COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 05/06 pp 59−70
Volume 3, Number 1, March 2023
DOI: 10.23919/CSMS.2022.0024

A Coevolutionary Algorithm for Many-Objective Optimization Problems with Independent and Harmonious Objectives

Fangqing Gu*, Haosen Liu, and Hailin Liu

Abstract: Evolutionary algorithm is an effective strategy for solving many-objective optimization problems. At present, most evolutionary many-objective algorithms are designed for solving many-objective optimization problems where the objectives conflict with each other. In some cases, however, the objectives are not always in conflict. It consists of multiple independent objective subsets and the relationship between objectives is unknown in advance. The classical evolutionary many-objective algorithms may not be able to effectively solve such problems. Accordingly, we propose an objective set decomposition strategy based on the partial set covering model. It decomposes the objectives into a collection of objective subsets to preserve the nondominance relationship as much as possible. An optimization subproblem is defined on each objective subset. A coevolutionary algorithm is presented to optimize all subproblems simultaneously, in which a nondominance ranking is presented to interact information among these sub-populations. The proposed algorithm is compared with five popular many-objective evolutionary algorithms and four objective set decomposition based evolutionary algorithms on a series of test problems. Numerical experiments demonstrate that the proposed algorithm can achieve promising results for the many-objective optimization problems with independent and harmonious objectives.

Key words: many-objective optimization; decomposition; objective conflict; evolutionary algorithm; set covering model

1 Introduction

Researchers have proposed various evolutionary multi-objective optimization algorithms (EMOs) during the last decades[1−3]. However, the efficiency of these EMOs will decrease for problems with more than three objectives[4]. Such problems are known as many-objective optimization problems (MaOPs)[5, 6]. To alleviate this issue, researchers have developed various many-objective evolutionary optimization algorithms (MaOEAs)[7]. These algorithms can be broadly classified into three categories, i.e., Pareto-based algorithms[8−11], decomposition-based algorithms[12−14], and indicator-based algorithms[15−18]. Generally speaking, these EMOs are designed for solving many/multi-objective optimization problems whose objectives are conflicted.

However, the objectives are not always conflicted in practice. Some objectives may be harmonious or independent. As an example, Fig. 1 plots some non-dominated solutions in parallel coordinate graphs. Each line of Fig. 1 stands for a non-dominated solution. The x-axis stands for the objectives, while the y-axis stands for the values of the objective function of these non-dominated solutions. From Fig. 1 we can see that objectives f_5 and f_6 are harmonious. One can group f_5 and f_6 in a new compound objective[19], or remove one of f_5 and f_6 from the objective set by using objective reduction method[20, 21]. Objective reduction[22−24] and objective extraction[25, 26] are the representatives of such approaches which have demonstrated effectiveness in solving MaOPs with redundant or correlated

• Fangqing Gu, Haosen Liu, and Hailin Liu are with the School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou 510520, China. E-mail: fqgu@gdut.edu.cn.
∗ To whom correspondence should be addressed.
※ This article was recommended by Associate Editor Rui Wang.
Manuscript received: 2022-09-08; revised: 2022-10-04; accepted: 2022-11-06

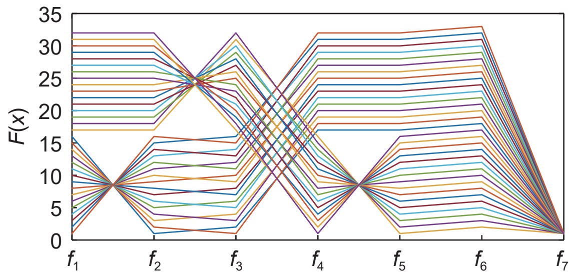


Fig. 1 A set of non-dominated solutions of an example.

objectives. Objective reduction methods improve the efficiency of EMOs by reducing the number of objectives. Nevertheless, it is a little powerless for these optimization problems with multiple independent objective subsets.

MaOP consists of multiple independent objective subsets in the case that the Pareto solution set (PS) of the original problem can be covered by the PSs of several subproblems with these objective subsets. As shown in Fig. 1, we can easily notice that the PS of the original problem consists of the PSs of the subproblem with objectives $\{f_1, f_2\}$ and the subproblem with objectives $\{f_3, f_4\}$. That is, we can obtain the PS of the original MaOP by solving two subproblems $\{f_1, f_2\}$ and $\{f_3, f_4\}$. The relationship between objectives is unknown in advance. Thus, it is very important to identify the relationship between the objectives[27].

Nowadays, some research attempts to decompose an MaOP into several subproblems and then optimize these subproblems independently. For example, an adaptive divide-and-conquer method[28] has been proposed which uses the Kendall *K* method to measure the correlation among the objectives. The objectives are grouped into some subsets by statistical analysis, and each objective subset owns a sub-population[29]. Accordingly, the population is divided into some sub-populations. After division, each sub-population evolves independently and repeats this step in each generation. Reference [30] investigated three different strategies to decompose the objective set, i.e., random, fixed, and shift. However, the above methods do not adequately consider the conflict between objectives, which may group the conflict or non-conflict objectives in the same subset. Based on the consideration of conflict information, an objective set decomposition method was proposed in Ref. [30]. It utilizes the Pearson-correlation coefficient to measure the degree of conflict between objects. The offsprings are selected by NSGA-II according to the decomposed objective subsets. These MaOEAs show their great performance in solving MaOPs with independent objective subsets. Nevertheless, these algorithms involve a predetermined number of objective subsets to decompose the original objective set equally. This may lead to the algorithm not being able to solve MaOPs with independent objective subsets with different numbers of objectives. In addition, the evolution of the population does not make full use of the information interaction between different objective subsets.

Consequently, this paper proposes an adaptive objective set decomposition strategy based on the partial set covering model. It is intended to find a minimum size sub-collection of the objective subset to maintain the nondominance relationship of the solutions. After that, we define a subproblem for each objective subset and propose a coevolutionary algorithm based on the decomposed objective subsets (DOS-CEA). The algorithm uses a novel selection operation that considers other objective subsets while considering the current objective subset. Due to the lack of good expression of the features of MaOPs with independent objective subsets, a series of test problems are constructed. Numerical experiments on the existing test problems and newly proposed test problems show the efficiency of the proposed DOS-CEA. The main contributions of this paper are summarized as follows.

(1) We propose an objective set decomposition strategy based on the partial set covering model to decompose an MaOP into several subproblems and maintain the nondominance relationship of solutions as much as possible.

(2) We propose a coevolutionary algorithm for solving the many-objective optimization problems with independent and harmonious objectives.

The rest of this paper is organized as follows. We describe the proposed objective set decomposition strategy based on a partial set covering model in Section 2. Section 3 describes the proposed DOS-CEA. In Section 4, we compare the proposed DOS-CEA with four objective set decomposition based methods and five popular MaOEAs on the existing test problems and constructed test problems. The paper ends in Section 5 by presenting some conclusions.

2 Proposed Objective Set Decomposition Strategy

2.1 Representing the objectives' relationships by using a partial set covering model

An MaOP can be formulated as follows:

$$\min_{\boldsymbol{x}\in\boldsymbol{\Omega}} \ \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ represents the decision variables and n is the dimension of the $\boldsymbol{x}$. $\boldsymbol{\Omega}$ represents the decision space. $f_i : \boldsymbol{\Omega} \rightarrow \mathbb{R}$ represents the i-th objective function, and M is the total number of objectives. The objective set of Eq. (1) is denoted as a set $\mathcal{F} = \{f_1, f_2, \ldots, f_M\}$.

A solution $\boldsymbol{x}^1$ is dominated by solution $\boldsymbol{x}^2$ in regard to objective subset $\mathcal{F}_k (\subseteq \mathcal{F})$ if and only if for any $f_i \in \mathcal{F}_k$, such that $f_i(\boldsymbol{x}^1) \leqslant f_i(\boldsymbol{x}^2)$ and there exists $f_i \in \mathcal{F}_k$ satisfing $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$, denoted as $\boldsymbol{x}^1 \prec_{\mathcal{F}_k} \boldsymbol{x}^2$. $\boldsymbol{x}^* \in \boldsymbol{\Omega}$ is known as a Pareto optimal solution with respect to $\mathcal{F}_k$ if no solution $\boldsymbol{x} \in \boldsymbol{\Omega}$ such that $\boldsymbol{x} \prec_{\mathcal{F}_k} \boldsymbol{x}^*$. All Pareto optimal solutions compose the Pareto optimal solution set (PS), denoted as $E(\mathcal{F}_k, \boldsymbol{\Omega})$. For an objective subset $\mathcal{F}_k \subset \mathcal{F}$, we have shown that $E(\mathcal{F}_k, \boldsymbol{\Omega})$ may not be a subset of $E(\mathcal{F}, \boldsymbol{\Omega})$[25]. Therefore, we propose a concept of non-dominant hold set for describing the relationship of objectives. For any $\boldsymbol{x}^* \in E(\mathcal{F}, \boldsymbol{\Omega})$, if no solution $\boldsymbol{x} \in \boldsymbol{\Omega}$ such that $\boldsymbol{x} \prec_{\mathcal{F}_k} \boldsymbol{x}^*$, the nondominance relationship of $\boldsymbol{x}^*$ is maintained. The set of all the solutions whose nondominance relationship is maintained concerning $\mathcal{F}_k$ is called a non-dominant hold set of $\mathcal{F}_k$ and also denoted as $E(\mathcal{F}_k, \boldsymbol{\Omega})$§. For any two objective subsets $\mathcal{F}_k, \mathcal{F}_j \subseteq \mathcal{F}$, if $\mathcal{F}_k \subseteq \mathcal{F}_j$, we have $E(\mathcal{F}_k, \boldsymbol{\Omega}) \subseteq E(\mathcal{F}_j, \boldsymbol{\Omega})$. In addition, we provide three definitions of the relationship between the objectives.

Definition 1: If $E(\mathcal{F}_k, \boldsymbol{\Omega}) \cup E(\mathcal{F}_j, \boldsymbol{\Omega}) = E(\mathcal{F}_k \cup \mathcal{F}_j, \boldsymbol{\Omega})$, then $\mathcal{F}_k$ and $\mathcal{F}_j$ are independent.

Definition 2: If $E(\mathcal{F}_k, \boldsymbol{\Omega}) = E(\mathcal{F}_j, \boldsymbol{\Omega}) = E(\mathcal{F}_k \cup \mathcal{F}_j, \boldsymbol{\Omega})$, then $\mathcal{F}_k$ and $\mathcal{F}_j$ are harmonious.

Definition 3: If $E(\mathcal{F}_k, \boldsymbol{\Omega}) \cup E(\mathcal{F}_j, \boldsymbol{\Omega}) \subset E(\mathcal{F}_k \cup \mathcal{F}_j, \boldsymbol{\Omega})$, then $\mathcal{F}_k$ and $\mathcal{F}_j$ are conflicted.

Let $\mathcal{S} = \{S_1, S_2, \ldots, S_{\mathcal{M}}\}$ be a collection of subsets of $E(\mathcal{F}, \boldsymbol{\Omega})$, where $S_i = E(\mathcal{F}_i, \boldsymbol{\Omega})$ is the non-dominant hold set of $\mathcal{F}_i \subset \mathcal{F}$ and $\mathcal{M}$ is the number of the subsets. Obviously, we have $S_i \subseteq E(\mathcal{F}, \boldsymbol{\Omega})$ and $\cup_{i=1}^{\mathcal{M}} S_i = E(\mathcal{F}, \boldsymbol{\Omega})$. If we find a sub-collection $\mathcal{T} \subseteq \mathcal{S}$, such that every element of $E(\mathcal{F}, \boldsymbol{\Omega})$ belongs to at least one subset in $\mathcal{T}$, then Eq. (1) can be decomposed into some subproblems with objectives $\mathcal{F}_i$ corresponding to $S_i \in \mathcal{T}$. Therefore, we can obtain $E(\mathcal{F}, \boldsymbol{\Omega})$ by solving these subproblems with these objective subsets.

Currently, most evolutionary algorithms work well for multi-objective optimization problems. Thus, we limit the number of objectives of a subproblem smaller than four, i.e., $|\mathcal{F}_i| < 4$. This brings about a consequence that $\cup_{|\mathcal{F}_i|<4} S_i$ may not be able to cover all elements of $E(\mathcal{F}, \boldsymbol{\Omega})$. Objective decomposition intends to find minimum size sub-collection of objective subsets such that the PS of the original MaOP is covered by that of these subproblems with these objective subsets as much as possible. Accordingly, the objective decomposition can be modeled as a partial set covering problem.

Since $E(\mathcal{F}, \boldsymbol{\Omega})$ is unknown in advance, we can only decompose the objective set according to the current non-dominated solutions and periodically re-decompose the objective set. Suppose X is a non-dominated solution set of Eq. (1). Analogously, $S_i = E(\mathcal{F}_i, X)$ is the non-dominated solution set in regard to objective set $\mathcal{F}_i$. Objective set decomposition strategy decomposes Eq. (1) into several subproblems aiming to maintain the nondominance relationship of X as much as possible. That is to find a minimum size sub-collection $\mathcal{T} \subseteq \mathcal{S}$, such that $|\{\boldsymbol{x} \in X : \boldsymbol{x} \in \cup S_i, S_i \in \mathcal{T}\}| \geqslant qN'$, where q is a constant between 0 and 1, $N' = |\cup S_i|$ with $|\mathcal{F}_i| < 4$. It is a partial set covering problem.

2.2 Proposed greedy strategy for the partial set covering problem

It is well-known that the partial set covering problem is an NP-hard problem. In this paper, we propose a greedy strategy for objective set decomposition based on the partial set covering problem. Algorithm 1

Algorithm 1 Greedy strategy for partial set covering problem

Require:

(1) A non-dominated solution set X;

(2) A real number $0 < \epsilon < 1$;

(3) Original objective vectors $\boldsymbol{F}(\boldsymbol{x}), \boldsymbol{x} \in X$.

Ensure:

A collection of objective subsets $\mathcal{R}$.

Initialization:

Compute $\mathcal{S} \leftarrow \{S_i\}$, where $S_i = E(\mathcal{F}_i, X)$ with $|\mathcal{F}_i| = 2, 3$.

$\mathcal{R} \leftarrow \varnothing$, $q' \leftarrow 1$, $X' \leftarrow \varnothing$, $N' \leftarrow |\cup S_i|$.

while $q' > \epsilon$ & $\mathcal{S} \neq \varnothing$ **do**

$k \leftarrow \arg\max_i |S_i|, S_i \in \mathcal{S}$.

$\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{F}_k$, $\mathcal{S} \leftarrow \mathcal{S} \setminus S_k$.

Remove those elements covered by S_k from each $S_i \in \mathcal{S}$.

Add S_k into the covered solution set $X' \leftarrow X' + S_k$, $q' \leftarrow \frac{|X'|}{N'}$.

end while

§ The remaining parts of this paper, $E(\mathcal{F}_k, \boldsymbol{\Omega})$ is the non-dominant hold set of $\mathcal{F}_k$.

provides the pseudo-code of the proposed objective set decomposition strategy. Specifically, let X be the current non-dominated solutions obtained by an EMO algorithm, and ϵ is a tolerance of the elements which are uncovered. In the algorithm initialization, we identify the non-dominated solution set S_i for each objective subset $\mathcal{F}_i$ with $|\mathcal{F}_i| < 4$. X' is the covered solution set and $\mathcal{R}$ is the selected objective subsets. In the loop of the proposed, we find the subset S_k which has the largest size. That is, it can cover the most uncovered elements of X. And then add $\mathcal{F}_k$ into the selected collection of the objective subset, i.e., $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{F}_k$, and remove the solution subset S_k from the collection of the solution subset $\mathcal{S} \leftarrow \mathcal{S} \setminus S_k$. Those elements which have been covered by S_k have to be removed from each set. Finally, we output a sub-collection $\mathcal{R}$ of the objective subsets.

3 Proposed Coevolutionary Algorithm

3.1 Redefinition subproblems

For an objective subset $\mathcal{F}_k \subset \mathcal{F}$, we have shown that the non-dominated solutions of $\mathcal{F}_k$, i.e., $E(\mathcal{F}_k, \boldsymbol{\Omega})$ may not be that of the original optimization problem. Thus, we can not select the next generation population only according to objective subset $\mathcal{F}_k$ in the evolution process. Therefore, after the objective set is decomposed into a collection of objective subset $\mathcal{R} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_s\}$, an optimization subproblem is redefined on each objective subset $\mathcal{F}_k$. The k-th subproblem with respect to objective subset $\mathcal{F}_k$ is given as

$$\min_{\boldsymbol{x} \in \boldsymbol{\Omega}} G_k(\boldsymbol{x}) = \left(g_k^1(\boldsymbol{x}), g_k^2(\boldsymbol{x}), \ldots, g_k^{k_h}(\boldsymbol{x})\right)^{\mathrm{T}} \quad (2)$$

where $g_k^i(\boldsymbol{x}) = f_k^i(\boldsymbol{x}) + \varepsilon \sum_{f_j \notin \mathcal{F}_k} f_j(\boldsymbol{x})$, $f_k^i(\boldsymbol{x})$ is the i-th objective function of objective subset $\mathcal{F}_k$ and $k_h = |\mathcal{F}_k|$ is the number of objectives of $\mathcal{F}_k$, $k = 1, 2, \ldots, s$. ε is a small positive constant which is set as 1×10^{-3} in this paper. We can easily show that a dominated solution of Eq. (1) must be a dominated solution of Eq. (2). Furthermore, a non-dominated solution of Eq. (2) is one non-dominated solution of Eq. (1).

3.2 Framework of the proposed coevolutionary algorithm

In this paper, a coevolutionary algorithm based on the objective set decomposition is put forward. Algorithm 2 shows the general framework of the proposed co-evolutionary algorithm. First, we randomly sample N solutions in the decision space and compute the values

Algorithm 2 Proposed DOS-CEA

Require:
(1) Eq. (1);
(2) N: The population size;
(3) gen_max: Maximum value of the number of generations;
(4) T: Frequency of objective set decomposition.
Ensure:
The final population P_t.
Initialization:
Set the population $P_0 \leftarrow \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$, set $t \leftarrow 0$;
while $t <$ gen_max **do**
  $Q_t \leftarrow \varnothing$;
  for each $\boldsymbol{x} \in P_t$ **do**
    Randomly choose a solution $\boldsymbol{y}$ from P_t;
    Generate an offspring z by applying genetic operators on $\boldsymbol{x}$ and $\boldsymbol{y}$;
    $Q_t \leftarrow Q_t \cup z$.
  end for
  if $\mathrm{mod}(t, T) = 0$ **then**
    Decompose the objectives into several objective subsets $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_s$ by Algorithm 1;
    Redefine s MOPs $G_1, G_2, \ldots, G_s$ on $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_s$ by Eq. (2).
  end if
  N best solutions are selected from $P_t \cup Q_t$ by Algorithm 3 to form P_{t+1}, $t \leftarrow t+1$.
end while

of the objective function. These solutions form the initial population P_0. In the proposed algorithm, we decompose the objective set into several objective subsets every T generations by Algorithm 1. Then, an MaOP is decomposed into s subproblems and redefined as $G_1, G_2, \ldots, G_s$ by Eq. (2). Each sub-problem G_k is optimized by a subpopulation with a size $N_k = \frac{|G_k|}{\sum_{i=1}^{s} |G_i|} N$, $k = 1, 2, \ldots, s$. After that, we select N_k best solutions according to G_k to update P_{t+1}^k by Algorithm 3.

3.3 Selection mechanism

The information interaction among the sub-populations selected by multiple subproblems plays an important role. When updating sub-population P_{t+1}^k, we prefer these non-dominated solutions according to current subproblem G_k, which are also non-dominated according to other subproblems. Therefore, a non-dominated rank is presented to evaluate the performance of the solutions. The non-dominated rank ND_k of $\boldsymbol{x}_i$ according to the subproblem G_k is calculated as

Algorithm 3 Selection mechanism

Require:

(1) A series of subproblems $G_k, k = 1,2,\ldots,s$;

(2) Population $P_t \cup Q_t$.

Ensure:

The population of next generation P_{t+1}.

for each $G_k, k = 1,2,\ldots,s$ **do**

$P_t \cup Q_t$ is divided into $\{\text{Fr}_1, \text{Fr}_2, \ldots\}$ according to G_k by using non-dominated sorting, $P^k_{t+1} = \varnothing$, $i = 1$;

while $|P^k_{t+1}| + |\text{Fr}_i| < N_j$ **do**

$P^k_{t+1} \leftarrow P^k_{t+1} \cup \text{Fr}_i$, $i \leftarrow i+1$;

end while

Use non-dominated rank sorting, i.e., Eq. (3), to divide solutions in Fr_{i+1} into $\{R_1, R_2, \ldots\}$ according to G_k, $h = 1$;

while $|P^k_{t+1}| + |R_h| < N_j$ **do**

$P^k_{t+1} \leftarrow P^k_{t+1} \cup R_h$, $h \leftarrow h+1$;

end while

Calculate each solution's crowding distance in R_{h+1} and descending sort as CD;

$P^k_{t+1} \leftarrow P^k_{t+1} \cup \text{CD}[1 : N_j - |P^k_{t+1}|]$;

end for

$P_{t+1} \leftarrow \cup^s_{k=1} P^k_{t+1}$.

$$\text{ND}_k(\boldsymbol{x}_i) = \frac{\sum_{j=1, j\neq k}^{j=s} I(G_j(\boldsymbol{x}_i))}{s-1} \tag{3}$$

$I(G_j(\boldsymbol{x}_i))$ is given as follows:

$$I(G_j(\boldsymbol{x}_i)) = \begin{cases} 1, & \boldsymbol{x}_i \text{ is a non-dominated} \\ & \text{solution according to } G_j; \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The non-dominated rank sorting divides the population into $\{R_1, R_2, \ldots\}$ based on ND_k descending order for subproblem G_k.

The pseudo-code of the selection mechanism in the t-th generation is provided in Algorithm 3. For subproblem G_k, we select N_k solutions following three criteria, i.e., (1) non-dominated sorting, (2) non-dominated rank sorting, and (3) crowding distance sorting, in order. From the selection mechanism, the solutions which are non-dominated according to most subproblems are easier to retain. This is beneficial to the information interaction between these sub-populations.

4 Numerical Experiments

4.1 Compared algorithms

We compared the proposed DOS-CEA with nine MaOEAs to investigate the performance of the proposed algorithm. Among them, four MaOEAs are objective set decomposition based methods that use the NSGA-II[8] with different objective set decomposition strategies, i.e., random[30], fixed[30], shift[30], and conflict degree[31], called NSGA-II-random, NSGA-II-fixed, NSGA-II-shift, and NSGA-II-conflict in this paper. We also consider other five popular MaOEAs, i.e., NSGA-III[32], MOEA/D[12], RVEA[33], PREA[16], and MaOEA-IGD[17]. The main ideas of these algorithms are given as follows:

- NSGA-II-random[30] uses a random strategy to decompose the objective set. In each cycle, NSGA-II-random contains two phases, i.e., the approximation phase and the decomposition phase. In the approximation phase, it uses NSGA-II to update the population according to the original objective set. In the decomposition phase, NSGA-II-random first randomly decomposes the objective set into several objective subsets equally and then updates the sub-population according to each objective subset.
- NSGA-II-fixed[30] uses a fixed strategy to decompose the objective set. Differing from the NSGA-II-random, NSGA-II-fixed sequentially decomposes the objective set equally.
- NSGA-II-shift[30] uses a shift strategy to decompose the objective set. Differing from the NSGA-II-random, NSGA-II-shift shifts the last objective in each objective subset to the next objective subset in each cycle.
- NSGA-II-conflict[30] uses the conflict degree to decompose the objective set. In the NSGA-II-conflict, the conflict degree is measured by the Person correlation coefficient. Differing from the NSGA-II-random, NSGA-II-conflict decomposes the objective set evenly and maximizes the conflict degree in each objective subset.

4.2 Test problems and performance metrics

Twenty-one MaOPs with independent and harmonious objectives are considered in investigating the performance of the proposed DOS-CEA. c-ZDT1(m)[34] which is the existing test suite with independent objective subsets is considered. We also constructed a series of new test problems called MaOPIOS1-18. The main features of these test problems are as follows:

- c-ZDT1(m) is concatenated by ZDT1 m times. In c-ZDT1(m), a pair of sequences $(f_{2i-1}, f_{2i}), i = 1,2,...,m$ is an independent objective subset. The number of objectives in each independent objective subset is equal. In this paper, we construct three test problems c-ZDT1(2), c-ZDT1(3), and c-ZDT1(4).

● The constructed test problems, i.e., MaOPIOS1-18, consist of two types of multiple independent objective subsets with the same and different numbers of objectives. In addition, the Pareto front of each objective subset has its shape, i.e., linear, nonlinear, convex, and concave. Table 1 shows the characteristics of the test problems constructed in this paper. Due to the space limit, the detailed description of MaOPIOS1-18 can be found in the supplementary document.

We use inverted generational distance (IGD)[35] and hypervolume (HV)[36] to evaluate the performance of the compared algorithm. The following is a brief description of IGD and HV:

● Inverted generational distance (IGD): The value of IGD is calculated as

$$\mathrm{IGD}(P,P^*)=\frac{\sum_{\boldsymbol{v}\in P^*}d(\boldsymbol{v},P)}{|P^*|} \tag{5}$$

where P^* is a set of points that are uniformly distributed on the Pareto front, and P is a set of solutions obtained by an algorithm. $d(\boldsymbol{v},P)$ denotes the Euclidean distance from solution $\boldsymbol{v}$ to P, i.e., $d(\boldsymbol{v},P)=\min_{\boldsymbol{u}\in P}d(\boldsymbol{v},\boldsymbol{u})$. $|P^*|$ denotes the size of set P^*. The smaller $\mathrm{IGD}(P,P^*)$ value is, the better approximation is to the Pareto front for P. For each test problem in the experiment, 100 000 points are generated by using the Das and Dennis method[37].

● Hypervolume (HV): The value of HV for a given solution set P is calculated as

$$\mathrm{HV}(P|\boldsymbol{y})=\mathrm{Vol}\left(\bigcup_{p\in P}[f_1(\boldsymbol{p}),y_1]\times[f_2(\boldsymbol{p}),y_2]\times\cdots\times[f_M(\boldsymbol{p}),y_M]\right) \tag{6}$$

where $\boldsymbol{y}=(y_1,y_2,\ldots,y_M)$ is a reference point, dominated by all Pareto solutions. $\mathrm{Vol}(\cdot)$ denotes the Lebesgue measure. The larger value of $\mathrm{HV}(P|\boldsymbol{y})$ is, the better approximation P gives to the Pareto front. For each test problem in the experiment, the reference point $\boldsymbol{y}=1.1\times(z_1^*,z_2^*,\ldots,z_M^*)$ with z_i^* is the maximum value of the i-th objective of the Pareto solutions.

4.3 Parameter settings

● **Population size**: For all compared algorithms on each test problem, the size of the population is $N=100$.

● **Termination condition**: All algorithms terminate when the maximum number of evolution generation gen_max is reached and gen_max = 1500 for all test problems.

● **Operators**: Simulated binary crossover[38] and polynomial mutation[39] are applied to generate offspring. They are also used in compared algorithms for generating offspring.

● **Special parameters**: For the proposed DOS-CEA,

Table 1 Characteristics description of each test problem.

Problem	M	m	M_i	Pareto front shape
MaOPIOS1	5	2	2, 3	Linear, linear
MaOPIOS2	5	2	2, 3	Nonlinear, nonlinear
MaOPIOS3	5	2	2, 3	Convex, convex
MaOPIOS4	5	2	2, 3	Concave, concave
MaOPIOS5	5	2	2, 3	Linear, nonlinear
MaOPIOS6	5	2	3, 2	Linear, nonlinear
MaOPIOS7	5	2	2, 3	Linear, convex
MaOPIOS8	5	2	3, 2	Linear, convex
MaOPIOS9	5	2	2, 3	Convex, concave
MaOPIOS10	5	2	3, 2	Convex, concave
MaOPIOS11	4	2	2, 2	Linear, linear
MaOPIOS12	6	2	3, 3	Linear, linear
MaOPIOS13	4	2	2, 2	Convex, convex
MaOPIOS14	6	2	3, 3	Convex, convex
MaOPIOS15	4	2	2, 2	Nonlinear, nonlinear
MaOPIOS16	6	2	3, 3	Nonlinear, nonlinear
MaOPIOS17	4	2	2, 2	Concave, concave
MaOPIOS18	6	2	3, 3	Concave, concave

Note: M represents the number of objectives, m represents the number of independent objective subsets, and M_i denotes the number of objectives in each independent objective subset sequentially.

the generation of objective set decomposition T is set to 50. For the other algorithm, the number of objective subsets is set to 2.

● **Number of runs**: All algorithms run 30 times independently on each test problem.

4.4 Results and analysis of the experiments

Table 2 lists the average values of IGD obtained by the proposed algorithm and NSGA-II-random, NSGA-II-fixed, NSGA-II-shift, and NSGA-II-conflict in the 30 independent runs on all test problems. Table 3 lists the average values of IGD obtained by the proposed algorithm and five popular MaOEAs, i.e., NSGA-III, MOEA/D, RVEA, PREA, and MaOEA-IGD. Moreover, the comparison results of mean HV values are list in Tables 4 and 5. In Tables 2−5, "+", "=", and "–" represent that the proposed DOS-CEA is inferior to, similar to, and better than other algorithms in regard to the Wilcoxon rank sum test at 5% significance level, respectively. Tables 2−5 indicate the proposed DOS-CEA is significantly superior to the other algorithms in terms of IGD and HV on these test problems. Then, we try to analyze the detailed reasons why the proposed DOS-CEA is superior to its counterparts on these test problems.

For the MaOPs whose independent objective subsets have the same number of objectives, i.e., c-ZDT1(m) and MaOPIOS11-18, the proposed DOS-CEA achieves a good performance from Table 2 to Table 5. The mean HV values obtained by NSGA-II-fixed and NSGA-II-shift on most test problems are 0, which means all the solutions obtained are dominated by the reference point $\boldsymbol{y}$. The reason for this phenomenon is that there are few combinations of objective subsets by objective set decomposition for NSGA-II-fixed and NSGA-II-shift. In this case, it is difficult for these algorithms to accurately decompose the objective set. In addition, the HV values obtained by the compared algorithms NSGA-II-conflict and NSGA-II-shift on the part of test problems are 0. Although NSGA-II-conflict and NSGA-II-shift consider more combinations of objective subsets, the decomposition of the objective set may still be incorrect. Compared to the conventional MaOEAs, the proposed DOS-CEA algorithm is also superior to these algorithms on the test problems. These algorithms have been demonstrated to be

Table 2 Comparison results of the IGD mean value of the five algorithms running 30 independently on all test problems.

Problem	IGD mean value				
	NSGA-II-conflict	NSGA-II-fixed	NSGA-II-random	NSGA-II-shift	DOS-CEA
c-ZDT1(2)	3.10×10^{0} –	2.27×10^{1} –	5.63×10^{0} –	1.78×10^{1} –	9.70×10^{-2}
c-ZDT1(3)	5.53×10^{0} –	6.93×10^{1} –	3.39×10^{0} –	3.60×10^{0} –	2.47×10^{-1}
c-ZDT1(4)	9.02×10^{0} –	3.52×10^{0} –	5.86×10^{0} –	3.15×10^{0} –	4.03×10^{-1}
MaOPIOS1	2.02×10^{0} –	6.56×10^{0} –	8.77×10^{0} –	4.51×10^{0} –	3.30×10^{-1}
MaOPIOS2	1.59×10^{0} –	5.72×10^{0} –	1.30×10^{1} –	3.97×10^{0} –	2.94×10^{-1}
MaOPIOS3	1.74×10^{0} –	5.22×10^{0} –	1.22×10^{1} –	3.98×10^{0} –	2.45×10^{-1}
MaOPIOS4	1.84×10^{0} –	4.72×10^{0} –	6.08×10^{0} –	3.40×10^{0} –	3.57×10^{-1}
MaOPIOS5	1.53×10^{0} –	5.94×10^{0} –	1.10×10^{1} –	3.72×10^{0} –	2.70×10^{-1}
MaOPIOS6	7.72×10^{1} –	2.26×10^{2} –	2.64×10^{0} –	6.81×10^{1} –	3.00×10^{-1}
MaOPIOS7	2.18×10^{0} –	6.45×10^{0} –	2.79×10^{0} –	4.20×10^{0} –	2.19×10^{-1}
MaOPIOS8	1.06×10^{2} –	2.21×10^{2} –	7.74×10^{0} –	6.15×10^{1} –	2.83×10^{-1}
MaOPIOS9	1.87×10^{0} –	5.20×10^{0} –	2.81×10^{1} –	4.18×10^{0} –	3.09×10^{-1}
MaOPIOS10	8.82×10^{1} –	2.24×10^{2} –	4.41×10^{0} –	4.50×10^{1} –	2.31×10^{-1}
MaOPIOS11	5.89×10^{1} –	1.93×10^{2} –	2.01×10^{1} –	5.67×10^{1} –	2.27×10^{-1}
MaOPIOS12	3.90×10^{1} –	2.79×10^{2} –	8.43×10^{0} –	2.36×10^{0} –	4.00×10^{-1}
MaOPIOS13	9.14×10^{1} –	2.21×10^{2} –	3.69×10^{1} –	6.60×10^{1} –	2.08×10^{-1}
MaOPIOS14	7.13×10^{1} –	2.61×10^{2} –	6.69×10^{0} –	2.08×10^{0} –	2.61×10^{-1}
MaOPIOS15	7.59×10^{1} –	2.07×10^{2} –	3.27×10^{1} –	7.36×10^{1} –	1.75×10^{-1}
MaOPIOS16	4.40×10^{1} –	2.54×10^{2} –	1.03×10^{1} –	1.66×10^{0} –	3.56×10^{-1}
MaOPIOS17	1.02×10^{2} –	2.08×10^{2} –	3.51×10^{1} –	7.09×10^{1} –	2.26×10^{-1}
MaOPIOS18	5.83×10^{1} –	2.63×10^{2} –	1.70×10^{1} –	1.85×10^{0} –	5.37×10^{-1}
+/–/=	0/21/0	0/21/0	0/21/0	0/21/0	–

Table 3 Comparison results of the IGD mean value of the six algorithms running 30 independently on all test problems.

Problem	IGD mean value					
	NSGA-III	MOEA/D	RVEA	PREA	MaOEA-IGD	DOS-CEA
c-ZDT1(2)	2.22×10^{-1} –	1.15×10^{-1} –	3.05×10^{-1} –	1.35×10^{-1} –	3.80×10^{0} –	9.70×10^{-2}
c-ZDT1(3)	5.31×10^{-1} –	2.21×10^{-1} +	4.85×10^{-1} –	3.67×10^{-1} –	2.28×10^{0} –	2.47×10^{-1}
c-ZDT1(4)	4.40×10^{0} –	4.06×10^{-1} =	7.62×10^{-1} –	7.43×10^{-1} –	3.32×10^{0} –	4.03×10^{-1}
MaOPIOS1	2.07×10^{0} –	3.38×10^{-1} =	9.46×10^{-1} –	2.99×10^{0} –	1.22×10^{0} –	3.30×10^{-1}
MaOPIOS2	1.53×10^{0} –	2.68×10^{-1} =	5.87×10^{-1} –	2.73×10^{0} –	5.88×10^{-1} –	2.94×10^{-1}
MaOPIOS3	1.50×10^{0} –	3.10×10^{-1} –	6.74×10^{-1} –	2.96×10^{0} –	4.92×10^{-1} –	2.45×10^{-1}
MaOPIOS4	1.99×10^{0} –	1.07×10^{0} –	1.05×10^{0} –	2.34×10^{0} –	1.26×10^{0} –	3.57×10^{-1}
MaOPIOS5	1.60×10^{0} –	2.98×10^{-1} =	7.96×10^{-1} –	2.46×10^{0} –	9.18×10^{-1} –	2.70×10^{-1}
MaOPIOS6	1.95×10^{0} –	3.54×10^{-1} =	9.01×10^{-1} –	2.90×10^{0} –	1.09×10^{0} –	3.00×10^{-1}
MaOPIOS7	1.84×10^{0} –	3.25×10^{-1} –	8.27×10^{-1} –	2.83×10^{0} –	8.16×10^{-1} –	2.19×10^{-1}
MaOPIOS8	1.85×10^{0} –	4.12×10^{-1} –	1.02×10^{0} –	3.07×10^{0} –	1.02×10^{0} –	2.83×10^{-1}
MaOPIOS9	1.72×10^{0} –	5.12×10^{-1} –	8.34×10^{-1} –	2.56×10^{0} –	1.06×10^{0} –	3.09×10^{-1}
MaOPIOS10	1.62×10^{0} –	5.63×10^{-1} –	6.25×10^{-1} –	2.40×10^{0} –	8.26×10^{-1} –	2.31×10^{-1}
MaOPIOS11	1.61×10^{0} –	3.55×10^{-1} –	1.22×10^{0} –	2.11×10^{0} –	1.14×10^{0} –	2.27×10^{-1}
MaOPIOS12	2.64×10^{0} –	3.93×10^{-1} =	8.74×10^{-1} –	4.47×10^{0} –	1.28×10^{0} –	4.00×10^{-1}
MaOPIOS13	1.36×10^{0} –	2.58×10^{-1} –	8.88×10^{-1} –	2.03×10^{0} –	8.15×10^{-1} –	2.08×10^{-1}
MaOPIOS14	1.95×10^{0} –	2.31×10^{-1} =	4.80×10^{-1} –	4.11×10^{0} –	3.76×10^{-1} –	2.61×10^{-1}
MaOPIOS15	1.16×10^{0} –	2.45×10^{-1} –	7.23×10^{-1} –	1.98×10^{0} –	6.79×10^{-1} –	1.75×10^{-1}
MaOPIOS16	1.78×10^{0} –	2.60×10^{-1} +	4.69×10^{-1} –	3.65×10^{0} –	5.83×10^{-1} –	3.56×10^{-1}
MaOPIOS17	1.35×10^{0} –	8.20×10^{-1} –	1.17×10^{0} –	1.72×10^{0} –	1.13×10^{0} –	2.26×10^{-1}
MaOPIOS18	2.15×10^{0} –	1.23×10^{0} –	8.75×10^{-1} –	3.15×10^{0} –	1.38×10^{0} –	5.37×10^{-1}
+/–/=	0/21/0	2/12/7	0/21/0	0/21/0	0/21/0	–

Table 4 Comparison results of the HV mean value of the five algorithms running 30 independently on all test problems.

Problem	HV mean value				
	NSGA-II-conflict	NSGA-II-fixed	NSGA-II-random	NSGA-II-shift	DOS-CEA
c-ZDT1(2)	2.25×10^{-2} –	0.00×10^{0} –	5.79×10^{-3} –	0.00×10^{0} –	4.77×10^{-1}
c-ZDT1(3)	0.00×10^{0} –	0.00×10^{0} –	5.03×10^{-5} –	0.00×10^{0} –	2.73×10^{-1}
c-ZDT1(4)	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	1.37×10^{-1}
MaOPIOS1	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	2.19×10^{-1}
MaOPIOS2	3.00×10^{-3} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	5.58×10^{-1}
MaOPIOS3	2.11×10^{-3} –	0.00×10^{0} –	6.61×10^{-6} –	0.00×10^{0} –	5.56×10^{-1}
MaOPIOS4	5.67×10^{-4} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	6.06×10^{-2}
MaOPIOS5	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	3.68×10^{-1}
MaOPIOS6	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	4.06×10^{-1}
MaOPIOS7	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	3.85×10^{-1}
MaOPIOS8	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	4.02×10^{-1}
MaOPIOS9	6.61×10^{-7} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	2.44×10^{-1}
MaOPIOS10	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	1.87×10^{-1}
MaOPIOS11	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	1.71×10^{-1}
MaOPIOS12	0.00×10^{0} –	0.00×10^{0} –	1.46×10^{-5} –	0.00×10^{0} –	3.35×10^{-1}
MaOPIOS13	0.00×10^{0} –	0.00×10^{0} –	5.04×10^{-4} –	0.00×10^{0} –	4.62×10^{-1}
MaOPIOS14	3.08×10^{-4} –	0.00×10^{0} –	4.80×10^{-3} –	7.07×10^{-4} –	7.49×10^{-1}
MaOPIOS15	0.00×10^{0} –	0.00×10^{0} –	8.08×10^{-4} –	0.00×10^{0} –	5.80×10^{-1}
MaOPIOS16	6.17×10^{-4} –	0.00×10^{0} –	4.79×10^{-4} –	2.50×10^{-3} –	6.24×10^{-1}
MaOPIOS17	0.00×10^{0} –	0.00×10^{0} –	9.27×10^{-7} –	0.00×10^{0} –	4.50×10^{-2}
MaOPIOS18	8.98×10^{-5} –	0.00×10^{0} –	0.00×10^{0} –	0.00×10^{0} –	8.53×10^{-2}
+/–/=	0/21/0	0/21/0	0/21/0	0/21/0	–

Table 5 Comparison results of the HV mean value of the six algorithms running 30 independently on all test problems.

Problem	HV mean value					
	NSGA-III	MOEA/D	RVEA	PREA	MaOEA-IGD	DOS-CEA
c-ZDT1(2)	3.26×10^{-1} –	4.58×10^{-1} –	2.60×10^{-1} –	4.14×10^{-1} –	1.35×10^{-2} –	4.77×10^{-1}
c-ZDT1(3)	1.24×10^{-1} –	2.70×10^{-1} –	1.45×10^{-1} –	1.22×10^{-1} –	4.74×10^{-3} –	2.73×10^{-1}
c-ZDT1(4)	0.00×10^{0} –	1.26×10^{-1} –	4.58×10^{-2} –	5.95×10^{-3} –	2.14×10^{-3} –	1.37×10^{-1}
MaOPIOS1	0.00×10^{0} –	1.84×10^{-1} =	3.52×10^{-3} –	0.00×10^{0} –	1.87×10^{-3} –	2.19×10^{-1}
MaOPIOS2	1.14×10^{-3} –	5.38×10^{-1} =	1.66×10^{-1} –	0.00×10^{0} –	2.15×10^{-1} –	5.58×10^{-1}
MaOPIOS3	4.62×10^{-3} –	4.28×10^{-1} –	1.42×10^{-1} –	0.00×10^{0} –	2.38×10^{-1} –	5.56×10^{-1}
MaOPIOS4	0.00×10^{0} –	7.48×10^{-3} –	1.03×10^{-4} –	0.00×10^{0} –	1.13×10^{-3} –	6.06×10^{-2}
MaOPIOS5	1.93×10^{-5} –	2.63×10^{-1} –	2.30×10^{-2} –	0.00×10^{0} –	8.14×10^{-3} –	3.68×10^{-1}
MaOPIOS6	6.68×10^{-6} –	3.17×10^{-1} –	3.91×10^{-2} –	4.09×10^{-6} –	1.65×10^{-2} –	4.06×10^{-1}
MaOPIOS7	2.46×10^{-4} –	2.49×10^{-1} –	2.39×10^{-2} –	0.00×10^{0} –	2.51×10^{-2} –	3.85×10^{-1}
MaOPIOS8	9.45×10^{-4} –	2.41×10^{-1} –	1.83×10^{-2} –	0.00×10^{0} –	1.93×10^{-2} –	4.02×10^{-1}
MaOPIOS9	4.10×10^{-4} –	1.34×10^{-1} –	1.72×10^{-2} –	0.00×10^{0} –	1.26×10^{-2} –	2.44×10^{-1}
MaOPIOS10	0.00×10^{0} –	4.95×10^{-2} –	1.68×10^{-2} –	0.00×10^{0} –	1.31×10^{-2} –	1.87×10^{-1}
MaOPIOS11	2.55×10^{-5} –	9.05×10^{-2} –	7.53×10^{-6} –	0.00×10^{0} –	8.16×10^{-5} –	1.71×10^{-1}
MaOPIOS12	0.00×10^{0} –	2.23×10^{-1} –	1.97×10^{-2} –	0.00×10^{0} –	2.21×10^{-3} –	3.35×10^{-1}
MaOPIOS13	2.89×10^{-3} –	4.05×10^{-1} –	2.98×10^{-2} –	0.00×10^{0} –	1.41×10^{-1} –	4.62×10^{-1}
MaOPIOS14	2.01×10^{-3} –	5.44×10^{-1} –	3.27×10^{-1} –	0.00×10^{0} –	2.67×10^{-1} –	7.49×10^{-1}
MaOPIOS15	1.14×10^{-2} –	5.03×10^{-1} –	9.57×10^{-2} –	0.00×10^{0} –	1.62×10^{-1} –	5.80×10^{-1}
MaOPIOS16	5.82×10^{-4} –	5.72×10^{-1} –	2.36×10^{-1} –	0.00×10^{0} –	1.88×10^{-1} –	6.24×10^{-1}
MaOPIOS17	0.00×10^{0} –	6.22×10^{-3} –	0.00×10^{0} –	0.00×10^{0} –	4.77×10^{-4} –	4.50×10^{-2}
MaOPIOS18	0.00×10^{0} –	4.94×10^{-3} –	4.80×10^{-3} –	0.00×10^{0} –	1.85×10^{-3} –	8.53×10^{-2}
+/–/=	0/21/0	0/19/2	0/21/0	0/21/0	0/21/0	–

effective in solving MaOPs. Therefore, different from the comparison objective set decomposition based MaOEAs, these conventional MaOEAs, especially MOEA/D, are competitive on test problems. The decomposition-based MaOEAs use uniform weights to aggregate all objectives. This makes it possible to find some PF fragments.

For the MaOPs whose independent objective subsets possess different numbers of objectives, i.e., MaOPIOS1-10, not surprising, the proposed DOS-CEA acquires the best performance from Table 2 to Table 5. Since the compared objective set decomposition based algorithms decompose the objective set into a predetermined number of objective subsets with the same number of objectives, it is difficult to deal with such problems. In contrast to the objective set average decomposition, DOS-CEA adaptively decomposes the objective set according to the partial set covering model. In this way, different objective subsets may have different numbers of objectives. Therefore, the objective set decomposition in DOS-CEA is more reasonable. Compared with the MaOPs with the same number of objectives in each independent objective subset, the conventional MaOEAs deteriorated in this type of problem. This can be attributed to two reasons. One is that these problems are all 5-objective test problems, which are more difficult than the previous 4-objective test problems. Another is that the numbers of each independent objective subset are different, and it requires different resources for each independent objective subset.

4.5 Further analysis

From the above experimental results, we have observed the superior performance of DOS-CEA for solving MaOPs with independent objective subsets. In order to better understand coevolution in DOS-CEA, we compare DOS-CEA with its variant. The variant is the same as DOS-CEA, except for the different selection of the next generation population in each generation. The variant uses the NSGA-II to select offspring according to each objective subset.

We show the average change of IGD values obtained by DOS-CEA and its variant on three test problems c-ZDT1(4), MaOPIOS1, and MaOPIOS9 in 30 independent runs in Fig. 2. From Fig. 2, we can see that

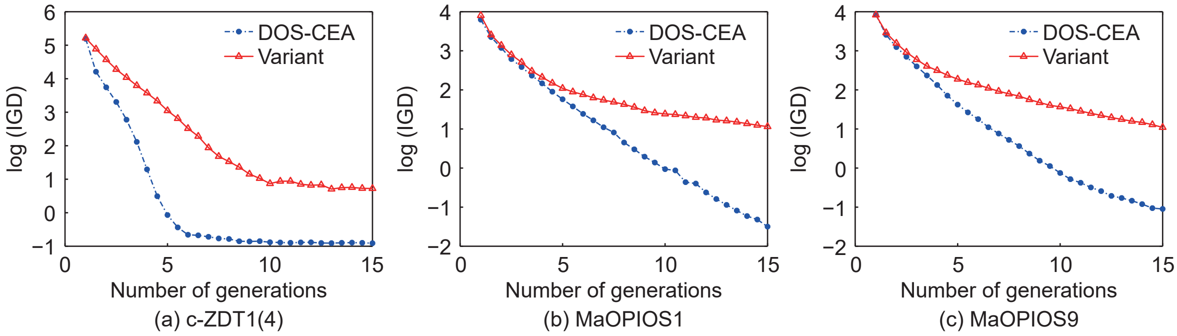


Fig. 2 Change of IGD values obtained by DOS-CEA and its variant on three test problems according to generation.

DOS-CEA performs better than its variant on three representative test problems. The poor performance of the variant can be due to the neglect of the other objectives while selecting the population according to the current objective subset. As for our proposed algorithm DOS-CEA, it redefines the subproblems according to all objective subsets. Besides, it uses a new proposed selection operator. In this way, more effective information interaction can be carried out among population selection according to each subset, which realizes coevolution.

4.6 Sensitivity of *T* in DOS-CEA

In the proposed algorithm DOS-CEA, *T* is a control parameter for objective set decomposition. In the sensitivity analysis experiments, we compare four different *T* values, i.e., 20, 30, 50, and 80, on three test problems c-ZDT1(4), MaOPIOS1, and MaOPIOS9 to analyze the influences of this parameter. All other settings are the same as used in Section 4.3. Figure 3 shows the histograms of different *T* values on three test problems in the 30 independent runs. Figure 3 indicates that the DOS-CEA with different *T* values gets a similar performance. This means the proposed DOS-CEA is not sensitive to the control parameter *T*.

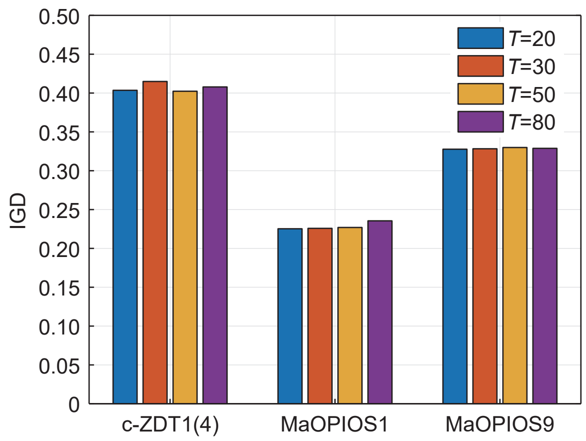


Fig. 3 IGD values obtained by DOS-CEA with different *T* values on three test problems.

5 Conclusion

We have proposed an adaptive objective set decomposition and coevolutionary algorithm (DOS-CEA) for solving MaOPs with independent and harmonious objectives. In the objective set decomposition, we proposed an adaptive objective set decomposition strategy based on a partial set covering model to cover the nondominance relationship of the solutions as much as possible. And in the selection of the population, we proposed a new selection mechanism that considers the information interaction among objective subsets. The performance of DOS-CEA has been studied on a series of test problems with independent and objective subsets. The empirical results fully demonstrate its effectiveness on MaOPs with independent objective subsets. In the future, we will apply the proposed algorithm to solve some practical problems.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 62172110), the Natural Science Foundation of Guangdong Province (Nos. 2021A1515011839 and 2022A1515010130), and the Programme of Science and Technology of Guangdong Province (No. 2021A0505110004).

References

[1] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, An efficient approach to nondominated sorting for evolutionary multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.

[2] W. Fang, L. Zhang, S. Yang, J. Sun, and X. Wu, A multiobjective evolutionary algorithm based on coordinate transformation, *IEEE Transactions on Cybernetics*,

vol. 49, no. 7, pp. 2732–2743, 2019.

[3] H. -L. Liu, F. Gu, Y. -M. Cheung, S. Xie, and J. Zhang, On solving WCDMA network planning using iterative power control scheme and evolutionary multiobjective algorithm, *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, pp. 44–52, 2014.

[4] B. Li, J. Li, K. Tang, and X. Yao, Many-objective evolutionary algorithms: A survey, *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[5] H. Bai, T. Fan, Y. Niu, and Z. Cui, Multi-UAV cooperative trajectory planning based on many-objective evolutionary algorithm, *Complex System Modeling and Simulation*, vol. 2, no. 2, pp. 130–141, 2022.

[6] L. Pan, L. Li, C. He, and K. C. Tan, A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization, *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3477–3490, 2020.

[7] K. Qiao, J. Liang, B. Qu, K. Yu, C. Yue, and H. Song, Differential evolution with level-based learning mechanism, *Complex System Modeling and Simulation*, vol. 2, no. 1, pp. 35–58, 2022.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[9] E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in *Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, Athens, Greece, 2001, pp. 95–100.

[10] L. B. Said, S. Bechikh, and K. Ghedira, The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making, *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801–818, 2010.

[11] S. Yang, M. Li, X. Liu, and J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[12] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[13] H. Liu, F. Gu, and Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[14] L. Chen, H. -L. Liu, K. C. Tan, Y. -M. Cheung, and Y. Wang, Evolutionary many-objective algorithm using decomposition-based dominance relationship, *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4129–4139, 2019.

[15] J. Bader and E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[16] J. Yuan, H. -L. Liu, F. Gu, Q. Zhang, and Z. He, Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions, *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 75–86, 2021.

[17] Y. Sun, G. G. Yen, and Z. Yi, IGD indicator-based evolutionary algorithm for many-objective optimization problems, *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2019.

[18] J. G. Falcón-Cardona and C. A. Coello, Indicator-based multi-objective evolutionary algorithms: A comprehensive survey, *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35, 2021.

[19] A. R. R. Freitas, P. J. Fleming, and F. G. Guimarães, A non-parametric harmony-based objective reduction method for many-objective optimization, in *Proc. 2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, UK, 2013, pp. 651–656.

[20] H. K. Singh, A. Isaacs, and T. Ray, A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems, *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.

[21] Y. Li, H. L. Liu, and E. D. Goodman, Hyperplane-approximation-based method for many-objective optimization problems with redundant objectives, *Evolutionary Computation*, vol. 27, no. 2, pp. 313–344, 2019.

[22] D. Brockhoff and E. Zitzler, Objective reduction in evolutionary multiobjective optimization: Theory and applications, *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.

[23] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, Objective reduction in many-objective optimization: Linear and nonlinear algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[24] Y. -M. Cheung and F. Gu, Online objective reduction for many-objective optimization problems, in *Proc. 2014 IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 1165–1171.

[25] Y. -M. Cheung, F. Gu, and H. -L. Liu, Objective extraction for many-objective optimization problems: Algorithm and test problems, *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 755–772, 2016.

[26] A. Zhou, Y. Wang, and J. Zhang, Objective extraction via fuzzy clustering in evolutionary many-objective optimization, *Information Sciences*, vol. 509, pp. 343–355, 2020.

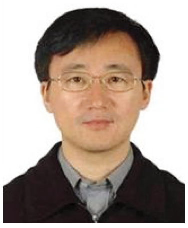
[27] X. Guo, Y. Wang, and X. Wang, An objective reduction algorithm using representative Pareto solution search for many-objective optimization problems, *Soft Computing*, vol. 20, no. 12, pp. 4881–4895, 2016.

[28] R. C. Purshouse and P. Fleming, An adaptive divide-and-conquer methodology for evolutionary multi-criterion optimization, in *Proc. the 2nd International Conference*

Evolutionary Multi-Criterion Optimization, Faro, Portugal, 2003, pp. 133–147.
[29] N. Luo, X. Li, and Q. Lin, Objective reduction for many-objective optimization problems using objective subspace extraction, *Soft Computing*, vol. 22, no. 4, pp. 1159–1173, 2018.
[30] H. Aguirre and K. Tanaka, Space partitioning with adaptive ϵ-ranking and substitute distance assignments: A comparative study on many-objective mnk-landscapes, in *Proc. the 11th Annual Conference on Genetic and Evolutionary Computation*, Montreal, Canada, 2009, pp. 547–554.
[31] A. L. Jaimes, C. A. Coello, H. Aguirre, and K. Tanaka, Objective space partitioning using conflict information for solving many-objective problems, *Information Sciences*, vol. 268, pp. 305–327, 2014.
[32] K. Deb and H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
[33] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
[34] E. Zitzler, K. Deb, and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
[35] P. Bosman and D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
[36] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[37] I. Das and J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
[38] K. Deb and R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems*, vol. 9, no. 3, pp. 115–148, 1995.
[39] K. Deb and M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

Fangqing Gu received the BS degree from Changchun University, Jilin, China in 2007, the MS degree from Guangdong University of Technology, Guangzhou, China in 2011, and the PhD degree from Hong Kong Baptist University in 2016. He is currently an associate professor at the School of Mathematics and Statistics, Guangdong University of Technology. His research interests include data mining, machine learning, and evolutionary computation.

Haosen Liu received the BS and MS degrees from Guangdong University of Technology, Guangzhou, China in 2019 and 2022, respectively. His current research interests include evolutionary computation and machine learning.



Hailin Liu received the BS degree in mathematics from Henan Normal University, Xinxiang, China, the MS degree in applied mathematics from Xidian University, Xi’an, China, and the PhD degree in control theory and engineering from South China University of Technology, Guangzhou, China. He was a postdoctoral researcher at South China University of Technology, Guangzhou, China. He is currently a professor at the School of Mathematics and Statistics, Guangdong University of Technology. His research interests include evolutionary computation and optimization, and blind source separation.