SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2021, Vol. 64 202201:1–202201:16
https://doi.org/10.1007/s11432-020-3092-y

Interactive multiobjective evolutionary algorithm based on decomposition and compression

Lu CHEN[1], Bin XIN[1,2*] & Jie CHEN[1,2]

[1] *School of Automation, Beijing Institute of Technology, Beijing* 100081*, China;*
[2] *State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing* 100081*, China*

Received 24 March 2020/Revised 7 September 2020/Accepted 2 October 2020/Published online 15 September 2021

Abstract Many real-world optimization problems involve multiple conflicting objectives. Such problems are called multiobjective optimization problems (MOPs). Typically, MOPs have a set of so-called Pareto optimal solutions rather than one unique optimal solution. To assist the decision maker (DM) in finding his/her most preferred solution, we propose an interactive multiobjective evolutionary algorithm (MOEA) called iDMOEA-εC, which utilizes the DM's preferences to compress the objective space directly and progressively for identifying the DM's preferred region. The proposed algorithm employs a state-of-the-art decomposition-based MOEA called DMOEA-εC as the search engine to search for solutions. DMOEA-εC decomposes an MOP into a series of scalar constrained subproblems using a set of evenly distributed upper bound vectors to approximate the entire Pareto front. To guide the population toward only the DM's preferred part on the Pareto front, an adaptive adjustment mechanism of the upper bound vectors and two-level feasibility rules are proposed and integrated into DMOEA-εC to control the spread of the population. To ease the DM's burden, only a small set of representative solutions is presented in each interaction to the DM, who is expected to specify a preferred one from the set. Furthermore, the proposed algorithm includes a two-stage selection procedure, allowing to elicit the DM's preferences as accurately as possible. To evaluate the performance of the proposed algorithm, it was compared with other interactive MOEAs in a series of experiments. The experimental results demonstrated the superiority of iDMOEA-εC over its competitors.

Keywords multiobjective optimization, interactive decision making, preference incorporation, decomposition, compression

Citation Chen L, Xin B, Chen J. Interactive multiobjective evolutionary algorithm based on decomposition and compression. Sci China Inf Sci, 2021, 64(10): 202201, https://doi.org/10.1007/s11432-020-3092-y

1 Introduction

Multiobjective optimization problems (MOPs) are common in many areas, including energy management [1], motion planning [2–4], resource allocation [5,6], and edge computing [7]. Conflicting objectives mean that MOPs often do not have a unique optimal solution, but rather have a set of so-called Pareto optimal solutions. The ultimate goal of multiobjective optimization is to assist the decision maker (DM) in finding the most preferred solution (MPS). Many multiobjective evolutionary algorithms (MOEAs) have been proposed to solve MOPs (e.g., see [8,9] for a comprehensive survey). MOEAs are characterized by a population-based search paradigm. The population of an MOEA evolves generation by generation to approximate the entire Pareto front of an MOP, which is very appealing for multiobjective optimization. However, the number of solutions required for approximating the entire Pareto front grows exponentially with the increase in the number of objectives, which leads to a high computational cost and imposes a heavy burden on the DM, who is required to choose the MPS among many solutions.

Several studies have proposed interactive MOEAs by incorporating the DM's preferences into MOEAs [10]. The search is concentrated only on the DM's preferred part of the Pareto front, so the computational cost is reduced in this case. Moreover, the DM can learn about the MOP to be solved during the solution process and adjust his/her preferences to refine solutions. Generally, two interaction patterns

[*] Corresponding author (email: brucebin@bit.edu.cn)

can be identified in interactive MOEAs [10]. The first involves interaction after a complete MOEA runs. This means that an MOEA runs fully between every two adjacent interactions until its stopping criteria is met to find approximate Pareto optimal solutions. The other pattern involves interaction during an MOEA runs, which means that the MOEA runs only once and pauses its evolution process multiple times to interact with the DM. The second interaction pattern is less time consuming, and the entire process is more DM-oriented [11].

Interactive MOEAs adopting the second interaction pattern do not usually get close to the Pareto front until the later period of the search process. When it is time to interact, these algorithms often provide a set of intermediate solutions for the DM. The DM is required to express his/her preferences for these solutions. Typically, preferences are expressed in the form of pairwise comparison, classifying solutions into multiple categories, or selecting the best one among them. Compared with pairwise comparison, specifying the best solution among a set of solutions requires less effort from the DM. For example, $m(m-1)/2$ comparisons would need to be made to compare m solutions pair-wisely; however, $m-1$ comparisons are enough for the DM to select the best one from m solutions. This study considers the type of preferences in the form of selecting the best solution among a set of solutions with the aim to reduce the DM's burden.

The idea of decomposing an MOP into a series of subproblems has been popular in handling MOPs [12, 13]. DMOEA-εC [13], AMOEA/D [14], and MOEA/D-AAP [15] are three state-of-the-art decomposition-based MOEAs. Incorporating the DM's preferences into these MOEAs can provide new ways of solving MOPs with the aim of obtaining the DM's preferred solutions.

For interactive MOEAs, compression is a good way of cutting down the search space and focusing the search process on the region of interest. Several interactive MOEAs that ask the DM to provide the best solution from multiple solutions utilize the DM's preferences in different ways to reduce the objective space directly or indirectly. For example, iTDEA defines a territory for each solution and uses the favorable weights of the DM's current best solution to determine a new preferred weight region [16]. Solutions whose favorable weights are located in the preferred weight region are assigned a smaller territory size to obtain denser solutions around the best one. iMOEA/D also uses the weighting vector of the best solution to renew the preferred weight region that shrinks gradually with the interactions [12]. These two algorithms compress the objective space indirectly by reducing the weight space. PI-NSGA-II-PC uses the DM's preferred solution to construct a polyhedral cone in the objective space [17]. Solutions inside the cone are considered to be more preferable than those outside the cone. While the search space is compressed progressively, the shape of the cone is not fixed, making it hard to control the size of the preferred region (i.e., preferred region in the objective space).

This paper presents an interactive decomposition-based MOEA with an ε-constraint framework, called iDMOEA-εC, which uses the DM's current best solution to decompose MOPs and compress the objective space directly. DMOEA-εC is used as a search engine to search for solutions. It decomposes an MOP into a range of scalar constrained subproblems using a set of evenly spread upper bound vectors, and is designed to approximate the entire Pareto front. An adaptive adjustment mechanism of the upper bound vectors and two-level feasibility rules are developed and incorporated into DMOEA-εC to adapt DMOEA-εC for finding the DM's preferred solutions. To reduce the DM's burden, only a small sample set of solutions is provided in every interaction for the DM, who is expected to select the one (s)he prefers most. Considering the difficulty of capturing the DM's preferences using a small number of solutions, a two-stage selection procedure is proposed to help elicit the DM's preferences for a better localization of the DM's preferred region. Owing to these characteristics, iDMOEA-εC can use the DM's preferences to approximate the DM's MPS well without imposing a heavy burden on the DM. In addition, iDMOEA-εC's computational complexity is close to that of DMOEA-εC; hence, it can assist the DM in finding the MPS within a reasonable time.

The remainder of the paper is organized as follows. Section 2 presents related concepts of MOPs and the algorithmic framework of DMOEA-εC. In Section 3, the proposed iDMOEA-εC is introduced in detail. Section 4 is devoted to the verification of the effectiveness of iDMOEA-εC through the comparison of iDMOEA-εC with several other interactive MOEAs. Further investigations on the behavior of iDMOEA-εC are provided in Section 5. Section 6 concludes this paper.

Table 1 Acronyms used in this paper

Acronym	Full name
MOP	Multiobjective optimization problem
DM	Decision maker
MPS	Most preferred solution
MOEA	Multiobjective evolutionary algorithm
iDMOEA-εC	Interactive decomposition-based MOEA with the ε-constraint framework
AMOEA/D	Adaptive MOEA based on decomposition
MOEA/D-AAP	Decomposition-based MOEA with angle-based adaptive penalty scheme
iTDEA	Interactive territory defining evolutionary algorithm
iMOEA/D	Interactive decomposition-based multiobjective evolutionary algorithm
NSGA-II	Non-dominated sorting genetic algorithm with elitism strategy
PI-NSGA-II-PC	Progressively interactive NSGA-II based on polyhedral cones
ICB-MOEA/D	Interactive classification-based MOEA/D
DRA	Dynamic resource allocation
EP	Elite population
NFE	Number of function evaluations

2 Preliminaries

This section is devoted to introducing some basic concepts of multiobjective optimization, the ε-constraint method, and the framework of DMOEA-εC. In order to help readers get the meanings of the acronyms used in this paper quickly, Table 1 presents these acronyms and their full names.

2.1 Multiobjective optimization problem

Generally, an MOP can be formulated as follows:

$$\begin{array}{ll} \min & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})) \\ \text{s.t.} & \boldsymbol{x} \in S \subset \mathbb{R}^n, \end{array} \tag{1}$$

where the decision vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ belongs to the feasible region $S \subset \mathbb{R}^n$, $f_i : S \to \mathbb{R}$ for all $i \in \{1, 2, \ldots, k\}$ are k ($k \geqslant 2$) objective functions which are to be minimized simultaneously.

A vector $\boldsymbol{u}$ is said to dominate another vector $\boldsymbol{v}$, denoted by $\boldsymbol{u} \prec \boldsymbol{v}$, if and only if $u_i \leqslant v_i, \forall i \in \{1, 2, \ldots, k\}$ and $u_j < v_j, \exists j \in \{1, 2, \ldots, k\}$. A decision vector $\boldsymbol{x}$ is said to be Pareto optimal if and only if there is no decision vector which dominates it. The corresponding objective vector is called a Pareto optimal objective vector. The set of all Pareto optimal decision vectors is denoted by E. The image of E in the objective space is called the Pareto front.

When providing preferences, it is often useful for the DM to know the ranges of the objective functions. The ideal point, denoted by $\boldsymbol{z}^* = (z_1^*, z_2^*, \ldots, z_k^*)$, is composed of the minimum value of each objective in the feasible region, i.e., $z_i^* = \min_{\boldsymbol{x} \in S} f_i(\boldsymbol{x})$ for all $i = 1, 2, \ldots, k$. The nadir point $\boldsymbol{z}^{\text{nad}} = (z_1^{\text{nad}}, z_2^{\text{nad}}, \ldots, z_k^{\text{nad}})$ consists of the maximum value of each objective in the Pareto front, i.e., $z_i^{\text{nad}} = \max_{\boldsymbol{x} \in E} f_i(\boldsymbol{x})$ for all $i = 1, 2, \ldots, k$ [13].

2.2 The ε-constraint method

In the classic ε-constraint method introduced by Haimes et al. [18], one of the objective functions is selected as the main objective which is to be minimized. The other objectives are converted into constraints. The following constrained single-objective optimization problem is formed [13, 18]:

$$\begin{array}{ll} \min & f_{\text{main}}(\boldsymbol{x}) = f_s(\boldsymbol{x}) + \rho \displaystyle\sum_{j=1}^{k} f_j(\boldsymbol{x}) \\ \text{s.t.} & \begin{cases} \widetilde{f}_j \leqslant \varepsilon_j, \ \forall j \in \{1, 2, \ldots, k\}/\{s\}, \\ \boldsymbol{x} \in S, \end{cases} \end{array} \tag{2}$$

where $\widetilde{f}_j = (f_j(\boldsymbol{x}) - z_j^*)/(z_j^{\text{nad}} - z_j^*)$ for $j = 1, 2, \ldots, k$; $0 \leqslant \boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_{s-1}, \varepsilon_{s+1}, \ldots, \varepsilon_k\} \leqslant 1$ is an upper bound vector; ρ is a very small positive number. Generally, different upper bound vectors lead to

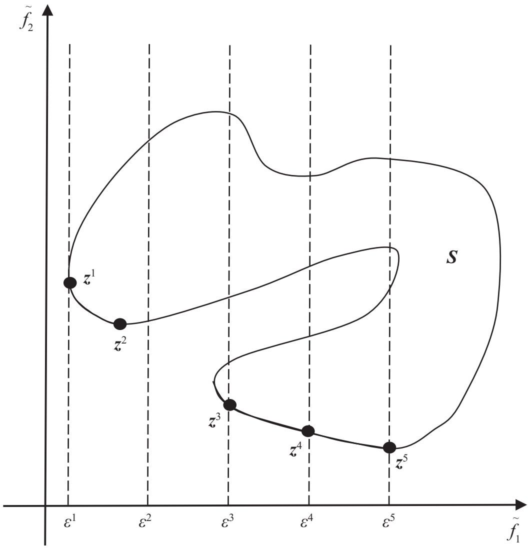

Figure 1 Different upper bounds of the ε-constraint method and corresponding Pareto optimal solutions.

different optimal solutions. It has been proved that each Pareto optimal solution of an MOP can be found by solving the ε-constraint method via altering the main objective and the upper bound vector [13, 19]. Figure 1 gives an example of a two-objective case with f_2 being the main objective. Different upper bounds and the corresponding Pareto optimal solutions (black points) are shown.

2.3 Framework of DMOEA-εC

Adopting the framework of the ε-constraint method, DMOEA-εC transforms an MOP into N constrained single-objective subproblems by using a set of evenly spread upper bound vectors $\boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2, \ldots, \boldsymbol{\varepsilon}^N$ which satisfy $0 \leqslant \varepsilon_j^i \leqslant 1, i \in \{1, 2, \ldots, N\}, j \in \{1, 2, \ldots, k\}/\{s\}$ [13]. DMOEA-εC is a population-based algorithm and each subproblem corresponds to a solution. All the subproblems are optimized collaboratively in a single run of DMOEA-εC. Each subproblem has a neighborhood denoted by $B(i)$. It consists of a set of subproblems whose upper bound vectors are close to $\boldsymbol{\varepsilon}^i$. In the following, we define some notations that will be used in DMOEA-εC:

- $G_{\max}$: The maximum allowable generations.
- $\mathrm{DRA}_{\mathrm{interval}}$: Generation interval of using the dynamic resource allocation scheme.
- IN_m: Generation interval of alternating the main objective function.
- $|\mathrm{EP}|_{\max}$: Maximum size of the external archive set EP; $|\cdot|$ means the size of a set.

The algorithmic framework of DMOEA-εC is presented in Algorithm 1. The dynamic resource allocation (DRA) scheme is to assign different amounts of computational effort to different subproblems [20]. In addition, the main objective is alternated periodically. The solution-to-subproblem matching procedure is used to match the solutions with subproblems after the subproblems are initialized or altered. The subproblem-to-solution matching procedure is to find a suitable subproblem for each newly generated solution. For every two solutions to be compared, whether they satisfy the constraints of the corresponding subproblem is considered. The following Deb's feasibility rules [21] are adopted for comparing them:

(1) Any feasible solution is preferred to any infeasible solution;

(2) For two feasible solutions, the one with better objective function value is preferred;

(3) For two infeasible solutions, the one having smaller constraint violation is preferred.

3 The proposed iDMOEA-εC

In the proposed iDMOEA-εC, the DM is allowed to specify the overall number of interactions, denoted by H. Then, H interaction stages, denoted by $h = 1, 2, \ldots, H$, are arranged at generations $G_h, h = 1, 2, \ldots, H$. At each interaction stage, iDMOEA-εC pauses for interaction and presents a small representative sample of nondominated solutions to the DM. We set $G_1 = \frac{G_{\max}}{3}$ as has been done in [16] to make the population not too far away from the Pareto front before interaction. The other interaction

Algorithm 1 Framework of DMOEA-εC

```
Input: The MOP to be solved and related parameters.
Output: An external archive population EP.
Initialize $N$ evenly spread upper bound vectors and a random population $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$; find all nondominated solutions to form EP; calculate $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\mathrm{nad}}$; set generation number gen $= 0$;
Use the solution-to-subproblem matching procedure to match solutions with subproblems;
while gen $\leqslant G_{\max}$ do
    if gen $> 0$ and mod(gen, DRA$_{\mathrm{interval}}$) $== 0$ then
        Use the dynamic resource allocation scheme to update the indices of the subproblems $I$ which will be processed in next generations;
    end if
    if gen $> 0$ and mod(gen, IN$_m$) $== 0$ then
        Alternate the main objective and form new subproblems;
        Rematch solutions with new subproblems by using the solution-to-subproblem matching procedure;
    end if
    for $i \in I$ do
        Generate a new solution $\boldsymbol{y}$ by using the reproduction operator;
        Update the approximated ideal point $\boldsymbol{z}^*$;
        Use the subproblem-to-solution matching procedure to find a subproblem $j$ for $\boldsymbol{y}$;
        Compare $\boldsymbol{y}$ with the solutions of $B(j)$ and update these solutions by using Deb's feasibility rules;
        Update EP; if $|\mathrm{EP}| > |\mathrm{EP}|_{\max}$, use the farthest-candidate approach to prune it;
        Update the approximated nadir point $\boldsymbol{z}^{\mathrm{nad}}$;
    end for
    gen = gen + 1.
end while
```

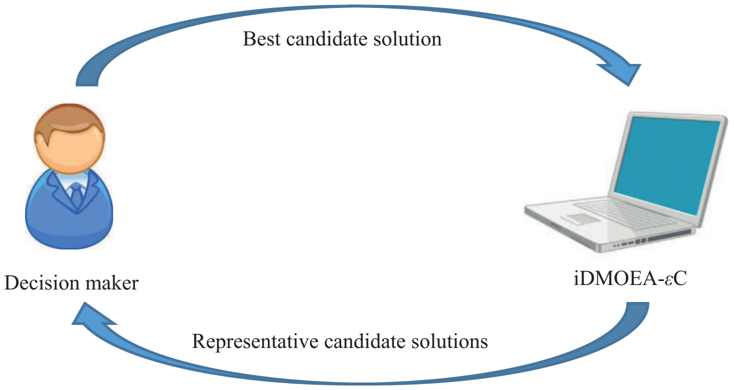

Figure 2 (Color online) The interaction between the decision maker and iDMOEA-εC.

stages are set as $G_h = \frac{G_{\max}}{3} + \frac{2G_{\max}\cdot(h-1)}{3(H-1)}$ for $h = \{2, 3, \ldots, H\}$. When gen $< G_1$, iDMOEA-εC performs as DMOEA-εC does. After then, the population evolves under the guidance of the DM's preferences.

Similar to DMOEA-εC, iDMOEA-εC also uses an archive set EP to store elite nondominated solutions. In each interaction, a representative set of solutions from EP is presented to the DM who is required to select the solution that (s)he prefers most, denoted by $\boldsymbol{z}^{b,h}$. According to this solution, a preferable upper bound vector whose components act as the upper bounds for the corresponding objectives is determined. We use $\boldsymbol{z}^{p,h+1}$ to denote this vector and define the region upper bounded by it as the preferred region. Solutions outside the preferred region are less preferred. Particularly, to obtain more and denser elite nondominated solutions in the preferred region, EP only stores nondominated solutions in the identified preferred region. The algorithmic framework of iDMOEA-εC is presented in Algorithm 2. Figure 2 shows the interaction between the decision maker and iDMOEA-εC.

3.1 Two-stage selection procedure

In reality, it is usually hard for the DM to choose the best solution from a large amount of solutions owing to the heavy burden. Thus, it is reasonable to present a representative sample of solutions to the DM. However, using a small number of solutions to represent the search space is difficult, especially in a high-dimensional space. The representative solutions can be very sparse, and it is hard for the algorithm to capture the DM's preferences accurately, which may cause the deviation from the real preferred region.

To handle this problem, a two-stage selection procedure is proposed in this paper. At each interaction stage, it uses a coarse-grained selection to determine a rough location of the DM's preferred solution and

Algorithm 2 Framework of iDMOEA-εC

```
Input: The MOP to be solved and related parameters.
Output: The solution that DM prefers most in the final interaction.
Initialize $N$ evenly spread upper bound vectors $\boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2, \dots, \boldsymbol{\varepsilon}^N$ and a random population $\{\boldsymbol{x}^1, \dots, \boldsymbol{x}^N\}$; find all nondominated solutions to form EP; calculate $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\rm nad}$; set gen = 0, $h = 0$;
Use the solution-to-subproblem matching procedure to match solutions with subproblems;
while gen $\leqslant G_{\max}$ do
    if gen > 0 and mod(gen, DRA$_{\rm interval}$) == 0 then
        Use the DRA scheme to update the indices of the subproblems $I$ to be processed in next generations;
    end if
    if gen == $G_h, \forall h \in \{1, 2, \dots, H\}$ then
        Let $h = h + 1$; the algorithm presents multiple nondominated solutions to the DM from EP, and the DM selects the most preferred one by using the two-stage selection procedure introduced in Subsection 3.1;
        if $h == H$ then
            break;
        end if
        Update the preferable upper bound vector $\boldsymbol{z}^{p,h+1}$ according to (3);
        Update EP by deleting solutions which do not dominate $\boldsymbol{z}^{p,h+1}$;
        if $h \geqslant 2$ then
            Update the initial upper bound vectors $\boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2, \dots, \boldsymbol{\varepsilon}^N$ by using (5);
        end if
    end if
    if gen > 0 and mod(gen, IN$_m$) == 0 then
        Alternate the main objective;
        if $h \geqslant 2$ then
            Update the initial upper bound vectors $\boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2, \dots, \boldsymbol{\varepsilon}^N$ by using (5);
        end if
        Rematch solutions with new subproblems by using the solution-to-subproblem matching procedure;
    end if
    for $i \in I$ do
        Generate a new decision vector $\boldsymbol{y}$ by using certain reproduction operator;
        Update the approximated ideal point $\boldsymbol{z}^*$;
        Using the subproblem-to-solution matching procedure to find a subproblem $j$ for $\boldsymbol{y}$;
        Compare the solutions of $B(j)$ with $\boldsymbol{y}$ and update them via the two-level feasibility rules[1] (see Subsection 3.4);
        Update EP; if |EP| > |EP|$_{\max}$, prune it by using the crowding distance [13];
        if gen < $G_1$ then
            Update the approximated nadir point $\boldsymbol{z}^{\rm nad}$;
        end if
    end for
    gen = gen + 1.
end while
```

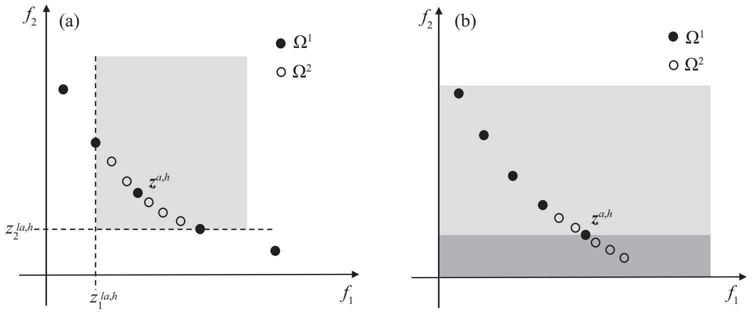

Figure 3 Examples to show how Ω^2 can be obtained according to the location of $\boldsymbol{z}^{a,h}$ in Ω^1. (a) Case 1: $\boldsymbol{z}^{a,h}$ is in the middle of Ω^1; (b) Case 2: $\boldsymbol{z}^{a,h}$ is at the boundary of Ω^1.

then a fine-grained selection to fine-tune the location. Suppose P solutions are to be selected from EP and given to the DM. At the first stage, a set of $P/2$ solutions, denoted by Ω^1, are selected from EP by clustering (e.g., k-means clustering [22]). The DM is expected to choose the best one, denoted by $\boldsymbol{z}^{a,h}$, from Ω^1.

Let $J = \{i | z_i^{a,h} = \min_{\boldsymbol{z} \in \Omega^1} z_i, i = 1, 2, \dots, k\}$. At the second stage, another set of $P/2$ solutions, denoted by Ω^2, is chosen from EP$\backslash\Omega^1$ according to J. The details are as follows.

Case 1. $J = \emptyset$. This means that for each objective i, there is at least one objective value that is smaller than $z_i^{a,h}$ in Ω^1. Let $\boldsymbol{z}^{la,h} = [z_1^{la,h}, \dots, z_k^{la,h}]$ where $z_i^{la,h} = \max\{z_i | \boldsymbol{z} \in \Omega^1, z_i < z_i^{a,h}\}$. We identify the set of all the solutions dominated by $\boldsymbol{z}^{la,h}$ from EP$\backslash\Omega^1$ and use Ω^3 to denote it. If $|\Omega^3| \geqslant P/2$, $P/2$ solutions

[1] When gen < G_1, there is no preference to follow and Deb's feasibility rules in DMOEA-εC are still used.

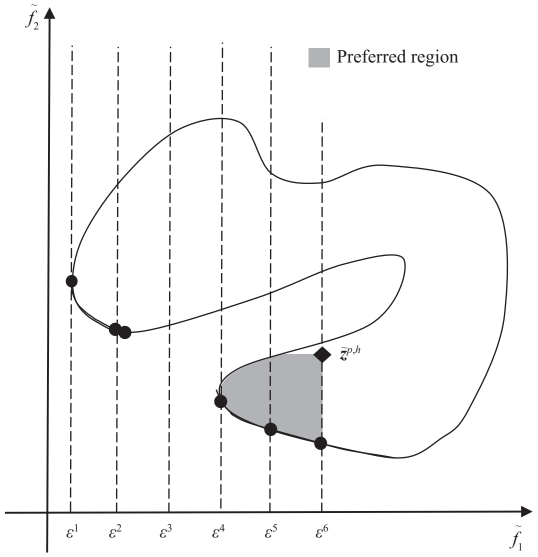

Figure 4 The preferred region and evenly spread upper bounds.

will be selected from Ω^3 by clustering. Otherwise, find the other $P/2-|\Omega^3|$ solutions closest to $\boldsymbol{z}^{a,h}$ from remaining solutions according to the Euclidean distance. Figure 3(a) displays possible locations of Ω^2 (black circles) according to $\boldsymbol{z}^{a,h}$ and Ω^1 (black points) in this case.

Case 2. $J \neq \emptyset$. This implies that $\boldsymbol{z}^{a,h}$ is at the boundary of Ω^1 and the solutions outside the range of Ω^1 may also be promising. Hence, we divide $\mathrm{EP}\backslash\Omega^1$ into two sets Ω^4 and Ω^5 where $\Omega^4 = \{\boldsymbol{z}|\boldsymbol{z} \in \mathrm{EP}\backslash\Omega^1, z_i \geqslant z_i^{a,h}, \forall i \in J\}$ and $\Omega^5 = \{\boldsymbol{z}|\boldsymbol{z} \in \mathrm{EP}\backslash\Omega^1, z_i < z_i^{a,h}, \exists i \in J\}$. According to this division, none of the solutions in Ω^5 are in Ω^1. Considering that if $\boldsymbol{z}^{a,h}$ is taken as the origin of coordinates, in the $|J|$-dimensional space determined by the objective functions associated with J, Ω^4 only locates in the first quadrant, we choose $\lfloor \frac{P}{2} \cdot \frac{1}{2^{|J|}} \rfloor$ ($\lfloor \cdot \rfloor$ means rounding down to an integer) solutions closest to $\boldsymbol{z}^{a,h}$ in the sense of the Euclidean distance from Ω^4. Meanwhile, $\frac{P}{2} - \lfloor \frac{P}{2} \cdot \frac{1}{2^{|J|}} \rfloor$ solutions are selected from Ω^5 by clustering. Figure 3(b) shows how Ω^2 is determined based on $\boldsymbol{z}^{a,h}$ in this case.

In both cases above, $P/2$ solutions are chosen from $\mathrm{EP}\backslash\Omega^1$ to form Ω^2. They together with $\boldsymbol{z}^{a,h}$ are sent to the DM who is asked to specify the current best solution $\boldsymbol{z}^{b,h}$.

3.2 Update of the preferable upper bound vector

Once the current best solution $\boldsymbol{z}^{b,h}$ has been determined by the DM, iDMOEA-εC uses it and the preferable upper bound vector $\boldsymbol{z}^{p,h}$ to calculate $\boldsymbol{z}^{p,h+1}$. Specifically, for each objective, if $z_i^{b,h} < z_i^{p,h}$, then we reduce $z_i^{p,h}$. Owing to the limited preference information on solutions, the DM's real MPS may occasionally be out of the preferred region constrained by $\boldsymbol{z}^{p,h}$ and there may be the case that $\boldsymbol{z}^{b,h}$ reaches $\boldsymbol{z}^{p,h}$ on some objectives. In this case, in order not to miss possible promising solutions, the values of $\boldsymbol{z}^{p,h}$ on these objectives are not reduced, but conversely increased. To sum up, $\forall i \in \{1,2,\ldots,k\}$, $z_i^{p,h+1}$ is determined by using the following formula:

$$z_i^{p,h+1} = \begin{cases} \alpha z_i^{p,h} + (1-\alpha) z_i^{b,h}, & z_i^{b,h} < z_i^{p,h}, \\ z_i^{p,h} + \alpha(z_i^{p,h} - z_i^*), & z_i^{b,h} = z_i^{p,h}, \end{cases} \tag{3}$$

where $\alpha \in \{0,1\}$ is the compression rate. A smaller α implies a faster contraction speed. Although some extension may be needed occasionally, in general, the preferred region is compressed gradually.

Note that $\boldsymbol{z}^{p,1}$ is set to $\boldsymbol{z}^{\mathrm{nad}}$. After the first interaction begins, $\boldsymbol{z}^{p,h+1}$ will replace $\boldsymbol{z}^{\mathrm{nad}}$ for normalizing solutions and $\boldsymbol{z}^{\mathrm{nad}}$ is not needed anymore. Let $\widetilde{\boldsymbol{z}}^{p,h+1}$ denote the normalized preferable upper bound vector. Obviously, $\widetilde{\boldsymbol{z}}^{p,h+1} = [1,1,\ldots,1]_{1\times k}$ and every upper bound ε_j^i does not exceed $\widetilde{z}_j^{p,h+1}$. In this way of normalization, the region dominating $\boldsymbol{z}^{p,h+1}$, i.e., the preferred region, is emphasized. Figure 4 displays the preferred region (marked in gray) and several evenly spread upper bounds.

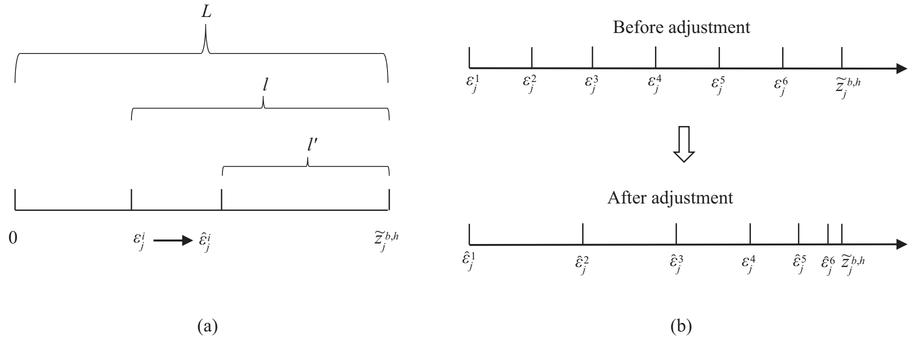

Figure 5 The adjustment of upper bound(s). (a) A single upper bound; (b) multiple upper bounds.

3.3 Adaptive adjustment of upper bound vectors

As the interaction goes on, the preferable upper bound vector $\boldsymbol{z}^{p,h}$ is getting closer and closer to the Pareto front. If evenly distributed upper bound vectors are still used, there will be a lot of subproblems which do not have feasible solutions in the preferred region. For example, in Figure 4, subproblems associated with $\boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2$ and $\boldsymbol{\varepsilon}^3$ have no feasible solutions in the preferred region. This will not only cause a waste of computational resources, but also lead to a sparse distribution of obtained solutions in the preferred region. Hence, we propose an adaptive adjustment mechanism for the initial evenly spread upper bound vectors. At the h-th ($h \geqslant 2$) interaction stage, the initial upper bound vectors are shifted towards the current normalized best solution which is defined as

$$\tilde{\boldsymbol{z}}^{b,h} = [\tilde{z}_1^{b,h}, \tilde{z}_2^{b,h}, \ldots, \tilde{z}_k^{b,h}], \quad \tilde{z}_j^{b,h} = \frac{z_j^{b,h} - z_j^*}{z_j^{\mathrm{nad}} - z_j^*}, \quad \forall j \in \{1, 2, \ldots, k\}, \tag{4}$$

and therefore the denser upper bounds near $\tilde{\boldsymbol{z}}^{b,h}$ are obtained.

Let C ($|C| = k-1$) denote the set of constraints of subproblems. For constraint j ($j \in C$), each upper bound ε_j^i which is smaller than $\tilde{z}_j^{b,h}$ is shifted towards $\tilde{z}_j^{b,h}$ to obtain a new upper bound $\hat{\varepsilon}_j^i$ by using the following equation:

$$\begin{aligned} \hat{\varepsilon}_j^i &= \tilde{z}_j^{b,h} - \tilde{z}_j^{b,h} \left(\frac{\tilde{z}_j^{b,h} - \varepsilon_j^i}{\tilde{z}_j^{b,h}} \right)^{\frac{1}{\gamma(h)}}, \\ \gamma(h) &= \frac{H-h}{H-1}, \quad h = 2, \ldots, H-1. \end{aligned} \tag{5}$$

Let $L = \tilde{z}_j^{b,h}, l = \tilde{z}_j^{b,h} - \varepsilon_j^i$, and $l' = \tilde{z}_j^{b,h} - \hat{\varepsilon}_j^i$. Then, we have

$$\frac{l'}{L} = \left(\frac{l}{L}\right)^{\frac{1}{\gamma}}, \quad 0 < l \leqslant L, \quad \gamma \in (0,1). \tag{6}$$

The geometric illustration of this equation is given in Figure 5(a).

Theorem 1. If l and l' satisfy the relation in (6), then, for a fixed γ: (1) $l' \leqslant l$ and the equal sign holds if and only if $l = L$; (2) l' strictly increases and increases faster with the growth of l.

Proof. (1) From (6), we can obtain $\frac{l'}{l} = (\frac{l}{L})^{\frac{1}{\gamma}-1}$. Since $\frac{1}{\gamma} - 1 > 0$ and $0 < \frac{l}{L} \leqslant 1$, $\frac{l'}{l}$ strictly increases with l and reaches the maximum value when $l = L$. Furthermore, $\frac{l'}{l} < 1$ when $l < L$ and $\frac{l'}{l} = 1$ when $l = L$. Thus, $l' \leqslant l$ and $l' = l$ holds only when $l = L$.

(2) Since $\frac{\mathrm{d}l'}{\mathrm{d}l} = \frac{1}{\gamma}(\frac{l}{L})^{\frac{1}{\gamma}-1} > 0$ and $\frac{\mathrm{d}^2 l'}{\mathrm{d}l^2} = \frac{1}{\gamma L}(\frac{1}{\gamma}-1)(\frac{l}{L})^{\frac{1}{\gamma}-2} > 0$, l' is strictly increasing and grows faster with the increase of l.

The first conclusion in Theorem 1 implies that $\hat{\varepsilon}_j^i$ is closer to $\tilde{z}_j^{b,h}$ than ε_j^i when $\varepsilon_j^i > 0$. When $\varepsilon_j^i = 0$, $\hat{\varepsilon}_j^i = \varepsilon_j^i = 0$, which is beneficial for the population to preserve the lower-boundary solutions of the preferred region. The second conclusion guarantees that all shifted upper bounds in the same dimension do not overlap and become denser near $\tilde{\boldsymbol{z}}^{b,h}$. Figure 5(b) shows the contrast between original upper bounds and the shifted ones.

Now we analyze the influence of γ. For a fixed l with $l < L$, $(l/L)^{1/\gamma}$ increases with γ. That is to say, the smaller γ is, the closer the shifted upper bound is to the corresponding component of $\tilde{\boldsymbol{z}}^{b,h}$.

Table 2 Time complexity analysis of iDMOEA-εC

Procedure	Time complexity
Two-stage selection	$O(P \cdot N)$
Adaptive adjustment of upper bound vectors	$O(k \cdot N)$
Search engine (DMOEA-εC)	$O(k \cdot N^2)$

This indicates that more subproblems can have the chance of owning feasible solutions in the preferred region with the decrease of γ. In (5), γ decreases gradually with the number of interactions h, which is beneficial for maintaining a relatively large coverage of the search space in the early interaction stages and obtaining more and more solutions in the preferred region in the later interaction stages.

3.4 Two-level feasibility rules

Once a new decision vector $\boldsymbol{y}$ is generated, it is matched with a suitable subproblem j ($j \in \{1, 2, \ldots, N\}$) and compared with the neighboring solutions of this subproblem. In order to push the population towards the preferred region, the two-level feasibility rules are proposed. In the higher level, the components of the preferable upper bound vector $\boldsymbol{z}^{p,h+1}$ are regarded as the upper bounds of k implicit soft constraints for all objectives. For any decision vector $\boldsymbol{x}$, its violation to these constraints, termed as "preference violation", is defined as

$$\mathrm{PV}(\boldsymbol{x}) = \sum_{i=1}^{k} \max\{f_i(\boldsymbol{x}) - z_i^{p,h+1}, 0\}.$$

In the lower level, the violation to the considered subproblem's constraints is taken into account. Let $\boldsymbol{x}^i$ ($i = 1, 2, \ldots, |B(j)|$) denote the i-th neighboring solution of the subproblem j. Then, $\boldsymbol{x}^i$ will be replaced by $\boldsymbol{y}$ under the following cases:

(1) $\mathrm{PV}(\boldsymbol{y}) < \mathrm{PV}(\boldsymbol{x}^i)$;

(2) $\mathrm{PV}(\boldsymbol{y}) = \mathrm{PV}(\boldsymbol{x}^i)$, while $\boldsymbol{y}$ is preferred to $\boldsymbol{x}^i$ according to Deb's feasibility rules in Subsection 2.3.

3.5 Computational complexity analysis

The time complexity of the main procedures of iDMOEA-εC is presented in Table 2. At each interaction stage, the time complexity of eliciting and utilizing the DM's preferences mainly comes from the two-stage selection procedure and the adaptive adjustment of upper bound vectors. In the process of searching for solutions, iDMOEA-εC uses DMOEA-εC as the search engine. The time complexity of DMOEA-εC at each generation is $O(kN^2)$ and is mainly introduced by updating EP [13]. In summary, iDMOEA-εC's time complexity is $O(kN^2G_{\max})$.

4 Numerical experiments

In order to verify the effectiveness of iDMOEA-εC, in this section, iDMOEA-εC is compared with five interactive MOEAs on a series of real-world problems and benchmark test problems.

4.1 Experimental settings

4.1.1 *Test problems*

We select 20 test problems of different types from the literature. Owing to space limitation, the formulations of these problems are given in Appendix A. Related information on these problems is summarized in Table 3. In this table, the serial number of each problem, the corresponding references, the number of objectives, and the number of decision variables are listed in the first to the fourth columns, respectively. Other columns will be introduced later. Among the 20 test problems, P1 is a simplified water resources planning problem [23–25]. P2 and P3 describe two water quality management problems [26,27]. A location problem is formulated in P4 [25, 28]. The other problems are common benchmark test problems [29–33].

Table 3 Information on test problems

Problem	Reference	k	n	N	NFE	Weights	Real MPS
P1	[23–25]	3	2	55	11000	$[1, 1, 1]$	$[44.1, 24.1749, -48.1301]$
P2	[26]	4	2	55	10000	$[1, 10, 1, 1]$	$[-5.7049, -2.984, -6.9672, 0.6647]$
P3	[27]	5	2	50	10000	$[1, 5, 1, 1, 1]$	$[-5.6894, -2.9447, -6.9881, 0.4082, 0.0634]$
P4	[25, 28]	5	2	50	10000	$[1, 1, 1, 1, 1]$	$[-0.0797, -7.4879, -3.364, -0.0131, -6.4414]$
P5	[29]	2	30	100	20000	$[1, 1]$	$[0.3478, 0.4102]$
P6	[30]	2	2	50	10000	$[1, 1]$	[0.35,0.2906]
P7	[30]	5	2	50	10000	$[1, 2, 1, 2, 1]$	$[3.0147, 5.5702, 34.6287, -1.2038, 8.3255]$
P8	[30]	7	2	112	22000	$[1, 1, 1, 1, 1, 1, 1]$	$[2.9390, 1.1611, 2.5191, 5.1745, 34.9714, -1.2132, 9.9984]$
P9	[31]	3	7	210	60000	$[1.2, 1.1, 1]$	[0.1519,0.1657,0.1823]
P10	[31]	4	8	204	60000	$[1, 1.2, 1.2, 1]$	$[0.1364, 0.1136, 0.1136, 0.1364]$
P11	[31]	3	12	210	60000	$[1, 1, 1]$	$[0.5, 0.6, 0.6245]$
P12	[31]	4	13	204	60000	$[1, 1, 1, 1]$	$[0.4, 0.4, 0.5, 0.6557]$
P13	[32]	2	30	200	100000	$[1, 1]$	$[0.3478, 0.4102]$
P14	[32]	2	30	200	100000	$[1, 1]$	$[0.5, 0.5]$
P15	[33]	2	11	100	20000	$[1, 1]$	$[0.382, 2.4]$
P16	[33]	2	11	100	20000	$[1, 1]$	$[1.2, 1.6]$
P17	[33]	2	11	100	20000	$[1, 1]$	$[1.5612, 2.5]$
P18	[33]	3	12	210	60000	$[1, 1, 1]$	$[0.206, 0.5584, 2.4]$
P19	[33]	3	12	210	60000	$[1, 1, 1]$	$[0.65, 1.3, 2.1]$
P20	[33]	3	12	210	60000	$[1, 1, 1]$	$[0.8375, 2.223, 4.309]$

4.1.2 *Interactive MOEAs for comparison*

Considering that iDMOEA-εC requires the DM to select the best solution from a sample set of solutions, four interactive MOEAs that also ask for such kind of preferences from the DM are chosen for comparison. They are iTDEA [16], PI-NSGA-II-PC [17], iMOEA/D [12], and ICB-MOEA/D [34]. In addition, another interactive MOEA, named I-SIBEA [35], which allows the DM to classify a sample set of solutions into preferred ones and non-preferred ones is also used for comparison. However, it is restricted to only ask for a preferred solution from the DM. Since I-SIBEA needs to calculate hypervolumes very frequently, it is very time-consuming. On MOPs with more than three objectives, its runtime can be hundreds of times more than that of the other algorithms. Thus, we only compare with it on two- and three-objective test problems.

4.1.3 *Utility function*

The following quadratic utility function which is to be minimized is used to simulate the real DM's preferences for evaluating solutions:

$$U(f_1, f_2, \ldots, f_k) = \sum_{i=1}^{k} w_i (f_i - \bar{z}_i)^2, \tag{7}$$

where $\boldsymbol{w} = (w_1, \ldots, w_k)$ is the weight vector whose values are given in Table 3. For P1–P10, the vector $\bar{\boldsymbol{z}} = (\bar{z}_1, \ldots, \bar{z}_k)$ is set to the ideal point $\boldsymbol{z}^*$ which is computed beforehand. For P11–P20, $\bar{\boldsymbol{z}}$ is a point on the Pareto front. The optimal solution of U in the feasible region is taken as the real MPS of U, denoted by $\boldsymbol{z}^{\mathrm{MPS}}$. The real MPS on each problem is displayed in Table 3. In each interaction, the current best solution of the DM is the one with the minimum utility value.

4.1.4 *Performance indicators*

Let $\boldsymbol{z}^{\mathrm{final}}$ denote the solution most preferred by the DM at the final interaction. The normalized Euclidean distance between $\boldsymbol{z}^{\mathrm{final}}$ and $\boldsymbol{z}^{\mathrm{MPS}}$ is used as a performance indicator to measure the closeness of $\boldsymbol{z}^{\mathrm{final}}$ to $\boldsymbol{z}^{\mathrm{MPS}}$:

$$d(\boldsymbol{z}^{\mathrm{final}}, \boldsymbol{z}^{\mathrm{MPS}}) = \sqrt{\sum_{i=1}^{k} \left(\frac{z_i^{\mathrm{final}} - z_i^{\mathrm{MPS}}}{z_i^{\mathrm{nad}} - z_i^{*}} \right)^2}. \tag{8}$$

Table 4 Statistical results (Mean(Std.)) of the six algorithms over 30 independent runs in terms of distance

Problem	iDMOEA-εC	iTDEA	PI-NSGA-II-PC	iMOEA/D	I-SIBEA	ICB-MOEA/D
P1	6.418E−3(1.313E−2)	5.400E−2(3.398E−2)$^{+}$	5.809E−2(3.711E−2)$^{+}$	**3.224E−3(2.216E−3)**$^{-}$	2.874E−2(4.201E−2)$^{+}$	1.257E−2(1.904E−2)$^{+}$
P2	**6.237E−3(1.280E−2)**	3.186E−2(1.357E−2)$^{+}$	1.685E−1(7.640E−2)$^{+}$	3.977E−2(4.028E−2)$^{+}$	–	8.030E−3(1.105E−2)$^{+}$
P3	**3.856E− 3(3.632E−3)**	4.092E−2(2.101E−2)$^{+}$	2.376E−1(1.097E−1)$^{+}$	1.524E−2(1.578E−2)$^{+}$	–	8.548E−3(2.979E−3)$^{+}$
P4	**2.533E−3(3.042E−3)**	8.821E−2(5.946E−2)$^{+}$	1.335E−1(7.273E−2)$^{+}$	3.695E−2(4.871E−2)$^{+}$	–	6.799E−3(1.996E−3)$^{+}$
P5	**8.385E−3(6.856E−3)**	2.035E−2(1.387E−2)$^{+}$	1.473E−2(9.352E−3)$^{+}$	1.148E−2(1.120E−2)$^{\approx}$	7.548E−2(6.863E−2)$^{+}$	9.427E−3(9.169E−3)$^{\approx}$
P6	**1.901E−2(5.470E−2)**	1.775E−1(1.408E−1)$^{+}$	4.302E−2(9.040E−2)$^{+}$	1.959E−1(2.510E−1)$^{+}$	1.123E−1(1.270E−1)$^{+}$	6.396E−2(1.191E−1)$^{+}$
P7	**6.252E−3(6.032E−3)**	4.738E−2(3.106E−2)$^{+}$	4.367E−2(2.799E−2)$^{+}$	9.849E−3(1.051E−2)$^{\approx}$	–	1.113E−2(1.098E−2)$^{+}$
P8	9.169E−3(3.256E− 2)	2.097E−1(1.491E−1)$^{+}$	2.548E−1(1.894E−1)$^{+}$	1.513E−2(1.769E−2)$^{+}$	–	**4.923E−3(1.382E−2)**$^{-}$
P9	**2.359E−2(1.532E−2)**	4.912E−2(2.219E−2)$^{+}$	4.824E−2(2.080E−2)$^{+}$	2.193E−2(8.582E−3)$^{\approx}$	5.850E−2(1.740E−2)$^{+}$	2.530E−2(1.341E−2)$^{\approx}$
P10	4.803E−2(4.996E−2)	7.693E−2(3.127E−2)$^{+}$	1.191E−1(4.319E−2)$^{+}$	**3.283E−2(1.706E−2)**$^{\approx}$	–	6.574E−2(8.283E−2)$^{+}$
P11	**6.599E−3(8.388E−3)**	8.988E−2(2.773E−2)$^{+}$	4.335E−2(1.807E−2)$^{+}$	1.964E−2(6.766E−3)$^{+}$	2.098E−1(3.938E−2)$^{+}$	2.164E−2(3.200E−2)$^{+}$
P12	**2.943E−2(2.285E−2)**	1.342E−1(3.193E−2)$^{+}$	1.222E−1(4.028E−2)$^{+}$	3.652E−2(2.620E−2)$^{\approx}$	–	4.229E−2(3.971E−2)$^{+}$
P13	**8.091E−2(6.567E−2)**	1.059E−1(7.447E−2)$^{+}$	1.255E−1(6.169E−2)$^{+}$	9.731E−2(7.343E−2)$^{\approx}$	1.240E−1(8.919E−2)$^{+}$	9.308E− 2(6.873E−2)$^{\approx}$
P14	1.590E−1(1.861E−1)	2.535E−1(1.280E−1)$^{+}$	7.501E−2(4.086E−2)$^{-}$	1.916E−1(2.337E−1)$^{+}$	2.180E−1(1.003E−1)$^{+}$	**5.213E−2(1.407E−1)**$^{-}$
P15	**3.252E−3(6.250E−4)**	9.263E−3(2.875E−3)$^{+}$	5.493E−3(1.291E−3)$^{+}$	5.877E−3(1.747E−3)$^{+}$	1.404E−2(1.048E−2)$^{+}$	3.440E−3(1.111E−3)$^{\approx}$
P16	**3.604E−3(6.781E−4)**	1.849E−2(9.890E−3)$^{+}$	7.898E−3(3.340E−3)$^{+}$	9.620E−3(3.910E−3)$^{+}$	2.460E−2(3.540E−2)$^{+}$	7.219E−3(8.597E−4)$^{+}$
P17	1.848E−2(3.171E−3)	1.815E−2(5.872E−3)$^{\approx}$	**1.067E−2(3.337E−3)**$^{-}$	2.386E−2(8.985E−3)$^{+}$	2.341E−2(1.974E−2)$^{+}$	2.107E−2(6.886E−3)$^{+}$
P18	**4.741E−3(1.578E−3)**	1.048E−1(4.319E−2)$^{+}$	2.664E−2(1.086E−2)$^{+}$	3.975E−2(3.051E−2)$^{+}$	1.105E−1(5.659E−2)$^{+}$	6.986E−3(2.067E−3)$^{+}$
P19	1.124E−2(5.157E−3)	9.221E−2(4.505E−2)$^{+}$	7.220E−2(2.059E−2)$^{+}$	2.922E−2(2.517E−2)$^{+}$	1.172E−1(6.550E−2)$^{+}$	**1.000E−2(5.577E−3)**$^{\approx}$
P20	**4.923E−2(2.341E−2)**	1.169E−1(3.752E−2)$^{+}$	7.337E−2(2.565E−2)$^{+}$	6.678E−2(2.002E−2)$^{\approx}$	1.509E−1(7.038E−2)$^{+}$	4.995E−2(2.684E−2)$^{\approx}$
+/ ≈ /−	–	19/1/0	18/0/2	12/7/1	13/0/0	12/6/2

For simplicity, we call this performance indicator "distance". Define the maximum value of U on the Pareto front and the minimum value of U as $U_{\max}$ and $U_{\min}$, respectively. The following normalized difference between $U_{\min}$ and $U(\boldsymbol{z}^{\text{final}})$ is also adopted as a performance indicator to measure the quality of $\boldsymbol{z}^{\text{final}}$ [16]:

$$\text{difference} = \frac{U(\boldsymbol{z}^{\text{final}}) - U_{\min}}{U_{\max} - U_{\min}}. \tag{9}$$

4.1.5 *Parameter settings*

Just like iDMOEA-εC, iTDEA, iMOEA/D, I-SIBEA, and ICB-MOEA/D all allow the DM to specify the overall number of interactions and they terminate when the maximum allowable number of generations is reached. Different from these algorithms, PI-NSGA-II-PC has its own stopping criterion, i.e., stopping when no enough improvement of solutions can be achieved. Its total number of interactions H and overall number of generations are not predefined. In order to test different algorithms under the same number of interactions, here we modified PI-NSGA-II-PC to make it have the same interaction stages as the other algorithms do. For all algorithms, H is set to 6 for two- and three-objective test problems and 8 for the other test problems. The moments for interaction are the same as introduced in Section 3. Besides, in each interaction, the number of nondominated solutions shown to the DM takes the value of $P = 2(k+3)$.

A two-layer weight vector generation method [36] is used for iMOEA/D to generate weight vectors. The population size N for each test problem is determined accordingly (see Table 3). The other algorithms adopt the same population size except for I-SIBEA. As a small population size is recommended for I-SIBEA [35], its population size is set to $N/2$. The two-stage upper bound vector generation procedure proposed in [37] is used for iDMOEA-εC to generate initial upper bound vectors. For fair comparison, the maximum number of function evaluations (NFE) is kept the same for all algorithms (see Table 3). The $|\text{EP}|_{\max}$ for iDMOEA-εC and PI-NSGA-II-PC is set to N. iTDEA also has an archive, but the size of this archive is determined by a pair of parameters (τ_0, τ_H). Based on some previous experiments on each test problem, we have determined suitable parameter values for iTDEA on each problem to make the mean size of its archive larger than N. These parameter values are $(0.2, 0.02)$ for P1 and P12, $(0.2, 0.05)$ for P2–P4, $(0.01, 0.0005)$ for P5 and P6, $(0.1, 0.01)$ for P7 and P8, $(0.1, 0.005)$ for P9 and P18, $(0.1, 0.02)$ for P10, P11, and P20, $(0.0001, 0.000001)$ for P13 and P14, $(0.01, 0.001)$ for P15, $(0.01, 0.005)$ for P16 and P17, and $(0.05, 0.01)$ for P19. For iDMOEA-εC, the compression rate α takes the value of 0.25 for test problems with two and three objectives and 0.3 for the other problems.

4.2 Experimental results and analysis

Each of the six interactive MOEAs is run 30 times independently in MATLAB environment on a PC with Intel(R) Core(TM) CPU i7-5557U 3.1 GHz, and 4 GB RAM. The means and standard deviations of each algorithm in terms of the distance and the difference indicators are shown in Tables 4 and 5, respectively. The numbers in parentheses represent the standard deviations. The value in bold means that the corresponding algorithm has the smallest mean value as compared with the other algorithms.

Table 5 Statistical results (Mean(Std.)) of the six algorithms over 30 independent runs in terms of difference

Problem	iDMOEA-εC	iTDEA	PI-NSGA-II-PC	iMOEA/D	I-SIBEA	ICB-MOEA/D
P1	1.510E−4(4.305E−4)	1.205E−2(9.582E−3)$^{+}$	6.850E−3(5.629E−3)$^{+}$	**1.173E−5(1.113E−5)**$^{-}$	4.116E−3(3.670E−3)$^{+}$	3.960E−4(3.996E−4)$^{+}$
P2	**1.853E−5(7.408E−5)**	1.367E−4(9.809E−5)$^{+}$	4.048E−3(2.885E− 3)$^{+}$	3.677E−4(8.880E−4)$^{+}$	–	5.605E−5(5.275E−5)$^{+}$
P3	**3.005E−6(7.086E−6)**	2.192E−4(2.002E−4)$^{+}$	4.897E−3(2.916E−3)$^{+}$	4.441E−5(7.385E−5)$^{+}$	–	4.740E−5(2.604E−5)$^{+}$
P4	**2.340E−5(4.510E−5)**	1.157E−2(1.156E−2)$^{+}$	2.786E−2(2.298E−2)$^{+}$	2.805E−3(5.069E−3)$^{+}$	–	1.508E−4(2.804E−4)$^{+}$
P5	**6.596E−4(4.739E−4)**	4.501E−3(1.274E−3)$^{+}$	5.551E−3(1.080E−3)$^{+}$	6.007E−3(2.440E−3)$^{+}$	1.995E−2(3.666E−2)$^{+}$	6.848E−4(6.796E−4)$^{\approx}$
P6	**6.071E−3(1.985E−2)**	1.057E−1(1.247E−1)$^{+}$	1.917E−2(6.106E−2)$^{+}$	1.965E−1(3.626E−1)$^{+}$	5.155E−2(1.055E−1)$^{+}$	2.940E−2(6.988E−2)$^{+}$
P7	**2.730E−5(3.547E−5)**	2.167E−3(2.254E−3)$^{+}$	1.848E−3(1.794E−3)$^{+}$	1.105E−4(1.877E−4)$^{\approx}$	–	1.098E−4(2.042E−4)$^{+}$
P8	7.530E−6(3.788E−5)	3.930E−4(3.594E−4)$^{+}$	7.504E−4(6.866E−4)$^{+}$	1.012E−5(1.876E−5)$^{+}$	–	**4.079E−6(4.483E−6)**$^{-}$
P9	**2.377E−3(1.630E−3)**	6.927E−3(2.725E−3)$^{+}$	9.619E−3(3.830E−3)$^{+}$	2.987E−3(8.235E−3)$^{+}$	6.888E−3(1.683E−3)$^{+}$	2.872E−3(1.921E−3)$^{+}$
P10	9.700E−3(1.448E−2)	2.053E−2(1.339E−2)$^{+}$	3.446E−2(1.523E−2)$^{+}$	**2.478E−3(1.843E−3)**$^{-}$	–	1.340E−2(2.777E−2)$^{\approx}$
P11	**1.139E−4(2.878E−4)**	8.848E−3(4.852E−3)$^{+}$	2.206E−3(1.629E−3)$^{+}$	4.317E−4(2.776E−4)$^{+}$	4.557E−2(1.768E−2)$^{+}$	5.433E−4(1.639E−3)$^{+}$
P12	**1.157E−3(1.411E−3)**	1.587E−2(7.089E−3)$^{+}$	1.379E−2(8.163E−3)$^{+}$	1.684E−3(2.557E−3)$^{\approx}$	–	2.253E−3(4.904E−3)$^{+}$
P13	**1.847E−2(2.576E−2)**	3.093E−2(4.302E−2)$^{+}$	4.047E−2(2.737E−2)$^{+}$	2.869E−2(3.365E−2)$^{+}$	4.416E−2(7.291E−2)$^{+}$	2.327E−2(2.988E−2)$^{+}$
P14	1.089E−1(1.738E−1)	1.751E−1(1.620E−1)$^{+}$	**2.896E−2(1.671E−2)**$^{-}$	1.802E−1(2.752E−1)$^{+}$	1.371E−1(8.902E−2)$^{+}$	4.324E−2(1.483E−1)$^{-}$
P15	**2.519E−6(1.449E−6)**	2.833E−5(7.542E−6)$^{+}$	2.174E−5(9.801E−6)$^{+}$	2.039E−5(2.444E−5)$^{+}$	5.276E−4(6.821E−4)$^{+}$	3.288E−6(2.311E−6)$^{\approx}$
P16	**4.290E−6(2.735E−6)**	1.167E−3(9.415E−4)$^{+}$	6.841E−5(5.304E−5)$^{+}$	4.654E−5(2.094E−5)$^{+}$	2.532E−3(8.529E−3)$^{+}$	4.085E−5(8.392E−5)$^{\approx}$
P17	3.241E−4(1.724E−4)	1.729E−3(8.189E−4)$^{+}$	**1.022E−4(4.852E−5)**$^{-}$	2.952E−4(1.489E−4)$^{\approx}$	1.581E−3(2.494E−3)$^{+}$	3.022E−4(3.163E−4)$^{\approx}$
P18	**2.586E−6(1.520E−6)**	3.661E−3(3.368E−3)$^{+}$	7.038E−4(7.227E−4)$^{+}$	1.225E− 3(1.591E−3)$^{+}$	9.800E−3(6.550E−3)$^{+}$	4.119E−6(6.708E−6)$^{+}$
P19	1.934E−5(2.309E−5)	3.823E−3(2.930E−3)$^{+}$	2.061E−3(9.092E−4)$^{+}$	2.639E−4(3.122E−4)$^{+}$	3.401E−3(1.823E−3)$^{+}$	**1.241E−5(1.695E−5)**$^{-}$
P20	**6.048E−4(7.512E−4)**	6.027E−3(3.618E−3)$^{+}$	1.869E−3(6.522E−4)$^{+}$	1.319E−3(1.190E−3)$^{+}$	2.054E−2(1.479E−2)$^{+}$	9.192E−4(1.307E−3)$^{+}$
+/ ≈ /−	–	20/0/0	18/0/2	15/3/2	13/0/0	12/5/3

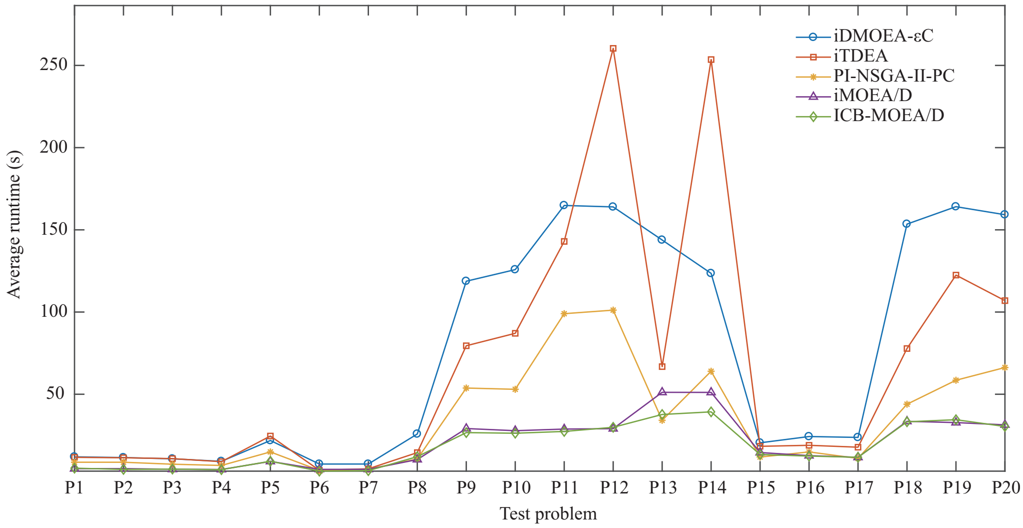

Figure 6 (Color online) The average runtime (s) of iDMOEA-εC, iTDEA, PI-NSGA-II-PC, iMOEA/D, and ICB-MOEA/D on each test problem.

In order to compare the significance of the differences between iDMOEA-εC and each of the other algorithms, the Wilcoxon rank sum test at a significance level of 0.05 is conducted. The three symbols, $+, \approx$, and $-$, represent that iDMOEA-εC performs statistically better than, similar to, or worse than its competitors, respectively. The numbers of them are listed in the last rows of Tables 4 and 5.

From Tables 4 and 5, iDMOEA-εC can approximate the DM's real MPS effectively. Compared with the other five algorithms, iDMOEA-εC has the minimum mean values on 14 out of 20 instances in terms of both the distance and the difference indicators, reflecting the superiority of iDMOEA-εC over its competitors in approximating the MPS. According to the results of the Wilcoxon rank sum test, no matter which indicator is used, iDMOEA-εC is significantly better than I-SIBEA on all test instances. In addition, iDMOEA-εC outperforms iTDEA and PI-NSGA-II-PC on all test problems except for P14 and P17. Compared with iMOEA/D and ICB-MOEA/D, iDMOEA-εC has significantly better or similar performance on most test problems. To sum up, iDMOEA-εC exhibits remarkable advantages over its competitors in finding the DM's MPS.

As the DM is involved in the solution process, the ability of assisting the DM to find the MPS quickly is important for an interactive algorithm. Figure 6 displays the average runtime of iDMOEA-εC, iTDEA, PI-NSGA-II-PC, iMOEA/D, and ICB-MOEA/D on each test problem[2)]. It can be seen that the runtime of the five algorithms has little difference on P1–P8 and P15–P17. On the other test problems, it is obvious that iMOEA/D and ICB-MOEA/D have smaller runtime. This is mainly because iMOEA/D

[2)] The runtime of I-SIBEA is much longer than that of the other algorithms and is not considered here.

and ICB-MOEA/D do not use EP to save elite solutions while the other three algorithms update EP at each generation. The population size used on P9–P14 and P18–P20 is larger than that used on the other test problems, and the differences between the runtime of different algorithms become evident. On all test problems, iDMOEA-εC's runtime is less than 3 min. This means that the waiting time of the DM between two adjacent interactions is usually a few seconds, which is acceptable in practice.

5 Further discussion

This section is devoted to a further study of iDMOEA-εC's behavior. Firstly, the influences of the overall number of interactions H and the compression rate α are analyzed. Secondly, the effects of the two-stage selection procedure and the adaptive adjustment mechanism of upper bound vectors are investigated.

5.1 Parameter sensitivity analysis on H and α

The total number of interactions H is closely related with the DM's burden. The DM will feel tired if too many interactions are carried out. For practical purposes, it is of great importance that iDMOEA-εC can get good performance within a small number of interactions. The compression rate α is also an important parameter which may affect the accuracy of the final solution. In this subsection, we study how iDMOEA-εC is sensitive to H and α.

For each k ($k = 2, 3, 4, 5, 7$), we take at most two test problems as examples and conduct 20 groups of tests on each selected problem by using iDMOEA-εC. The selected test problems are P1, P2, P4, P5, P7, P8, P9 and P10. H takes values of 4, 6, 8, 10, and α varies among $\{0.15, 0.2, 0.25, 0.3, 0.35\}$. Each group of tests corresponds to a pair of H and α and contains 30 independent runs. All the other parameters are kept the same as Section 4. The mean distance and mean difference values for each test group are calculated. For the convenience of observing and analyzing the results, the histograms of the mean distance and mean difference values are displayed in Figures 7 and 8, respectively. Each figure has eight subfigures corresponding to eight problems. The subtitle of each subfigure gives the name and the number of objectives of the problem.

Firstly, we analyze the influence of H. For P7 and P10, there are no remarkable differences when different values of H are used. On P5, iDMOEA-εC performs rather similar no matter which value of H is taken according to the distance indicator, while its performance in terms of the difference indicator has obvious difference between $H = 4$ and $H > 4$. As for P1, P2, P4, P8 and P9, iDMOEA-εC performs obviously better when $H > 4$ than $H = 4$ in terms of both indicators. This means that for some problems, the performance of iDMOEA-εC can be improved sufficiently when more than four interactions are allowed. Meanwhile, on P1 and P2, iDMOEA-εC has better mean values when $H > 6$ than $H = 6$, while $H = 8$ and $H = 10$ do not bring a significant difference. With regard to the other six problems, iDMOEA-εC performs similarly under $H = 6, 8, 10$. This reveals that six or eight interactions are enough for iDMOEA-εC, and more interactions may not bring significantly better results.

Next, the effects of α are discussed. From Figures 7 and 8, the influence of α varies with different H and different problems. For example, for the same problem P1, $\alpha = 0.35$ causes the largest mean distance value when $H = 8$, while $\alpha = 0.15$ leads to the the largest mean value when $H = 4$. While on P7, $\alpha = 0.15$ brings the smallest mean value when $H = 4$. Generally speaking, iDMOEA-εC is more sensitive to the change of α when $H = 4$. For other values of H, the results of iDMOEA-εC usually do not change a lot when using different α. Occasionally, some values of α lead to an obvious performance degradation. Since the worst mean values are often caused by $\alpha = 0.15$ or $\alpha = 0.35$, it can be claimed that iDMOEA-εC is more stable when α varies from 0.2 to 0.3.

5.2 Effects of the two-stage selection procedure and the adaptive adjustment of upper bound vectors

For the purpose of verifying the effectiveness of the proposed two-stage selection procedure, we compare iDMOEA-εC with its variant, named iDMOEA-εC-1Stage. This variant presents P solutions to the DM all at once. In addition, to exhibit the necessity of adaptively adjusting the upper bound vectors, another variant called iDMOEA-εC-NoAda is also formed. It uses the evenly spread upper bound vectors all the time. P1–P12 are used as examples. Following the same parameter setting in Section 4, iDMOEA-εC and its variants are run 30 times independently on each test problem. The experimental results in terms

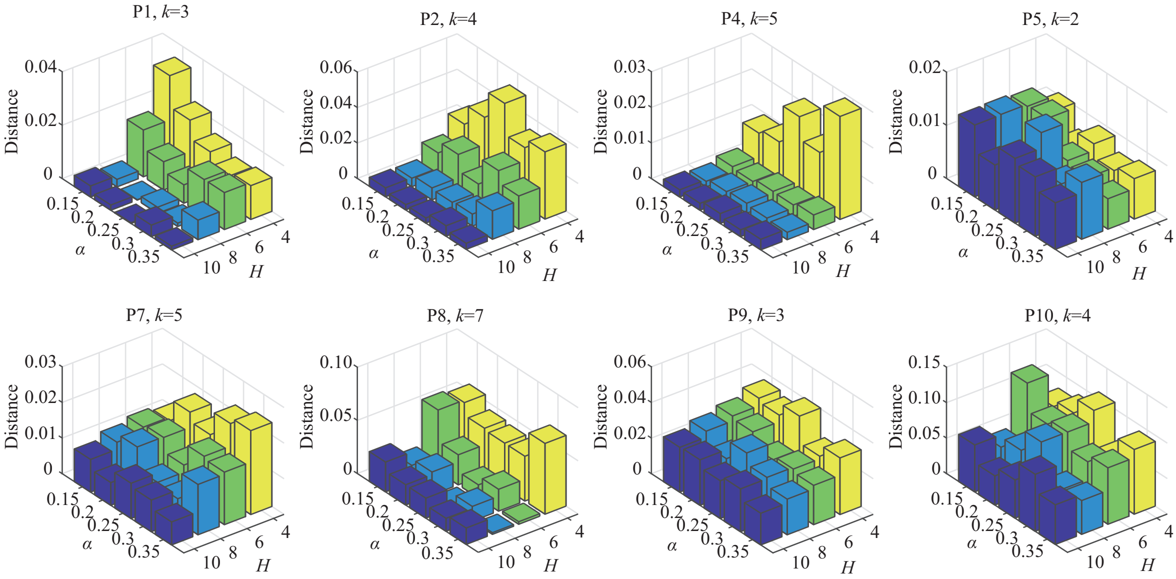

Figure 7 (Color online) Histograms of mean distance values for the 20 test groups on each problem.

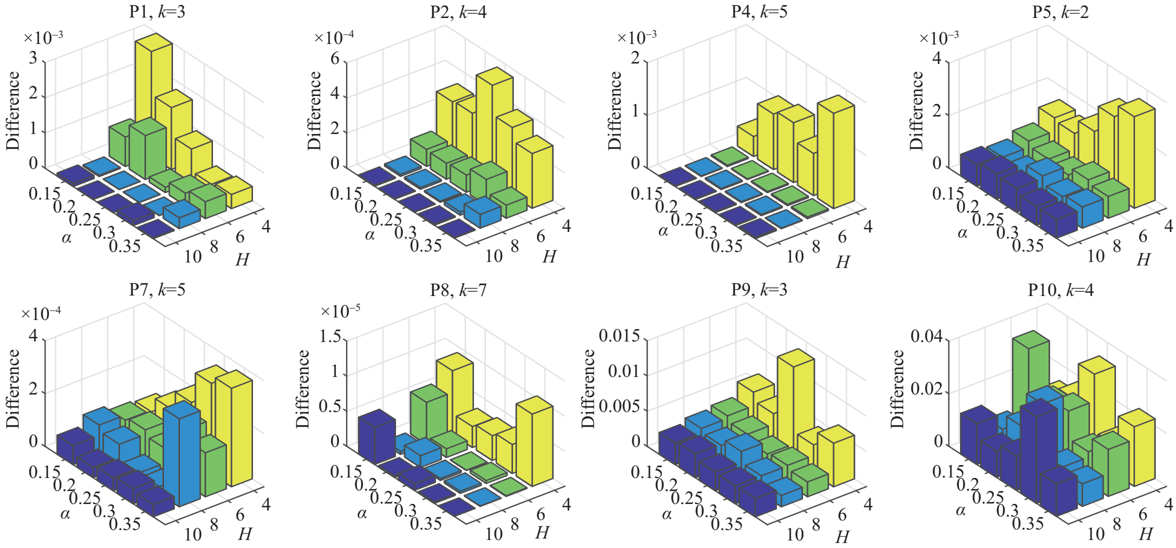

Figure 8 (Color online) Histograms of mean difference values for the 20 test groups on each problem.

of the mean distance values and the mean difference values are given in Tables 6 and 7, respectively. The smallest mean value for each problem is in bold. The Wilcoxon rank sum test at a significance level of 0.05 is conducted to test the significance of the differences between iDMOEA-εC and each of its variants.

From Tables 6 and 7, iDMOEA-εC outperforms its variants on all the 12 test problems in terms of the distance indicator and on 11 out of 12 instances in terms of the difference indicator. In addition, according to the Wilcoxon rank sum test, iDMOEA-εC performs significantly better than or similar to its variants on all test problems in terms of both indicators. This validates the effectiveness of the proposed two-stage selection procedure and the adaptive adjustment mechanism for the upper bound vectors.

6 Conclusion

This paper presented an interactive MOEA involving a decomposition framework and progressive and explicit compression of the DM's preferred region. Both decomposition and compression act on the objective space, which reflects the DM's preferences more directly than other spaces such as the weight space. The proposed algorithm employs a state-of-the-art MOEA to search for solutions. Furthermore, it

Table 6 Statistical results (Mean(Std.)) of iDMOEA-εC and its two variants in terms of the distance indicator

Problem	iDMOEA-εC	iDMOEA-εC-1Stage	iDMOEA-εC-NoAda
P1	**6.418E−3(1.313E−2)**	1.106E−2(1.741E−2)$^{+}$	1.633E−2(2.415E−2)$^{+}$
P2	**6.237E−3(1.280E−2)**	1.165E−2(1.674E−2)$^{+}$	1.085E−2(1.483E−2)$^{+}$
P3	**3.856E−3(3.632E−3)**	5.513E−3(4.088E−3)$^{+}$	4.353E−3(3.444E−3)$^{\approx}$
P4	**2.533E−3(3.042E−3)**	2.969E−3(1.863E−3)$^{+}$	3.427E−3(3.887E−3)$^{+}$
P5	**8.385E−3(6.856E−3)**	1.452E−2(1.114E−2)$^{+}$	1.500E−2(9.872E−3)$^{+}$
P6	**1.901E−2(5.470E−2)**	8.618E−2(2.550E−1)$^{+}$	4.381E−2(1.335E−1)$^{+}$
P7	**6.252E−3(6.032E−3)**	2.943E−2(9.690E−2)$^{+}$	9.072E−3(8.030E−3)$^{+}$
P8	**9.169E−3(3.256E−2)**	1.081E−2(3.333E−2)$^{+}$	1.964E−2(9.957E−2)$^{\approx}$
P9	**2.359E−2(1.532E−2)**	2.519E−2(1.621E−2)$^{\approx}$	3.573E−2(1.999E−2)$^{+}$
P10	**4.803E−2(4.996E−2)**	5.344E−2(7.157E−2)$^{\approx}$	7.383E−2(8.257E−2)$^{+}$
P11	**6.599E−3(8.388E−3)**	1.724E−2(2.063E−2)$^{+}$	1.411E−2(1.777E−2)$^{\approx}$
P12	**2.943E−2(2.285E−2)**	3.191E−2(3.615E−2)$^{\approx}$	2.952E−2(2.842E−2)$^{\approx}$

Table 7 Statistical results (Mean(Std.)) of iDMOEA-εC and its two variants in terms of the difference indicator

Problem	iDMOEA-εC	iDMOEA-εC-1Stage	iDMOEA-εC-NoAda
P1	**1.510E−4(4.305E−4)**	2.997E−4(5.602E−4)$^{\approx}$	5.975E−4(1.236E−3)$^{+}$
P2	**1.853E−5(7.408E−5)**	3.865E−5(1.279E−4)$^{+}$	3.025E−5(7.000E−5)$^{+}$
P3	**3.005E−6(7.086E−6)**	4.866E−6(6.517E−6)$^{+}$	3.497E−6(4.645E−6)$^{\approx}$
P4	2.340E−5(4.510E−5)	**5.107E−6(8.761E−6)**$^{\approx}$	3.876E−5(6.723E−5)$^{\approx}$
P5	**6.596E−4(4.739E−4)**	1.055E−3(9.467E−4)$^{+}$	5.192E−3(3.156E−3)$^{+}$
P6	**6.071E−3(1.985E−2)**	1.323E−1(5.959E−1)$^{+}$	3.791E−2(1.490E−1)$^{+}$
P7	**2.730E−5(3.547E−5)**	3.572E−3(1.869E−2)$^{+}$	8.063E−5(1.141E−4)$^{+}$
P8	**7.530E−6(3.788E−5)**	1.090E−5(5.333E−5)$^{+}$	4.187E−5(2.252E−4)$^{+}$
P9	**2.377E−3(1.630E−3)**	2.736E−3(1.848E−3)$^{\approx}$	5.313E−3(5.422E−3)$^{+}$
P10	**9.700E−3(1.448E−2)**	1.155E−2(2.355E−2)$^{\approx}$	1.939E−2(2.836E−2)$^{+}$
P11	**1.139E−4(2.878E−4)**	7.227E−4(1.494E−3)$^{+}$	5.149E−4(1.503E−3)$^{+}$
P12	**1.157E−3(1.411E−3)**	1.938E−3(4.176E−3)$^{+}$	2.323E−3(2.219E−3)$^{\approx}$
$+/\approx/-$		8/4/0	9/3/0

includes an adaptive adjustment mechanism of subproblems and two-level feasibility rules to control the spread of the population and guide the search process toward the preferred region. Finally, the proposed algorithm employs a two-stage selection procedure to elicit the DM's preferences on a small representative set of solutions. The algorithm was compared to other interactive MOEAs in a set of experiments. The experimental results demonstrated that the proposed algorithm can approximate the DM's MPS better than its competitors.

In the future, we intend to further improve the ways of eliciting the DM's preferences and compressing the objective space to capture the DM's preferences more precisely.

Acknowledgements This work was supported in part by National Outstanding Youth Talents Support Program (Grant No. 61822304), National Natural Science Foundation of China (Grant No. 61673058), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (Grant No. U1609214), Consulting Research Project of the Chinese Academy of Engineering (Grant No. 2019-XZ-7), Projects of Major International (Regional) Joint Research Program of NSFC (Grant No. 61720106011), Peng Cheng Laboratory, and Beijing Advanced Innovation Center for Intelligent Robots and Systems.

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Wang Y Y, Jiao X H. Multi-objective energy management for PHEV using Pontryagin's minimum principle and particle swarm optimization online. Sci China Inf Sci, 2021, 64: 119204

2 Gao G Q, Xin B. A-STC: auction-based spanning tree coverage algorithm formotion planning of cooperative robots. Front Inf Technol Electron Eng, 2019, 20: 18–31

3 Radmanesh M, Kumar M, Guentert P H, et al. Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study. Unmanned Syst, 2018, 6: 95–118

4 Hafez A T, Kamel M A. Cooperative task assignment and trajectory planning of unmanned systems via HFLC and PSO. Unmanned Syst, 2019, 7: 65–81

5 Xue Y, Zhang J, Gao X Q. Resource allocation for pilot-assisted massive MIMO transmission. Sci China Inf Sci, 2017, 60: 042302

6 Zhou H, Yu W, Yi P, et al. Quantized gradient-descent algorithm for distributed resource allocation. Unmanned Syst, 2019, 7: 119–136

7 Wang K H, Xiong Z H, Chen L, et al. Joint time delay and energy optimization with intelligent overclocking in edge computing. Sci China Inf Sci, 2020, 63: 140313

8 Li B, Li J, Tang K, et al. Many-objective evolutionary algorithms. ACM Comput Surv, 2015, 48: 1–35

9 von Lücken C, Barán B, Brizuela C. A survey on multi-objective evolutionary algorithms for many-objective problems. Comput Opt Appl, 2014, 25: 707–756

10 Xin B, Chen L, Chen J, et al. Interactive multiobjective optimization: a review of the state-of-the-art. IEEE Access, 2018, 6: 41256–41279

11 Deb K, Sinha A, Korhonen P J, et al. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. IEEE Trans Evol Computat, 2010, 14: 723–739

12 Gong M, Liu F, Zhang W, et al. Interactive MOEA/D for multi-objective decision making. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, 2011. 721–728

13 Chen J, Li J, Xin B. DMOEA-εC: decomposition-based multiobjective evolutionary algorithm with the ε-constraint framework. IEEE Trans Evol Comput, 2017, 21: 714–730

14 Zhou H, Qiao J. Multiobjective optimal control for wastewater treatment process using adaptive MOEA/D. Appl Intell, 2019, 49: 1098–1126

15 Qiao J, Zhou H, Yang C, et al. A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty. Appl Soft Comput, 2019, 74: 190–205

16 Koksalan M, Karahan I. An interactive territory defining evolutionary algorithm: iTDEA. IEEE Trans Evol Comput, 2010, 14: 702–722

17 Sinha A, Korhonen P, Wallenius J, et al. An interactive evolutionary multi-objective optimization method based on polyhedral cones. In: Proceedings of International Conference on Learning and Intelligent Optimization. Berlin: Springer, 2010. 318–332

18 Haimes Y Y, Lasdon L S, Wismer D A. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans Syst Man Cybern, 1971, 1: 296–297

19 Miettinen K. Nonlinear Multiobjective Optimization. Boston: Kluwer Academic Publishers, 1999

20 Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proceedings of the IEEE Congress on Evolutionary Computation, 2009. 203–208

21 Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng, 2000, 186: 311–338

22 Jain A K, Murty M N, Flynn P J. Data clustering. ACM Comput Surv, 1999, 31: 264–323

23 Chankong V, Haimes Y Y. Multiobjective Decision Making: Theory and Methodology. Amsterdam: Elsevier Science Publishing, 1983

24 Galperin E A, Wiecek M. Retrieval and use of the balance set in multiobjective global optimization. Comput Math Appl, 1999, 37: 111–123

25 Miettinen K, Mäkelä M M, Kaario K. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. Eur J Oper Res, 2006, 175: 931–947

26 Narula S C, Weistroffer H R. A flexible method for nonlinear multicriteria decision-making problems. IEEE Trans Syst Man Cybern, 1989, 19: 883–887

27 Miettinen K, Mäkelä M M. Interactive method NIMBUS for nondifferentiable multiobjective optimization problems. In: Multicriteria Analysis. Berlin: Springer, 1997. 310–319

28 Miettinen K, Lotov A V, Kamenev G K, et al. Integration of two multiobjective optimization methods for nonlinear problems. Opt Methods Softw, 2003, 18: 63–80

29 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. Evolary Comput, 2000, 8: 173–195

30 Miettinen K, Mäkelä M M. Interactive bundle-based method for nondifferentiable multiobjeective optimization: NIMBUS. Optimization, 1995, 34: 231–246

31 Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization. Berlin: Springer, 2005. 105–145

32 Zhang Q, Zhou A, Zhao S, et al. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, 2009

33 Huband S, Barone L, While L, et al. A scalable multi-objective test problem toolkit. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization, 2005. 280–295

34 Xin B, Li H, Wang L. ICB-MOEA/D: an interactive classification-based multi-objective optimization algorithm. In: Proceedings of the 37th Chinese Control Conference, 2018. 2500–2505

35 Chugh T, Sindhya K, Hakanen J, et al. An interactive simple indicator-based evolutionary algorithm (I-SIBEA) for multiobjective optimization problems. In: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2015. 277–291

36 Li K, Deb K, Zhang Q, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput, 2015, 19: 694–716

37 Li J, Xin B. An improved version of DMOEA-εC for many-objective optimization problems: iDMOEA-εC. In: Proceedings of the 38th Chinese Control Conference, 2019. 2212–2217