^-$	1.1E+7 (2.4E+5)$^-$	2.0E+7 (1.9E+5)$^-$	1.9E+7 (8.8E+4)$^-$	1.0E+7 (1.2E+5)$^-$
mQAP_30_3(0.0)	8.4E+6 (3.8E+4)	1.3E+7 (1.1E+5)$^-$	8.7E+6 (5.5E+4)$^-$	1.1E+7 (5.6E+4)$^-$	1.1E+7 (1.8E+5)$^-$	1.7E+7 (1.3E+5)$^-$	2.0E+7 (1.1E+5)$^-$	9.4E+6 (8.6E+4)$^\approx$
mQAP_30_3(0.3)	8.3E+6 (1.1E+5)	1.3E+7 (2.2E+5)$^-$	8.1E+6 (1.0E+5)$^\approx$	9.7E+6 (1.4E+5)$^-$	1.1E+7 (2.3E+5)$^-$	1.5E+7 (2.5E+5)$^-$	1.5E+7 (7.6E+4)$^-$	9.0E+6 (1.7E+5)$^-$
mQAP_40_3(-0.3)	1.5E+7 (7.6E+4)	2.4E+7 (1.6E+5)$^-$	1.6E+7 (1.0E+5)$^\approx$	1.9E+7 (1.1E+5)$^-$	2.1E+7 (2.5E+5)$^-$	4.2E+7 (3.8E+5)$^-$	3.3E+7 (8.8E+4)$^-$	1.6E+7 (1.4E+5)$^\approx$
mQAP_40_3(0.0)	1.7E+7 (1.3E+5)	2.6E+7 (2.1E+5)$^-$	1.7E+7 (1.6E+5)$^\approx$	2.2E+7 (1.6E+5)$^-$	2.6E+7 (2.2E+5)$^-$	4.9E+7 (2.8E+5)$^-$	4.0E+7 (9.5E+4)$^-$	1.8E+7 (2.0E+5)$^\approx$
mQAP_40_3(0.3)	1.7E+7 (1.0E+5)	2.4E+7 (2.0E+5)$^-$	1.7E+7 (1.0E+5)$^\approx$	2.2E+7 (1.5E+5)$^-$	2.5E+7 (3.7E+5)$^-$	4.7E+7 (3.2E+5)$^-$	4.0E+7 (1.3E+5)$^-$	1.8E+7 (1.6E+5)$^\approx$

The best IGD value is highlighted in gray.
'+', '−' and '≈' indicate that the result is significantly better, significantly worse and statistically similar to that of CoMOLS/D on this test instance, respectively (wilcoxon's rank sum test at the 0.05 significance level).

Table 3
Mean and standard deviation of DIR obtained by CoMOLS/D, MOEA/D-LS (WS, TCH, PBI and iPBI), ϵ-MOEA-LS, IBEA-LS and ND/DPP-LS.

Instance	CoMOLS/D	MOEA/D-LS [17]				ϵ-MOEA-LS [57]	IBEA-LS [58]	ND/DPP-LS [27]
		WS	TCH	PBI	iPBI			
ClusterAB100	2.25E-1 (7.2E-4)	2.46E-1 (1.2E-2)$^{-}$	2.26E-1 (5.7E-5)$^{-}$	2.96E-1 (2.9E-2)$^{-}$	2.43E-1 (1.3E-2)$^{-}$	3.02E-1 (7.1E-2)$^{-}$	4.69E-1 (9.1E-2)$^{-}$	2.59E-1 (6.2E-3)$^{-}$
ClusterAB300	2.26E-1 (7.5E-5)	2.53E-1 (1.2E-2)$^{-}$	2.48E-1 (1.0E-2)$^{-}$	2.95E-1 (2.1E-5)$^{-}$	2.52E-1 (2.7E-5)$^{-}$	5.07E-1 (1.5E-1)$^{-}$	5.78E-1 (1.2E-1)$^{-}$	2.75E-1 (1.2E-2)$^{-}$
euclidAB100	2.25E-1 (1.4E-4)	2.52E-1 (1.9E-2)$^{-}$	2.26E-1 (1.0E-4)$^{\approx}$	2.52E-1 (1.7E-5)$^{-}$	2.41E-1 (1.3E-2)$^{-}$	2.84E-1 (5.8E-2)$^{-}$	4.08E-1 (4.8E-2)$^{-}$	2.26E-1 (9.6E-3)$^{\approx}$
euclidAB300	2.26E-1 (6.7E-5)	2.48E-1 (1.4E-2)$^{-}$	2.52E-1 (3.3E-5)$^{-}$	2.79E-1 (2.1E-2)$^{-}$	2.54E-1 (1.5E-2)$^{-}$	4.72E-1 (1.5E-1)$^{-}$	4.85E-1 (4.7E-2)$^{-}$	2.76E-1 (7.0E-3)$^{\approx}$
kroAB100	2.26E-1 (4.1E-4)	2.53E-1 (2.8E-2)$^{-}$	2.26E-1 (5.8E-5)$^{\approx}$	2.56E-1 (1.3E-2)$^{-}$	2.41E-1 (1.3E-2)$^{\approx}$	3.04E-1 (6.1E-2)$^{-}$	4.13E-1 (5.7E-2)$^{-}$	2.62E-1 (1.4E-2)$^{-}$
kroAB200	2.26E-1 (6.1E-4)	2.61E-1 (2.7E-2)$^{-}$	2.43E-1 (1.3E-2)$^{\approx}$	2.60E-1 (1.7E-2)$^{-}$	2.45E-1 (1.2E-2)$^{-}$	4.47E-1 (1.3E-1)$^{-}$	4.77E-1 (1.1E-1)$^{-}$	2.76E-1 (2.1E-2)$^{-}$
kroAB300	2.26E-1 (7.8E-5)	2.48E-1 (1.6E-2)$^{-}$	2.52E-1 (1.9E-5)$^{-}$	2.83E-1 (2.0E-2)$^{-}$	2.56E-1 (1.3E-2)$^{-}$	4.42E-1 (1.4E-1)$^{-}$	5.61E-1 (1.2E-1)$^{-}$	2.75E-1 (7.9E-3)$^{-}$
euclidABC100	1.05E-1 (6.8E-3)	1.07E-1 (5.0E-3)$^{\approx}$	7.81E-2 (4.8E-3)$^{+}$	1.08E-1 (7.4E-3)$^{\approx}$	1.03E-1 (3.8E-3)$^{\approx}$	2.80E-1 (3.7E-2)$^{-}$	2.70E-1 (1.3E-2)$^{-}$	1.01E-1 (5.4E-3)$^{\approx}$
kroABC100	8.83E-2 (6.7E-3)	9.20E-2 (6.4E-3)$^{-}$	6.84E-2 (4.2E-3)$^{\approx}$	1.05E-1 (5.8E-3)$^{-}$	9.88E-2 (2.0E-3)$^{\approx}$	3.04E-1 (3.9E-2)$^{-}$	2.71E-1 (1.9E-2)$^{-}$	8.70E-2 (5.7E-3)$^{\approx}$
mQAP_50_2(-0.3)	3.27E-1 (4.2E-2)	3.10E-1 (3.1E-2)$^{\approx}$	3.32E-1 (3.3E-2)$^{-}$	3.98E-1 (5.0E-2)$^{-}$	3.42E-1 (4.3E-2)$^{-}$	3.25E-1 (3.5E-2)$^{\approx}$	4.07E-1 (4.4E-2)$^{-}$	3.63E-1 (3.6E-2)$^{\approx}$
mQAP_50_2(0.0)	3.45E-1 (3.3E-2)	3.43E-1 (3.0E-2)$^{\approx}$	3.41E-1 (2.8E-2)$^{-}$	4.31E-1 (2.9E-2)$^{-}$	3.71E-1 (3.6E-2)$^{-}$	3.59E-1 (3.4E-2)$^{\approx}$	3.96E-1 (2.8E-2)$^{-}$	3.77E-1 (3.2E-2)$^{\approx}$
mQAP_50_2(0.3)	3.72E-1 (3.5E-2)	3.79E-1 (3.5E-2)$^{\approx}$	3.95E-1 (2.9E-2)$^{-}$	4.70E-1 (4.8E-2)$^{-}$	3.75E-1 (2.8E-2)$^{\approx}$	3.91E-1 (5.2E-2)$^{-}$	4.90E-1 (3.6E-2)$^{-}$	4.27E-1 (3.5E-2)$^{\approx}$
mQAP_60_2(-0.3)	3.24E-1 (3.5E-2)	3.66E-1 (3.3E-2)$^{-}$	3.56E-1 (3.1E-2)$^{-}$	4.39E-1 (4.3E-2)$^{-}$	3.47E-1 (3.3E-2)$^{-}$	3.79E-1 (3.7E-2)$^{-}$	4.09E-1 (1.7E-2)$^{-}$	3.97E-1 (3.5E-2)$^{-}$
mQAP_60_2(0.0)	3.32E-1 (2.4E-2)	3.39E-1 (1.7E-2)$^{\approx}$	3.61E-1 (3.5E-2)$^{-}$	4.17E-1 (3.5E-2)$^{-}$	3.62E-1 (3.4E-2)$^{-}$	3.70E-1 (2.8E-2)$^{-}$	3.97E-1 (3.8E-2)$^{-}$	3.33E-1 (2.9E-2)$^{\approx}$
mQAP_60_2(0.3)	3.60E-1 (2.5E-2)	3.54E-1 (2.7E-2)$^{\approx}$	3.61E-1 (2.5E-2)$^{\approx}$	3.97E-1 (2.2E-2)$^{-}$	3.61E-1 (4.2E-2)$^{-}$	3.79E-1 (2.7E-2)$^{-}$	4.12E-1 (3.3E-2)$^{-}$	3.97E-1 (2.9E-2)$^{-}$
mQAP_30_3(-0.3)	1.57E-1 (1.3E-2)	1.67E-1 (1.3E-2)$^{\approx}$	1.25E-1 (1.5E-2)$^{+}$	1.28E-1 (1.5E-2)$^{+}$	1.15E-1 (5.1E-3)$^{+}$	1.95E-1 (1.4E-2)$^{-}$	2.34E-1 (2.3E-2)$^{-}$	1.54E-1 (1.6E-2)$^{\approx}$
mQAP_30_3(0.0)	1.65E-1 (9.3E-3)	1.68E-1 (1.9E-2)$^{\approx}$	1.36E-1 (1.5E-2)$^{+}$	1.41E-1 (1.2E-2)$^{\approx}$	1.16E-1 (3.7E-3)$^{+}$	2.07E-1 (2.7E-2)$^{-}$	2.55E-1 (2.1E-2)$^{-}$	1.64E-1 (1.8E-2)$^{\approx}$
mQAP_30_3(0.3)	1.81E-1 (9.6E-3)	1.99E-1 (2.1E-2)$^{\approx}$	1.47E-1 (1.5E-2)$^{+}$	1.51E-1 (2.2E-2)$^{+}$	1.16E-1 (5.4E-3)$^{+}$	2.12E-1 (2.0E-2)$^{-}$	2.60E-1 (1.5E-2)$^{-}$	1.79E-1 (2.0E-2)$^{\approx}$
mQAP_40_3(-0.3)	1.48E-1 (1.3E-2)	1.56E-1 (1.6E-2)$^{\approx}$	1.09E-1 (1.3E-2)$^{+}$	1.22E-1 (1.1E-2)$^{\approx}$	1.18E-1 (7.1E-3)$^{+}$	2.38E-1 (2.1E-2)$^{-}$	2.40E-1 (2.1E-2)$^{-}$	1.40E-1 (1.5E-2)$^{\approx}$
mQAP_40_3(0.0)	1.80E-1 (1.1E-2)	1.97E-1 (1.9E-2)$^{\approx}$	1.36E-1 (9.0E-3)$^{+}$	1.43E-1 (1.4E-2)$^{+}$	1.26E-1 (5.1E-3)$^{+}$	2.65E-1 (2.3E-2)$^{-}$	2.58E-1 (2.0E-2)$^{-}$	1.72E-1 (1.5E-2)$^{\approx}$
mQAP_40_3(0.3)	1.62E-1 (1.7E-2)	1.66E-1 (1.6E-2)$^{\approx}$	1.19E-1 (1.1E-2)$^{-}$	1.29E-1 (7.8E-3)$^{+}$	1.19E-1 (6.4E-3)$^{+}$	2.59E-1 (2.5E-2)$^{-}$	2.55E-1 (1.8E-2)$^{-}$	1.52E-1 (1.5E-2)$^{\approx}$

The best DIR value is highlighted in gray.
'+', '−' and '≈' indicate that the result is significantly better, significantly worse and statistically similar to that of CoMOLS/D on this test instance, respectively (wilcoxon's rank sum test at the 0.05 significance level).

Table 4
Comparisons of CoMOLS/D with MOEA/D-LS (WS, TCH, PBI and iPBI), ϵ-MOEA-LS, IBEA-LS and ND/DPP-LS in terms of C-metric.

Instance	MOEA/D-LS [17]								ϵ-MOEA-LS [57]		IBEA-LS [58]		ND/DPP-LS [27]	
	WS		TCH		PBI		iPBI							
	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)
ClusterAB100	**70.91**	13.15	27.29	**48.86**	**48.86**	26.30	31.66	**43.02**	**97.34**	3.38	**100**	0.00	26.17	**38.21**
ClusterAB300	**78.11**	14.89	**38.40**	27.17	**63.53**	21.44	**59.85**	23.24	**100**	0.00	**100**	0.00	**44.51**	25.33
euclidAB100	**95.22**	1.11	**53.47**	37.30	**67.74**	21.27	**60.71**	20.48	**97.45**	1.43	**100**	0.00	32.39	**49.43**
euclidAB300	**82.47**	8.58	**54.39**	19.09	**88.62**	7.33	**46.30**	33.20	**100**	0.00	**100**	0.00	**73.68**	20.30
kroAB100	**84.67**	6.40	**46.52**	39.52	**49.44**	28.64	**55.06**	36.00	**93.70**	3.30	**100**	0.00	**35.63**	28.46
kroAB200	**87.67**	6.32	**44.35**	27.12	**52.31**	30.12	**51.61**	30.78	**100**	0.00	**100**	0.00	**54.17**	34.04
kroAB300	**86.60**	8.60	**40.13**	28.57	**94.12**	2.04	**66.49**	16.18	**100**	0.00	**100**	0.00	**31.61**	29.38
euclidABC100	14.07	**29.10**	**64.70**	2.63	**64.51**	5.33	**99.68**	0.00	**97.81**	0.00	**100**	0.00	**57.21**	13.93
kroABC100	13.70	**28.86**	**68.29**	2.36	**59.71**	3.93	**97.31**	0.02	**100**	0.00	**100**	0.00	**52.49**	6.18
mQAP_50_2(-0.3)	**88.78**	3.17	**50.76**	42.25	**65.64**	26.76	**59.43**	31.69	**67.72**	32.66	**80.36**	8.08	**43.32**	37.42
mQAP_50_2(0.0)	**85.71**	7.62	**52.26**	41.59	**73.68**	20.00	33.22	**55.24**	**91.05**	3.17	**92.86**	4.52	**63.48**	33.59
mQAP_50_2(0.3)	**97.65**	0.36	43.79	**49.64**	**52.17**	32.14	42.77	**68.21**	**75.78**	21.45	**99.40**	0.00	32.36	**47.68**
mQAP_60_2(-0.3)	**95.24**	0.69	41.69	**57.04**	**54.07**	30.72	**42.45**	27.94	**100**	0.00	**100**	0.00	22.04	**44.07**
mQAP_60_2(0.0)	**88.37**	0.63	33.25	**57.99**	15.85	**61.76**	24.28	**63.32**	**95.45**	1.22	**62.88**	25.84	34.26	**53.42**
mQAP_60_2(0.3)	**88.37**	1.92	**51.63**	40.38	**75.73**	21.15	**44.77**	36.22	**97.42**	0.00	**91.49**	7.09	**52.68**	22.03
mQAP_30_3(-0.3)	**49.30**	18.79	**40.49**	9.97	**60.24**	7.22	**91.28**	0.44	**96.40**	0.08	**34.84**	9.44	**42.49**	10.37
mQAP_30_3(0.0)	**52.10**	13.97	**46.61**	13.37	**61.57**	10.55	**84.23**	0.94	**97.00**	0.00	**37.32**	9.69	**36.49**	16.37
mQAP_30_3(0.3)	**50.00**	18.80	**32.08**	20.78	**55.94**	14.19	**78.81**	2.69	**92.24**	0.20	**68.47**	4.72	**45.39**	22.45
mQAP_40_3(-0.3)	**49.76**	19.24	**75.43**	3.84	**84.29**	3.59	**94.05**	2.12	**99.16**	0.00	**33.30**	14.18	**85.33**	4.57
mQAP_40_3(0.0)	**47.24**	14.94	**55.85**	9.25	**69.43**	3.56	**96.46**	0.24	**100**	0.00	**36.51**	6.12	**45.85**	6.85
mQAP_40_3(0.3)	**45.25**	19.98	**58.99**	8.06	**74.13**	7.76	**97.38**	0.05	**100**	0.00	**32.46**	9.74	**62.34**	5.26

Better C-metric values are highlighted in boldface.
A corresponds to CoMOLS/D; B corresponds to the compared algorithm.

approach. First, a new population TQ is constructed by associating a random solution in TP to each direction vector in W (i.e. W_1 or W_2). The individuals in TP are seen as the updating information for TQ. As the algorithm shows, then every subproblem in TQ is checked to see if the solution can be replaced by any individuals in population TP based on the aggregate function.

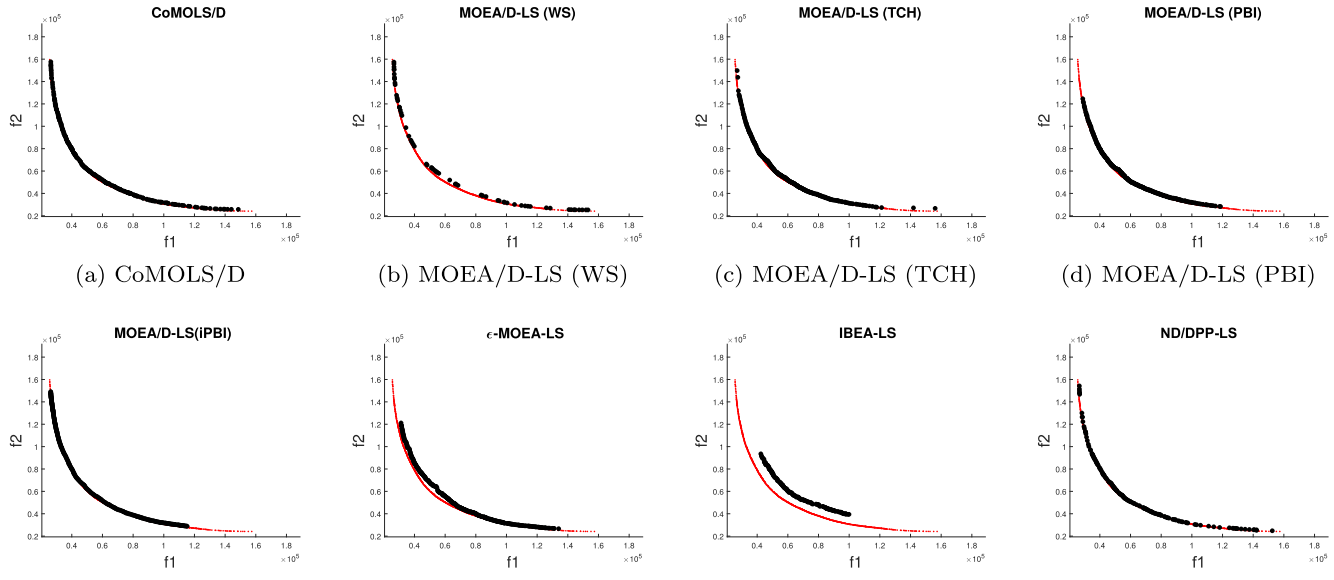
4. Experimental setup

This section is devoted to the experimental setup. Two combinatorial MOPs, i.e., multiobjective traveling salesman problem (mTSP) and multiobjective quadratic assignment problem (mQAP) are firstly introduced, followed by the explanation of the performance metrics adopted in the experimental studies. Finally, the parameter settings of the compared algorithms are also given in detail.

4.1. Multiobjective traveling salesman problem

In the traveling salesman problem (TSP), a salesman tends to find the shortest route of n cities, starting and ending at the same city and visiting each of the other cities only once. The TSP could be defined as a graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of cities, and $E = \{e = (i, j) \mid i, j \in V\}$ is the set of edges.

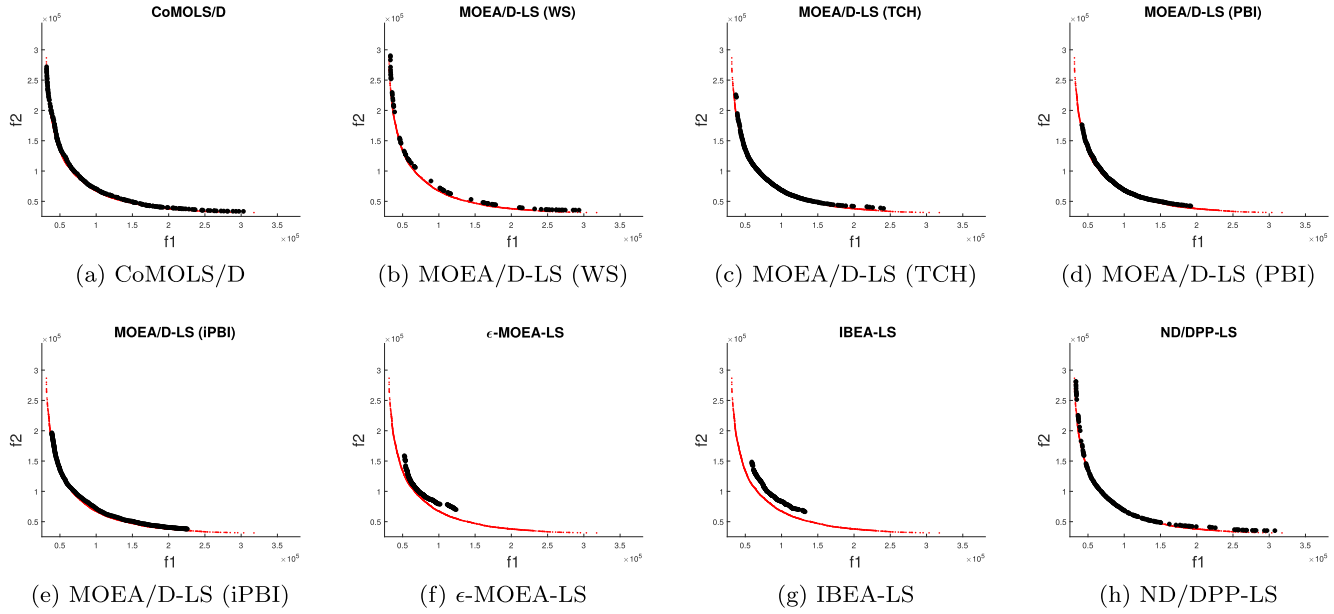
For multiobjective TSP (mTSP), each edge e has m cost values. Each feasible solution contains all the cities in V which can form a Hamiltonian cycle. Mathematically, the mTSP can be formulated as:



(a) CoMOLS/D (b) MOEA/D-LS (WS) (c) MOEA/D-LS (TCH) (d) MOEA/D-LS (PBI)

(e) MOEA/D-LS (iPBI) (f) ϵ-MOEA-LS (g) IBEA-LS (h) ND/DPP-LS

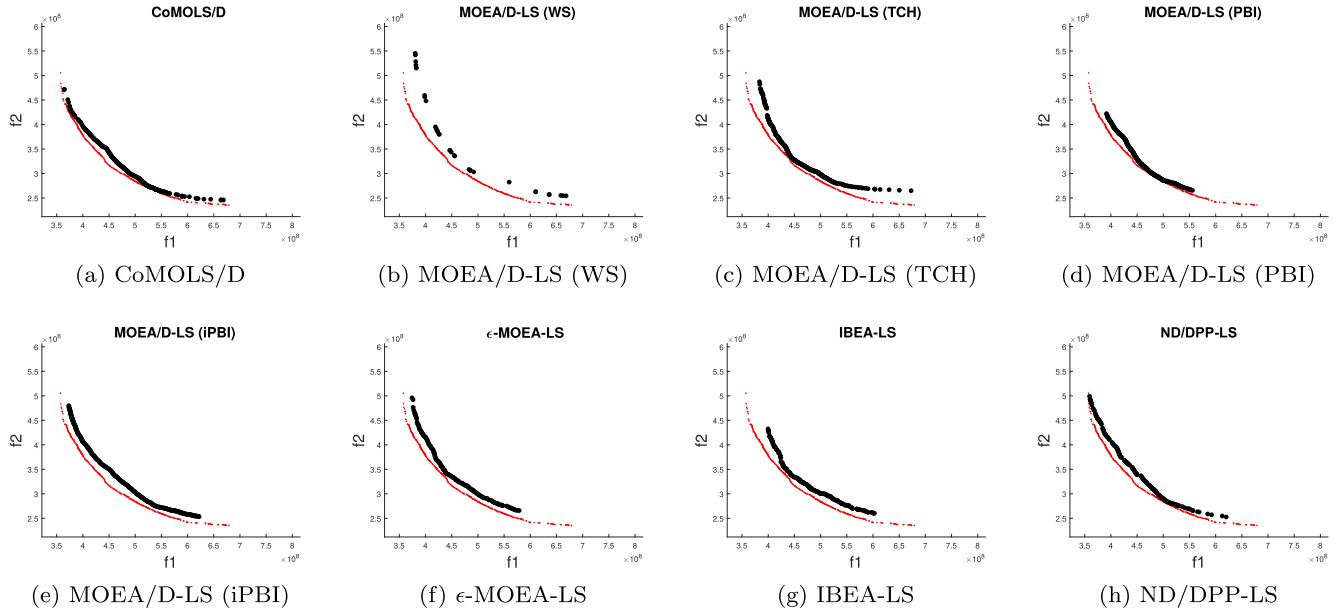
Fig. 3. The nondominated solutions obtained by eight algorithms in the run with the median HV values on euclidAB100.



(a) CoMOLS/D (b) MOEA/D-LS (WS) (c) MOEA/D-LS (TCH) (d) MOEA/D-LS (PBI)

(e) MOEA/D-LS (iPBI) (f) ϵ-MOEA-LS (g) IBEA-LS (h) ND/DPP-LS

Fig. 4. The nondominated solutions obtained by eight algorithms in the run with the median HV value on kroAB200.



(a) CoMOLS/D (b) MOEA/D-LS (WS) (c) MOEA/D-LS (TCH) (d) MOEA/D-LS (PBI)

(e) MOEA/D-LS (iPBI) (f) ϵ-MOEA-LS (g) IBEA-LS (h) ND/DPP-LS

Fig. 5. The nondominated solutions obtained by eight algorithms in the run with the median HV value on mQAP_50_2(-0.3).

$$\text{minimize} \quad \mathcal{F}_k(\boldsymbol{\pi}) = \sum_{j=1}^{n-1} c^{(k)}_{\pi(j),\pi(j+1)} + c^{(k)}_{\pi(1),\pi(n)}, \qquad (9)$$

$$\text{subject to} \quad k = 1, \ldots, m$$

where $\boldsymbol{\pi} = (\pi(1), \ldots, \pi(n))$ is a permutation of cities; $c^{(k)}_{s,t}$ is the kth$(1 \le k \le m)$ cost of the edge between city s and city t; and m is the number of objectives to be optimized. In our experiments, m is equal to 2 or 3.

4.2. Multiobjective quadratic assignment problem

The multiobjective quadratic assignment problem (mQAP) models any sort of facilities layout problem in which the minimization of multiple simultaneous flows is required. Given n facilities and n locations, the mQAP consists of a single $n \times n$ *distance* matrix A where $a_{i,j}$ is the distance between locations i and j, and m distinct $n \times n$ *flow* matrices $B^{(k)}$ $(1 \le k \le m)$, where $b^{(k)}_{r,s}$ is the kth flow between facilities r and s. The mQAP can be stated as follows:

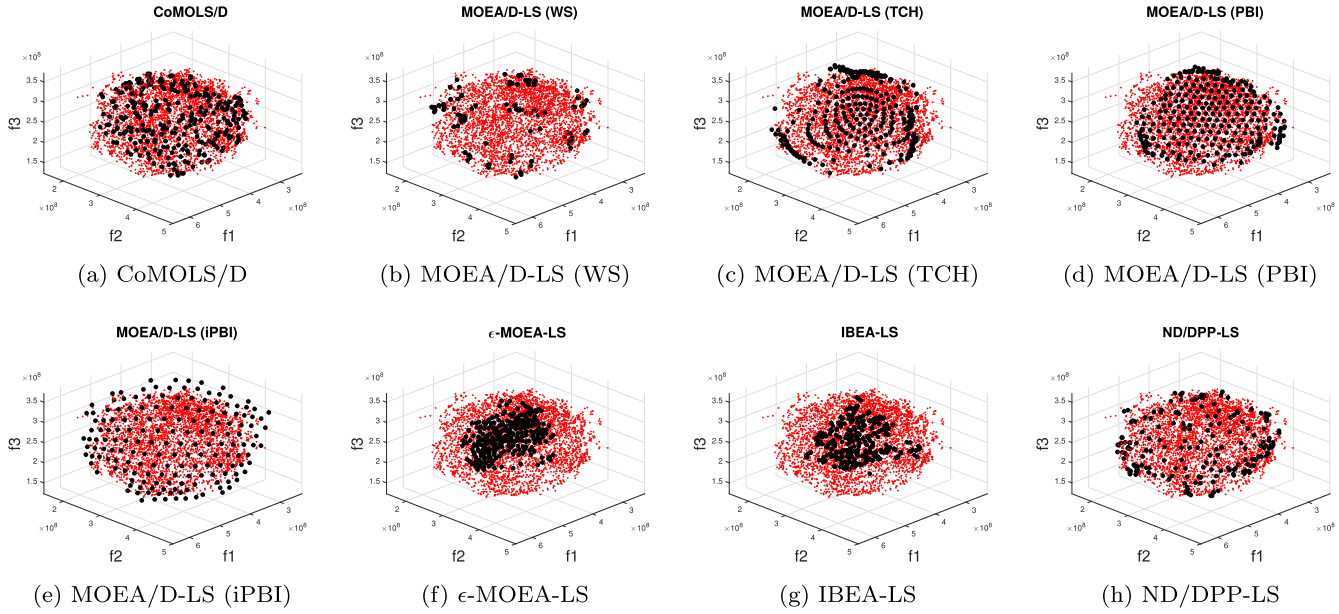
$$\text{minimize} \quad \mathcal{F}_k(\boldsymbol{\pi}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b^{(k)}_{\pi(i),\pi(j)}, \qquad (10)$$

$$\text{subject to} \quad k = 1, \ldots, m$$

where $\boldsymbol{\pi}$ is a feasible solution constructed by a permutation of n facilities.

4.3. Test instances

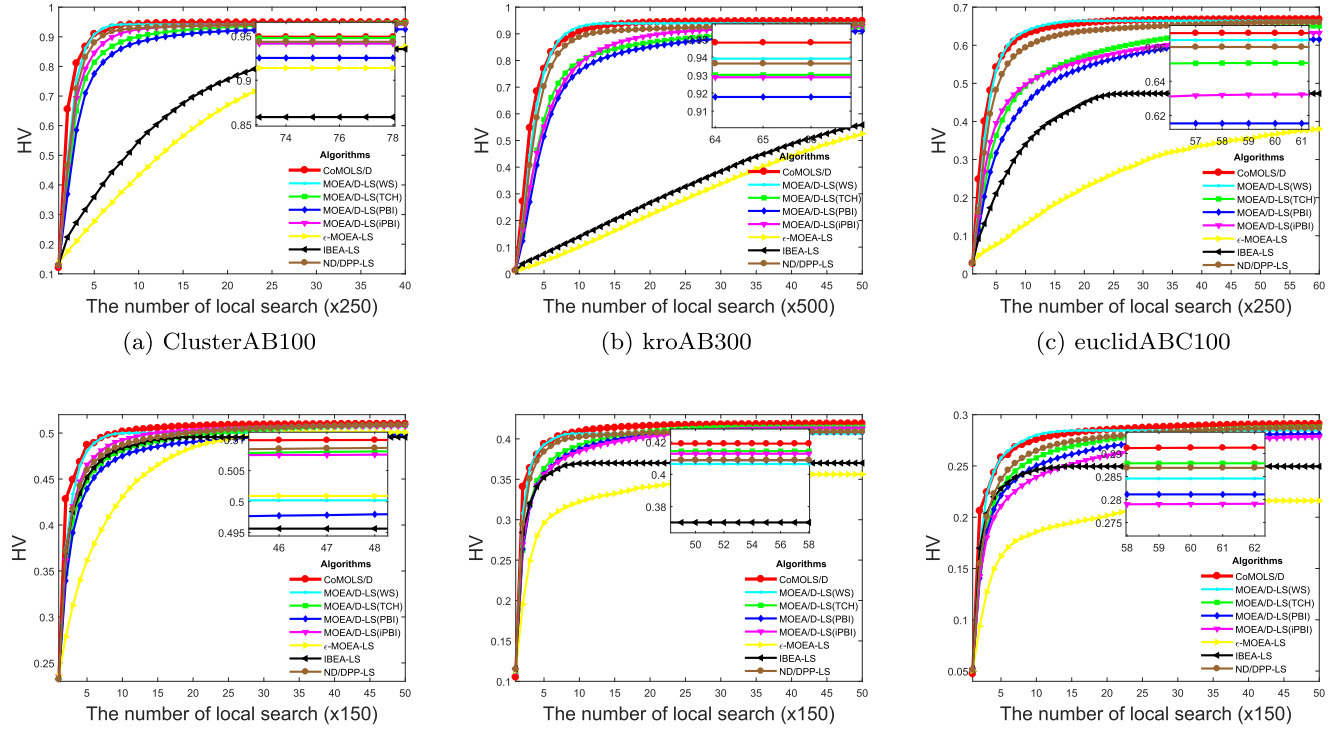
Nine mTSP instances are considered in the experiments. All these instances with the prefix “kro”, “euclid” or “Cluster” are collected from



(a) CoMOLS/D (b) MOEA/D-LS (WS) (c) MOEA/D-LS (TCH) (d) MOEA/D-LS (PBI)

(e) MOEA/D-LS (iPBI) (f) ϵ-MOEA-LS (g) IBEA-LS (h) ND/DPP-LS

Fig. 6. The nondominated solutions obtained by eight algorithms in the run with the median HV value on mQAP_40_3(-0.3).



(a) ClusterAB100 (b) kroAB300 (c) euclidABC100

(d) mQAP_60_2(0.0) (e) mQAP_30_3(-0.3) (f) mQAP_40_3(0.3)

Fig. 7. The average normalized HV values vs. the number of local search times for eight algorithms over 31 runs.

Refs. [21,46,47] as the test instances.[2] They are named as follows: for example, "kroABC100" means that the instance is "kro" with 100 cities and three objectives ('A', B′ and 'C'). All these instances are symmetric version of the TSP, which means that the distance from node i to node j is the same as that from node j to node i.

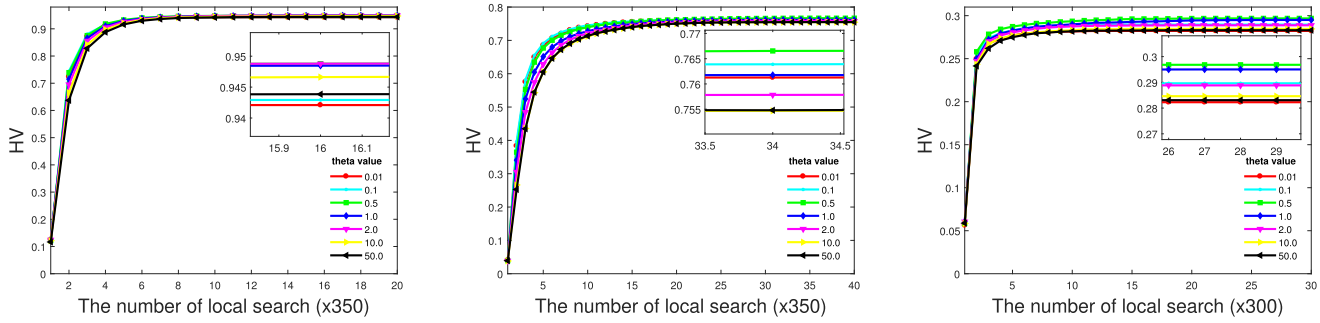
For mQAP, all the instances were generated according to Ref. [48] with $n = 50, 60$ for bi-objective and $n = 30, 40$ for tri-objective. Moreover, the parameter ξ can be used to control the correlation between the flow matrices in mQAP instances [49] ($\xi \in \{-0.3, 0, 0.3\}$). An instance named "mQAP_60_2 (0.3)" indicates that it is a bi-objective QAP, in which the number of facilities (locations) n is 60 and the correlation parameter ξ is set to 0.3.

[2] Files are downloaded from https://sites.google.com/site/thibautlust/research/multiobjective-tsp.

Table 5
The average CPU time (in second) spent by eight compared algorithms.

Instance	CoMOLS/D	MOEA/D-LS(WS) [17]				ϵ-MOEA-LS [57]	IBEA-LS [58]	ND/DPP-LS [27]
		WS	TCH	PBI	iPBI			
ClusterAB100	53.1	14.0	66.5	50.4	66.4	131.7	195.2	82.5
ClusterAB300	156.6	40.1	247.8	199.2	258.9	772.0	735.0	294.7
euclidAB100	59.9	13.6	65.7	59.6	68.0	177.1	218.3	90.2
euclidAB300	177.9	33.8	228.2	218.5	240.7	604.8	692.1	317.0
kroAB100	49.1	11.9	63.8	64.6	69.2	176.0	208.7	97.4
kroAB200	58.4	15.7	76.3	82.8	87.5	508.8	460.7	108.7
kroAB300	134.1	40.4	212.4	225.3	260.0	725.7	687.2	282.4
euclidABC100	194.7	70.5	201.7	151.9	255.9	247.4	219.5	269.5
kroABC100	159.2	75.2	187.0	187.2	334.6	241.4	217.0	174.6
mQAP_50_2(-0.3)	8.6	1.3	7.2	7.2	6.7	79.8	101.2	7.5
mQAP_50_2(0.0)	7.2	1.2	7.6	7.2	6.0	79.3	106.2	8.1
mQAP_50_2(0.3)	6.9	1.2	5.8	5.7	5.9	74.6	96.9	6.7
mQAP_60_2(-0.3)	18.0	2.7	15.5	15.0	13.8	153.0	135.6	19.3
mQAP_60_2(0.0)	19.1	2.8	18.4	13.6	19.0	154.1	141.0	20.8
mQAP_60_2(0.3)	16.8	2.6	16.8	14.1	14.0	149.7	144.3	17.3
mQAP_30_3(-0.3)	2.5	0.6	2.1	2.2	3.1	81.7	86.2	2.2
mQAP_30_3(0.0)	3.2	0.5	2.2	2.1	2.9	77.6	83.8	2.1
mQAP_30_3(0.3)	1.7	0.5	1.8	1.7	2.4	55.6	78.7	1.8
mQAP_40_3(-0.3)	9.4	1.8	6.6	6.4	8.1	156.0	127.9	5.9
mQAP_40_3(0.0)	8.4	1.7	6.2	6.4	7.6	170.9	121.8	6.3
mQAP_40_3(0.3)	10.3	1.8	7.4	6.1	8.3	174.0	123.0	7.6

All the algorithms were tested on a PC with intel(R) Core(TM) i7-7700 CPU 3.6 GHz.
All the algorithms were coded in Java and based on the framework of jMetal [56].



(a) ClusterAB100 (b) kroABC100 (c) mQAP_30_3(0.0)

Fig. 8. Sensitivity test of the penalty parameter θ in CoMOLS/D.

4.4. Performance metrics

4.4.1. Hypervolume (HV)

Suppose that $\mathbf{r}^* = (r_1^*, r_2^*, \ldots, r_m^*)^T$ is a reference point in the objective space which is dominated by any Pareto optimal objective vectors in a PF approximation S. The HV value of S (with regard to $\mathbf{r}^*$) measures the volume of region in the objective space dominated by S and bounded by $\mathbf{r}^*$ [50].

$$HV(S) = Leb\left(\bigcup_{x\in S}[f_1(x), r_1^*] \times \ldots [f_m(x), r_m^*]\right) \tag{11}$$

where $Leb(\bullet)$ indicates the Lebesgue measure. HV can measure the approximation in terms of both convergence and diversity. Obviously, the higher the HV value, the better the approximation is. In this paper, the HV value of S is calculated based on Eq. (11) with reference point $(1.1, 1.1, \ldots)^T$ after normalizing S.

4.4.2. Set coverage (C-metric)

Let $\mathbb{A}$ and $\mathbb{B}$ be two different sets of PF approximation of a MOP, $C(\mathbb{A}, \mathbb{B})$ indicates the percentage of the solutions in $\mathbb{B}$ that are dominated by at least one solution in $\mathbb{A}$:

$$C(\mathbb{A}, \mathbb{B}) = \frac{|\{u \in \mathbb{B} \mid \exists v \in \mathbb{A} : v \prec u\}|}{|\mathbb{B}|} \times 100\% \tag{12}$$

where $C(\mathbb{A}, \mathbb{B}) = 0$ means that no solution in $\mathbb{B}$ is dominated by any one in $\mathbb{A}$ while $C(\mathbb{A}, \mathbb{B}) = 100$ means that each solution in $\mathbb{B}$ are dominated by at least one in $\mathbb{A}$. The indicators of $C(\mathbb{A}, \mathbb{B})$ and $C(\mathbb{B}, \mathbb{A})$ together can show the relative convergence between $\mathbb{A}$ and $\mathbb{B}$ [50].

4.4.3. Inverted generational distance (IGD)

Let P^* be a set of points uniformly sampled over the true PF, and S be the set of solutions obtained by an EMO algorithm. The IGD [51,52] value of S is computed as:

$$IGD(S, P^*) = \frac{\sum_{x\in P^*} dist(x, S)}{|P^*|} \tag{13}$$

where $dist(x, S)$ is the Euclidean distance between a point $x \in P^*$ and its nearest neighbor in S, and $|P^*|$ is the cardinality of P^*. IGD calculates an average minimum distance from each point in P^* to those in S, which measures both convergence and diversity of a solution set S. The lower the IGD value is, the better the quality of S is. IGD could measure both convergence and diversity like HV, but it requires the true Pareto

fronts. For COMPs, the true PFs are always unknown, so we set all the nondominated solutions delivered by all the eight compared algorithms over all the 31 runs as the approximation PFs.

4.4.4. Diversity indicator based on reference vectors (DIR)

Given a Pareto approximation S that contains N solutions in m-dimensional objective space, DIR [53] is used to estimate the diversity of S. Let $V = \{\lambda^1, \lambda^2, \ldots, \lambda^M\}$ be a set of uniformly generated reference vectors. The coverage of each solution in S is obtained by computing the number of nearest reference vectors. Suppose a coverage vector $c = (c_1, c_2, \ldots, c_N)^T$ with regard to S is obtained, the DIR value of S is computed as:

$$DIR = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(c_i - mean(c))^2}}{\frac{M}{N}\sqrt{N-1}} \tag{14}$$

where $mean(c)$ is the average value of c. DIR is a unary diversity indicator to estimate the diversity of PF approximations for multi/many-objective optimization.

4.5. Parameter settings

The population sizes of all the compared algorithms are set to 300 for both bi-objective and tri-objective problems. The maximal number of iterations is set to 100 for bi-objective problems and 200 for tri-objective ones. Each algorithm was run 31 times independently for each instance.

In the decomposition-based approaches, a set of direction vectors $W: \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$ is uniformly generated by Das and Dennis's method [54]. The number of direction vectors N is controlled by a parameter H: $N = \binom{H+m-1}{m-1}$, where m is the number of objectives with a uniform spacing $\delta = 1/H$. For the bi-objective problems, if 300 direction vectors is needed, we set $H = 299$. As for tri-objective ones ($m = 3$), if 300 direction vectors are required, then we set $H = 23$.

To maintain a similar population size with MOEA/D-LS, IBEA-LS and CoMOLS/D, ϵ in ϵ-MOEA-LS is set to 100 for bi-objective mTSP instances and 1,000 for tri-objective mTSP instances. For mQAP, ϵ is set to 700,000 for bi-objective instances and 3,000,000 for tri-objective ones. NSGA-II and MOEA/D(TCH) are chosen as one Pareto-based and one decomposition-based method embedded in ND/DPP-LS, where the Pareto-based population size is set to 150 for all the test instances, the decomposition-based population size is set to 150 for bi-objective instances and 153 for tri-objective instances for uniformity. In the decomposition-based algorithms, the local search is conducted on the solution of each subproblem for replacing its neighbors. In the non-decomposition-based algorithms, N solutions are selected from the offspring obtained by conducting local search on their parents.

For mTSP and mQAP instances, 2-opt neighborhood approach [22,55], which swaps two elements of the permutation for generating the neighboring solutions, is adopted for the neighborhood $N(x)$ of each solution x for all the compared algorithms. The penalty parameter is set to 5.0 for MOEA/D-LS (PBI); 1.0 for MOEA/D-LS (iPBI) and CoMOLS/D on all the bi-objective problems and 0.5 on all the tri-objective problem.

All the compared algorithms are implemented based on the open source MOEA platform, jMetal [56]. The fundamental data structures are the same for all the compared algorithms.

5. Experimental results and discussions

In this section, the following experiments are conducted to test the performance of CoMOLS/D:

- comparisons with MOEA/D-LS (WS, TCH, PBI and iPBI), ϵ-MOEA-LS [57] and IBEA-LS [58] in terms of HV, IGD, DIR, C-metric and CPU time;

Table 6
Sensitivity test on some instances over different population size $|P|$.

Instance	$\frac{N}{4}$	$\frac{N}{3}$	$\frac{N}{2}$	$\frac{2N}{3}$	$\frac{3N}{4}$	N
ClusterAB100	9.25E-01 (2.81E-03)	9.34E-01 (2.33E-03)	9.49E-01 (1.42E-03)	9.41E-01 (4.16E-03)	9.41E-01 (1.70E-03)	9.49E-01 (1.54E-03)
euclidAB300	9.14E-01 (3.31E-03)	9.10E-01 (1.53E-03)	9.12E-01 (1.53E-03)	9.11E-01 (2.80E-03)	9.07E-01 (1.76E-03)	9.04E-01 (1.62E-03)
kroAB200	9.52E-01 (2.01E-03)	9.59E-01 (2.58E-03)	9.74E-01 (1.66E-03)	9.73E-01 (1.81E-03)	9.71E-01 (1.38E-03)	9.65E-01 (1.41E-03)
euclidABC100	6.14E-01 (2.14E-03)	6.51E-01 (1.49E-03)	6.68E-01 (1.76E-03)	6.83E-01 (4.35E-03)	6.76E-01 (1.98E-03)	6.66E-01 (2.14E-03)
mQAP_60_2(0.0)	4.99E-01 (1.04E-03)	5.09E-01 (1.73E-03)	5.10E-01 (5.94E-03)	5.24E-01 (2.47E-03)	5.16E-01 (1.29E-03)	5.08E-01 (1.04E-03)
mQAP_40_3(0.3)	2.79E-01 (1.44E-03)	2.87E-01 (1.76E-03)	2.91E-01 (4.02E-03)	2.87E-01 (2.80E-03)	2.82E-01 (1.64E-03)	2.84E-01 (1.44E-03)

The best HV value is highlighted in gray.

Table 7
CoMOLS/D (iPBI) vs. CoMOLS/D (iTCH) in terms of *C*-metric and HV.

Instance	*C*-metric		HV	
	C(A,B)	C(B,A)	CoMOLS/D (iPBI)	CoMOLS/D (iTCH)
ClusterAB100	31.03	**50.32**	9.491E-01 (1.42E-03)	9.495E-01 (1.28E-03)≈
ClusterAB300	13.16	**81.51**	**9.750E-01 (1.91E-03)**	9.772E-01 (1.70E-03)+
euclidAB100	41.47	**43.81**	7.966E-01 (1.53E-03)	7.964E-01 (1.62E-03)≈
euclidAB300	19.77	**76.90**	**9.128E-01 (1.59E-03)**	9.144E-01 (1.41E-03)+
kroAB100	**51.13**	32.80	8.697E-01 (2.33E-03)	8.701E-01 (1.54E-03)≈
kroAB200	**51.11**	40.93	9.742E-01 (1.66E-03)	9.747E-01 (2.14E-03)≈
kroAB300	23.70	**61.22**	**9.482E-01 (1.63E-03)**	9.496E-01 (1.04E-03)+
euclidABC100	**22.30**	22.23	6.682E-01 (1.76E-03)	**6.669E-01 (1.44E-03)**-
kroABC100	**23.84**	20.38	7.671E-01 (1.45E-03)	**7.658E-01 (1.49E-03)**-
mQAP_50_2(-0.3)	50.96	**52.82**	5.715E-01 (7.07E-03)	5.706E-01 (6.11E-03)≈
mQAP_50_2(0.0)	**76.25**	9.21	5.437E-01 (5.41E-03)	5.404E-01 (6.17E-03)≈
mQAP_50_2(0.3)	**61.79**	43.21	5.625E-01 (8.83E-03)	5.606E-01 (8.36E-03)≈
mQAP_60_2(-0.3)	**65.51**	28.87	5.374E-01 (6.30E-03)	5.378E-01 (5.17E-03)≈
mQAP_60_2(0.0)	**73.75**	20.38	5.103E-01 (5.94E-03)	5.087E-01 (5.99E-03)≈
mQAP_60_2(0.3)	28.43	**63.46**	5.417E-01 (5.07E-03)	5.395E-01 (4.69E-03)≈
mQAP_30_3(-0.3)	29.86	**33.85**	4.193E-01 (5.79E-03)	4.185E-01 (6.25E-03)≈
mQAP_30_3(0.0)	**34.33**	31.40	2.953E-01 (3.60E-03)	2.951E-01 (4.61E-03)≈
mQAP_30_3(0.3)	30.11	**40.22**	5.904E-01 (9.00E-03)	5.884E-01 (9.65E-03)≈
mQAP_40_3(-0.3)	32.78	**34.60**	2.783E-01 (4.62E-03)	2.791E-01 (4.25E-03)≈
mQAP_40_3(0.0)	22.16	**36.29**	2.866E-01 (4.81E-03)	2.872E-01 (4.70E-03)≈
mQAP_40_3(0.3)	26.78	**37.22**	2.916E-01 (4.02E-03)	2.930E-01 (4.75E-03)≈

1. A corresponds to CoMOLS/D (iPBI). B corresponds to CoMOLS/D (iTCH).
2. '+', '−' or '≈' indicate that the HV values obtained by the corresponding algorithm is significantly better, significantly worse or similar to that of CoMOLS/D (iPBI) on this test instance, respectively (wilcoxon's rank sum test at the 0.05 significance level).

- sensitivity test on the penalty parameter θ in CoMOLS/D;
- sensitivity test on $|P|$ in CoMOLS/D;
- iPBI vs. iTCH in CoMOLS/D.

5.1. Performance results

The performance of the nondominated sets obtained by all the eight compared algorithms on nine mTSP and twelve mQAP instances, in terms of HV, IGD and DIR are presented in Tables 1–3 respectively. It can be observed that CoMOLS/D significantly outperforms other seven algorithms on all nine mTSP instances except for "euclidABC100" and "kroABC100" in terms of DIR. As for the mQAP instances, CoMOLS/D achieves the best performance on all the tri-objective ones except for "mQAP_30_3(0.3)" and "mQAP_40_3(-0.3)" in terms of HV and IGD. For bi-objective mQAP instances, CoMOLS/D, MOEA/D-LS (TCH), MOEA/D-LS (iPBI) and ND/DPP-LS have very similar performance.

The convergence performance of all the eight algorithms on all the test instances in terms of *C*-metric are shown in Table 4. It is clear to see that CoMOLS/D has better convergence performance on most of the test instances.

To visualize the final performance of all the eight compared algorithms, their final nondominated solutions on euclidAB100, kroAB200, mQAP 50 2(-0.3) and mQAP 40 3(-0.3) are illustrated in Figs. 3–6. In these figures, all the nondominated solutions delivered by all the eight compared algorithms over all the 31 runs, marked in red dots ("·") are plotted as the substitution of a reference PF. It can be seen clearly that the solution set obtained by CoMOLS/D are more widely and uniformly distributed. The solution sets derived by MOEA/D-LS (WS) are widely but unevenly distributed. The solution sets obtained by MOEA/D-LS (TCH, PBI and iPBI) are uniformly but narrowly distributed. Both the convergence and diversity of the nondominated solutions derived by ϵ-MOEA-LS and IBEA-LS are not very satisfactory (see Figs. 3–6).

In addition, the convergence plots among all the eight algorithms over "ClusterAB100", "kroAB300", "euclidABC100", "mQAP_60_2(0.0)", "mQAP_30_3(-0.3)", and "mQAP_40_3(0.3)" are illustrated in Fig. 7. It can be observed that CoMOLS/D converges very fast and always achieves the best final performance in terms of HV.

The CPU time used by each algorithm for each run is recorded and listed in Table 5. In general, all the decomposition-based algorithms run faster than the domination-based and indicator-based algorithms on all the test instances. Among them, MOEA/D-LS (WS) has the fastest convergence speed for all the instances. As for mTSP instances, CoMOLS/D uses less time than MOEA/D-LS (TCH, PBI and iPBI) except for "ClusterAB100", "euclidAB100" and "euclidABC100". For mQAP instances, CoMOLS/D and MOEA/D-LS (TCH, PBI and iPBI) consume very similar amount of CPU time.

5.2. Sensitivity test on penalty parameter θ

In CoMOLS/D, a penalty parameter θ is used to control the balance of diversity and convergence in PBI and iPBI. To investigate its sensitivity on CoMOLS/D, the performance of CoMOLS/D with θ = (0.01, 0.1, 0.5, 1.0, 2.0, 10.0, 50.0) is tested. Fig. 8 shows the performance of CoMOLS/D in terms of the mean HV values, with different θ values on "ClusterAB100", "kroABC100" and "mQAP_30_3(0.0)".

It can be observed that a larger value of θ usually leads to the slower convergence speed for CoMOLS/D. CoMOLS/D achieves the best performance with $\theta = 1.0 or 2.0$ for the bi-objective instances and $\theta = 0.5$ for the tri-objective ones.

5.3. Sensitivity test on $|P|$

The population in CoMOLS/D contains two subpopulations: P and Q. As $|Q| = N - |P|$, only the sensitivity test on $|P|$ is conducted as follows.

The performance of CoMOLS/D with $|P| = (\frac{N}{4}, \frac{N}{3}, \frac{N}{2}, \frac{2N}{3}, \frac{3N}{4}, N)$ is shown in Table 6. It can be observed that CoMOLS/D has the best performance when $\frac{N}{2}(N = 300)$ for most of the instances.

5.4. iPBI vs. iTCH in CoMOLS/D

The aggregation function iPBI is adopted for the secondary population Q in CoMOLS/D. Nevertheless, the inverted Tchebycheff method (iTCH) [59] has been proposed very recently. Similar to iPBI, the iTCH also uses the nadir point as its reference point. In addition, the iTCH

method does not have any additional parameter. Thus it is interesting to conduct experiments on CoMOLS/D using iTCH. In this section, the CoMOLS/D (iTCH), as a variant of CoMOLS/D, is compared with the original CoMOLS/D (iPBI).

The performance of two compared algorithms, in terms of *C*-metric and HV values, are presented in Table 7. From these empirical results, we have the following observations. Overall, two algorithms are competitive with each other. In particular, CoMOLS/D (iPBI) tends to perform better over the tri-objective TSP instances while CoMOLS/D (iTCH) is slightly better on the bi-objective TSP instances. Nevertheless, both of them have very similar performance on mQAP instances.

6. Conclusion

In this paper, we propose a coevolutionary algorithm with local search in MOEA/D framework for combinatorial multiobjective optimization problems. The proposed CoMOLS/D is compared with four classical decomposition based and two non-decomposition based approaches on two sets of test instances (mTSP and mQAP). The experimental results show that the overall performance of CoMOLS/D outperforms all the compared algorithms. The sensitivity analyses of the penalty parameter and the first population size are also investigated. Moreover, an alternative inverted decomposition method (iTCH) is compared with inverted PBI in CoMOLS/D framework. The feature research directions include the extension of CoMOLS/D for the combinatorial many-objective optimization problems and applications on real-world problems.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant 61732006, 61300159, 61876185 and 61473241, by the Natural Science Foundation of Jiangsu Province of China under grant SBK2018022017, by China Postdoctoral Science Foundation under grant 2015M571751 and by Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (NO. CAAC-ITRB-201703).

References

[1] V.A. Shim, K.C. Tan, C.Y. Cheong, A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem, IEEE Trans. Syst., Man, Cybern., Part C 42 (5) (2012) 682–691.
[2] A. Gaspar-Cunha, A multi-objective evolutionary algorithm for solving traveling salesman problems: application to the design of polymer extruders, in: Adaptive and Natural Computing Algorithms, Springer, 2005, pp. 189–193.
[3] W. Peng, Q. Zhang, H. Li, Comparison between Moea/d and Nsga-Ii on the Multi-Objective Travelling Salesman Problem, Multi-objective memetic algorithms, 2009, pp. 309–324.
[4] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover, 1998.
[5] K. Tan, Y. Chew, L. Lee, A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows, Comput. Optim. Appl. 34 (1) (2006) 115–151.
[6] J. Arroyo, V. Armentano, Genetic local search for multi-objective flowshop scheduling problems, Eur. J. Oper. Res. 167 (3) (2005) 717–738.
[7] H. Sato, H.E. Aguirre, K. Tanaka, Local dominance and local recombination in MOEAs on 0/1 multiobjective knapsack problems, Eur. J. Oper. Res. 181 (3) (2007) 1708–1723.
[8] X. Cai, O. Wei, A hybrid of decomposition and domination based evolutionary algorithm for multi-objective software next release problem, in: 10th IEEE International Conference on Control and Automation, 2013.
[9] J.J. Durillo, Y. Zhang, E. Alba, M. Harman, A.J. Nebro, A study of the bi-objective next release problem, Empir. Softw Eng. 16 (1) (2011) 29–60.
[10] A. Alsheddy, E.E.K. Tsang, Guided pareto local search based frameworks for biobjective optimization, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, 2010, pp. 1–8.
[11] E. Ulungu, J. Teghem, P. Fortemps, D. Tuyttens, MOSA method: a tool for solving multiobjective combinatorial optimization problems, J. Multi-Criteria Decis. Anal. 8 (4) (1999) 221–236.
[12] Y.-C. Liang, M.-H. Lo, Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm, J. Heuristics 16 (3) (2010) 511–535.
[13] C. Garca-Martnez, O. Cordn, F. Herrera, A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP, Eur. J. Oper. Res. 180 (1) (2007) 116–148.
[14] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: Amosa, IEEE Trans. Evol. Comput. 12 (3) (2008) 269–283.
[15] L. Ke, Q. Zhang, R. Battiti, Hybridization of decomposition and local search for multiobjective optimization, IEEE Trans. Cybern. 44 (10) (2014) 1808–1820.
[16] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, IEEE Trans. Evol. Comput. 19 (2) (2015) 264–283.
[17] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[18] H. Ishibuchi, N. Akedo, Y. Nojima, A study on the specification of a scalarizing function in moea/d for many-objective knapsack problems, in: International Conference on Learning and Intelligent Optimization, Springer, 2013, pp. 231–246.
[19] X. Cai, Z. Yang, Z. Fan, Q. Zhang, Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (9) (2017) 2824–2837.
[20] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston, 1999.
[21] T. Lust, J. Teghem, Two-phase pareto local search for the biobjective traveling salesman problem, J. Heuristics 16 (3) (2010) 475–510.
[22] T. Lust, A. Jaszkiewicz, Speed-up techniques for solving large-scale biobjective tsp, Comput. Oper. Res. 37 (3) (2010) 521–533.
[23] J. Dubois-Lacoste, M. Lpez-Ibez, T. Sttzle, A hybrid tp+ pls algorithm for bi-objective flow-shop scheduling problems, Comput. Oper. Res. 38 (8) (2011) 1219–1236.
[24] 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.
[25] L. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm, IEEE Trans. Evol. Comput. 20 (3) (2016) 475–480.
[26] H. Sato, Analysis of inverted pbi and comparison with other scalarizing functions in decomposition based moeas, J. Heuristics 21 (6) (2015) 819–849.
[27] K. Li, S. Kwong, K. Deb, A dual-population paradigm for evolutionary multiobjective optimization, Inf. Sci. 309 (2015) 50–72.
[28] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, L. Jiao, On the use of two reference points in decomposition based multiobjective evolutionary algorithms, Swarm Evol. Comput. 34 (2017) 89–102.
[29] 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.
[30] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired co-evolutionary algorithms using weight vectors, Eur. J. Oper. Res. 243 (2) (2015) 423–441.
[31] M. Li, X. Yao, What Weights Work for You? Adapting Weights for Any Pareto Front Shape in Decomposition-Based Evolutionary Multi-Objective Optimisation, 2017. arXiv preprint arXiv:1709.02679.
[32] X. Cai, H. Sun, Q. Zhang, Y. Huang, A grid weighted sum pareto local search for combinatorial multi and many-objective optimization, IEEE Trans. Cybern. 99 (2018) 1–13.
[33] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.
[34] P.R. Ehrlich, P.H. Raven, Butterflies and plants: a study in coevolution, Evolution 18 (4) (1964) 586–608.
[35] J. Paredis, Coevolutionary computation, Artif. Life 2 (4) (1995) 355–375.
[36] L.M. Antonio, C.A.C. Coello, Coevolutionary multiobjective evolutionary algorithms: survey of the state-of-the-art, IEEE Trans. Evol. Comput. 22 (6) (2018) 851–865.
[37] A. Carlos, A. Coello, G.B. Lamont, V. Van, in: C. Coello, G.B. Lamont, A. David, V. Veldhuizen (Eds.), Evolutionary Algorithms for Solving Multi-Objective Problems, Carlos, Springer, 2007, p. 800.
[38] L.M. Antonio, C.A.C. Coello, A non-cooperative game for faster convergence in cooperative coevolution for multi-objective optimization, in: CEC, 2015, pp. 109–116.
[39] M. Laumanns, G. Rudolph, H.-P. Schwefel, A spatial predator-prey approach to multi-objective optimization: a preliminary study, in: International Conference on Parallel Problem Solving from Nature, Springer, 1998, pp. 241–249.
[40] M.d.A.C. e Silva, L. d. S. Coelho, L. Lebensztajn, Multiobjective biogeography-based optimization based on predator-prey approach, IEEE Trans. Magn. 48 (2) (2012) 951–954.
[41] J. Wang, W. Zhang, J. Zhang, Cooperative differential evolution with multiple populations for multiobjective optimization, IEEE Trans. Cybern. 46 (12) (2016) 2848–2861.
[42] L.M. Antonio, C.A.C. Coello, Indicator-based cooperative coevolution for multi-objective optimization, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 991–998.
[43] M. Gong, H. Li, E. Luo, J. Liu, J. Liu, A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing, IEEE Trans. Evol. Comput. 21 (2) (2017) 234–248.
[44] M. Li, S. Yang, X. Liu, Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 645–665.
[45] R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley Professional, 2011.

[46] G. Reinelt, Tspliba traveling salesman problem library, ORSA J. Comput. 3 (4) (1991) 376–384.
[47] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60.
[48] J.D. Knowles, D. Corne, Instance generators and test suites for the multiobjective quadratic assignment problem, in: EMO, Springer, 2003, pp. 295–310.
[49] J.D. Knowles, D. Corne, Towards landscape analyses to inform the design of hybrid local search for the multiobjective quadratic assignment problem, HIS 87 (2002) 271–279.
[50] 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.
[51] P.A. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188.
[52] M. Li, X. Yao, Quality evaluation of solution sets in multiobjective optimisation: a survey, ACM Comput. Surv. 52 (2) (2019) 26.
[53] X. Cai, H. Sun, Z. Fan, A diversity indicator based on reference vectors for many-objective optimization, Inf. Sci. 430 (2018) 467–486.
[54] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.
[55] . Taillard, Robust taboo search for the quadratic assignment problem, Parallel Comput. 17 (45) (1991) 443–455.
[56] A.J. Nebro, J.J. Durillo, M. Vergne, Redesigning the jmetal multi-objective optimization framework, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, 2015, pp. 1093–1100.
[57] K. Deb, M. Mohan, S. Mishra, Evaluating the -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions, Evol. Comput. 13 (4) (2005) 501–525.
[58] E. Zitzler, S. Knzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842.
[59] S. Jiang, S. Yang, An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts, EEE Trans. Cybern. 46 (2) (2016) 421–437.