A Localized Decomposition Evolutionary Algorithm for Imbalanced Multi-objective Optimization

Yulong Ye[1], Qiuzhen Lin[1]*, Ka-Chun Wong[2], Jianqiang Li[1], Zhong Ming[1], Carlos A. Coello Coello[3,4]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR. China*

[2]*Department of Computer Science, City University of Hong Kong, Hong Kong, China*

[3]*CINVESTAV-IPN, Department of Computer Science, Mexico, D.F. 07360, Mexico*

[4]*Basque Center for Applied Mathematics (BCAM) & Ikerbasque, Spain*

Abstract:

Multi-objective evolutionary algorithms based on decomposition (MOEA/Ds) convert a multi-objective optimization problem (MOP) into a set of scalar subproblems, which are then optimized in a collaborative manner. However, when tackling imbalanced MOPs, the performance of most MOEA/Ds will evidently deteriorate, as a few solutions will replace most of the others in the evolutionary process, resulting in a significant loss of diversity. To address this issue, this paper suggests a localized decomposition evolutionary algorithm (LDEA) for imbalanced MOPs. A localized decomposition method is proposed to assign a local region for each subproblem, where the inside solutions are associated and the solution update is restricted inside (i.e., solutions are only replaced by offspring within the same local region). Once off-spring are generated within an originally empty region, the best one is reserved for this subproblem to extend diversity. Meanwhile, the subproblem with the largest number of associated solutions will be found and one of its associated solutions with the worst aggregated value will be removed. Moreover, to speed up convergence for each subproblem while balancing the population's diversity, LDEA only evolves the best-associated solution in each subproblem and correspondingly tailors two decomposition methods in the environmental selection. When compared to nine competitive MOEAs, LDEA has shown the advantages in tackling two benchmark sets of imbalanced MOPs, one benchmark set of balanced yet complicated MOPs, and one real-world MOP.

Keywords: Multi-objective optimization; Evolutionary algorithm; Localized decomposition.

1. Introduction

During recent decades, a number of multi-objective evolutionary algorithms (MOEAs) have been reported, which have shown very promising performance when tackling different kinds of multi-objective optimization problems (MOPs) [1]. According to the environmental selection mechanism, most MOEAs can be classified into three main categories, i.e., Pareto-based MOEAs [2]-[5], indicator-based MOEAs [6]- [11], and decomposition-based MOEAs (MOEA/Ds) [12]-[15]. When compared to Pareto-based and indicator-based MOEAs, MOEA/Ds are more flexible in balancing the convergence and diversity in their environmental selection mechanisms [16]-[17], showing the advantages for tackling some complicated MOPs [15], [18]. An MOP in MOEA/Ds is decomposed into a set of scalar subproblems, which are then optimized by the associated solutions in a collaborative manner. During the recent years, this framework has triggered a large number of research studies to further enhance different components of MOEA/Ds, such as the adjustment of weight vectors [19]-[21], modified decomposition methods [22]-[27], dynamic mating selection [28]-[31], enhanced evolutionary operators [32]-[35], and improved environmental selection [36]-[44]. Particularly, for more details, interested readers can refer to a review of MOEA/Ds [45] and some of their progress closely related to the methods proposed in this paper is detailed in Section 2.3.

Although these MOEA/Ds show excellent performance in solving regular MOPs, they still fail to achieve satisfactory results when confronted with MOPs with imbalanced features. Specifically, imbalanced MOPs offer different degrees of search difficulty in different regions of the PF. As a result, some regions of the PF are relatively easy to search, while other regions are more challenging to explore. In addition, the easy-to-find regions of PF tend to dominate a relatively large area in the search space, making it difficult to discover "supporting" solutions for the remaining parts of the PF. The imbalance of search difficulty in MOPs poses an enormous challenge to MOEAs, as they may converge prematurely to suboptimal solutions in the easier-to-find regions, neglecting the more challenging parts of the PF. Consequently, the obtained PF may be biased and the algorithm may miss the true optimal solutions. As experimentally studied in [15] and [46], the solution association methods in most MOEA/Ds will deteriorate their performance for solving imbalanced MOPs, as a few solutions may replace most of the others, leading to a significant loss of diversity. It is always a very challenging task to balance convergence and diversity in designing MOEAs [46]. In recent years, some MOEAs have been designed based on the reference vectors (same as weight vectors), e.g., NSGA-III [47], θ-DEA [48], and SPEA/R [49], which are mostly used to solve MOPs with more than three objectives. In these MOEAs, solutions are associated to the closest reference vectors and the reference vectors may be associated with none, one, or multiple solutions, which naturally provides a flexible method for solution association. However, their collaborative capability may be weakened, as solutions from the neighboring subproblems may not exist for coevolution.

As inspired by the constrained decomposition reported in [25], [37] and the reference vectors used in [47]-[49], this paper suggests a localized decomposition evolutionary algorithm (LDEA) for tackling imbalanced MOPs. A localized decomposition method is presented to strictly emphasize diversity first for each subproblem, which modifies the solution association methods in most MOEA/Ds and allows the subproblems associated with none, one, or multiple solutions. Moreover, in order to maintain the collaborative capability in MOEA/Ds, the matting selection, evolutionary operators, and environmental selection are accordingly modified to cooperate well with the above association mechanism. To summarize, the main contributions of this paper are the following:

1) A localized decomposition evolutionary algorithm is designed for solving imbalanced MOPs, which associates solutions to their closest weight vector (subproblem). A localized decomposition method is run based on the switch of weighted sum (WS) and Tchebycheff (TCH) aggregated functions to follow the principle of

diversity first and convergence second, which is very promising for solving imbalanced MOPs. The experimental results indicate that LDEA not only performs well on imbalanced MOPs (e.g., the *MOP* [15] and *IMB* [46] test suites), but also shows promising performance on balanced yet complicated MOPs (e.g., the *UF* [18] test suite).

2) The mating selection, evolutionary operators, and environmental selection are accordingly modified to cooperate well with the localized decomposition method. Only the best offspring of each subproblem is evolved in mating selection and the dynamical search step sizes are embedded in evolutionary operators to speed up convergence, while the best solution of each subproblem is compulsorily saved to maintain diversity. In this way, our LDEA can obtain evenly distributed and very close approximations to the true PFs on most of the test problems adopted.

The rest of this paper is organized as follows. In Section 2, the related background and motivation of our approach are introduced. The details of LDEA are clarified in Section 3, while the experimental results of LDEA with other competitive MOEAs and some discussions are provided in Section 4. Our conclusions and future work are presented in Section 5.

2. Related background and motivations

2.1.Multi-objective optimization problem

Multi-objective optimization problems (MOPs) involve more than one conflicting objective and give rise to a set of Pareto-optimal solutions. In this paper, MOPs without constraints are considered, as follows:

$$\underset{\mathbf{x}\in\Omega}{\text{Min}}\,\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})), \tag{1}$$

where $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ is a decision vector having *n* decision variables in $\Omega \subset \mathbf{R}^n$, and $\Omega = \prod_{i=1}^{n}[L_i, U_i]$ denotes the *n*-dimensional feasible decision space. L_i and U_i are the lower bound and upper bound of x_i (i = 1, 2, …, n), respectively. $\mathbf{F}(\mathbf{x})$ defines *m* objective functions mapping from the decision space Ω to the objective space $\mathbf{R}^m$. Due to the potential conflicts among different objectives, there exists a Pareto-optimal set (PS) with equally optimal solutions when considering all the objectives, and the mapping of the PS on the objective space is termed the Pareto-optimal front (PF) [1].

2.2.Decomposition approaches

In this paper, two popular decomposition approaches (WS [12] and TCH [50]) are used, which are respectively defined as follows:

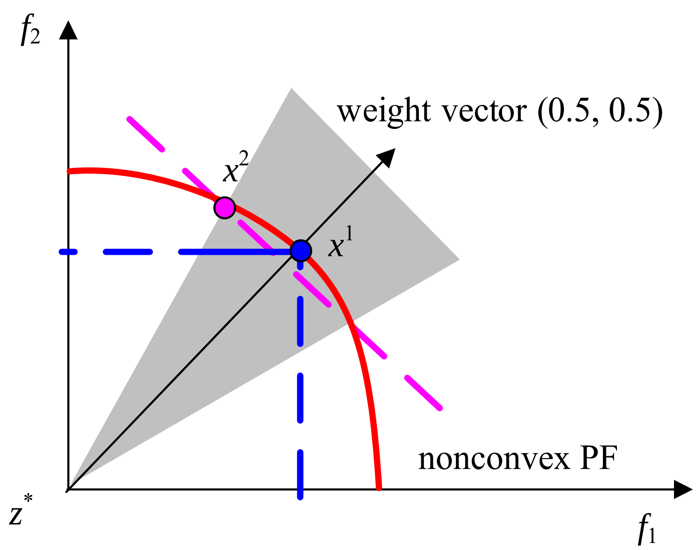
$$g^{\text{WS}}(\mathbf{x}\,|\,\mathbf{w}) = \sum_{i=1}^{m} f_i(\mathbf{x}) \times w_i, \tag{2}$$

$$g^{\text{TCH}}(\mathbf{x}\,|\,\mathbf{w},\mathbf{z}^*) = \max_{1\le i\le m}((f_i(\mathbf{x}) - z_i^*)/w_i), \tag{3}$$

where *m* is the number of objectives, $\mathbf{w}=(w_1,...,w_m)^T$ is an *m*-dimensional weight vector with $\sum_{i=1}^{m} w_i = 1$

($w_i \geq 0$), and $\mathbf{z}^* = \{z_1^*, ..., z_m^*\}$ is an ideal point with $z_i^* = \{\min f_i(\mathbf{x}) \mid \mathbf{x} \in \mathbf{\Omega}\}$ for $i = 1, ..., m$.

As experimentally studied in [51], WS would work well for MOPs with convex PFs while TCH was better on MOPs with nonconvex PFs. In [52], WS and TCH were simultaneously used when tackling MOPs with different shapes of PFs. Each subproblem was associated with two solutions, which requires a population size twice as large and consequently wastes computational resources. Recently, a localized WS [25] was suggested to enhance diversity for each subproblem, by embedding the constraints on the original WS. However, this localized WS is unable to find the exact solution associated to each subproblem in solving MOPs with nonconvex PFs, as plotted in Fig. 1. Two solutions x^1 (marked by blue) and x^2 (marked by pink) are compared to be associated with the weight vector (0.5, 0.5). The blue line and the pink line respectively give the contour lines of TCH and the localized WS constrained by the shaded background for x^1 and x^2. The solutions under the contour line are better, while those above the contour line are worse when compared to x^1 or x^2. Obviously, x^2 will be selected if the localized WS is used, while x^1 will be selected if TCH is used.



In fact, x^1 is better than x^2 when considering diversity as measured by the perpendicular distance to the weight vector. Thus, although the localized WS can speed up convergence toward the PFs, it has to be replaced by TCH at a latter evolutionary stage in order to maintain good diversity. Motivated by this observation, our localized decomposition method is designed to use WS at the beginning and then activate TCH when WS cannot bring any improvement for a subproblem in a long-running period.

Fig. 1. The difference between TCH and the localized WS when selecting the associated solutions in MOEA/D.

2.3. Related studies

Here, a summary of some representative decomposition-based MOEAs (MOEA/Ds) is provided in Table 1. Initially, the original MOEA/Ds were designed to consider convergence first and diversity second in their environmental selection methods. For example, MOEA/D [12] uses the aggregated function value for each subproblem in its environmental selection, and MOEA/D-DE [53] further sets the maximal replacement number of each offspring when replacing the parents. However, this method is only suitable for solving MOPs with uniformly distributed Pareto-optimal fronts (PFs). Thus, with further studies of MOEA/Ds, some of them were suggested to strike a balance of convergence and diversity. For example, MOEA/D-AGR [36] designs an adaptive neighborhood update strategy to better achieve co-evolution, and MOEA/D-IR [16] develops a mutual preference interrelationship for improving optimization efficiency. Besides, MOEA/D-STM [17] devises a stable matching model (STM) to obtain a balance between convergence and diversity, which is further modified in AMOSTM and AOOSTM [41] by restricting incomplete matching lists for STM. These MOEA/Ds have shown very promising performance on some complicated MOPs, such as the *UF* test suite [18]. However, they experienced a significant performance deterioration for solving imbalanced MOPs, as experimentally shown in [15], [46]. Thus, some recent MOEA/Ds have been designed based on the principle of diversity first and convergence second. For example, MOEA/D-ACD [37] and MOEA/D-LWS [25] embed the constraints to limit the updating region of each subproblem, and MOEA/D-M2M [15] restricts the solving of MOPs within each sub-region. In these MOEA/Ds, when a subproblem has no associated solution, MOEA/D-ACD will relax the constraints to associate solutions, MOEA/D-LWS will associate solutions randomly, and MOEA/D-M2M will borrow solutions from other sub-regions. Thus, some solutions may still be associated to a far-away subproblem. Under this case, the diversity of each subproblem cannot be truly reflected, and thus correct neighboring information cannot be provided to run a collaborative search, which may slow down the convergence speed and easily get trapped in local PFs, as MOEA/Ds are designed to be an essentially collaborative framework for the subproblems. Based on the above analysis of the existing studies, this paper proposes a Localized Decomposition Evolutionary Algorithm, called LDEA, to better collaborate subproblem solving and thus improve the optimization performance when solving imbalanced MOPs. More details of our proposed LDEA are provided in Section 3.

Table 1
Summary of characteristics for decomposition-based MOEAs

Algorithm	Feature	Strengths	Limitations
MOEA/D [12]	Decompose the MOP and co-optimize subproblems	Prioritizes convergence	Best for uniformly distributed PFs
MOEA/D-DE [53]	Limit the maximal offspring replacement	Better offspring-parent strategy	
MOEA/D-AGR [36]	Adaptive neighborhood update	Enhanced co-evolution	Struggle in solving imbalanced MOPs
MOEA/D-IR [16]	Mutual preference for optimization	High optimization efficiency	
MOEA/D-STM [17]	Stable matching model (STM)	Convergence-diversity balance	
MOEA/D-ACD [37]	Limit the region for subproblem updates	Prioritizes diversity	Can misrepresent subproblem diversity
MOEA/D-LWS [25]			
MOEA/D-M2M [15]	Borrow solutions from other sub-regions if needed	Solution borrowing mechanism	Risk of local PF traps, slow convergence

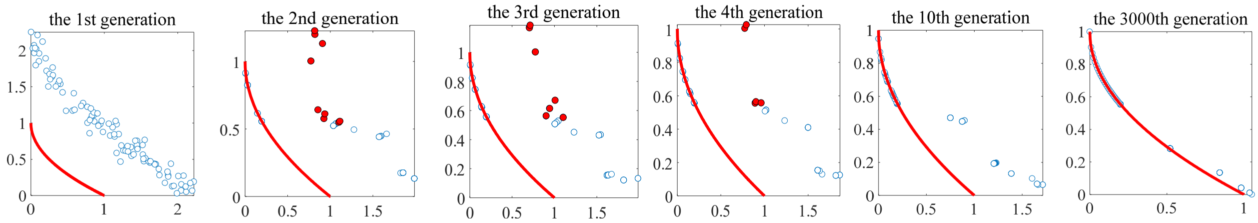
2.4.Difficulty in solving imbalanced MOPs

In the early study of MOEA/Ds [18], the imbalanced feature in PSs was used to challenge their performance, in which solutions in a small segment of PS are projected to the majority of the PF. Thus, it is easy to converge towards a segment of the PF, but it is difficult to determine its majority. In [15], this imbalanced feature was introduced into the PF, which brings great challenges for most MOEA/Ds, as a strong diversity maintenance capability in MOEA/Ds would be required. Recently, this imbalanced feature was further explained in [46] and the imbalanced property was described as follows:

1) The complexity of finding a specific subset of the PF (called the favored subset) is significantly lower than that of finding the other part of the PF (called the unfavored subset).

2) The PS of the favored subset dominates a significantly larger part of the feasible variable space than the PS of another unfavored subset.

Thus, a severe imbalance exists when achieving convergence and diversity maintenance in MOEA/Ds. Some solutions can easily replace most of the other solutions using the decomposition approach thus converging very quickly, which will significantly destroy the population's diversity. A case study of this imbalanced feature is given in Fig. 2 to show the population evolution of MOEA/D [53] at different generations on *IMB1* [46]. In this case, the population size and the total number of generations are respectively set to 100 and 3000. In Fig. 2(a), the population was randomly initialized as represented by the circles, while the PF of *IMB1* was marked using a red curve. In the second generation of Fig. 2(b), four solutions were found to have converged very closely towards the PF, which causes a replacement of other solutions (indicated by the red circles) using TCH. In Figs. 2(c)-2(e), this imbalanced evolutionary trend is obvious, as the number of solutions in the unexplored regions (above the right segment of the PF) is reduced with an increasing number of solutions having converged towards the left segment of the PF. At last, in the 3000th generation of Fig. 2(f), most solutions converge towards the left segment of the PF, while only five solutions are distributed above the right segment of the PF. Thus, the imbalanced feature of *IMB1* in Figs. 2(a)-2(f) may significantly deteriorate the population's diversity and weaken the collaborative capability in MOEA/D.



(a) (b) (c) (d) (e) (f)

Fig. 2. The population evolution of MOEA/D at different generations on *IMB1*.

Furthermore, to visually demonstrate the performance of using different aggregation functions in MOEA/D when solving MOPs with imbalanced features, the population evolution of MOEA/D with the TCH and WS aggregation functions at different generations on *IMB3* are plotted in Figs. 3 and 4, respectively. As observed from Figs. 3 and 4, MOEA/D with the TCH aggregation function performs better in terms of diversity, while MOEA/D with the WS aggregation function exhibits faster convergence in the early stage. Thus, in our proposed LDEA, WS and TCH are incorporated by a collaborative switching mechanism to balance convergence and diversity, the details of which are provided in Section 3.

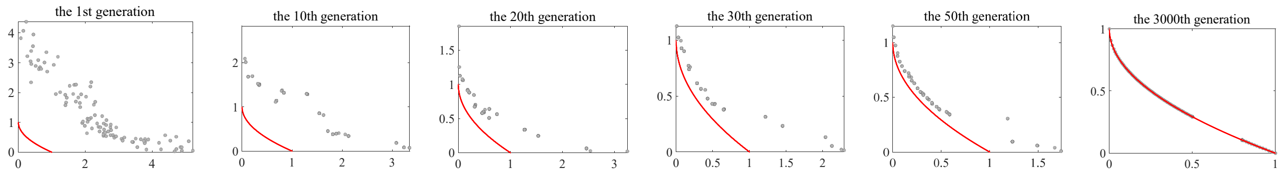


Fig. 3 The population evolution of MOEA/D with TCH aggregation function at different generations on *IMB3*.

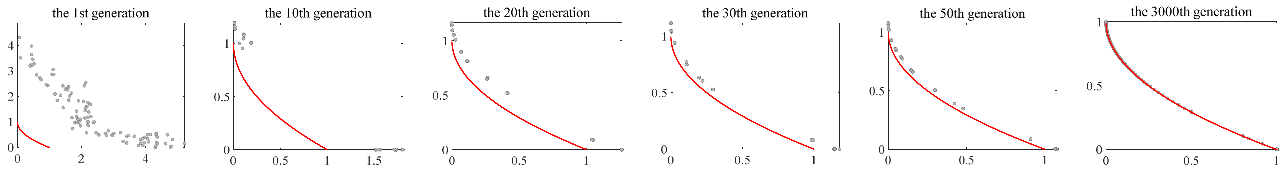


Fig. 4 The population evolution of MOEA/D with WS aggregation function at different generations on *IMB3*.

In [46], it was pointed out that the imbalanced features in MOPs would present significant challenges for MOEA/Ds and most MOEA/Ds could not solve the *IMB* test suite, while an MOEA enhanced by the multi-objective to multi-objective (M2M) method [15] can properly tackle these imbalanced MOPs. Specifically, the M2M converts an MOP into a set of MOPs by dividing the region for co-evolution, and thus effectively maintains the diversity of the population. In this way, it can effectively avoid falling into local optimum, which makes it well suited for solving imbalanced MOPs. However, in the M2M method, an empty sub-region is still assigned with solutions borrowed from other sub-regions, which cannot truly reflect diversity in each sub-region and cannot provide correct neighboring information to run the collaborative search in MOEA/Ds. Inspired by the M2M method, this paper suggests a localized decomposition evolutionary algorithm for solving imbalanced MOPs, which assigns a local region for each subproblem and performs a localized decomposition. Moreover, the mating selection, evolutionary operators, and environmental selection are accordingly modified to cooperate with the proposed localized decomposition method. The experimental results in Section 4 validate that LDEA can properly address the imbalanced MOPs (the *MOP* [15] and *IMB* [46] test suites) and balanced yet complicated MOPs (the *UF* test suite [18]), and also has promising performance on real-world MOP.

3. The details of our algorithm

In this section, the details of LDEA are introduced. First, the framework of LDEA is introduced to have an overview of the way it works. Then, the two main components of LDEA (the modified evolutionary operators and the localized decomposition method) are respectively introduced to clarify the implementation of LDEA. To be specific, the improved evolutionary operators are modified by embedding the dynamically reduced search steps, which can provide the course-grained search and fine-grained search respectively at the early and latter evolutionary stages when compared to the original version [15], [54]. Then, the localized decomposition method is run based on the switch of constrained WS and TCH, which only makes the solution update within the same local region of each subproblem.

3.1. The main framework of LDEA

Algorithm 1: LDEA

```
1  generate W={w^1,w^2,...,w^N} and C={c_1,c_2,...,c_N}.
2  initialize g=1 and get an initial population P randomly.
3  set S={Λ_1^p, Λ_2^p,..., Λ_N^p} by Eq. (4).
4  while g<G_max
5     O = Evolution(S).
6     [S, C] = Localized Decomposition(P, O, C).
8     collect all solutions in S as P.
9     g = g + 1.
10 end while
11 return P.
```

The pseudo-code of LDEA is given in **Algorithm 1**. In line 1, N weight vectors $\boldsymbol{W}=\{\mathbf{w}^1,\mathbf{w}^2,...,\mathbf{w}^N\}$ (N is the population size) are uniformly generated using the method in [55] and the performance monitoring vector $\boldsymbol{C}=\{c_1,c_2,...,c_N\}$ for WS is initialized as 0 for each c_i ($i\in[1,N]$). Then, the value of the generation counter g is set to 1 and the initial population $\boldsymbol{P}$ is randomly generated within the search space in line 2. In line 3, the set $\boldsymbol{S}$ is obtained by classifying N subsets $\Lambda_1^p,\Lambda_2^p,...,\Lambda_N^p$ from $\boldsymbol{P}$ and each subset Λ_i^p ($i\in[1,N]$) includes the solutions in $\boldsymbol{P}$ that are closest to the weight vector $\mathbf{w}^i$, as follows:

$$\Lambda_i^p=\{\mathbf{x}\in\boldsymbol{P}\,|\,\langle\mathbf{F}(\mathbf{x})-\mathbf{z}^*,\mathbf{w}^i\rangle\le\langle\mathbf{F}(\mathbf{x})-\mathbf{z}^*,\mathbf{w}^j\rangle\}\quad\text{for}\quad\forall j\in[1,N], \tag{4}$$

where $\mathbf{z}^*=(z_1^*,z_1^*,...,z_m^*)$ (m is the number of objectives) is an ideal objective vector that is estimated by the minimal values of all the objectives from $\boldsymbol{P}$, i.e., $z_k^*=\{\min f_k(\mathbf{x})\,|\,\mathbf{x}\in\boldsymbol{P}\}$ for each $k\in[1,m]$, and $\langle\mathbf{F}(\mathbf{x})-\mathbf{z}^*,\mathbf{w}^i\rangle$ returns the acute angle of two vectors $\mathbf{F}(\mathbf{x})-\mathbf{z}^*$ and $\mathbf{w}^i$, as follows:

$$\langle\mathbf{F}(\mathbf{x})-\mathbf{z}^*,\mathbf{w}^i\rangle=arccos\left|\frac{\sum_{k=1}^{m}\left(f_k(\mathbf{x})-z_k^*\right)\cdot w_k^i}{\sqrt{\sum_{k=1}^{m}\left(f_k(\mathbf{x})-z_k^*\right)^2}\cdot\sqrt{\sum_{k=1}^{m}\left(w_k^i\right)^2}}\right|, \tag{5}$$

The termination condition (e.g., the generation counter g is smaller than the maximum number of generations G_{max} in this paper) will be checked in line 4. If it is not reached, the main evolutionary process of LDEA will be run in lines 5-9. In line 5, the modified evolutionary operators (**Algorithm 2**) are performed on $\boldsymbol{S}$ to get an offspring population $\boldsymbol{O}$, the details of which are given in Section 3.2. Then, the proposed localized decomposition method (**Algorithm 3**) is executed in line 6 to update the set $\boldsymbol{S}$ and the performance monitoring vector $\boldsymbol{C}$, the details of which are provided in Section 3.3. In line 8, $\boldsymbol{P}$ is updated to include all the solutions in $\boldsymbol{S}$ and the generation number g is increased by 1 in line 9.

Finally, after the completion of the above evolutionary process in lines 5-9, the final population ***P*** will be reported as the approximation set in line 11. For ease of understanding, the outline of LDEA is provided in Fig. 5, which visually illustrates the running process of LDEA.

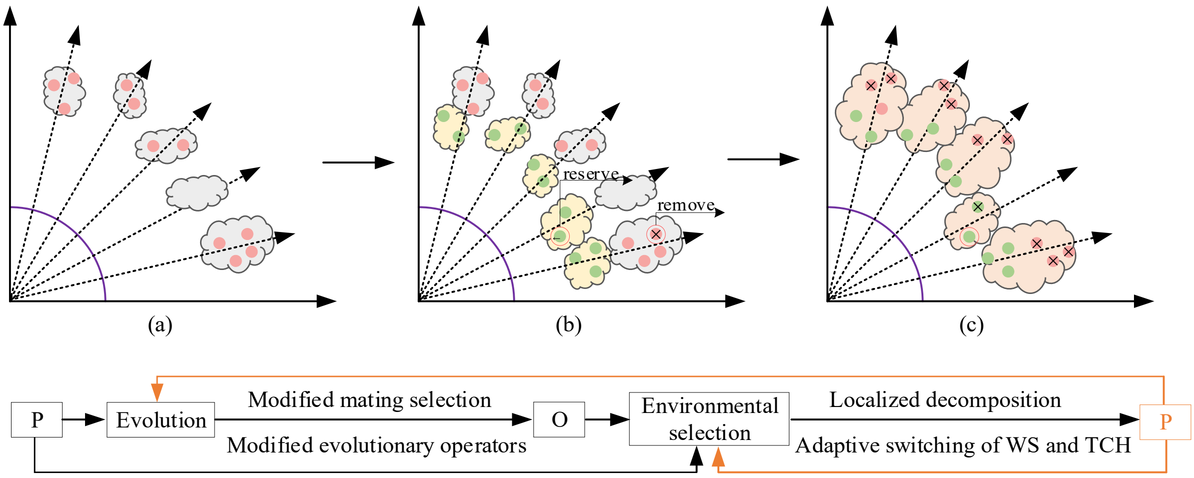


Fig. 5. Outline of the proposed LDEA for solving imbalanced MOPs.

3.2. The modified evolutionary operators

The evolutionary operators used in this paper are modified from [15] by embedding the dynamically reduced search steps, which try to cooperate well with the localized decomposition method and enhance the overall search capability of LDEA, as run in line 5 of **Algorithm 1**. Here, its details are introduced by providing the pseudo-code in **Algorithm 2** with the input: ***S*** (the subsets respectively associated to the subproblems) and the output: ***O*** (the generated offspring population from ***S***).

Algorithm 2: Evolution(*S*)

1 set ***O*** as an empty set and obtain the mating neighborhood size T by Eq. (6).
2 **for** i =1 to N
3   collect the best solutions in T neighbors of Λ_i^p as a mating pool ***A***.
4   **for** j =1 to $|\Lambda_i^p|$.
5   set the best solution of Λ_i^p as x^1.
6   **if** $r <= g/G_{max}$ or $|\boldsymbol{A}| \neq T$
7   select x^2 and x^3 from ***A*** randomly.
8   **Else**
9   select x^2 and x^3 from ***S*** randomly.
10   **end if**
11   get an offspring with x^1, x^2 and x^3 using Eqs. (7)-(8).
12   add this offspring into ***O***, evaluate it and update $\mathbf{z}^*$.
13   **end for**
14 **end for**
15 **return *O***.

In line 1, the offspring population ***O*** is first initialized as an empty set and then the mating neighborhood size T is obtained. As each subset Λ_i^p ($i \in [1, N]$) in ***S*** collects the solutions associated to the ith subproblem (i.e., the weight vector $\mathbf{w}^i$), some subsets may be empty according to Eq. (4) due to the imbalanced distribution of solutions. Thus, the traditional mating selection in most MOEA/Ds [12], [37], [53] cannot be properly used, as they associate each subproblem with one solution. Here, the mating selection in LDEA is modified. At first, the mating neighborhood size *T* is set as follows:

$$T = \lfloor T_m \times (G_{max} - g) / G_{max} \rfloor + 1, \quad (6)$$

where T_m is a pre-set maximal integer for the neighborhood size, $\lfloor s \rfloor$ is a floor operator returning the maximal integer no larger than *s*, while G_{max} and g are respectively the maximal generation and the current generation. T_m is determined by the population size *N*, usually set to $0.1 \times N$. Using Eq. (6), the mating neighborhood size *T* will be gradually reduced from T_m to 1. By setting a larger neighborhood size *T* at early search stages, more neighbors are used to encourage a global search. In contrast, a smaller neighborhood size *T* is used at the latter stages of the search by only including a few close neighbors to emphasize a local search.

Then, for each *i*th subset of ***S*** in line 2, the best solution in each of *T* neighbors of Λ_i^p (i.e., the *T* subsets with their weight vectors closest to the weight vector $\mathbf{w}^i$) is collected in line 3 as a mating pool ***A***. For each Λ_i^p in line 4, the best solution x^1 in Λ_i^p is selected as a representative for this subset, which is evolved during $|\Lambda_i^p|$ times in lines 6-12 by the modified evolutionary operators. To be specific, in lines 6-10, when a randomly generated number *r* in (0, 1) is smaller than g/G_{max} or at least one neighboring subproblem has no associated solution (i.e., $|\boldsymbol{A}| \neq T$), two mating parents x^2 and x^3 are randomly selected from ***A*** in line 7. Otherwise, two mating parents x^2 and x^3 are randomly selected from ***S*** in line 9.

After that, an offspring can be generated by x^1, x^2 and x^3, as follows:

$$y_i = x_i^1 + r_1(1-(r_2)^\rho)(x_i^2 - x_i^3), \quad (7)$$

where r_1 and r_2 are two random real numbers respectively generated from (-1, 1) and (0, 1), and ρ is an index that is set to $-(1-g/G_{max})^{0.7}$ as suggested in [15] and [49]. This operator is called generation-related differential evolution (*g*-DE) here.

Then, the mutation operator modified from [15] and [49], called generation-related mutation (*g*-MUT) here, is defined as follows:

$$y_i = y_i + r_1\sigma(1-(r_2)^\rho)(u_i - l_i), \quad (8)$$

where r_1 and r_2 are two random real numbers respectively generated from (-0.5, 0.5) and (0, 1), and ρ is an index that is set to $-(1-g/G_{max})^{0.7}$ as suggested in [15] and [49]. The mutation step size σ is also dynamically reduced with the generation number, and is set to $(1-g/G_{max})$ in this paper.

In line 12 of **Algorithm 2**, this offspring is added into ***O*** and its objective values are evaluated to update the ideal point $\mathbf{z}^*$ used in TCH. Finally, when *N* offspring are produced in lines 2-14, the offspring population ***O*** is returned in line 15.

3.3.The localized decomposition method

Algorithm 3: [*S*, *C*]= Localized Decomposition(*P*, *O*, *C*)

```
1  get each Λ_i^p and Λ_i^o respectively from P and O by Eqs. (4)-(5).
2  obtain the number of solutions n_i for each subset Λ_i^p by Eq. (9).
3  set sum = Σ_{i=1}^N n_i .
4  while sum > N
5     find a kth subproblem with the maximal number of solutions by Eq. (10).
6     set n_k = n_k − 1 and sum = sum − 1 .
7  end while
8  set Λ_i^p=Λ_i^p ∪ Λ_i^o for each i ∈ [1, N] .
9  for i=1 to N
10    if n_i > 0
11        decide the decomposition approach for ith subproblem.
12        reserve n_i best solutions from Λ_i^p for ith subproblem.
13        update the performance monitoring value c_i by Eq. (11).
14    end if
15 end for
16 set S={ Λ_1^p, Λ_2^p,..., Λ_N^p }.
17 return S, C.
```

The pseudo-code of our localized decomposition method is given in **Algorithm 3** with the inputs: the original population ***P***, the offspring population ***O***, and the performance monitoring vector ***C*** for WS in each subproblem, which is run in line 6 of **Algorithm 1** in order to update the original population. In line 1, *N* subsets $\Lambda_1^p, \Lambda_2^p, ..., \Lambda_N^p$ from ***P*** and *N* subsets $\Lambda_1^o, \Lambda_2^o, ..., \Lambda_N^o$ from ***O*** are obtained by using Eqs. (4)-(5). In this way, each subset Λ_i^p (or Λ_i^o) ($i \in [1, N]$) includes the solutions in ***P*** (or ***O***) that are closest to the weight vector $\mathbf{w}^i$, which has been used in some constrained decomposition methods [25], [37] and some MOEAs based on the reference vectors [47]-[49]. Once offspring are generated to associate with an empty subproblem in ***P***, the best offspring from ***O*** should be added into ***P*** to extend diversity. Thus, in line 2, the number of solutions n_i in each subset Λ_i^p is updated by including the best offspring from ***O*** in the originally empty local region, as follows:

$$n_i = \begin{cases} 1, & \text{if } |\Lambda_i^p| == 0 \text{ and } |\Lambda_i^o| \text{ !=0} \\ |\Lambda_i^p|, & \text{otherwise} \end{cases}. \tag{9}$$

Then, in line 3, the summation of all n_i ($i \in [1, N]$) is obtained by getting $sum = \sum_{i=1}^{N} n_i$. If sum is larger than the population size *N* in line 4, the *k*th subproblem with the maximal number of associated solutions is identified in line 5, as follows:

$$k = \{i : \arg\max |\Lambda_i^p|,\ i = 1, 2, ..., N\}. \tag{10}$$

If there is more than one subproblem in Eq. (10), only one of them is randomly selected. After that, the solution with the worst aggregated value will be removed from the subset $|\Lambda_k^p|$. Then, the subset size n_k indicating the number of associated solutions and the value of the *sum* are reduced by 1 in line 6, which won't affect the convergence of all subproblems, as their best solutions are still reserved. After the running of lines 4-7, the total number of associated solutions in all the subsets is *N*. Then, the solutions in each subset Λ_i^o are respectively combined into the subset Λ_i^p ($i \in [1, N]$) in line 8 for selecting the n_i solutions with the best-aggregated values from each *i*th subproblem.

In lines 9-15, for each subset Λ_i^p, its aggregated function is decided in line 11. If the performance monitoring value c_i is smaller than a preset value (i.e., ke^m in Eq. (11), where *k* is set to 10 in this paper and *m* is the number of objectives), WS is used for the *i*th subproblem; otherwise, TCH is selected. By introducing the performance monitoring vector ***C***, LDEA can adaptively switch the aggregated functions between WS and TCH to balance convergence and diversity, which is highly important for solving imbalanced MOPs. Then, in line 12, the n_i best solutions are reserved into the subset Λ_i^p based on their aggregated function values, followed by updating the corresponding performance monitoring value c_i for WS in line 13, as follows:

$$c_i = \begin{cases} 0, & \text{if } g^{\mathrm{WS}}(p^{best} \mid \mathbf{w}^i) > g^{\mathrm{WS}}(o^{best} \mid \mathbf{w}^i) \text{ and } c_i < ke^m \\ c_i + 1, & \text{otherwise} \end{cases}, \tag{11}$$

where p^{best} and o^{best} denote the best solutions respectively in Λ_i^p and Λ_i^o. At last, all the solutions in $\Lambda_1^p, \Lambda_2^p, \ldots, \Lambda_N^p$ and the performance monitoring vector ***C*** are returned in line 17.

4. Experimental studies

4.1. Benchmark problems and parameter settings

In this study, two imbalanced test suites (*MOP1-MOP7* [15] and *IMB1-IMB10* [46]) and one balanced yet complicated test suite (*UF1-UF10* [18]) were used to assess the performance of LDEA. The test problems used in our experimental study include eighteen two-objective problems, i.e., *IMB1-IMB3*, *IMB7-IMB9*, *MOP1-MOP5*, and *UF1-UF7*, and nine test problems with three objectives, namely, *IMB4-IMB6*, *IMB10*, *MOP6-MOP7*, and *UF8-UF10*. These adopted test problems have very complicated mathematical features on their PSs and PFs, which can challenge the capabilities of MOEAs for achieving convergence and diversity maintenance. For illustration purposes, a summary of the basic characteristics of these test problems and the corresponding parameter settings is presented in Table 2.

Table 2
Summary of characteristics for test problems and corresponding parameter settings

Test problem	*m*	*n*	*N*	MaxFE	Type	PF Geometry
IMB1-IMB3	2	10	100	3×10^4	Imbalanced MOPs	convex/linear/concave
IMB7-IMB9	2	10	100	3×10^4		convex/linear/concave
IMB4-IMB6/IMB10	3	10	300	9×10^4		linear/concave/linear/concave
MOP1-MOP5	2	10	100	3×10^4		convex/concave/concave/disconnected/convex
MOP6-MOP7	3	10	300	9×10^4		linear/concave
UF1-UF3	2	30	100	3×10^4	Complicated MOPs	convex/convex/convex/
UF4-UF7						concave/disconnected/ disconnected/linear
UF8-UF10	3	30	300	9×10^4		disconnected/disconnected/disconnected
M3O	2	4	50	1×10^3	Real-world MOPs	—

As suggested in [15], the population size was set to 100 for the two-objective test problems and 300 for the three-objective test problems, while the maximum number of generations was set to 3000 for each problem. The number of decision variables was set to 30 for all the *UF* test problems and it was set to 10 for all the *IMB* and *MOP* test problems. As shown in the experiments of [15], some traditional MOEAs emphasizing convergence were found to be unsuitable for solving these imbalanced MOPs due to their loss of diversity. Thus, six competitive MOEAs with more emphasis on diversity i.e., MSEA [56], AOOSTM [41], AMOSTM [41], OPE [57], ACD [37], and SPEA/R [49] were selected for comparison with LDEA. As suggested in the original references, SPEA/R employed the evolutionary operators used in [54] with the same parameters adopted in Eqs. (7)-(8), where the crossover probability is 1.0. In particular, the parameters of MSEA and OPE are all the same as those of their original references [56], [57]. The other three compared MOEAs used *DE/rand/1* (DE) with the scaling factor F=0.5 and the crossover rate CR=1.0, and polynomial-based mutation (PM) with the mutation index set to 20 [4]. LDEA executed g-DE in Eq. (7) and g-MUT in Eq. (8), respectively, as the crossover and mutation operators, in which the crossover probability was set to 1.0 and the mutation was randomly run in one decision variable of each solution. After the evolutionary operators, if a decision variable was offside, it was reset to the closest boundary value. Each algorithm was run 30 times for each test problem to collect the mean results and standard deviations for comparison. Other parameters of all the compared algorithms are given in Table 3, where δ is the probability of selecting the neighbors for generating offspring, T means the neighborhood range for mating selection and environmental selection in MOEA/Ds, and T_m is only the mating neighborhood size. In AOOSTM & AMOSTM, the generation interval to update the utility of each subproblem [28] was set to 30 [41]. In SPEA/R, the size of the mating pool M_T was set to select the 20 closest solutions [49]. For LDEA, k in Eq. (11) was set to 10. Please note that LDEA was run in Matlab, MSEA, OPE, and SPEA/R were run in PlatEMO[1] [58], while AOOSTM, AMOSTM, and ACD were run in jMetal[2] [59].

4.2. Performance indicators

In order to provide a comprehensive assessment on the performance of all the compared algorithms, two widely used performance indicators (inverted generational distance (IGD) [60] and Hypervolume (HV) [61]) were adopted to assess the convergence and diversity of the final solution set. Specifically, the specific definitions of IGD and HV can be found in [4] and [62], respectively. In general, a smaller value of IGD and a larger value of HV indicate a better performance to closely approximate the PF with a more uniform distribution. No less than 500 sampling points from the true PF were used to compute IGD, while the reference points were respectively set to $(1.1,1.1)^T$ and $(1.1,1.1,1.1)^T$ for the two-objective and the three-objective test problems, in order to compute HV, as suggested in [63]-[64].

---

[1] https://github.com/BIMK/PlatEMO.

[2] https://github.com/MelonNg/EMOStudy_jMetal.

Table 3
Parameter settings of all the compared algorithms

Algorithm	Parameter settings
MSEA	——
OPE	——
AOOSTM&AMOSTM	T=0.1N, δ=0.9
ACD	T=0.1N, δ=0.9
SPEA/R	M_T=20
LDEA	k=10, T_m=0.1N

4.3.Performance comparisons with six competitive MOEAs

Table 4 provides the mean results and standard deviations of the compared algorithms on all the test problems regarding IGD, where the statistically best mean results for each problem are highlighted in **boldface**. The Wilcoxon's rank sum test was run with a 5% significance level to show whether there exists a statistically significant difference on the results obtained by LDEA and each competitor. The signs "—", "+" and "=" in the second to last row of Table 4 summarize the number of test problems in which LDEA respectively performed better than, worse than, and similarly to each competitor, while the last row of Table 4 concludes the percentages of obtaining the statistically best mean results on all the test problems.

On the *MOP* and *IMB* test problems with imbalanced features, the advantages of LDEA were obvious, as it obtained the best IGD results on *IMB1-IMB3*, *IMB7*, *IMB10,* and *MOP1-MOP7*. AMOSTM achieved the best IGD results on *IMB4-IMB6*, while SPEA/R produced the best IGD results on *IMB7-IMB9*. Other compared algorithms could not perform best on any of *IMB* and *MOP* problems. AMOSTM, AOOSTM, and ACD employed DE and PM as their evolutionary operators, and they were designed to emphasize diversity, which produced approximations with IGD values mostly under an accuracy of 10^{-2}. Similarly, MSEA places greater emphasis on diversity by designing a multi-stage evolutionary strategy, but ignores the importance of convergence, which also produces approximations with IGD values mostly under 10^{-2} accuracy. Although OPE tries to balance convergence and diversity by designing an adaptive resource allocation strategy, its performance on imbalanced MOPs is still not satisfactory. In contrast, SPEA/R obtained approximations with IGD values under an accuracy of 10^{-3} on *IMB7-IMB9* and *MOP2-MOP4*. From these experimental results, SPEA/R seemed better than MSEA, AMOSTM, AOOSTM, OPE, and ACD on the imbalanced test MOPs, as it adopted evolutionary operators in [54], which encouraged a fine-grained search at the latter stages of the search. LDEA inherited the advantages of M2M [15] and SPEA2/R [54] by using the localized decomposition method based on WS and TCH. Moreover, LDEA modified the evolutionary operators, which also run the fine-grained search at the latter stages of the search and could find approximate solutions closer to the true PF. Thus, LDEA could obtain better approximations with IGD

Table 4
The IGD comparison results of the compared algorithms on all the test problems

Problem/IGD		MSEA	AOOSTM	AMOSTM	OPE	ACD	SPEA/R	LDEA
IMB1	Mean	1.98E-01 —	1.09E-2 —	1.13E-2 —	2.85E-2 —	1.31E-2 —	1.28E-2 —	**4.75E-3**
	Std	3.34E-04	6.00E-4	1.02E-3	1.77e-3	6.84E-4	2.35E-3	**5.13E-5**
IMB2	Mean	1.14E-01 —	4.39E-2 —	4.53E-2 —	3.93E-2 —	6.13E-2 —	1.29E-2 —	**4.97E-3**

	Std	1.85E-04	2.22E-2	2.14E-2	7.21e-3	2.11E-2	1.17E-3	**5.80E-5**
IMB3	Mean	2.17E-01 —	1.73E-2 —	1.66E-2 —	6.60E-2 —	2.30E-2 —	2.05E-2 —	**6.28E-3**
	Std	1.90E-04	1.46E-3	1.52E-3	4.33e-3	2.13E-3	9.67E-4	**1.23E-4**
IMB4	Mean	1.04E-1 —	2.35E-2 +	**2.34E-2 +**	6.02E-2 —	2.79E-2 —	5.34E-2 —	2.47E-2
	Std	1.50E-2	1.98E-4	**1.63E-4**	2.55e-3	9.07E-4	3.22E-3	2.19E-4
IMB5	Mean	8.09E-2 —	3.03E-2 +	**3.03E-2** +	6.76E-2 —	7.47E-2 —	1.09E-1 —	3.14E-2
	Std	7.76E-5	2.79E-4	**2.72E-4**	6.16e-3	7.82E-3	5.00E-3	1.92E-4
IMB6	Mean	4.40E-2 —	2.34E-2 +	**2.34E-2** +	3.07E-2 —	2.50E-2 —	5.29E-2 —	2.41E-2
	Std	6.13E-5	1.53E-4	**1.24E-4**	5.35e-4	3.37E-4	2.39E-3	1.51E-4
IMB7	Mean	2.90E-2 —	2.74E-2 —	2.56E-2 —	2.23E-2 —	2.90E-2 —	**5.40E-3** =	**5.38E-3**
	Std	2.27E-4	7.06E-3	8.90E-3	7.42e-3	6.21E-3	**4.20E-4**	**1.38E-4**
IMB8	Mean	3.40E-2 —	2.82E-2 —	2.82E-2 —	2.74E-2 —	3.53E-2 —	**5.55E-3** +	5.86E-3
	Std	4.34E-4	1.08E-2	1.13E-2	7.20e-3	4.12E-3	**4.97E-4**	2.22E-4
IMB9	Mean	3.23E-2 —	3.68E-2 —	3.56E-2 —	2.99E-2 —	3.92E-2 —	**5.44E-3** +	7.12E-3
	Std	3.66E-4	7.02E-3	8.97E-3	5.34e-3	1.04E-3	**5.19E-4**	2.24E-4
IMB10	Mean	3.96E-1 —	3.07E-2 —	3.13E-2 —	2.00E-1 —	3.15E-2 —	5.43E-2 —	**2.73E-2**
	Std	5.52E-2	9.06E-4	9.49E-4	5.79e-3	9.51E-4	2.80E-3	**3.01E-4**
MOP1	Mean	3.63E-1 —	2.56E-2 —	2.55E-2 —	1.43E-2 —	2.71E-2 —	1.20E-2 —	**7.98E-3**
	Std	4.50E-3	1.97E-3	2.00E-3	8.90e-4	2.67E-3	3.25E-4	**1.62E-4**
MOP2	Mean	3.55E-1 —	2.69E-2 —	1.77E-2 —	3.60E-2 —	1.63E-2 —	7.26E-3 —	**4.45E-3**
	Std	1.71E-1	5.13E-2	3.46E-2	6.35e-2	2.02E-2	4.04E-4	**8.19E-4**
MOP3	Mean	4.36E-1 —	7.92E-3 —	9.51E-3 —	5.86E-2 —	9.18E-3 —	7.95E-3 —	**4.48E-3**
	Std	2.15E-2	1.01E-2	1.03E-2	9.77e-2	1.43E-2	4.63E-4	**1.62E-4**
MOP4	Mean	3.20E-1 —	1.28E-2 —	1.96E-2 —	2.57E-2 —	6.09E-2 —	5.03E-3 —	**4.21E-3**
	Std	8.75E-3	5.36E-3	2.35E-2	1.92e-2	6.07E-2	5.02E-4	**5.00E-4**
MOP5	Mean	2.88E-1 —	2.15E-2 —	2.11E-2 —	2.18E-2 —	2.34E-2 —	1.95E-2 —	**7.36E-3**
	Std	2.90E-2	2.40E-3	1.86E-3	1.16e-3	1.58E-3	2.74E-3	**1.64E-4**
MOP6	Mean	3.09E-1 —	4.01E-2 —	3.95E-2 —	4.25E-2 —	4.99E-2 —	5.91E-2 —	**2.69E-2**
	Std	1.54e-6	8.62E-4	8.50E-4	5.65e-4	1.68E-3	3.78E-3	**2.48E-4**
MOP7	Mean	3.57E-1 —	6.84E-2 —	6.76E-2 —	8.58E-2 —	2.34E-1 —	1.13E-1 —	**3.58E-2**
	Std	1.10E-6	1.53E-3	1.85E-3	2.90e-3	2.36E-2	5.51E-3	**3.48E-4**
UF1	Mean	8.61E-2 —	4.99E-3 —	4.89E-3 —	3.43E-2 —	6.53E-3 —	8.33E-3 —	**4.42E-3**
	Std	1.05E-2	3.61E-4	1.89E-4	4.06e-3	2.94E-4	3.19E-3	**1.57E-4**
UF2	Mean	3.89E-2—	9.89E-3 —	1.31E-2 —	1.76E-2 —	1.44E-2 —	1.36E-2 —	**8.12E-3**
	Std	1.31E-2	4.22E-3	1.91E-2	3.26e-3	2.15E-3	2.39E-3	**1.03E-3**
UF3	Mean	2.79E-1 —	**9.72E-3** =	2.04E-2 —	8.18E-2 —	**1.26E-2** =	2.18E-2 —	**1.17E-2**
	Std	3.17E-2	**6.24E-3**	2.32E-2	1.01e-2	**6.74E-3**	1.71E-2	**3.94E-3**
UF4	Mean	4.25E-2 —	6.26E-2 —	6.60E-2 —	4.74E-2 —	7.14E-2 —	4.21E-2 —	**4.00E-2**
	Std	1.46E-3	5.72E-3	6.01E-3	1.99e-3	8.06E-3	4.28E-4	**5.00E-4**
UF5	Mean	2.73E-1—	3.04E-1 —	2.81E-1 —	2.70E-1 —	4.60E-1 —	**1.64E-1** =	**1.60E-1**
	Std	8.37E-2	1.04E-1	1.34E-1	6.62e-2	1.31E-1	**1.98E-2**	**1.27E-2**
UF6	Mean	1.97E-1 —	1.47E-1 —	1.91E-1 —	1.52E-1	1.90E-1 —	1.02E-1 —	**1.20E-2**
	Std	1.02E-1	8.83E-2	1.57E-1	7.10e-2	8.62E-2	2.26E-2	**5.99E-3**
UF7	Mean	1.57E-1 —	**6.09E-3** +	7.62E-3 +	1.45E-2 —	8.52E-3 =	9.08E-3 —	8.12E-3
	Std	1.52E-1	**1.00E-3**	4.81E-3	2.70e-3	2.58E-3	1.53E-3	1.54E-3
UF8	Mean	2.36E-1 —	**4.30E-2** +	4.69E-2 +	5.73E-2 +	5.82E-2 +	9.53E-2 —	8.15E-2
	Std	6.64E-2	**1.09E-2**	1.21E-2	7.22e-3	1.05E-2	2.05E-2	3.13E-2
UF9	Mean	1.66E-1 —	1.54E-1 —	1.21E-1 —	**3.17E-2 +**	1.25E-1 —	1.37E-1 —	3.71E-2
	Std	6.98E-2	6.31E-2	7.91E-2	**3.55e-2**	4.27E-2	5.07E-2	1.01E-2
UF10	Mean	3.58E-1 =	4.42E-1 —	8.33E-1 —	**3.20E-1 =**	4.84E-1 —	4.95E-1 —	**3.12E-1**
	Std	1.03E-1	9.70E-2	3.31E-1	**7.12e-2**	6.91E-2	5.44E-2	**4.79E-2**
—/=/+		26/1/0	21/1/5	22/0/5	24/1/2	24/2/1	23/2/2	
best/all		0/27	3/27	3/27	2/27	1/27	4/27	19/27

values having an accuracy of 10^{-3} for all the two-objective problems (*IMB1-IMB3*, *IMB7-IMB10*, and *MOP1-MOP5*) and with an accuracy of 10^{-2} for all the three-objective problems (*IMB4-IMB5*, and *MOP6-MOP7*), which are actually very close to the true PF of each problem. To visually show the performance of LDEA, Fig. 6 plots the final non-dominated solution sets obtained by LDEA with the best IGD results, when solving all the *IMB* and *MOP* test problems. These plots show that the final solutions (indicated by the blue circles) of LDEA are distributed evenly and very close to the true PFs (indicated by the red lines).

To visually show the running mechanisms of LDEA, the evolution of the population (indicated by the blue circles) is plotted in Fig. 7 for tackling *IMB8* with its PF indicated by the red lines. Most of the subproblems were optimized by LDEA with WS in the 500th generation of Fig. 7(a). Then, in Figs. 7(b)-5(c), the solutions close to the right and left sides of the PF in the 1000th and 1500th generations gradually used

TCH to converge with good diversity in the upper and lower parts of the PF, while the solutions in the central region were still far away, which still adopted WS to speed up their convergence. In Figs. 7(d)-5(e), the solutions in the central region also gradually evolved to approximate the central parts of PF. When the evolution was terminated in Fig. 7(f), LDEA could produce a good approximation set that can entirely cover the true PF.

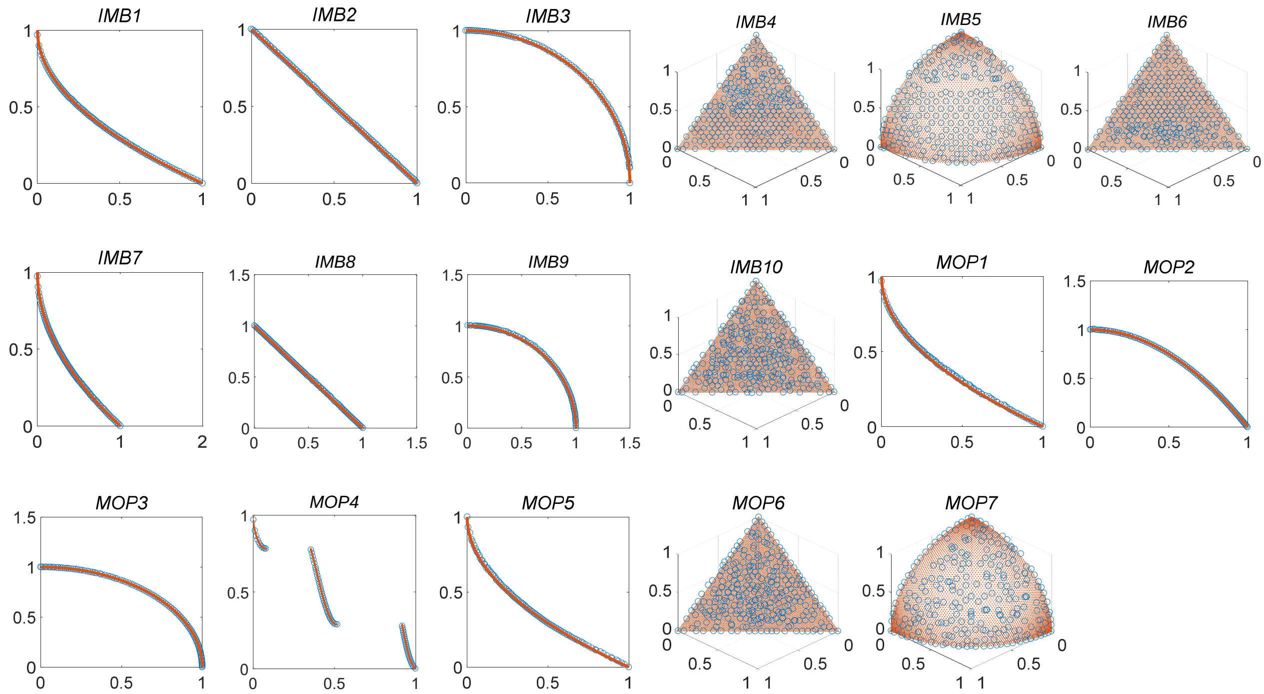


Fig. 6. The final non-dominated solution sets of LDEA with the best IGD values on all the *IMB* and *MOP* test problems.

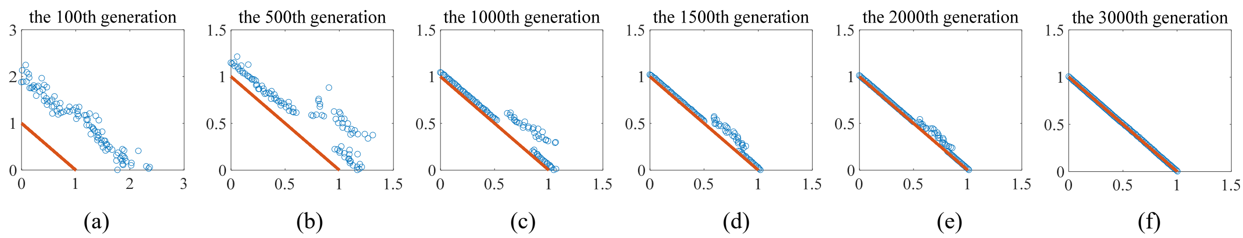


Fig. 7. The population evolution of LDEA at different generations on *IMB8*.

For the UF test problems with balanced yet complicated features in PSs, MSEA, AOOSTM, AMOSTM, OPE, ACD, SPEA/R, and LDEA obtained the statistically best IGD results respectively on 0, 3, 0, 2, 1, 1, and 7 cases. Thus, LDEA still showed obvious advantages on these UF test problems, as it was only worse on UF7-UF9. Due to the contradicting (imbalanced and balanced) features in the IMB, MOP, and UF test problems, it is very challenging to solve all the used test problems simultaneously. For example, AMOSTM and SPEA/R performed relatively better than MSEA, AOOSTM, OPE, and ACD on most of the IMB and MOP test problems, but they performed relatively worse than AOOSTM, OPE, and ACD on most of the UF test problems. However, LDEA showed obvious advantages on both types of test MOPs, which validated the effectiveness of our localized decomposition method in cooperation with the modified evolutionary operators.

Table 5
The HV comparison results of all the algorithms on all the test problems

Problem/HV		MSEA	AOOSTM	AMOSTM	OPE	ACD	SPEA/R	LDEA
IMB1	Mean	5.17E-1 —	7.11E-1 —	7.10E-1 —	6.90E-01 —	7.09E-1 —	7.07E-1 —	**7.18E-1**
	Std	4.41E-4	6.58E-4	1.07E-3	1.74E-03	7.08E-4	3.10E-3	**7.09E-5**
IMB2	Mean	4.54E-1 —	5.29E-1 —	5.27E-1 —	5.34E-01 —	5.08E-1 —	5.68E-1 —	**5.79E-1**
	Std	2.16E-4	2.85E-2	2.78E-2	9.23E-03	2.61E-2	1.68E-3	**9.17E-5**
IMB3	Mean	2.05E-1 —	3.27E-1 —	3.28E-1 —	2.80E-01 —	3.24E-1 —	3.30E-1 —	**3.42E-1**
	Std	1.02E-4	1.85E-3	1.93E-3	1.83E-03	1.97E-3	4.84E-4	**2.18E-4**
IMB4	Mean	7.85E-1 —	8.54E-1 +	**8.54E-1** +	8.02E-01 —	8.46E-1 —	8.12E-1 —	8.50E-1
	Std	9.36E-3	4.35E-4	**3.71E-4**	3.41E-03	1.26E-3	3.72E-3	3.35E-4
IMB5	Mean	5.48E-1 —	5.77E-1 +	**5.77E-1** +	5.46E-01 —	5.42E-1 —	4.46E-1 —	5.76E-1
	Std	2.19E-4	8.31E-4	**8.22E-4**	1.48E-03	8.83E-4	6.85E-3	2.24E-4
IMB6	Mean	8.43E-1 —	8.54E-1 +	**8.54E-1** +	8.33E-01 —	8.52E-1 —	8.13E-1 —	8.53E-1
	Std	1.14E-4	2.70E-4	**2.34E-4**	1.02E-03	3.99E-4	3.01E-3	1.79E-4
IMB7	Mean	6.97E-1 —	6.97E-1 —	6.98E-1 —	7.01E-01 —	6.95E-1 —	**7.17E-1** =	**7.17E-1**
	Std	4.95E-04	6.28E-3	7.98E-3	5.59E-03	5.52E-3	**7.33E-4**	**2.48E-4**
IMB8	Mean	5.46E-1 —	5.52E-1 —	5.51E-1 —	5.52E-01 —	5.44E-1 —	**5.78E-1** =	**5.78E-1**
	Std	1.11E-3	1.24E-2	1.32E-2	6.75E-03	4.91E-3	**8.72E-4**	**3.68E-4**
IMB9	Mean	3.15E-1 —	3.13E-1 —	3.13E-1 —	3.16E-01 —	3.12E-1 —	**3.43E-1** +	3.40E-1
	Std	8.21E-4	5.96E-3	7.50E-3	3.40E-03	1.66E-3	**1.10E-3**	3.95E-4
IMB10	Mean	3.92E-1 —	8.45E-1 —	8.44E-1 —	6.34E-01 —	8.44E-1 —	8.15E-1 —	**8.47E-1**
	Std	1.08E-1	1.03E-3	1.08E-3	1.11E-02	1.05E-3	3.07E-3	**3.76E-4**
MOP1	Mean	2.38E-1 —	6.93E-1 —	6.94E-1 —	7.06E-01 —	6.92E-1 —	7.08E-1 —	**7.14E-1**
	Std	9.23E-3	2.35E-3	2.18E-3	9.88E-04	2.80E-3	4.04E-4	**2.10E-4**
MOP2	Mean	1.74E-1 —	4.18E-1 —	4.28E-1 —	4.15E-01 —	4.27E-1 —	4.39E-1 —	**4.44E-1**
	Std	5.70E-2	5.37E-2	3.39E-2	5.53E-02	1.98E-2	5.75E-4	**3.38E-4**
MOP3	Mean	1.11E-1 —	3.41E-1 —	3.38E-1 —	2.98E-01 —	3.39E-1 —	3.41E-1 —	**3.46E-1**
	Std	1.97E-2	1.45E-2	1.50E-2	8.44E-02	2.08E-2	5.18E-4	**1.65E-4**
MOP4	Mean	2.51E-1 —	5.87E-1 —	5.79E-1 —	5.78E-01 —	5.28E-1 —	5.93E-1 —	**5.96E-1**
	Std	1.35E-2	4.43E-3	2.75E-2	2.32E-02	7.34E-2	1.52E-3	**1.22E-3**
MOP5	Mean	4.05E-1 —	6.97E-1 —	6.97E-1 —	6.97E-01 —	6.94E-1 —	6.97E-1 —	**7.14E-1**
	Std	1.15E-2	2.77E-3	2.13E-3	1.33E-03	1.96E-3	3.50E-3	**2.20E-4**
MOP6	Mean	6.23E-1 —	8.35E-1 —	8.36E-1 —	8.31E-01 —	8.28E-1 —	8.13E-1 —	**8.49E-1**
	Std	2.70E-4	8.27E-4	8.40E-4	9.69E-04	1.61E-3	4.09E-3	**3.08E-4**
MOP7	Mean	4.09E-1 —	5.32E-1 —	5.34E-1 —	5.26E-01 —	4.95E-1 —	5.07E-1 —	**5.66E-1**
	Std	6.43E-6	2.38E-3	4.38E-3	1.88E-03	1.49E-2	3.44E-3	**4.53E-4**
UF1	Mean	6.18E-1 —	7.16E-1 —	7.17E-1 —	6.73E-01 —	7.14E-1 —	7.11E-1 —	**7.18E-1**
	Std	1.49E-2	1.21E-3	5.04E-4	6.89E-03	5.26E-4	4.34E-3	**2.90E-4**
UF2	Mean	6.86E-1 —	7.11E-1 —	7.09E-1 —	7.01E-01 —	7.04E-1 —	7.04E-1 —	**7.14E-1**
	Std	8.10E-3	3.30E-3	1.28E-2	4.31E-03	3.08E-3	3.71E-3	**1.12E-3**
UF3	Mean	4.19E-1 —	**7.07E-1** +	6.97E-1 —	6.02E-01 —	7.02E-1 —	6.89E-1 —	7.06E-1
	Std	3.07E-2	**1.55E-2**	2.17E-2	1.44E-02	1.04E-2	3.21E-2	6.69E-3
UF4	Mean	**3.90E-1 =**	3.58E-1 —	3.53E-1 —	3.78E-01 —	3.46E-1 —	3.88E-1 —	**3.91E-1**
	Std	1.83E-3	7.24E-3	7.63E-3	3.18E-03	1.07E-2	5.92E-4	**5.69E-4**
UF5	Mean	2.59E-1 —	1.91E-1 —	2.16E-1 —	1.96E-01 —	1.01E-1 —	3.36E-1 =	3.28E-1
	Std	5.44E-2	7.45E-2	7.39E-2	7.50E-02	6.57E-2	2.56E-2	1.62E-2
UF6	Mean	3.04E-1 —	3.22E-1 —	3.02E-1 —	2.97E-01 —	2.93E-1 —	3.61E-1 —	**5.09E-1**
	Std	7.03E-2	6.15E-2	8.74E-2	8.20E-02	8.72E-2	3.22E-2	**1.05E-2**
UF7	Mean	4.44E-1 —	**5.77E-1** +	5.75E-1 =	5.63E-01 —	5.72E-1 —	5.72E-1 —	5.75E-1
	Std	1.07E-1	**1.57E-3**	7.10E-3	4.83E-03	4.63E-3	2.36E-3	2.42E-3
UF8	Mean	3.58E-1 —	**5.44E-1** +	5.36E-1 +	5.10E-01 +	5.17E-1 +	4.44E-1 —	4.79E-1
	Std	4.73E-2	**2.31E-2**	2.43E-2	1.59E-02	1.94E-2	2.58E-2	4.85E-2
UF9	Mean	6.51E-1 —	6.46E-1 —	6.92E-1 —	**7.85E-01 +**	6.73E-1 —	6.41E-1 —	7.77E-1
	Std	6.19E-2	8.77E-2	1.02E-1	**3.37E-02**	4.73E-2	6.39E-2	1.93E-2
UF10	Mean	1.92E-1 —	7.47E-2 —	8.29E-3 —	2.01E-01 =	8.83E-2 —	6.59E-2 —	**2.80E-1**
	Std	6.23E-2	4.30E-2	1.73E-2	1.10E-01	1.92E-2	3.97E-2	**7.49E-2**
—/=/+		26/1/0	21/0/6	22/1/4	24/1/2	26/0/1	24/3/1	
best/all		1/27	3/27	3/27	1/27	0/27	3/27	18/27

To summarize from the last to second row of Table 4, LDEA performed better on 26, 21, 22, 24, 24, and 23 cases, but only worse on 0, 5, 5, 2, 1, and 2 cases, when respectively compared to MSEA, AOOSTM, AMOSTM, OPE, ACD, and SPEA/R on IGD. Moreover, to summarize from the last row in Table 4, LDEA performed best on 19 out of a total of 27 cases regarding IGD, while MSEA, AOOSTM, AOOSTM, OPE, ACD, and SPEA/R respectively obtained the best IGD results on 0, 3, 3, 2, 1, and 4 cases. These summarized results on IGD indicated the obvious advantages of LDEA over the other compared algorithms.

Similar conclusions can be found in Table 5 when considering the HV results on all the *IMB*, *MOP,* and *UF* test problems. As observed from the last to second row in Table 5, LDEA performed better on 26, 21, 22, 24, 26, and 24 cases, but only worse on 0, 6, 4, 2, 1, and 1 cases, when respectively compared to MSEA, AOOSTM, AMOSTM, OPE, ACD, and SPEA/R on HV. Moreover, as observed from the last row in Table 5, LDEA performed best on 18 out of total 27 cases using HV, while MSEA, AMOSTM, AOOSTM, OPE, and SPEA/R respectively obtained the best HV results on 1, 3, 3, 1, and 3 cases. ACD could not find the best HV result for any test problem. These summarized HV results further confirmed the advantages of LDEA over the other compared algorithms.

Moreover, to quantify the performance of each optimizer on all the test problems (i.e., *IMB1-IMB10*, *MOP1-MOP7*, and *UF1-UF10*), a comparative analysis was conducted using Friedman's test. The resulting ranks are presented in Fig. 8, where a lower rank value indicates a superior performance of the optimizer. From Fig. 8, it is evident that LDEA outperforms its six competitors with a rank value of 1.52, which is significantly better than MSEA (6.22), AOOSTM (3.22), AMOSTM (3.44), OPE (4.33), ACD (4.96), and SPEA/R (3.7). These results establish that LDEA is a more effective optimizer for solving imbalanced and complicated MOPs. This is mainly attributed to its localized decomposition method and adaptive switching of aggregated functions between WS and TCH, which balance the convergence and diversity of the population effectively.

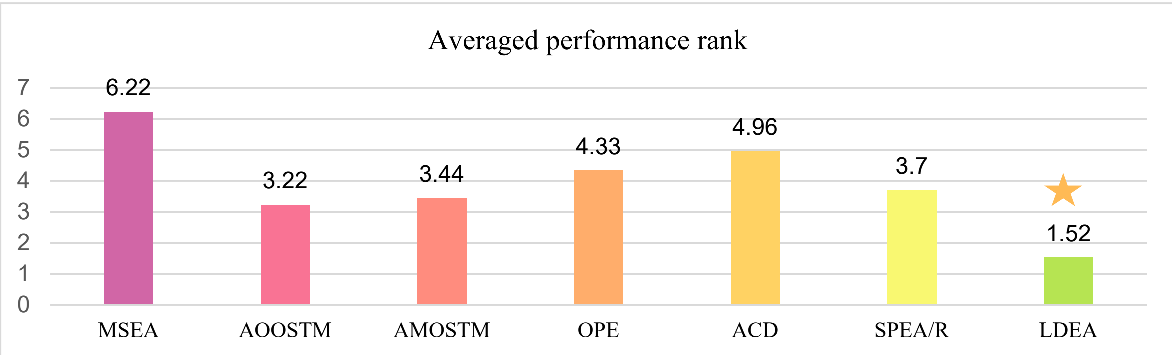


Fig. 8. Illustration of the Friedman ranks of all the compared algorithms on the *IMB*, *MOP*, and *UF* test problems.

4.4.Discussion on the effect of parameter *k* for controlling *WS*

As indicated in [52], it is very challenging to select WS and TCH for each subproblem. In this paper, the performance monitoring vector $\boldsymbol{C}=\{c_1,c_2,\ldots,c_N\}$ is employed to record the running period that the subproblems stop to be improved by WS. Once the c_i value ($i\in[1,N]$) is larger than ke^m in Eq. (11), TCH will be activated to replace WS for the *i*th subproblem. Thus, the setting of parameter *k* in Eq. (11) determines the running of WS and TCH for each subproblem, which may affect the performance of LDEA. To study its influence, the LDEA variants with the parameter *k* set as 2.5, 5, 7.5, 10, 12.5, 15, and 17.5 are experimentally compared. A smaller value of *k* indicates a shorter running period to replace WS using TCH. The other experimental settings in LDEA are kept the same.

Table 6
The IGD comparison results for LDEA with different k values on all the test problems

Problem/IGD		k=2.5	k=5	k=7.5	k=10	k=12.5	k=15	k=17.5
IMB1	Mean	4.75E-3	4.74E-3	4.73E-3	4.75E-3	4.74E-3	4.77E-3	4.84E-3
	Std	5.81E-5	6.40E-5	3.04E-5	5.13E-5	5.47E-5	4.60E-5	9.39E-5
IMB2	Mean	4.99E-3	4.98E-3	4.98E-3	4.97E-3	4.97E-3	5.01E-3	5.10E-3
	Std	6.36E-5	7.55E-5	8.70E-5	5.80E-5	8.08E-5	6.37E-5	9.36E-5
IMB3	Mean	6.29E-3	6.26E-3	6.28E-3	6.28E-3	6.32E-3	6.32E-3	6.37E-3
	Std	9.18E-5	8.30E-5	1.09E-4	1.23E-4	6.33E-5	1.05E-4	1.11E-4
IMB4	Mean	2.40E-2	2.40E-2	2.43E-2	2.47E-2	2.50E-2	2.53E-2	2.57E-2
	Std	7.32E-5	6.01E-5	1.32E-4	2.19E-4	1.98E-4	1.88E-4	2.38E-4
IMB5	Mean	3.05E-2	3.05E-2	3.08E-2	3.14E-2	3.21E-2	3.36E-2	3.53E-2
	Std	8.42E-5	7.71E-5	1.64E-4	1.92E-4	3.64E-4	4.30E-4	4.93E-4
IMB6	Mean	2.31E-2	2.32E-2	2.37E-2	2.41E-2	2.44E-2	2.48E-2	2.51E-2
	Std	7.12E-5	9.40E-5	1.26E-4	1.51E-4	1.04E-4	2.44E-4	1.61E-4
IMB7	Mean	5.37E-3	5.38E-3	5.38E-3	5.38E-3	5.36E-3	5.25E-3	5.28E-3
	Std	1.44E-4	1.26E-4	9.63E-5	1.38E-4	1.25E-4	9.99E-5	1.13E-4
IMB8	Mean	5.99E-3	5.94E-3	5.91E-3	5.86E-3	5.85E-3	5.75E-3	5.87E-3
	Std	1.60E-4	1.92E-4	1.89E-4	2.22E-4	1.58E-4	1.66E-4	2.55E-4
IMB9	Mean	7.20E-3	7.19E-3	7.14E-3	7.12E-3	6.82E-3	7.12E-3	7.55E-3
	Std	2.75E-4	2.24E-4	2.60E-4	2.24E-4	2.16E-4	2.14E-4	3.46E-4
IMB10	Mean	2.57E-2	2.57E-2	2.64E-2	2.73E-2	2.77E-2	2.79E-2	2.78E-2
	Std	1.11E-4	1.32E-4	1.70E-4	3.01E-4	2.94E-4	2.99E-4	2.73E-4
MOP1	Mean	7.96E-3	7.93E-3	7.94E-3	7.98E-3	7.99E-3	8.02E-3	8.07E-3
	Std	1.32E-4	2.06E-4	1.58E-4	1.62E-4	1.67E-4	1.55E-4	2.19E-4
MOP2	Mean	4.42E-3	4.31E-3	4.46E-3	4.45E-3	4.44E-3	4.40E-3	4.86E-3
	Std	7.48E-4	6.56E-4	7.54E-4	8.19E-4	7.28E-4	4.30E-4	7.22E-4
MOP3	Mean	4.53E-3	4.54E-3	4.47E-3	4.48E-3	4.59E-3	4.66E-3	4.82E-3
	Std	3.49E-4	3.67E-4	3.33E-4	1.62E-4	7.90E-4	2.73E-4	3.30E-4
MOP4	Mean	4.23E-3	4.27E-3	4.20E-3	4.21E-3	4.25E-3	4.23E-3	4.26E-3
	Std	5.18E-4	4.71E-4	4.72E-4	5.00E-4	5.23E-4	3.88E-4	1.85E-4
MOP5	Mean	7.34E-3	7.37E-3	7.39E-3	7.36E-3	7.36E-3	7.42E-3	7.39E-3
	Std	1.58E-4	1.46E-4	1.52E-4	1.64E-4	1.43E-4	1.74E-4	2.24E-4
MOP6	Mean	2.52E-2	2.53E-2	2.59E-2	2.69E-2	2.75E-2	2.77E-2	2.77E-2
	Std	1.34E-4	1.43E-4	1.66E-4	2.48E-4	3.03E-4	2.56E-4	3.09E-4
MOP7	Mean	3.40E-2	3.40E-2	3.46E-2	3.58E-2	3.67E-2	3.71E-2	3.73E-2
	Std	1.92E-4	2.25E-4	2.79E-4	3.48E-4	4.45E-4	4.93E-4	3.97E-4
UF1	Mean	4.90E-3	4.87E-3	4.62E-3	4.42E-3	4.26E-3	4.19E-3	4.13E-3
	Std	7.82E-4	6.15E-4	1.95E-4	1.57E-4	1.81E-4	1.52E-4	1.64E-4
UF2	Mean	8.63E-3	8.37E-3	8.11E-3	8.12E-3	7.84E-3	7.67E-3	7.60E-3
	Std	1.13E-3	1.20E-3	1.37E-3	1.03E-3	1.19E-3	1.16E-3	1.33E-3
UF3	Mean	1.61E-2	1.68E-2	1.27E-2	1.17E-2	1.05E-2	8.90E-3	8.77E-3
	Std	6.71E-3	6.43E-3	4.10E-3	3.94E-3	4.18E-3	3.56E-3	4.23E-3
UF4	Mean	4.09E-2	4.06E-2	4.03E-2	4.00E-2	3.97E-2	3.93E-2	3.90E-2
	Std	4.94E-4	5.03E-4	5.54E-4	5.00E-4	5.92E-4	5.83E-4	6.09E-4
UF5	Mean	1.61E-1	1.65E-1	1.62E-1	1.60E-1	1.66E-1	1.57E-1	1.61E-1
	Std	1.48E-2	1.98E-2	2.41E-2	1.27E-2	1.88E-2	2.12E-2	1.54E-2
UF6	Mean	1.59E-2	1.60E-2	2.17E-2	1.20E-2	1.10E-2	1.05E-2	2.00E-2
	Std	5.95E-3	7.10E-3	4.62E-2	5.99E-3	6.89E-3	4.47E-3	5.10E-2
UF7	Mean	7.72E-3	7.79E-3	7.85E-3	8.12E-3	8.59E-3	8.78E-3	8.65E-3
	Std	1.00E-3	9.61E-4	1.30E-3	1.54E-3	1.36E-3	1.45E-3	1.58E-3
UF8	Mean	7.87E-2	7.96E-2	8.11E-2	8.15E-2	7.87E-2	7.97E-2	7.96E-2
	Std	2.53E-2	3.15E-2	3.14E-2	3.13E-2	2.62E-2	3.20E-2	3.21E-2
UF9	Mean	5.20E-2	4.89E-2	4.09E-2	3.71E-2	4.14E-2	3.69E-2	3.65E-2
	Std	2.95E-2	2.97E-2	1.24E-2	1.01E-2	2.47E-2	1.06E-2	9.80E-3
UF10	Mean	2.21E-1	1.96E-1	2.12E-1	1.95E-1	2.10E-1	2.00E-1	2.06E-1
	Std	3.05E-2	2.41E-2	5.36E-2	2.76E-2	4.97E-2	2.93E-2	3.15E-2

Table 6 and Table 7 respectively provide the mean results and standard deviations of IGD and HV, which were obtained by LDEA with different k values in solving all the test problems. In Table 6 and Table 7, the statistically best mean results of each problem using the Wilcoxon's rank sum test with a 5% significance level are all highlighted with a shaded background.

For the *IMB* test problems, a smaller k value is better for LDEA to produce better IGD results on *IMB1*-*IMB6* and *IMB10* in Table 6. However, a larger k value is preferred by LDEA to better solve *IMB7*, and LDEA with k=15 and k=12.5 are respectively best on *IMB8* and *IMB9*. Moreover, the HV results of LDEA on *IMB1*-

IMB3 and *IMB7* are statistically similar in Table 7. A smaller *k* value is better for LDEA to solve *IMB4-IMB6* and *IMB10*, while a larger *k* value is more suitable for LDEA on *IMB8-IMB9*.

Table 7
The HV comparison results of LDEA with different *k* values on the test problems

Problem/HV		*k*=2.5	*k*=5	*k*=7.5	*k*=10	*k*=12.5	*k*=15	*k*=17.5
IMB1	Mean	7.18E-1	7.18E-1	7.18E-1	7.18E-1	7.18E-1	7.18E-1	7.18E-1
	Std	8.55E-5	9.56E-5	5.11E-5	7.09E-5	8.21E-5	5.52E-5	1.15E-4
IMB2	Mean	5.79E-1	5.79E-1	5.79E-1	5.79E-1	5.79E-1	5.79E-1	5.79E-1
	Std	1.01E-4	1.17E-4	1.39E-4	9.17E-5	1.31E-4	1.01E-4	1.42E-4
IMB3	Mean	3.42E-1	3.42E-1	3.42E-1	3.42E-1	3.42E-1	3.42E-1	3.42E-1
	Std	1.45E-4	1.46E-4	1.67E-4	2.18E-4	1.24E-4	1.78E-4	1.79E-4
IMB4	Mean	8.51E-1	8.51E-1	8.51E-1	8.50E-1	8.48E-1	8.47E-1	8.45E-1
	Std	1.14E-4	7.82E-5	1.70E-4	3.35E-4	9.06E-4	9.20E-4	1.08E-3
IMB5	Mean	5.76E-1	5.76E-1	5.76E-1	5.76E-1	5.75E-1	5.73E-1	5.71E-1
	Std	1.78E-4	1.81E-4	2.18E-4	2.24E-4	6.09E-4	5.85E-4	7.04E-4
IMB6	Mean	8.53E-1	8.53E-1	8.53E-1	8.53E-1	8.52E-1	8.51E-1	8.51E-1
	Std	1.21E-4	1.30E-4	1.58E-4	1.79E-4	2.42E-4	3.65E-4	5.18E-4
IMB7	Mean	7.17E-1	7.17E-1	7.17E-1	7.17E-1	7.17E-1	7.17E-1	7.17E-1
	Std	2.42E-4	2.16E-4	1.71E-4	2.48E-4	2.32E-4	1.94E-4	2.29E-4
IMB8	Mean	5.77E-1	5.77E-1	5.77E-1	5.78E-1	5.78E-1	5.78E-1	5.78E-1
	Std	2.62E-4	3.11E-4	3.07E-4	3.68E-4	2.55E-4	2.83E-4	4.19E-4
IMB9	Mean	3.40E-1	3.40E-1	3.40E-1	3.40E-1	3.41E-1	3.41E-1	3.41E-1
	Std	4.83E-4	4.21E-4	4.53E-4	3.95E-4	4.26E-4	4.45E-4	6.09E-4
IMB10	Mean	8.48E-1	8.48E-1	8.48E-1	8.47E-1	8.45E-1	8.43E-1	8.41E-1
	Std	1.53E-4	1.41E-4	1.95E-4	3.76E-4	6.70E-4	1.39E-3	1.40E-3
MOP1	Mean	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1
	Std	1.66E-4	2.44E-4	1.84E-4	2.10E-4	2.07E-4	1.96E-4	2.74E-4
MOP2	Mean	4.44E-1	4.44E-1	4.44E-1	4.44E-1	4.44E-1	4.44E-1	4.44E-1
	Std	4.12E-4	7.30E-4	8.13E-4	3.38E-4	4.49E-4	1.75E-4	3.45E-4
MOP3	Mean	3.46E-1	3.46E-1	3.46E-1	3.46E-1	3.46E-1	3.46E-1	3.46E-1
	Std	3.49E-4	3.66E-4	3.47E-4	1.65E-4	6.73E-4	2.47E-4	3.49E-4
MOP4	Mean	5.96E-1	5.96E-1	5.96E-1	5.96E-1	5.96E-1	5.96E-1	5.96E-1
	Std	1.12E-3	7.15E-4	6.44E-4	1.22E-3	1.44E-3	4.35E-4	1.93E-4
MOP5	Mean	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.14E-1
	Std	2.22E-4	2.09E-4	1.93E-4	2.20E-4	1.90E-4	2.31E-4	3.22E-4
MOP6	Mean	8.50E-1	8.50E-1	8.49E-1	8.49E-1	8.47E-1	8.46E-1	8.46E-1
	Std	1.26E-4	1.62E-4	1.63E-4	3.08E-4	5.65E-4	1.01E-3	9.68E-4
MOP7	Mean	5.67E-1	5.67E-1	5.67E-1	5.66E-1	5.65E-1	5.65E-1	5.64E-1
	Std	4.16E-4	3.47E-4	3.69E-4	4.53E-4	5.27E-4	4.89E-4	6.93E-4
UF1	Mean	7.17E-1	7.17E-1	7.18E-1	7.18E-1	7.18E-1	7.19E-1	7.19E-1
	Std	1.22E-3	8.34E-4	3.60E-4	2.90E-4	3.55E-4	3.21E-4	4.43E-4
UF2	Mean	7.14E-1	7.14E-1	7.14E-1	7.14E-1	7.15E-1	7.15E-1	7.15E-1
	Std	1.32E-3	1.26E-3	1.49E-3	1.12E-3	1.35E-3	1.25E-3	1.56E-3
UF3	Mean	7.00E-1	7.00E-1	7.06E-1	7.06E-1	7.08E-1	7.11E-1	7.11E-1
	Std	9.82E-3	8.87E-3	5.66E-3	6.69E-3	6.52E-3	5.82E-3	6.55E-3
UF4	Mean	3.90E-1	3.91E-1	3.91E-1	3.91E-1	3.91E-1	3.91E-1	3.92E-1
	Std	7.43E-4	6.18E-4	7.09E-4	5.69E-4	7.09E-4	7.79E-4	7.55E-4
UF5	Mean	3.32E-1	3.27E-1	3.35E-1	3.28E-1	3.25E-1	3.38E-1	3.28E-1
	Std	2.73E-2	2.33E-2	2.76E-2	1.62E-2	2.10E-2	2.66E-2	1.98E-2
UF6	Mean	5.02E-1	5.03E-1	5.04E-1	5.09E-1	5.09E-1	5.10E-1	4.99E-1
	Std	8.38E-3	9.15E-3	2.77E-2	1.05E-2	1.14E-2	1.17E-2	2.77E-2
UF7	Mean	5.75E-1	5.75E-1	5.75E-1	5.75E-1	5.74E-1	5.74E-1	5.74E-1
	Std	1.55E-3	1.49E-3	2.07E-3	2.42E-3	2.17E-3	2.20E-3	2.46E-3
UF8	Mean	4.77E-1	4.78E-1	4.79E-1	4.79E-1	4.83E-1	4.83E-1	4.82E-1
	Std	3.71E-2	4.72E-2	4.76E-2	4.85E-2	4.02E-2	5.00E-2	5.02E-2
UF9	Mean	7.60E-1	7.62E-1	7.72E-1	7.77E-1	7.72E-1	7.78E-1	7.80E-1
	Std	3.87E-2	3.72E-2	2.03E-2	1.93E-2	3.44E-2	1.81E-2	1.79E-2
UF10	Mean	2.34E-1	2.86E-1	2.52E-1	2.80E-1	2.63E-1	2.66E-1	2.53E-1
	Std	6.29E-2	6.08E-2	7.97E-2	7.49E-2	7.83E-2	7.85E-2	8.07E-2

For the *MOP* test problems, the IGD results of LDEA in Table 6 are slightly better when *k*<15 for *MOP1*. For *MOP2*, the best IGD result is obtained by *k*=5, while a smaller *k* value (i.e., *k*<12.5) will bring a better IGD result for *MOP3*. The performance on *MOP4* is not impacted by *k*. For *MOP6* and *MOP7*, their best IGD results are obtained by LDEA with *k*=2.5 and *k*=5. On the other hand, the HV results of LDEA in Table 7 are statistically similar on *MOP1-MOP5*, which are not influenced by *k*, while a smaller *k* value is preferred by LDEA to better solve *MOP6-MOP7*.

For the *UF* test problems, a larger *k* value can provide better IGD results for LDEA on *UF1-UF4* and *UF9-UF10*, and all the IGD results on *UF5-UF6* and *UF8* are statistically similar in Table 6. Only a smaller *k* value (*k*<12.5) is preferred by LDEA in solving *UF7*. From Table 7, a similar conclusion can be observed from the HV results on all the *UF* problems.

To summarize, the impact of the parameter *k* can be roughly classified as three cases. For *MOP4*, *UF5-UF6,* and *UF8*, the impact of *k* is very small and LDEA with different *k* values can properly address these problems. For *MOP6-MOP7*, *IMB4-IMB6*, *IMB10,* and *UF7*, a smaller *k* value will generally improve the performance of LDEA, indicating a short running period to activate TCH at the evolutionary process. Since their PF shapes are non-convex, the activation of TCH in LDEA can help to avoid the issue as plotted in Fig. 1. For *IMB8-IMB9* and *UF1-UF4*, a larger *k* value will be better for LDEA, which will run more WS in their evolutionary process. This is because the PF shapes of *UF1-UF3* are convex and thus the switch to TCH is not necessary, and WS can speed up the convergence when solving *UF4* and *IMB8-IMB9*.

Table 8
The IGD and HV comparison results for LDEA and its variants on all the test problems

Problem/IGD		LDEA-I	LDEA-II	LDEA	Problem/HV		LDEA-I	LDEA-II	LDEA
IMB1	Mean	1.30E-2 −	7.70E-3 −	**4.75E-3**	*IMB1*	Mean	7.08E-1 −	7.15E-1 −	**7.18E-1**
	Std	4.38E-4	1.54E-4	**5.13E-5**		Std	4.76E-4	1.78E-4	**7.09E-5**
IMB2	Mean	4.11E-2 −	9.10E-3 −	**4.97E-3**	*IMB2*	Mean	5.33E-1 −	5.74E-1 −	**5.79E-1**
	Std	1.65E-2	1.86E-4	**5.80E-5**		Std	2.13E-2	2.50E-4	**9.17E-5**
IMB3	Mean	2.34E-2 −	1.22E-2 −	**6.28E-3**	*IMB3*	Mean	3.21E-1 −	3.34E-1 −	**3.42E-1**
	Std	6.30E-4	2.56E-4	**1.23E-4**		Std	7.74E-4	3.03E-4	**2.18E-4**
IMB4	Mean	3.88E-2 −	2.85E-2 −	**2.47E-2**	*IMB4*	Mean	8.34E-1 −	8.45E-1 −	**8.50E-1**
	Std	4.51E-4	2.34E-4	**2.19E-4**		Std	6.04E-4	2.83E-4	**3.35E-4**
IMB5	Mean	3.93E-2 −	3.33E-2 −	**3.14E-2**	*IMB5*	Mean	5.61E-1 −	5.71E-1 −	**5.76E-1**
	Std	4.79E-4	1.94E-4	**1.92E-4**		Std	8.20E-4	4.14E-4	**2.24E-4**
IMB6	Mean	2.73E-2 −	2.45E-2 −	**2.41E-2**	*IMB6*	Mean	8.49E-1 −	8.52E-1 −	**8.53E-1**
	Std	2.07E-4	1.69E-4	**1.51E-4**		Std	1.90E-4	1.61E-4	**1.79E-4**
IMB7	Mean	2.64E-2 −	8.07E-3 −	**5.38E-3**	*IMB7*	Mean	6.97E-1 −	7.14E-1 −	**7.17E-1**
	Std	1.00E-2	3.66E-4	**1.38E-4**		Std	8.61E-3	5.72E-4	**2.48E-4**
IMB8	Mean	2.78E-2 −	9.38E-3 −	**5.86E-3**	*IMB8*	Mean	5.50E-1 −	5.73E-1 −	**5.78E-1**
	Std	1.17E-2	3.68E-4	**2.22E-4**		Std	1.36E-2	5.73E-4	**3.68E-4**
IMB9	Mean	3.46E-2 −	1.14E-2 −	**7.12E-3**	*IMB9*	Mean	3.13E-1 −	3.34E-1 −	**3.40E-1**
	Std	1.08E-2	4.84E-4	**2.24E-4**		Std	7.85E-3	8.70E-4	**3.95E-4**
IMB10	Mean	6.02E-2 −	3.62E-2 −	**2.73E-2**	*IMB10*	Mean	8.09E-1 −	8.36E-1 −	**8.47E-1**
	Std	1.07E-3	4.30E-4	**3.01E-4**		Std	1.00E-3	5.63E-4	**3.76E-4**
MOP1	Mean	3.27E-2 −	1.73E-2 −	**7.98E-3**	*MOP1*	Mean	6.83E-1 −	7.02E-1 −	**7.14E-1**
	Std	1.09E-3	5.95E-4	**1.62E-4**		Std	1.10E-3	6.46E-4	**2.10E-4**
MOP2	Mean	1.66E-2 −	5.59E-3 −	**4.45E-3**	*MOP2*	Mean	4.24E-1 −	4.41E-1 −	**4.44E-1**
	Std	8.84E-3	1.84E-3	**8.19E-4**		Std	1.41E-2	2.72E-3	**3.38E-4**
MOP3	Mean	3.28E-2 −	5.63E-3 −	**4.48E-3**	*MOP3*	Mean	3.05E-1 −	3.43E-1 −	**3.46E-1**
	Std	3.40E-2	8.50E-4	**1.62E-4**		Std	4.49E-2	9.76E-4	**1.65E-4**
MOP4	Mean	2.69E-2 −	4.42E-3 −	**4.21E-3**	*MOP4*	Mean	5.71E-1 −	5.95E-1 −	**5.96E-1**
	Std	2.59E-2	2.86E-4	**5.00E-4**		Std	3.02E-2	3.89E-4	**1.22E-3**
MOP5	Mean	2.78E-2 −	1.53E-2 −	**7.36E-3**	*MOP5*	Mean	6.85E-1 −	7.02E-1 −	**7.14E-1**
	Std	7.69E-4	3.85E-4	**1.64E-4**		Std	9.06E-4	4.40E-4	**2.20E-4**
MOP6	Mean	5.48E-2 −	3.45E-2 −	**2.69E-2**	*MOP6*	Mean	8.19E-1 −	8.39E-1 −	**8.49E-1**
	Std	7.47E-4	3.95E-4	**2.48E-4**		Std	7.53E-4	3.81E-4	**3.08E-4**
MOP7	Mean	7.62E-2 −	4.68E-2 −	**3.58E-2**	*MOP7*	Mean	5.14E-1 −	5.47E-1 −	**5.66E-1**
	Std	1.23E-3	3.91E-4	**3.48E-4**		Std	1.59E-3	7.11E-4	**4.53E-4**
UF1	Mean	4.98E-3 −	9.41E-3 −	**4.42E-3**	*UF1*	Mean	7.16E-1 −	7.11E-1 −	**7.18E-1**
	Std	4.51E-4	3.79E-3	**1.57E-4**		Std	7.40E-4	6.64E-3	**2.90E-4**
UF2	Mean	9.44E-3 −	**6.65E-3** +	8.12E-3	*UF2*	Mean	7.13E-1 −	**7.16E-1** +	7.14E-1
	Std	1.70E-3	**9.40E-4**	1.03E-3		Std	1.87E-3	**1.02E-3**	1.12E-3
UF3	Mean	8.14E-2 −	1.32E-2 −	**1.17E-2**	*UF3*	Mean	6.40E-1 −	7.02E-1 −	**7.06E-1**
	Std	3.20E-2	4.02E-3	**3.94E-3**		Std	3.24E-2	5.74E-3	**6.69E-3**

UF4	Mean	5.98E-2 —	4.13E-2 —	**4.00E-2**	*UF4*	Mean	3.62E-1 —	3.89E-1 —	**3.91E-1**
	Std	4.65E-3	5.13E-4	**5.00E-4**		Std	5.94E-3	6.25E-4	**5.69E-4**
UF5	Mean	3.72E-1 —	1.73E-1 —	**1.60E-1**	*UF5*	Mean	1.67E-1 —	**3.27E-1 =**	**3.28E-1**
	Std	1.77E-1	2.17E-2	**1.27E-2**		Std	7.88E-2	**2.56E-2**	**1.62E-2**
UF6	Mean	2.63E-1 —	5.76E-2 —	**1.20E-2**	*UF6*	Mean	3.85E-1 —	4.83E-1 —	**5.09E-1**
	Std	5.27E-2	1.76E-1	**5.99E-3**		Std	3.20E-2	8.72E-2	**1.05E-2**
UF7	Mean	**7.90E-3 =**	1.14E-2 —	**8.12E-3**	*UF7*	Mean	5.74E-1 —	5.70E-1 —	**5.75E-1**
	Std	**2.86E-3**	1.36E-3	**1.54E-3**		Std	4.07E-3	2.10E-3	**2.42E-3**
UF8	Mean	**6.14E-2 +**	8.54E-2 —	8.15E-2	*UF8*	Mean	**5.10E-1 +**	4.71E-1 —	4.79E-1
	Std	**9.89E-3**	3.93E-2	3.13E-2		Std	**2.16E-2**	6.03E-2	4.85E-2
UF9	Mean	5.37E-2 —	9.19E-2 —	**3.71E-2**	*UF9*	Mean	7.62E-1 —	7.02E-1 —	**7.77E-1**
	Std	3.54E-2	4.89E-2	**1.01E-2**		Std	4.68E-2	6.76E-2	**1.93E-2**
UF10	Mean	3.16E-1 —	2.94E-1 —	**1.95E-1**	*UF10*	Mean	1.54E-1 —	2.09E-1 —	**2.80E-1**
	Std	4.48E-2	1.08E-1	**2.76E-2**		Std	2.66E-2	7.78E-2	**7.49E-2**
—/=/+		25/1/1	26/0/1		—/=/+		26/0/1	25/1/1	

4.5.Discussion on the modified evolutionary operators

The evolutionary operators play a key role in the performance of MOEAs. In this paper, to cooperate well with the localized decomposition method, the adopted evolutionary operators were modified from [15]. Here, the effectiveness of our modified evolutionary operators in Eqs. (7)-(8) was experimentally studied by comparing LDEA with its two variants. One LDEA variant (LDEA-I) used the traditional DE and PM, while the other LDEA variant (LDEA-II) adopted the original evolutionary operators in M2M. In LDEA-I, *DE/rand/1* was run with F=0.5 and CR=1.0, and PM was randomly run in one decision variable with the mutation index set as 20 [4]. LDEA-II used the same parameter settings as LDEA in Eqs. (7)-(8). Please note that if a decision variable was out of bounds after applying the evolutionary operators, it was reset to the closest boundary value, as done in Section 4.1. Table 8 provides the mean results and standard deviations obtained by LDEA and its two variants on all the test problems, which shows the IGD and HV results respectively in the left and right columns. The Wilcoxon's rank sum test with a 5% significance level was run to show the statistically significant difference on the results of LDEA with each variant. In Table 8, the statistically best mean results of each problem are all highlighted in **boldface**. The signs "—", "+" and "=" in Table 8 summarize the numbers of test problems in which LDEA respectively performed better than, worse than, and similarly to its variants.

As observed from Table 8, LDEA obtained the best IGD and HV results on 25 out of 27 total cases, which validated the effectiveness of our modified evolutionary operators in LDEA. To be more specific, regarding IGD, LDEA performed better than LDEA-I and LDEA-II respectively on 25 and 26 cases, but only worse than LDEA-I and LDEA-II on 1 case each. For HV, LDEA performed better than LDEA-I and LDEA-II respectively on 26 and 25 cases, but only worse than LDEA-I and LDEA-II respectively on 1 case each. These one-by-one comparison results have evidently confirmed the effectiveness of the modified evolutionary operators in Eqs. (7)-(8) for enhancing the performance of LDEA.

4.6.Running time analysis

When evaluating computational efficiency, we considered several compared algorithms implemented in MATLAB, including DEAGNG, SPEA/R, NSGAIIARSBX, GFMMOEA, MSEA, OPE, and the proposed

LDEA. However, AOOSTM, AMOSTM, and ACD, developed in Java, are excluded from Fig. 9 due to potential performance disparities arising from differences in programming languages.

As shown in Fig. 9, GFMMOEA and OPE show prolonged computational durations for most test problems, while NSGAIIARSBX runs faster than other compared algorithms. In general, the overall performance of LDEA is moderate on these test problems. This is because the localized decomposition strategy inevitably consumes a certain amount of computational resources. However, given the excellent performance of LDEA regarding the IGD and HV indicators, it is still very competitive with other algorithms.

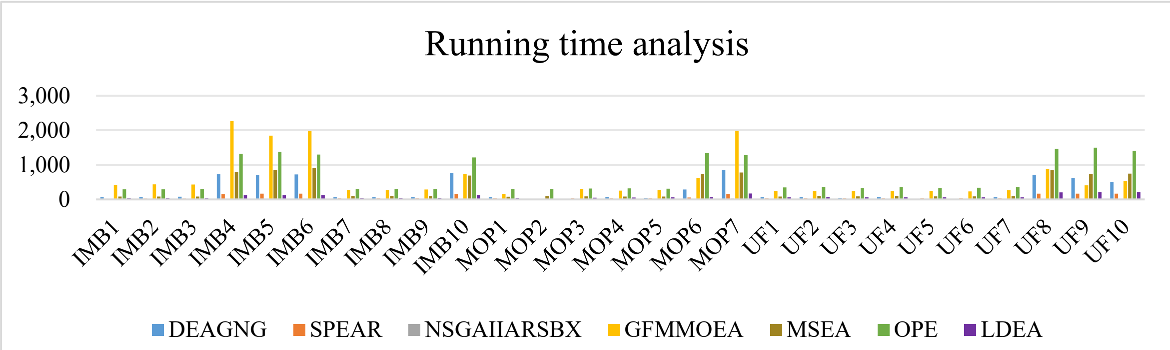
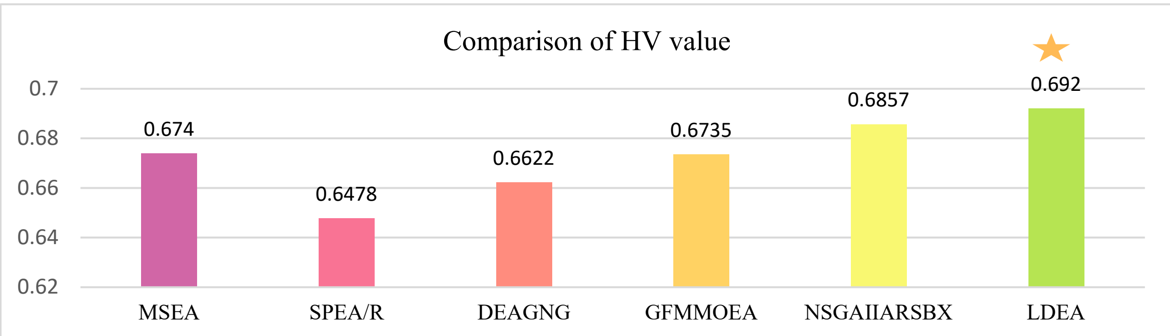


Fig. 9. The average running times (seconds) of LDEA and its six competitors on *IMB*, *MOP*, and *UF* test problems.

4.7.Real-world Application

There is a wide range of multi-objective optimization scenarios in the real-world [66]-[67]. Therefore, to evaluate the effectiveness of LDEA in real-world scenarios, a Multi-Objective Optimal Operations (M3O) problem is studied here [67], which is a real-world MOP built from a water reservoir system. In the experiment, the parameter settings for the M3O problem are referenced from [19], where the population size *N*, dimension of decision variables *n*, number of objective *m*, and maximal function evaluations are set to 50, 4, 2, and 1000, respectively. To ensure comparability among all the algorithms in PlatEMO[2] [58], MSEA [56], SPEA/R [49], and three additional algorithms (DEAGNG [68], GFMMOEA [69] and NSGAIIARSBX [70]) are chosen as comparison algorithms for solving the M3O problem. The HV values obtained by the six compared algorithms are plotted in Fig. 10, where the best results are marked with a star. Observing from Fig. 10, LDEA is superior to its five competitors, achieving the best HV value. This confirmed the effectiveness of LDEA in solving the real-world M3O problem. Moreover, to visually illustrate the comparison, the final solutions obtained by all the compared algorithms are plotted in Fig. 8. As shown in Fig. 11, a set of solutions with better diversity and



convergence is obtained by LDEA, outperforming its five competitors in solving the M3O problem. This can be attributed to the excellent optimization mechanism of LDEA in balancing diversity and convergence. Notably, an additional performance comparison between the three newly introduced compared algorithms (i.e., DEAGNG, GFMMOEA, and NSGAIIARSBX) and LDEA on the artificial benchmark test problems (i.e., *IMB1-IMB10* [46], *MOP1-MOP7* [15], and *UF1-UF10* [18]) is also studied, and the experimental results are provided in Table 9, which show that LDEA greatly outperforms these three compared algorithms on artificial benchmark test problems in terms of IGD and HV [71] metrics.

Fig. 10. Comparison of HV values obtained by six compared algorithms on the M3O problem.

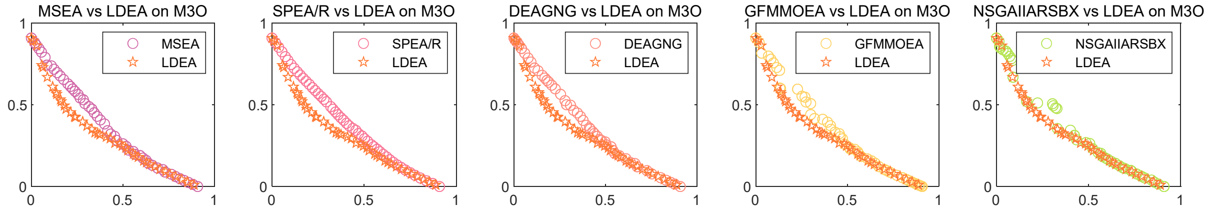


Fig. 11. Final solutions obtained by LDEA and its five competitors on the M3O problem.

Table 9
The IGD and HV results obtained by LDEA and three additional compared algorithms on all the test problems

Problem		IGD				HV			
		DEAGNG	GFMMOEA	NSGAIIARSBX	LDEA	DEAGNG	GFMMOEA	NSGAIIARSBX	LDEA
IMB1	Mean	1.19E-1 -	1.05E-1 -	9.47E-2 -	**4.75E-3**	5.86E-1 -	6.05E-1 -	6.13E-1 -	**7.18E-1**
	Std	3.63E-02	1.58E-02	3.43E-02	**5.13E-5**	3.57E-02	2.10E-02	3.45E-02	**7.09E-5**
IMB2	Mean	1.55E-1 -	1.42E-1 -	1.31E-1 -	**4.97E-3**	3.99E-1 -	4.08E-1 -	4.19E-1 -	**5.79E-1**
	Std	1.46E-02	2.15E-02	1.25E-02	**5.80E-5**	1.78E-02	2.09E-02	1.94E-02	**9.17E-5**
IMB3	Mean	2.28E-1 -	2.17E-1 -	2.23E-1 -	**6.28E-3**	1.42E-1 -	1.52E-1 -	1.76E-1 -	**3.42E-1**
	Std	2.88E-02	4.42E-02	7.22E-03	**1.23E-4**	2.41E-02	2.90E-02	2.91E-02	**2.18E-4**
IMB4	Mean	1.28E-1 -	1.21E-1 -	1.26E-1 -	**2.47E-2**	7.63E-1 -	7.70E-1 -	7.60E-1 -	**8.50E-1**
	Std	6.73E-03	6.63E-03	1.14E-02	**2.19E-4**	4.75E-03	2.22E-03	9.22E-03	**3.35E-4**
IMB5	Mean	1.18E-1 -	8.22E-2 -	8.93E-2 -	**3.14E-2**	5.09E-1 -	5.46E-1 -	5.35E-1 -	**5.76E-1**
	Std	1.18E-02	2.90E-04	7.63E-04	**1.92E-4**	1.11E-02	4.65E-04	2.21E-03	**2.24E-4**
IMB6	Mean	4.69E-2 -	4.47E-2 -	5.35E-2 -	**2.41E-2**	8.37E-1 -	8.42E-1 -	8.37E-1 -	**8.53E-1**
	Std	1.09E-03	1.60E-04	1.07E-03	**1.51E-4**	1.30E-03	1.52E-04	9.85E-04	**1.79E-4**
IMB7	Mean	2.94E-2 -	2.89E-2 -	2.89E-2 -	**5.38E-3**	6.96E-1 -	6.97E-1 -	6.97E-1 -	**7.17E-1**
	Std	8.39E-04	1.85E-04	6.78E-05	**1.38E-4**	9.51E-04	5.36E-04	8.22E-05	**2.48E-4**
IMB8	Mean	3.43E-2 -	3.41E-2 -	3.39E-2 -	**5.86E-3**	5.45E-1 -	5.45E-1 -	5.47E-1 -	**5.78E-1**
	Std	4.27E-04	4.05E-04	5.69E-05	**2.22E-4**	1.07E-03	9.89E-04	6.66E-05	**3.68E-4**
IMB9	Mean	3.28E-2 -	3.27E-2 -	3.22E-2 -	**7.12E-3**	3.13E-1 -	3.13E-1 -	3.16E-1 -	**3.40E-1**
	Std	4.78E-04	5.10E-04	1.02E-04	**2.24E-4**	1.13E-03	1.29E-03	7.93E-05	**3.95E-4**
IMB10	Mean	2.93E-1 -	2.39E-1 -	6.12E-2 -	**2.73E-2**	5.55E-1 -	6.22E-1 -	8.14E-1 -	**8.47E-1**
	Std	8.08E-02	3.07E-02	6.50E-03	**3.01E-4**	7.47E-02	3.19E-02	6.23E-03	**3.76E-4**
MOP1	Mean	3.52E-1 -	3.77E-1 -	3.15E-1 -	**7.98E-3**	2.61E-1 -	2.09E-1 -	3.24E-1 -	**7.14E-1**
	Std	3.51E-03	1.87E-02	4.39E-02	**1.62E-4**	6.69E-03	4.04E-02	5.66E-02	**2.10E-4**
MOP2	Mean	3.54E-1 -	3.54E-1 -	3.54E-1 -	**4.45E-3**	1.73E-1 -	1.73E-1 -	1.73E-1 -	**4.44E-1**
	Std	1.51E-16	1.62E-16	1.79E-16	**8.19E-4**	7.54E-17	5.65E-17	2.92E-17	**3.38E-4**
MOP3	Mean	5.19E-1 -	4.90E-1 -	4.20E-1 -	**4.48E-3**	9.09E-2 -	9.36E-2 -	1.40E-1 -	**3.46E-1**
	Std	4.79E-02	5.98E-02	1.49E-02	**1.62E-4**	7.06E-17	8.32E-03	3.41E-02	**1.65E-4**
MOP4	Mean	3.01E-1 -	2.92E-1 -	2.60E-1 -	**4.21E-3**	2.75E-1 -	2.84E-1 -	3.22E-1 -	**5.96E-1**
	Std	2.16E-02	1.78E-02	2.21E-02	**5.00E-4**	1.75E-02	1.78E-02	1.52E-02	**1.22E-3**
MOP5	Mean	2.87E-1 -	3.55E-1 -	2.54E-1 -	**7.36E-3**	4.10E-1 -	3.53E-1 -	4.30E-1 -	**7.14E-1**
	Std	2.87E-02	1.66E-01	7.27E-02	**1.64E-4**	2.22E-02	1.33E-01	7.36E-02	**2.20E-4**
MOP6	Mean	3.19E-1 -	3.09E-1 -	2.90E-1 -	**2.69E-2**	6.17E-1 -	6.23E-1 -	6.44E-1 -	**8.49E-1**
	Std	3.04E-02	1.60E-06	3.92E-02	**2.48E-4**	1.42E-02	4.37E-06	4.50E-02	**3.08E-4**
MOP7	Mean	4.52E-1 -	3.57E-1 -	3.57E-1 -	**3.58E-2**	3.70E-1 -	4.08E-1 -	4.08E-1 -	**5.66E-1**
	Std	9.71E-02	1.01E-06	3.49E-06	**3.48E-4**	4.74E-02	5.83E-06	2.94E-05	**4.53E-4**
UF1	Mean	1.01E-1 -	1.05E-1 -	9.06E-3 -	**4.42E-3**	5.99E-1 -	5.97E-1 -	7.11E-1 -	**7.18E-1**
	Std	2.18E-02	1.95E-02	3.63E-04	**1.57E-4**	2.57E-02	1.64E-02	5.16E-04	**2.90E-4**
UF2	Mean	4.01E-2 -	3.99E-2 -	1.31E-2 -	**8.12E-3**	6.79E-1 -	6.82E-1 -	7.06E-1 -	**7.14E-1**
	Std	1.66E-02	1.44E-02	6.93E-04	**1.03E-3**	1.05E-02	9.70E-03	1.10E-03	**1.12E-3**
UF3	Mean	2.55E-1 -	2.43E-1 -	**1.35E-2 =**	**1.17E-2**	4.41E-1 -	4.47E-1 -	**7.05E-1 =**	**7.06E-1**
	Std	5.35E-02	3.81E-02	**5.81E-03**	**3.94E-3**	5.11E-02	2.86E-02	**7.35E-03**	**6.69E-3**

UF4	Mean	4.08E-2 -	4.15E-2 -	4.46E-2 -	**4.00E-2**	3.90E-1 -	3.90E-1 -	3.85E-1 -	**3.91E-1**
	Std	8.93E-04	3.16E-03	3.99E-04	**5.00E-4**	1.05E-03	3.35E-03	7.39E-04	**5.69E-4**
UF5	Mean	3.07E-1 -	2.96E-1 -	2.15E-1 -	**1.60E-1**	2.49E-1 -	2.52E-1 -	3.19E-1 -	**3.28E-1**
	Std	8.63E-02	1.06E-01	1.30E-01	**1.27E-2**	5.11E-02	5.62E-02	5.76E-02	**1.62E-2**
UF6	Mean	1.80E-1 -	1.87E-1 -	2.00E-1 -	**1.20E-2**	3.23E-1 -	3.26E-1 -	3.33E-1 -	**5.09E-1**
	Std	9.86E-02	1.10E-01	2.02E-01	**5.99E-3**	5.07E-02	6.35E-02	9.47E-02	**1.05E-2**
UF7	Mean	1.62E-1 -	1.51E-1 -	1.46E-2 -	**8.12E-3**	4.37E-1 -	4.49E-1 -	**5.65E-1 =**	**5.75E-1**
	Std	1.40E-01	1.33E-01	5.91E-03	**1.54E-3**	1.01E-01	9.70E-02	**8.00E-03**	**2.42E-3**
UF8	Mean	2.39E-1 -	2.40E-1 -	3.05E-1 -	**8.15E-2**	3.34E-1 -	3.40E-1 -	2.23E-1 -	**4.79E-1**
	Std	4.00E-02	7.86E-02	5.65E-02	**3.13E-2**	2.57E-02	5.87E-02	5.78E-02	**4.85E-2**
UF9	Mean	1.18E-1 -	1.53E-1 -	2.04E-1 -	**3.71E-2**	6.89E-1 -	6.38E-1 -	5.65E-1 -	**7.77E-1**
	Std	6.57E-02	8.97E-02	1.04E-01	**1.01E-2**	5.69E-02	1.06E-01	1.19E-01	**1.93E-2**
UF10	Mean	3.74E-1 -	3.42E-1 -	1.20E+0 -	**3.12E-1**	1.68E-1 -	2.09E-1 -	3.69E-4 -	**2.80E-1**
	Std	8.85E-02	9.48E-02	3.27E-01	**4.79E-2**	4.25E-02	6.58E-02	2.02E-03	**7.49E-2**
-/=/+		27/0/0	27/0/0	26/1/0	—	27/0/0	27/0/0	25/2/0	—
best/all		0/27	0/27	1/27	27/27	0/27	1/27	2/27	27/27

5. Conclusions and future work

In this paper, a localized decomposition evolutionary algorithm was proposed for solving imbalanced MOPs. Using the localized decomposition method, the solutions' diversity is maintained by only updating the solutions under the same local region of each subproblem, which is extended when offspring are generated in the originally empty region. At the beginning, the WS decomposition is used to speed up convergence, while the TCH decomposition will be activated to maintain diversity when WS cannot bring any improvement for a long-running period. Moreover, the mating selection and the evolutionary operators are accordingly modified to cooperate well with the localized decomposition method, which can further improve the performance of LDEA. When compared to nine competitive MOEAs (MSEA, AOOSTM, AMOSTM, OPE, ACD, SPEA/R, DEAGNG, GFMMOEA, and NSGAIIARSBX), LDEA showed the advantages on two benchmark sets with imbalanced features, one benchmark set with balanced yet complicated features, and one real-world MOP built from a water reservoir system. More experiments were run to study the impact of parameters in LDEA, and the effectiveness of our modified evolutionary operators and localized decomposition approach.

Notably, even though LDEA has demonstrated its excellent competence in balancing convergence and diversity when solving imbalanced MOPs, the localized decomposition method may inevitably slow down the convergence speed, thereby reducing its effectiveness in limited evaluation scenarios. As part of our future work, the performance of this localized decomposition method will be further studied by using an adaptive adjustment strategy on the weight vectors, which can let LDEA handle more complicated MOPs or many-objective optimization problems with disconnected or incomplete PFs. Also, the use of LDEA in some real-world engineering problems will be part of our future work.

Acknowledgements

This work was supported by the Shenzhen Science and Technology Program under Grant JCYJ20220531101411027 and JCYJ20190808174801673, in part by the Natural Science Foundation of

Guangdong Province under Grant 2023A1515011238, and in part by the Guangdong Regional Joint Foundation Key Project under Grant 2022B1515120076. Prof. Carlos A. Coello Coello was partially supported by CONACyT grant no. 2016-01-1920 (Investigacion en Fronteras de la Ciencia 2016).

Reference

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*.NewYork, NY, USA: John Wiley, 2001.

[2] H. Kahraman, M. Akbel, and S. Duman, "Optimization of optimal power flow problem using multi-objective Manta Ray Foraging optimizer," *Appl. Soft Comput.*, vol. 116, pp. 108334, 2022.

[3] Y. Su, N. Luo, Q. Lin, and X. Li, "Many-objective optimization by using an immune algorithm," *Swarm Evol. Comput.*, vol. 69, p. 101026, 2022.

[4] Q. Lin, J. Chen, Z. Zhan, W. Chen, C.A. Coello Coello, Y. Yin, C. Lin, J. Zhang, "A hybrid evolutionary immune algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 711-729, 2016.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.

[6] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y.C. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609-622, 2017.

[7] S. Jiang, J. Zhang, Y. S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202-2213, 2015.

[8] J. L. L. García, R. Monroy, V. A. Sosa Hernández, and C. A. Coello Coello, "COARSE-EMOA: An indicator-based evolutionary algorithm for solving equality constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 67, p. 100983, 2021.

[9] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45-76, 2011.

[10] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: theoretical foundations and practical implications," *Theoret. Comput. Sci.*, vol. 425, pp. 75-103, 2012.

[11] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924-938, 2016.

[12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, 2007.

[13] X. Chen, J. Yin, D. Yu, and X. Fan, "A decomposition-based many-objective evolutionary algorithm with adaptive weight vector strategy", *Appl. Soft Comput.*, p. 109412, 2022.

[14] Y. Su, Q. Lin, J. Wang, J. Li, J. Chen and Z. Ming, “A constrained solution update strategy for multiobjective evolutionary algorithm based on decomposition,” *Complexity*, vol. 2019.

[15] H. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450-455, 2014.

[16] K. Li, S. Kwong, Q. Zhang, and K. Deb, “Interrelationship-based selection for decomposition multiobjective optimization,” *IEEE Trans. Cybern*., vol. 45, no. 10, pp. 2076-2088, 2015.

[17] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, “Stable matching-based selection in evolutionary multiobjective optimization,” *IEEE Trans. Evol. Comput*., vol. 18, no. 6, pp. 909-923, 2014.

[18] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” Technical Report CES-487, The School of Computer Science and Electronic Engineering, University of Essex, Tech. Rep., 2008.

[19] Y. Su, Q. Lin, Z. Ming, and K. C. Tan, “Adapting Decomposed Directions for Evolutionary Multiobjective Optimization,” *IEEE Trans. on Cybern*., pp. 1-14, 2022. doi: 10.1109/TCYB.2022.3165557.

[20] J. Hamidzadeh, N. Kashefi, and M. Moradi. “Combined weighted multi-objective optimizer for instance reduction in two-class imbalanced data problem,” *Eng. Appl. Artif. Intell*., 90 (2020) 103500.

[21] H. Liu, L. Chen, Q. Zhang, and K. Deb, “Adaptively allocating search effort in challenging many-objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433-448, 2018.

[22] C. Yue, P. N. Suganthan, J. Liang, B. Qu, K. Yu, Y. Zhu, and L. Yan, “Differential evolution using improved crowding distance for multimodal multiobjective optimization,” *Swarm Evol. Comput*., vol. 62, p. 100849, 2021.

[23] Z. Wang, M. Gong, P. Li, J. Gu, and W. Tian, “A hypervolume distribution entropy guided computation resource allocation mechanism for the multiobjective evolutionary algorithm based on decomposition,” *Appl. Soft Comput*., vol. 116, p. 108297, 2022.

[24] J. Zhou, Y. He, J. Zheng, D. Gong, Q. Yang, L. Fu, and T. R. Pei, “Hierarchical preference algorithm based on decomposition multiobjective optimization,” *Swarm Evol. Comput*., vol. 60, p. 100771, 2021.

[25] R. Wang, Z. B. Zhou, and H. Ishibuchi, “Localized weighted sum method for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3-18, 2018.

[26] Y. Su, J. Wang, L. Ma, X. Wang, Q. Lin, C. A. Coello Coello, and J. Chen, “A hybridized angle-encouragement-based decomposition approach for many-objective optimization problems”, *Appl. Soft Comput.*, vol. 78, pp. 355-372, 2019.

[27] N.T. Tam, T.H. Hung, H.T. Binh, “A decomposition-based multi-objective optimization approach for balancing the energy consumption of wireless sensor networks” *Appl. Soft Comput*., vol. 107, p. 107365, 2021.

[28] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances,” in Proc. *Congr. Evol. Comput*., 2009, pp. 203-208.

[29] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing and X. Zhu, “An Adaptive Resource Allocation Strategy for Objective Space Partition-Based Multiobjective Optimization,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 3, pp. 1507-1522, March 2021.

[30] Q. Lin, G. Jin, Y. Ma, K. Wong, C.A. Coello Coello, J. Li, J. Chen and J. Zhang, “A diversity-enhanced resource allocation strategy for decomposition-based multiobjective evolutionary algorithm,” *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2388-2401, 2018.

[31] V. Palakonda, R. Mallipeddi, and P. N. Suganthan, “An ensemble approach with external archive for multi- and many-objective optimization with adaptive mating mechanism and two-level environmental selection,” *Inf. Sci*, vol. 555, pp. 164–197, 2021.

[32] L. Dong, Q. Lin, Y. Zhou, and J. Jiang, “Adaptive operator selection with test-and-apply structure for decomposition-based multi-objective optimization,” *Swarm Evol. Comput.*, vol. 68, p. 101013, 2022.

[33] T. Zhou, Z. Hu, Q. Zhou, and S. Yuan “A novel grey prediction evolution algorithm for multimodal multiobjective optimization,” *Eng. Appl. Artif. Intell.*, 100 (2021) 104173.

[34] Y. Xiang, Y. Zhou, L. Tang and Z. Chen, “A Decomposition-based many-objective artificial bee colony algorithm,” *IEEE Trans. Cybern.*, vol. 49, no. 1, 2019

[35] Q. Lin, Z. Liu, Q. Yan, Z. Du, C. Coello, Z. Liang, W. Wang and J. Chen, “Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm,” *Inf. Sci.* vol. 339, pp. 332-352, 2016.

[36] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, “Adaptive replacement strategies for MOEA/D,” *IEEE Trans. Cybern.*, vol. 46, no. 2, pp, 474-486, 2016.

[37] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, “Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475-480, 2016.

[38] A. Kumar, P. P. Biswas, and P. N. Suganthan, “Differential evolution with orthogonal array-based initialization and a novel selection strategy,” *Swarm Evol. Comput.*, vol. 68, p. 101010, 2022.

[39] S. Brindha and S. Miruna Joe Amali, “A robust and adaptive fuzzy logic based differential evolution algorithm using population diversity tuning for multi-objective optimization,” *Eng. Appl. Artif. Intell.*, 102 (2021) 104240.

[40] X. Cai, Z. Yang, Z. Fan, and Q. Zhang, “Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization,” *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2824-2837, 2017.

[41] M.Y. Wu, K. Li, S. Kwong, Y. Zhou, and Q.F. Zhang, “Matching-based selection with incomplete lists for decomposition multi-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 714-730, 2017.

[42] J. Shi, Q. Zhang, and J. Sun, “PPLS/D: Parallel Pareto local search based on decomposition,” *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1060-1071, 2020.

[43] X. He, Y. Zhou, Z. Chen and Q. Zhang, "Evolutionary many-objective optimization based on dynamical decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 361-375, 2019.

[44] L. Chen, H. Liu, and K. C. Tan, "Decomposition based dominance relationship for evolutionary many-objective algorithm," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-6.

[45] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440-462, 2017.

[46] H. Liu, L. Chen, K. Deb, and E. D. Goodman, "Investigating the effect of imbalance between convergence and diversity in evolutionary multi-objective algorithms," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp, 408-425, 2017.

[47] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577-601, 2014.

[48] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16-37, 2016.

[49] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329-346, 2017.

[50] J. Wang, Y. Su, Q. Lin, L. Ma, D. Gong, J. Li, and Z. Ming, "A survey of decomposition approaches in multiobjective evolutionary algorithms," *Neurocomputing*, vol. 408, pp. 308-330, 2020.

[51] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: an adaptive scalarizing function-based multiobjective evolutionary algorithm," *Lecture Notes in Computer Science*, vol. 5467, pp. 438-452, 2009.

[52] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ser. GECCO '10. New York, NY, USA: ACM, 2010, pp. 519-526.

[53] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol.13, no.2, pp.284–302, 2009.

[54] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in Proc. Congr. Evol. Comput., vol. 15. 2009, pp. 361-384.

[55] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694-716, 2015.

[56] T. Ye, C. He, R. Cheng, and X. Zhang. "A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization." *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 9, pp. 5880-5894, 2021.

[57] H. Chen, G. Wu, W. Pedrycz, P. Suganthan, L. Xing, and X. Zhu. “An adaptive resource allocation strategy for objective space partition-based multiobjective optimization.” *IEEE Trans. Syst. Man, Cybern. Syst*., vol. 51, no. 3, pp. 1507-1522, 2021.

[58] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum], *IEEE ComputIntell Mag.*, 2017, 12(4): 73-87.

[59] J.J. Durillo, A.J. Nebro, E. Alba, “The jMetal framework for multi-objective optimization: Design and architecture,” in Proc. IEEE Congr. Evol. Comput., Barcelona, Spain, 2010, pp. 1-8.

[60] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, 2000.

[61] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in Evolutionary Multiobjective Optimization (Advanced Information and Knowledge Processing Series). Berlin, Germany: Springer, 2005, pp. 105-145.

[62] M. Wu, K. Li, S. Kwong, and Q. Zhang, “Evolutionary many-objective optimization based on adversarial decomposition,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 753-764, 2018.

[63] M. Li, S. Yang, and X. Liu, “Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization,” *IEEE Trans. Evol. Comput*., vol. 20, no. 5, pp. 645-665, 2016.

[64] Q. Lin, S. Liu, K. Wong, M. Gong, C.A. Coello Coello, J. Chen and J. Zhang, “A clustering-based evolutionary algorithm for many-objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391-405, 2019.

[65] W. Li, X. Yao, T. Zhang, R. Wang, and L. Wang, “Hierarchy Ranking Method for Multimodal Multiobjective Optimization With Local Pareto Fronts,” *IEEE Trans. Evol. Comput*., vol. 27, no. 1, pp. 98-110, 2023.

[66] K. Li, T. Zhang, and R. Wang, “Deep reinforcement learning for multiobjective optimization,” *IEEE Trans. Cybern*., vol. 51, no. 6, pp. 3103-3114, 2020.

[67] M. Giuliani, Y. Li, A. Cominola, S. Denaro, E. Mason, and A. Castelletti, “A MATLAB toolbox for designing multi-objective optimal operations of water reservoir system,” *Environ. Model. Softw*., vol. 85, pp. 293-298, Nov. 2016.

[68] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, “Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts,” *IEEE Trans. Evol. Comput*., vol. 24, no. 3, pp. 439-453, 2019.

[69] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, “Guiding evolutionary multi-objective optimization with generic front modeling,” *IEEE Trans. Cybern.,* vol. 50, no. 3, pp. 1106-1119, 2018.

[70] L. Pan, W. Xu, L. Li, C. He, and R. Cheng, “Adaptive simulated binary crossover for rotated multi-objective optimization,” *Swarm Evol. Comput*., vol. 60, pp. 100759, 2021.

[71] H. Kahraman, M. Akbel, S. Duman, M. Kati, and H. Sayan,“Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms,” *Swarm Evol. Comput*., vol. 75, pp. 101196, 2022.