A Survey of Weight Vector Adjustment Methods for Decomposition based Multi-objective Evolutionary Algorithms

Xiaoliang Ma, Yanan Yu, Xiaodong Li, *Fellow, IEEE*, Yutao Qi, and Zexuan Zhu, *Member, IEEE*

***Abstract*—Multi-objective evolutionary algorithms based on decomposition (MOEA/Ds) have attracted tremendous attention and achieved great success in the fields of optimization and decision-making. MOEA/Ds work by decomposing the target multi-objective optimization problem (MOP) into multiple single-objective subproblems based on a set of weight vectors. The subproblems are solved cooperatively in an evolutionary algorithm framework. Since weight vectors define the search directions and, to a certain extent, the distribution of the final solution set, the configuration of weight vectors is pivotal to the success of MOEA/Ds. The most straightforward method is to use predefined and uniformly distributed weight vectors. However, it usually leads to deteriorated performance of MOEA/Ds on solving MOPs with irregular Pareto fronts. To deal with this issue, many weight vector adjustment methods have been proposed by periodically adjusting the weight vectors in a random, predefined, or adaptive way. This work focuses on weight vector adjustment on a simplex and presents a comprehensive survey of these weight vector adjustment methods covering the weight vector adaptation strategies, theoretical analyses, benchmark test problems, and applications. The current limitations, new challenges, and future directions of weight vector adjustment are also discussed.**

***Index Terms*—Weight vector adjustment, decomposition-based MOEA, MOEA/D, Multi-objective evolutionary algorithms.**

I. INTRODUCTION

A multi-objective problem (MOP) [1] can be formulated as follows:

$$\begin{cases} \min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})) \\ \text{subject to} : \mathbf{x} \in \Omega \end{cases} \quad (1)$$

where Ω is the decision space, $\mathbf{x} \in \Omega$ is a decision vector, and $\mathbf{F}(\mathbf{x}) : \Omega \rightarrow \mathbf{R}^m$ is a vector of m objective functions. Given two decision vectors $\mathbf{x}_a$ and $\mathbf{x}_b$, $\mathbf{x}_a$ is said to dominate $\mathbf{x}_b$ (denoted as $\mathbf{x}_a \prec \mathbf{x}_b$), if and only if $\forall i \in \{1, \cdots, m\}, f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b)$ and $\mathbf{F}(\mathbf{x_a}) \neq \mathbf{F}(\mathbf{x_b})$. A solution $\mathbf{x}^* \in \Omega$ is called Pareto optimal if it is not dominated by any other solution. All Pareto optimal solutions constitute the Pareto optimal set (PS), i.e., $PS = \{\mathbf{x}^* \mid \nexists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. The mapping of the PS on the objective space is called the Pareto optimal front (PF), i.e., $PF = \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in PS\}$.

Multi-objective evolutionary algorithms based on decomposition (MOEA/Ds) are among the most well-known and effective methods to solve MOPs [2–8]. They are characterized by problem decomposition, low computational complexity, explicit neighborhood relationship, and natural diversity maintenance. Particularly, MOEA/Ds decompose a target MOP into multiple single-objective optimization subproblems based on a set of weight/reference vectors (for the sake of simplicity, weight/reference vectors are referred to as weight vectors in the following text), and the subproblems are solved cooperatively by evolving a population of solution individuals [2, 9].

Among the algorithmic components, the appropriate setting of weight vectors is extremely important to the performance of MOEA/Ds, because the weight vectors determine the search directions and, to a certain extent, the distribution of the final solution set. A basic assumption in the original MOEA/D [2] is that diverse weight vectors result in diverse Pareto optimal solutions. However, this is unnecessarily true especially on irregular PFs [10, 11]. Many studies have demonstrated that the performance of MOEA/Ds using predefined weight vectors deteriorates on MOPs with irregular PFs (e.g., disconnected, degenerate [12], inverted [13, 14], constrained, badly-scaled [15], and complex [11] PFs). MOEA/Ds also suffer on many real-world MOPs of which the PF shapes are usually irregular due to the reality constraints and highly nonlinear objective functions [3, 14, 16–18]. Without a priori knowledge of the PF shape, it could be very challenging to obtain a set of optimal weight vectors in advance [13, 19, 20].

An intuitive solution to handle irregular PFs in MOEA/Ds is to adjust the weight vectors adaptively during the evolution process. Many interesting attempts have been made along this line [11, 15, 21, 22], where the distribution of the weight vectors is tuned periodically to fit the PF shape. Thanks to the promising performance of weight vector adjustment in MOEA/Ds, there has been a rapid growth of studies on this topic in the last two decades, as the trend of yearly publications

This work was supported in part by the National Natural Science Foundation of China, under Grants 61976143, 61871272, 61911530218, 61603259, and 61772392, the Natural Science Foundation of Guangdong Province, under Grants 2019A1515010869 and 2020A151501946, the Shenzhen Fundamental Research Program, under Grant JCYJ20190808173617147, the Scientific Research Foundation of Shenzhen University for newly introduced teachers, under Grants 2019048 and 860/000002110312, the Science Basic Research Plan in Shaanxi Province of China, under Grant 2018JM6009, and the Zhejiang Labs International Talent Fund for Young Professionals. This work was supported by the National Engineering Laboratory for Big Data System Computing Technology.(*Corresponding author: Zexuan Zhu.*)

X. Ma and Y. Yu are with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: maxiaoliang@szu.edu.cn; 1810272005@email.szu.edu.cn).

X. Li is with the School of Science, RMIT University, Melbourne, 3001, Australia (e-mail: xiaodong.li@rmit.edu.au).

Y. Qi is with the School of Computer Science and Technology, Xidian University, Xi'an, China (e-mail: qi_yutao@163.com).

Z. Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China; Shenzhen Pengcheng Laboratory, Shenzhen 518055, China; and also with the SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen University, Shenzhen 518060, China (e-mail: zhuzx@szu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2020.2978158, IEEE Transactions on Evolutionary Computation

on weight vector adjustment methods demonstrated in Fig. 1.

The development of weight vector adjustment methods over the last two decades can be roughly chronicled in three stages as follows. The first stage ranging from the 1990s to 2006 is characterized by random/predefined weight vector adjustment methods. The representatives include evolution strategy with rotation matrix adaptation [23], bang-bang weight aggregation (BMA) [24], multiple-objective genetic local search (MOGLS) [25], and Pareto simulated annealing (PSA) [26]. The second stage covering 2007$\sim$2013 is featured by population/archive-guided, neighbor-weight-vector-based, and preference-based methods, e.g., MOEA/D [2], MOEA/D based on transformation (T-MOEA/D) [27], MOEA/D with novel weight design (W-MOEA/D) [28], evolutionary multi-objective approach based on simulated annealing (EMOSA) [10], MOEA/D with asymmetric Pareto-adaptive scheme ($pa\lambda$-MOEA/D) [29], MOAE/D based on reference point (R-MEAD) [30], and preference-inspired coevolutionary algorithm (PICEA-g) [31]. The third stage starting from 2014 to present is dedicated to handling different kinds of irregular PFs.

This work focuses on weight vector adjustment on a simplex and presents a comprehensive survey of the studies on weight vector adjustment methods for MOEA/Ds. The survey is expected to motivate the reader to appreciate the importance of weight vector adjustment in MOEA/Ds and provide a better understanding of the principle of weight vector adjustment. With this in mind, we review the existing weight vector adjustment methods and categorize them into six groups, namely random/predefined adjustment, fitting-based adjustment, local-population-guided adjustment, local-archive-guided adjustment, neighbor-weight-vector-guided adjustment, and preference-based adjustment. The test problems and real-world applications of the weight vector adjustment methods are also discussed in this survey followed by the limitations, challenges, and future directions.

The rest of this paper is organized as follows. Section II introduces the general framework of weight vector adjustment methods. Section III reviews the weight vector adjustment strategies. Section IV presents the test problems and real-world applications of weight vector adjustment methods. Section V discusses the potential directions for future work. Finally, Section V concludes this paper. For the convenience of the readers, the common nomenclature used in this work is provided below.

NOMENCLATURE

- $[\mathbf{x}^i, \mathbf{F}(\mathbf{x}^i), \mathbf{w}^i]$: the i-th subproblem in P_t
- $[\mathbf{x}^i, \mathbf{F}(\mathbf{x}^i)]$: the i-th individual in P_t
- $\mathbf{R}$: the current reference vector set
- $\mathbf{r}$: a reference vector
- $\mathbf{W}$: the current weight vector set
- $\mathbf{w}$: a weight vector subject to $\sum_{i=1}^{m} w_i = 1$, $w_i \geq 0$, and $i = 1, ..., m$
- $\mathbf{w}_b$: the best weight vector specified by the preference
- $\mathbf{x}$: a decision vector
- $\mathbf{x}^i$: the i-th solution in P_t
- G_{max}: the maximal number of evolution generation
- N: the population size
- O_t: the generated offspring population
- P_t: the current evolutionary population
- P_{t+1}: the next parent population

II. THE GENERAL FRAMEWORK OF MOEA/DS WITH WEIGHT VECTOR ADJUSTMENT

A. Decomposition of MOPs

The majority decomposition methods to convert an MOP into multiple single-objective optimization subproblems are based on a set of weight vectors. A weight vector is a carrier in MOEA/D to host a corresponding subproblem and the subproblem formulation can somehow influence the weight vector. The most widely used subproblem formulation methods include weighted sum (WS) [32], Tchebycheff method [32–34], and penalty-based boundary intersection (PBI) [2].

1) WS methods [32]: Let $\mathbf{w} = (w_1, \ldots, w_m)$ be a weight vector, subject to $w_i \geq 0$ and $\sum_{i=1}^{m} w_i = 1$, a subproblem in WS methods is defined in the form

$$\min_{\mathbf{x}\in\Omega} g^{ws}(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^{m} w_i \times f_i(\mathbf{x}).$$

WS methods can work well for convex PFs but not for nonconvex ones [32]. Localized WS method [33] and Tchebycheff method [32] could be used to overcome this issue.

2) Tchebycheff method [32]: In this method, a subproblem is defined as

$$\min_{\mathbf{x}\in\Omega} g^{tch}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1\leq i\leq m} \left\{ w_i \times (f_i(\mathbf{x}) - z_i^*) \right\},$$

where $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$ is the ideal point subject to $z_i^* \leq \min\{f_i(\mathbf{x})|\mathbf{x} \in \Omega\}$. Tchebycheff method succeeds in handling nonconvex PFs, however, it suffers from a nonlinear mapping between the subproblem optimal solution and the subproblem weight vector [11].

3) PBI method [2]: A PBI subproblem can be obtained via

$$\begin{aligned} \min_{\mathbf{x}\in\Omega} \quad & g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = d_1 + \theta \times d_2, \\ \text{s.t. } d_1 = & \frac{\mathbf{w}^T[\mathbf{F}(\mathbf{x}) - \mathbf{z}^*]}{\|\mathbf{w}\|_2}, \; d_2 = \|\mathbf{F}(\mathbf{x}) - \mathbf{z}^* - d_1 \times \mathbf{w}\|_2. \end{aligned}$$

where θ is a penalty parameter that should be tuned properly.

B. General framework of MOEA/Ds with weight vector adjustment

By using the aforementioned decomposition methods, an MOP can be decomposed into a set of single-objective or multi-objective optimization subproblems [2, 35]. These subproblems are related to each other and can be solved in a collaborative manner. Each subproblem is associated with a weight vector that defines the neighbor relationship among subproblems. A subproblem is optimized using information mainly from its neighboring subproblems. The configuration of weight vectors is the key to the success of MOEA/Ds. Using uniformly distributed weight vectors usually leads to deteriorated performance of MOEA/Ds on MOPs with irregular PFs. Recently, many approaches have been proposed to deal with this issue by adaptively adjusting the weight vectors.

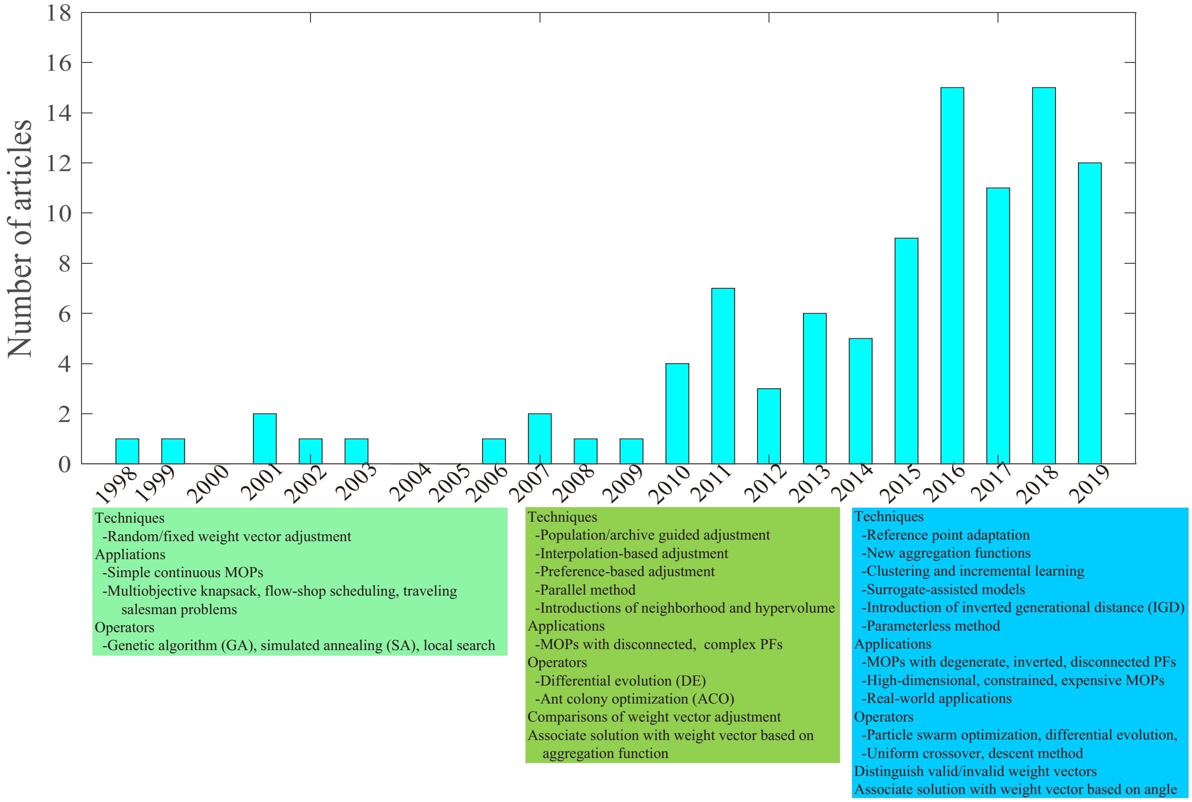


Fig. 1. The number and features of the published articles on weight vector adjustment methods collected the authors since the 1990s.

Algorithm 1 The general framework of MOEA/Ds with weight vector adjustment

---

Require: N: number of the subproblems; m: number of the objectives; $\mathbf{z}^*$: ideal point;
Ensure: The final population.
Step 1 **Initialization:**
Step 1.1 $t \leftarrow 1$.
Step 1.2 $P_t \leftarrow$ RandomInitialize(N).
Step 1.3 $\mathbf{W} \leftarrow$ Uniformly/randomly weight/reference vectors.
Step 1.4 $\mathbf{z}^* = (z_1^*, \ldots, z_m^*), z_i^* = \min_{p \in P_t} f_i(p), i = 1, ..., m$.
Step 2 **Evolution:**
Step 2.1 Use evolutionary operator(s) to generate offspring population O_t based on the parent population P_t.
Step 2.2 Use O_t to update the ideal point $\mathbf{z}^*$ and P_t.
Step 2.3 $t \leftarrow t + 1$.
Step 3 **Weight vector adjustment:**
If the adjustment criteria are met,
Delete the old ineffective/crowded weight vectors.
Add the new potential weight vectors.
End If
Step 4 **Stopping criterion:**
If the stopping criterion is met, output the obtained individuals.
Otherwise, go to Step 2.

Algorithm 1 shows the general framework of MOEA/Ds with weight vector adjustment. At the beginning, the population P_t is randomly initialized in Step 1.2 and then the evolution of the population is guided by uniformly/randomly distributed weight vectors (Step 1.3), which are usually generated by the simplex-lattice methods [36–38] or uniform design [39, 40]. The ideal point $\mathbf{z}^*$ is initialized by the minimal values of the individuals in P_t in Step 1.4. After initialization, evolutionary operators are performed on the current population P_t to generate offspring population O_t, which is then used to update P_t and the ideal point $\mathbf{z}^*$ in Step 2.2. If some triggering condition of weight vector adjustment is satisfied, the weight vectors are adjusted periodically by the deletion-then-addition process in Step 3. The old ineffective/crowded weight vectors are first deleted and then new potential weight vectors are added based on the current population, archive, neighbor weight vectors, and/or preference. The algorithm is terminated in Step 4 if some stopping criterion is reached.

MOEA/Ds with uniformly distributed weight vectors can obtain good performance on an MOP with a regular PF as shown in Fig. 2 (a). However, their performance deteriorates on MOPs with irregular (e.g., disconnected, degenerate [12], inverted [13, 14], constrained, badly-scaled [15], and complex [11]) PFs as shown in Fig. 2 (b)-(d). On an MOP with disconnected, degenerate, or inverted PF, some weight vectors may have no intersection with the PF as shown in Fig. 2 (b)-(d). This could make multiple weight vectors correspond to the same Pareto optimal solution, leading to the reduction of subproblem diversity and waste of computing efforts [21, 22, 41]. On an MOP with long peaked and tailed PF, using uniformly distributed weight vectors would result in crowded distribution of the subproblem optimal solutions in the middle part of the PF whereas sparse distributions on both ends of the PF as shown in Fig. 2 (e). The above examples show the difficulties of using predefined weight vectors in MOEA/Ds to handle irregular PFs. A natural idea to solve these issues is to adjust the weight vectors progressively to fit the PFs based on the

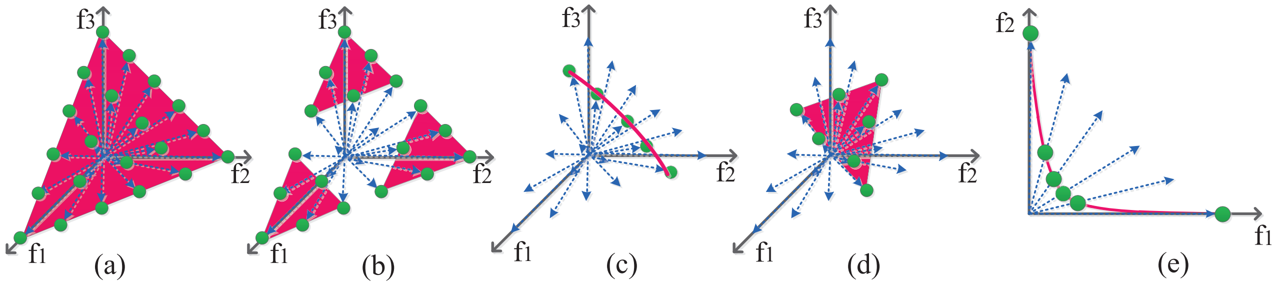


Fig. 2. Mismatch between weight vectors and the PF [13, 22]. (a) Regular PF, (b) Discontinuous PF, (c) Degenerate PF [12], (d) Inverted PF [13, 14], (e) PF with long peak and tail [11]
.

TABLE I
SUMMARY OF THE DIFFERENCES AMONG SIX WEIGHT VECTOR ADJUSTMENT STRATEGIES.

Weight vector adjustment strategy	**Heuristic information source**
Random/predefined adjustment	No feedback information usually
Fitting-based adjustment	The whole population/archive
Local-population-guided adjustment	Local population individuals
Local-archive-guided adjustment	Local archive individuals
Neighbor-weight-vector-based adjustment	Neighbor weight vectors
Preference-based adjustment	Preference information

assistant of the current population, archive, neighbor weight vectors, and/or preference.

III. WEIGHT VECTOR ADJUSTMENT STRATEGIES

Various weight vector adjustment strategies for MOEA/Ds have been proposed in the literature. We provided a review of these strategies in this section. Based on the source of the heuristic information used in the adjustment, the existing weight vector adjustment strategies for MOEA/Ds can be classified into random/predefined, fitting-based, local-population-guided, local-archive-guided, neighbor-weight-vector-guided, and preference-based adjustments. The overview of the strategies and the corresponding heuristic information sources are summarized in Table I. Note that some methods could fall within multiple strategies. We categorize and review such methods in one most related strategy.

A. Random/predefined adjustment

This type of adjustment methods changes weight vector(s) in a random/predefined way to guide the evolution of the population. They can be divided into three groups, namely random, predefined, and predefined-then-random adjustment methods.

Random weight vector adjustment is the most commonly used method in the early studies. For example, MOGLS [25, 44] generates a weight vector randomly at each generation to guide the elite solutions to perform a local search. The study [42] uses a single individual to solve an MOP guided by a search direction modified randomly as shown in Fig. 3 (a). The work [23] assigns a random weight vector to each individual in each generation to approximate the PF as shown in Fig. 3 (b).

Predefined weight vector adjustment was proposed in BWA [24]. Particularly, in BWA, the weight vector $\mathbf{w}(t) = [w_1(t), w_2(t)]$ is changed periodically in a fixed way, i.e. $w_1(t) = |\sin(2\pi t/H)|$ and $w_2(t) = 1 - w_1(t)$, to guide the population search as shown in Fig. 3 (c), where t is the generation index and H is the frequency of the weight change.

Predefined-then-random scheme uses predefined weight vectors in the initialization and then introduces random weight vectors in the later stage. For example, RVEA* [4] uses random reference vectors to replace the invalid reference vectors associated with no solution in the evolutionary process. A-IM-MOEA [45] adopts an adaptive reference vector approach composed of exploration and exploitation stages. In the exploration stage, the reference vector associated with the maximal number of candidate solutions is replaced by a random reference vector to improve the population diversity. In the exploitation stage, the reference vectors associated with no solutions are replaced by random reference vectors to fit the PF. MOEA/D-RW [46] introduces random weight vectors if the population evolution is stagnated. PICEA-w [43] co-evolves the population and the weight vectors by periodically adding random weight vectors. Similarly, GP-A-NSGA-III [47] also co-evolves the population and the reference vectors using a particle swarm optimization (PSO) and the fitness values of the reference vectors are defined as the number of population individuals associated with them. The algorithm ranks candidate individuals on all subproblems and assigns individual fitness by its smallest rank among all subproblems. As the example shown in Fig. 3 (d), the individuals I_1, I_2, and I_6 survive because each of them performs the best on two subproblems, whereas I_3, I_4, and I_5 are discarded because they fail to perform the best on any subproblem. For each survived individual, the selected weight vector must be the one that ranks the survived individual as the best. If more than one weight vector ranks the survived individual as the best, the one that is the furthest from the survival individual is selected, i.e., $\mathbf{w}_{r_1}$ for I_1, $\mathbf{w}_2$ for I_2, and $\mathbf{w}_{r_3}$ for I_6 survive in Fig. 3 (d).

Compared with other adjustment strategies, random/predefined weight vector adjustment is the simplest and easiest to implement, yet it is difficult to ensure the convergence and uniformity of the obtained solutions [4, 23, 46], especially on MOPs with discontinuous, badly-scale, or degenerated PFs [48]. A summary of random/predefined adjustment strategies is presented in Table II.

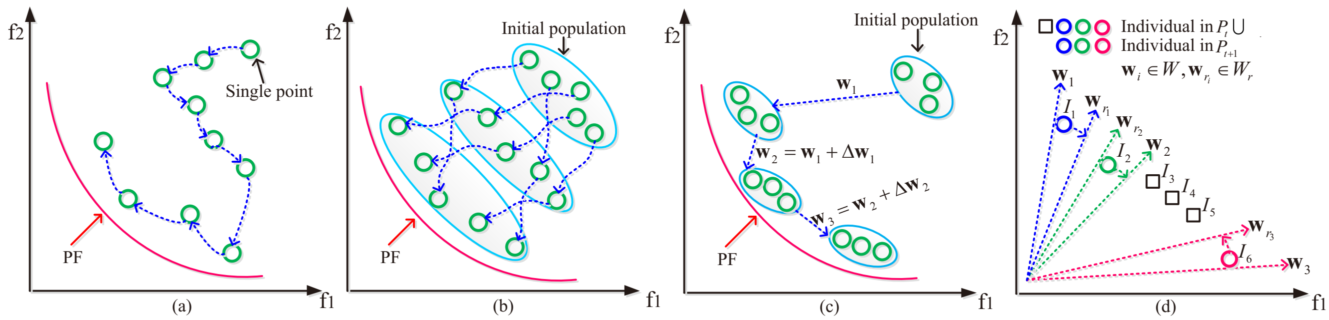


Fig. 3. Random/predefined adjustment. (a) Single point guided by a search direction modified randomly and slightly [42], (b) a random weight vector is assigned to each individual [23], (c) predefined and periodical search direction adjustment in BWA [24], (d) weight vectors co-evolved with the population in PICEA-w [43].

TABLE II
SUMMARY OF STUDIES ON RANDOM/PREDEFINED ADJUSTMENT STRATEGIES.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
MOGLS [44], IMMOGLS [25]	Generate a new random weight vector at each generation	Begin of evolution; every generation	Simple and easy to implement, no parameter introduced	Lack of diversity along PFs [46]
ES with RMA[23]	Assign a new random weight vector to each individual in each generation	Begin of evolution; every generation	Simple and easy to implement, no parameter introduced	Hard to balance the convergence and diversity of the obtained solutions [23]
[42]	Modify the weight vector randomly and slightly	Begin of evolution; from T_{max} to T_{min}	Simple and easy to implement	Hard to balance the convergence and diversity of the obtained solutions
MOEA/D-RW[46]	Introduce random weight vectors if the population evolution is stagnation	Evolution stagnation; every generation	Improve MOEA/D on MOPs with irregular PFs	Hard to balance the convergence and diversity of the obtained solutions on irregular PFs
PICEA-w [43]	Integrate $\mathbf{W}_r$ into $\mathbf{W}$ to select the elite solutions, which chooses the new weight vectors from $\mathbf{W} \cup \mathbf{W}_r$	Begin of evolution; every generation	Less sensitive to the problem geometry	Not very efficient on degenerate MaOPs by random generation of weight vectors [48]
RVEA* [4]	Replace an invalid reference vector by a random weight vector	Begin of evolution; every generation	Improve RVEA on MOPs with irregular PFs	No guarantee for local solution density [4], extra burden for the invalid reference vector [49]
A-IM-MOEA[4]	Use random reference vector to replace the most valid one in exploration stage and the invalid one in exploitation stage	Begin of evolution; every generation	Improve the performance of IM-MOEA on MOPs with discontinuous or degenerate PFs	Random insertion of new directions, introduction of parameter θ [22], and no guarantee for the uniformity of the obtained solutions [18]
GP-A-NSGA-III [47]	Co-evolve the population with the reference points	Begin of evolution; every generation	Improve NSGA-III on MOPs with irregular PFs	The coverage of reference points over PF may be unsatisfactory due to the strong randomness in PSO
BWA [24]	Change the weight vector gradually and periodically	Begin of evolution; every generation	Simple and easy to implement	Hard to balance the convergence and diversity of the obtained solutions

B. Fitting-based adjustment

Random/predefined adjustment usually uses no heuristic information and struggles to handle MOPs with irregular PFs. To deal with this issue, more sophisticated weight vector adjustment strategies were proposed by involving heuristic information. For example, fitting-based adjustment methods estimate the PF shape through the interpolation or fitting of the evolving population or archive, and resample uniform reference points on the estimated PF for weight vector design [28, 50–54].

On bi-objective optimization problems, W-MOEA/D [28] uses a piecewise linear interpolation of the non-dominated individuals to approximate the PF, and then uniformly samples a set of points on the estimated PF for weight vector generation. Taking Fig. 4 (a) as an example, the PF is estimated by two curves L_1 and L_2, which are piecewise linear interpolations of the non-dominated individuals. The uniformly sampled points $\mathbf{F^1}, \mathbf{F^2}, ..., \mathbf{F^7}$ on the estimated PF are used to generate new weight vectors. Cubic spline interpolation on the non-dominated solutions was proposed in [55] to estimate the PF shape, in which a set of reference vectors are sampled to generate new weight vectors. The above two adjustment methods have achieved success in many bi-objective problems, but it is nontrivial to extend them to MOPs with more objectives.

To deal with MOPs with more objectives, $[f_i(\mathbf{x})]^{1/p}$ is used to replace $f_i(\mathbf{x})$ in T-MOEA/D [27] for transforming the non-dominated individuals to approach $f_1+\cdots+f_m = 1$ as closely as possible. The algorithm samples uniform points on the transformed distribution and uses the inverse transformation to obtain the corresponding points in the original distribution. Taking Fig. 4 (b) for example, $\mathbf{F}(\mathbf{x^1}), ..., \mathbf{F}(\mathbf{x^5})$ are the non-dominated individuals used to estimate the parameter p of $\sum_{i=1}^{m}[f_i(\mathbf{x})]^{1/p} = 1$ and their corresponding transformed individuals are $\bar{\mathbf{F}}(\mathbf{x^1}), ..., \bar{\mathbf{F}}(\mathbf{x^5})$, i.e. $\bar{\mathbf{F}}(\mathbf{x^i}) = [\mathbf{F}(\mathbf{x^i})]^{1/p}, i = 1, ..., 5$. $\mathbf{F^1}, ..., \mathbf{F^5}$ are uniformly sampling points on the estimated PF $\sum_{i=1}^{m}[f_i(\mathbf{x})]^{1/p} = 1$ and their corresponding inverse transformed individuals $\hat{\mathbf{F}}^\mathbf{1}, ..., \hat{\mathbf{F}}^\mathbf{5}$, i.e., $[\hat{\mathbf{F}}^\mathbf{i}]^{1/p} = \mathbf{F^i}, i = 1, ..., 5.$ are used to generate weight vectors. Similarly, $pa\lambda$-MOEA/D [50] models the PF as $[f_1(\mathbf{x})]^p+\cdots+[f_m(\mathbf{x})]^p = 1$, where the parameter p is learned from the non-dominated solutions in the archive. The proposed algorithm resamples the reference vectors on the estimated PF by maximizing the hyper-volume (HV) metric as shown in Fig. 4 (c). Obviously, $pa\lambda$-MOEA/D and T-MOEA/D are suitable for MOPs with PF shapes close to $[f_1(\mathbf{x})]^p + \cdots + [f_m(\mathbf{x})]^p = 1$ but they fail to estimate PFs with more complex shapes. To alleviate this issue, the curve model $[f_1(\mathbf{x})]^p + [f_2(\mathbf{x})]^q = 1$ was proposed in the

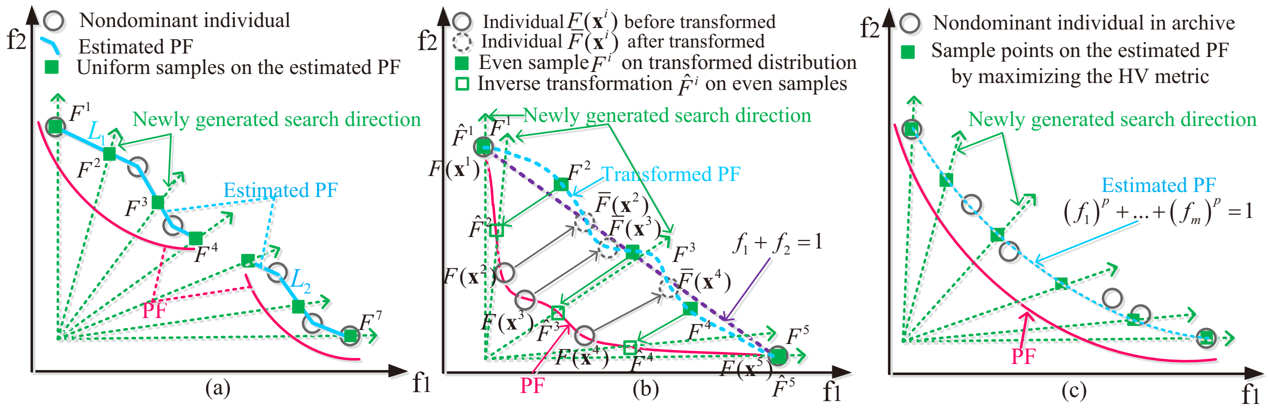


Fig. 4. Fitting-based weight vector adjustment. (a) Piecewise linear interpolation of the non-dominated individuals in W-MOEA/D [28], (b) objective transforming technique in T-MOEA/D [27], (c) using $f_1^p + \cdots + f_m^p = 1$ to approximate the distribution of the non-dominated individuals in $pa\lambda$-MOEA/D [50].

asymmetric Pareto-adaptive scheme [56] to fit bi-dimensional PF where the reference vectors are resampled on the estimated PF to maximize their HV metric. Piecewise hyperplane fitting of the obtained non-dominated individuals was introduced in [51] to fit more complicated PF shape. However, the piecewise hyperplane may fail to fit highly nonlinear PFs. Instead, a self-organizing mapping (SOM) network was used to fit the population and adjust the weight vectors based on the trained network neuron in [53]. To improve the analyzability of the fitting model, Gaussian process regression was introduced to learn the population distribution in [54, 57]. The proposed algorithms sample points randomly on the fitting model. Inferior samples and samples with large prediction variances are deleted to guarantee the performance, and the remaining samples are tailored with a diversity maintain strategy. DEA-GNG [58] learns the topological structure of the PF with a growing neural gas network. Both the reference vectors and the scalarizing functions are adjusted based on the learnt topological neighbors. DBEA-DS [59] uses two sets of reference vectors and selects the most suitable set at each generation based on the s-energy metric of the obtained population.

Different from other weight vector adjustment strategies, fitting-based adjustment is based on the whole population/archive distribution, instead of individual local density, and thus more robust to the effect of outliers. However, fitting-based adjustment calls for a prior model and the continuity of individual objective function [22], which might hamper the application of this strategy to real-world problems. Moreover, fitting-based adjustment is also easily misguided by the approximated solutions with poor convergence and diversity. A summary of fitting-based adjustment methods is presented in Table III.

C. Local-population-guided adjustment

Different from the previous fitting-based adjustment that uses global population distribution, local-population-guided adjustment is usually implemented based on local individual density.

Many local-population-guided adjustment methods are based on the angle between the search directions and the population individuals. For example, SDEA [62] divides the objective space into multiple subregions by uniform search directions and uses the weight-sum method to find the non-dominated solutions in each subregion independently. A sub-region with few non-dominated solutions is decomposed into multiple ones by adding new search directions. MOEA/D-AM2M [52, 63] periodically selects K representative individuals and uses them to decompose the objective space into K subregions based on their included angles. For example, as shown in Fig. 5 (a), three representative individuals $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in a bi-objective problem are selected and the objective space is accordingly divided into subregions $\{\mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3\}$. Search directions in each subregion are chosen to maximize their minimal included angles, e.g., $\{\mathbf{w}_1, \mathbf{w}_3, \mathbf{w}_5\}$ are selected as the new search directions of subregion Ω_2. Similarly, VaEA [64] applies the maximum-vector-angle-first principle to select elite individuals from the key layer of non-dominated sorting one by one to generate new search directions. OD-RVEA/AR [65] deletes invalid search directions that are not associated with any solutions, and selects individuals with the largest included angles to their nearest search directions to generate new search directions. A similar adjustment based on the cosine value of the angle between a population individual and its nearest search direction was proposed in EARPEA [66]. Angle-based weight vector adjustment suits for MOPs with disconnected, degenerate, or inverted PFs, but not for MOPs with long-tailed/peaked PFs.

Local-population-guided adjustment can also be based on individual distance and search directions. For instance, DDEA [49] initializes the next parent population P_{t+1} by m extreme individuals. It iteratively selects an individual $\mathbf{p}$ from $P_t \cup O_t$ with the maximal distance to P_{t+1} to generate a weight vector $\mathbf{w}_p$ for diversity maintenance, and adds the individual $\mathbf{s}$ with the optimal function value on $\mathbf{w}_p$ into P_{t+1} to improve population convergence until the size of P_{t+1} reaches N. Similarly, g-DEBA [22] iteratively deletes an invalid reference direction, and adds a new reference direction using the population individual with the largest perpendicular distance to the nearest reference direction of the deleted direction. A deleted reference directions has a chance to return back to active reference directions, if it belongs to the initial set of

TABLE III
SUMMARY OF STUDIES ON FITTING-BASED ADJUSTMENTS.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
W-MOEA/D[28]	Linear interpolation of non-dominated solutions+uniform sample on the estimated PF	$\geq G_{max}/3$ generations; $G_{max}/20$ generations	Suit to bi-objective problems	Easily overfitting due to the outliers[54], hard to extend to high-dimensional problems
tw-MOEA/D[55]	Spline interpolation of non-dominated solutions+uniform sample on the estimated PF	$\geq$ 50,000 fitness evaluations; 5,000 fitness evaluations	Suit to bi-objective problems	Easily overfitting due to the outliers[54], hard to extend to high-dimensional problems
DMOEA/D[51]	Piecewise linear fitting of non-dominated solutions+uniform sample on the estimated PF	Begin of evolution; 10 generations	Extension of [28] to tri-objective problems	Hard to estimate the PF and sample points on the estimated PF uniformly [48], easily overfitting due to the outliers[54]
T-MOEA/D[27]	Use an objective transform strategy	$\geq G_{max}/3$ generations; $G_{max}/20$ generations	Suit to MOPs whose PFs close to$\sum_{i=1}^{m}[f_1(\mathbf{x})]^{1/q} = 1$	Hard to deal with MOPs with inverted, degenerate, and disconnected PFs
$pa\lambda$-MOEA/D[29]	Approximate PF by $\sum_{i=1}^{m}[f_i(\mathbf{x})]^p = 1$, uniformly sample on the estimated PF	Archive size$\geq 2N$; every generation	Suit to the PF shape close to $\sum_{i=1}^{m}[f_i(\mathbf{x})]^p = 1$	Symmetry and continuity assumed on PF[22], expensive cost in calculating HV metric[60]
$apa\lambda$-MOEA/D [56]	Approximate PF by$[f_1(\mathbf{x})]^p + [f_2(\mathbf{x})]^q = 1$ uniformly sample on the estimated PF	Archive size$\geq 2N$; every generation	Suit to the PF shape close to $f_1^p + f_2^q = 1$	May fail on MOPs with discrete and high-dimensional PFs [61]
MOEA/D-AWG[57] MOEA/D-LTD[54]	Approximate PF by GP regression, uniformly sample on the estimated PF	$\geq 0.3G_{max}$ generations; 20 generations	Easy to learn PF shape, reduce the effect of outliers	High computational cost in GP regression
DBEA-DS[59]	Select better one from two alternative sets of reference vectors based on the s-energy metric	Begin of evolution; each generation	Suit to MOPs with regular or inverted PF	May not work well on discontinuous or degenerate PFs
MOEA/D-SOM, M2M-SOM[53]	Update weight vectors by the neurons' weight in the trained SOM network	Archive size$> 5N$; every generation	Easy to learn complex PFs	Expensive computational cost in training a SOM network
DEA-GNG[58]	Combining nodes of the learned GNG network and a set of uniform reference vectors	Begin of evolution; every generation	Both reference vectors and scalarizing functions are adjusted	Introduction of additional parameters

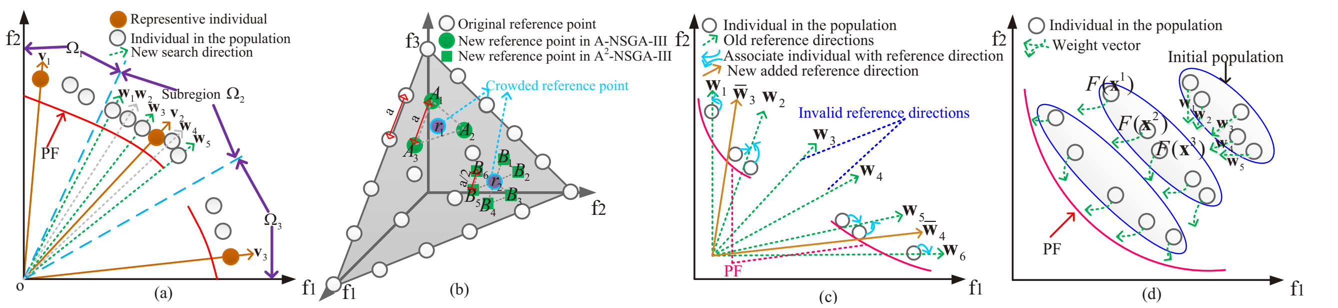


Fig. 5. Population guided weight vector adjustment. (a) Search direction adaptation in each subregion by maximizing the minimal angles among them in MOEA/D-AM2M [52, 63], (b) Addition of reference points in A-NSGA-III [3] and A^2-NSGA-III [17], (c) deleting invalid reference directions and adding new reference directions at the midpoint of two effective but distant neighboring direction vectors in MaOEA/D-2ADV [48], (d) search direction adaptation away from its nearest non-dominated neighbor in PSA [26].

uniform reference directions.

A few methods delete the ineffective and crowded weight vectors, and generate new vectors near the effective ones. For example, A-NSGA-III [3] periodically deletes invalid reference points associated with no individuals, and adds a simplex with m neighbor reference points for each crowded reference point. Taking Fig. 5 (b) for example, $\mathbf{r}_1$ is supposed to be a crowded reference point and a simplex $A_1A_2A_3$ neighboring $\mathbf{r}_1$ is added to reduce the crowdedness of $\mathbf{r}_1$. AMOEA/D [67] uses the same weight vector adjustment as A-NSGA-III except that the adjustment timing is based on the median of dispersion of the population (MDP). A^2-NSGA-III [17] extends A-NSGA-III by adding m neighboring simplices rather than one simplex for each crowded reference point to improve the diversity. As the example shown in Fig. 5 (b), $\mathbf{r}_2$ is a crowded reference point and $m = 3$ neighboring simplices are added, i.e., the reference points $B_1, ..., B_6$, are added by A^2-NSGA-III to improve the diversity of the reference points. To handle the randomness of the introduced reference vectors in RVEA* [4], iRVEA [68] replaces one inactive reference vector in each generation by a new one defined by a population individual that is least similar to the current active reference vectors. In [69], each non-dominated individual is considered as a tabu point. For the solutions close to any tabu point, a vector candidate is constructed based on the combination of two neighbor tabu vectors and then used to generate a new weight vector. TPEA-PBA [70] uses m corner reference directions for the sake of fast convergence in the early search stage and a set of uniformly distributed reference directions to increase population diversity in the middle stage. In the late stage, TPEA-PBA records the performance of each reference direction based on the number of its associated solutions or the dominated solutions. Poor reference directions are replaced by the variations of good reference directions. MaOEA/D-2ADV [48] improves TPEA-PBA by inserting new reference directions at the midpoint of two effective neighboring reference vectors to reduce the randomness introduced in the weight addition. As the example shown in Fig. 5 (c), MaOEA/D-2ADV firstly distinguishes the effective reference directions $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_5$, and $\mathbf{w}_6$ from the noneffective ones $\mathbf{w}_3$ and $\mathbf{w}_4$. Then, it deletes $\mathbf{w}_3$ and $\mathbf{w}_4$ and adds new reference directions $\overline{\mathbf{w}}_3$ and $\overline{\mathbf{w}}_4$ in the midpoint of two neighboring effective reference vectors. NSGA-MPBI [71] periodically identifies the valid reference points and adds the new reference points close to the valid reference points within a radius. PSA [26] associates each weight vector with an individual and modifies the vector slightly to push the individual away from its nearest non-dominated neighbor. As shown in Fig. 5 (d), given three individuals $\mathbf{F}(\mathbf{x^1}), \mathbf{F}(\mathbf{x^2})$, and $\mathbf{F}(\mathbf{x^3})$, their nearest non-dominated neighbors are $\mathbf{F}(\mathbf{x^2}), \mathbf{F}(\mathbf{x^3})$, and $\mathbf{F}(\mathbf{x^2})$, respectively. PSA adjusts the weight vectors $\mathbf{w}_1, \mathbf{w}_2$, and $\mathbf{w}_3$ to push their associated individuals $\mathbf{F}(\mathbf{x^1}), \mathbf{F}(\mathbf{x^2})$, and $\mathbf{F}(\mathbf{x^3})$ away from their corresponding nearest non-dominated neighbors. AWD-MOEA/D [72] periodically uses a set of predefined weight vectors to evolve the population and resets the search directions based on these non-dominated solutions obtained intermediately. AWA [73] is a multi-start framework of scalarized descent method by repeating two procedures, i.e., subdivision and relocation. The subdivision procedure generates new weight vectors filled by the midpoints of the current weight vector pairs iteratively. The relocation procedure adjusts each weight vector so that the optimum of its aggregation function gets close to the desired solution. AWA tends to suffer from prematurity. To relieve this issue, AWA-SSCAW [74] dynamically detects the round of weight vectors, controls the percentage of update weight vectors, and modifies the linear division ratio.

On one hand, local-population-guided adjustment is the most common way to adjust the distribution of weight vectors and it works well on various irregular PFs if the current population can provide a good approximation of the PF. On the other hand, local-population-guided adjustment is easily misguided by the population with poor convergence and diversity [74]. A summary of the studies on local-population-guided adjustments is presented in Table IV.

D. Local-archive-guided adjustment

Besides the evolution population, an external archive can serve as another reference for weight vector adjustment and the size of the external archive can be much larger than that of an evolution population [11, 21, 61, 76, 77].

Local-archive-guided adjustment is typically implemented in a delete-and-add scheme. For example, MOEA/D-AWA [11] periodically deletes weight vectors searching the overcrowded area, and adds new ones to search the sparse area of the current population with the help of an external archive. Taking Fig. 6 (a) for example, MOEA/D-AWA first detects the crowded population individuals $\mathbf{F}(\mathbf{x^3})$ and $\mathbf{F}(\mathbf{x^4})$ and then new weight vectors are added according to the sparse archived individuals $\mathbf{F}(\mathbf{x_a^1})$ and $\mathbf{F}(\mathbf{x_a^2})$. MOEA/D-URAW [76] improves MOEA/D-AWA by using a uniformly random method to generate the initial weight vectors for many-objective optimization problems (MaOPs). AdaW [15] periodically removes poorly performed weight vectors, and adds new ones to search the undeveloped promising regions identified by an archive.

Another way to implement local-archive-guided adjustment is to use non-dominated archive solutions for generating reference directions. For example, FV-MOEA/D [61] periodically resets weight vectors, whose subproblem optimal solutions are the non-dominated solutions in the archive as shown in Fig. 6 (b). Similarly, E-IM-MOEA [77] selects non-dominated individuals, with the largest products of the distances to their k-nearest neighbors, from the archive to generate the reference directions. AR-MOEA [21] resets weight vectors using the contributed archive individuals based on IGD-NS metric. AR-MOEA selects individuals from the contributed archive one by one to maximize the angle among them.

Compared to other weight vector adjustment strategies, local-archive-guided adjustment is based on local archive density and the number of adjustments tends to be smaller than that of other weight vector adjustments. The advantages and disadvantages of local-archive-guided adjustment are similar to local-population-guided adjustment. Combining the external archive with the population can provide a better estimation of the PF than using merely the population. A summary of the studies on local-archive-guided adjustment is presented in Table V.

E. Neighbor-weight-vector-based adjustment

A few weight vector updating methods have been invented to keep an adjusted weight vector away from or close to its nearest neighbor. For instance, EMOSA [75] first uses predefined uniform weight vectors to guide the search of the population. In the evolutionary process, it adjusts each search direction to keep the associated individual away from its nearest non-dominated neighbors as shown in Fig. 6 (c). The neighbor relationship between two individuals is defined by their weight vectors, i.e., $\mathbf{F}(\mathbf{x}^i)$ is a neighbor of $\mathbf{F}(\mathbf{x}^j)$ if the corresponding vector $\mathbf{w}^i$ is a neighbor of $\mathbf{w}^j$. New neighbor weight vectors are used to replace the parent ones in [10], if the new weight vectors can guide the corresponding individuals away from their nearest neighbors and are not too close to other parent weight vectors. In [80], each subproblem co-evolves a solution with its weight vector, which is adjusted to minimize the subproblem fitness and maximize the distance from itself to the nearest neighbor weight vector. For bi- and tri-objective optimization problems, MOEA-ABD [81] periodically groups the weight vectors into multiple subsets, and resets the number of weight vectors in each subset proportional to the scope formed by the boundary individuals. To extend MOEA-ABD to handle MaOPs, MOEA/D-AWVAM [82] estimates the search difficulty of each subregion according to the number of non-dominated solutions within the region, and reallocates the number of weight vectors proportional to the search difficulty of each subregion. Unlike the above methods [10, 75, 80] forcing the adjusted weight vector away from its neighbor weight vector(s), CLIA [83] introduces an incremental learning of reference vectors to regenerate denser alternative reference vectors close to the valid reference vectors and reduce the number of invalid reference vectors. MOEA/HD [84] layers subproblems into different hierarchies iteratively, and resets the search direction of a lower-hierarchy subproblem as the perpendicular bisector between the optimal individuals of two neighbor superior subproblems.

Neighbor-weight-vector-based adjustment is more likely a local tuning procedure, i.e., the uniformity of the obtained solutions is not well considered [10]. Moreover, neighbor-weight-vector-based adjustment tends to be slow [10] and may fail to deal with MOPs with degenerated, long-peaked, and/or long-tailed PFs [11]. A summary of the studies on neighbor-weight-vector-based adjustment is presented in Table VI.

TABLE IV
SUMMARY OF STUDIES ON LOCAL-POPULATION-GUIDED ADJUSTMENTS.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
EARPEA[66]	Delete invalid reference vectors, add individuals with good uniformity to $\mathbf{R}$ to fill $\mathbf{R}$	Begin of evolution; every generation	Suit to MaOPs with degenerate, inverted, and disconnected PFs	May not work well on MOPs with long peaked/tailed PF
OD-RVEA/AR [65]	Delete invalid reference vectors, add individuals with good uniformity to $\mathbf{R}$ to fill $\mathbf{R}$	Not mentioned	Suit to MOPs with irregular PFs	May not work well on MOPs with long peaked/tailed PF
MOEA/D-AM2M [52, 63]	Divide the objective space into K subregions, reset the search directions for each subregion by maximizing the minimal angles among them	Begin of evolution; 100 generations	Suit to MOPs with discontinuous, degenerated, and imbalanced PFs, no extra parameter introduced	Slow convergence on MOPs with regular PFs
SDEA[62]	Divide region with fewer non-dominated solutions into several subregions	Begin of evolution; every generation	Easy to learn the PF shape	May add too many subregions without deletion
VaEA[64]	Use Maximum-vector-angle-first rule to select individuals one by one as reference directions	Begin of evolution; every generation	Help to keep the good diversity of population	The convergence of the evolutionary population can be enhanced
DDEA[49]	Use Maximum-distance-first rule to select individuals one by one as reference directions	Begin of evolution; every generation	Robust to MaOPs with irregular PFs	Not good on CPTF problems
g-DEBA[22]	Delete invalid reference vectors and add new reference vectors close to the deleted ones	Begin of evolution; every generation	Balance the performance on regular PFs and irregular PFs	Slow adjustment on irregular PFs due to neighbor based weight replacement
A-NSGA-III[3]	Remove invalid reference points, add m points around each crowded reference point	Begin of evolution; every generation	Improve NSGA-III on MOPs with irregular PFs	No points introduced around the corners, low efficient on degenerate MaOPs[48]
AMOEA/D[67]	Remove invalid reference points, add m points around each crowded reference point	MDP indicator<threshold; every ΔT generations	Improve NSGA-III on MOPs with irregular PFs	No points introduced around the corners, low efficient on degenerate MaOPs[48]
A^2-NSGA-III[17]	Remove invalid reference points, add m simplexes, each of which having $m-1$ new points around each crowded reference point	Begin of evolution; every generation	Deal with several limits of A-NSGA-III on MOPs with irregular PFs	Extra parameters introduced to overcome the limit of A-NSGA-III [22] and low efficient on degenerate MaOPs [48]
iRVEA[68]	Replace an inactive reference vector by the one defined by the least similarity individual	Begin of evolution; every generation	Improve RVEA* on replacing an inactive reference vector	Slow adjustment due to one reference vector adjusted in each generation
[69]	Replace crowded individual by the combination of two neighbor non-dominated individuals to generate new weight vector	After convergence phase; every generation	Improve the performance of MOEA/Ds on irregular PFs	Uniformity of the obtained solutions should be improved
TPEA-PBA[70]	Delete invalid reference vectors, add new reference vectors near good reference vectors	Final 50 generations; only once update	No parameter/archive required, easily extended to MaOPs	Uniformity of reference direction adjustment should be improved [48]
MaOEA/D-2ADV [48]	Delete poor reference vectors, add new ones at the midpoint of two valid neighboring ones	$\Delta_t < 10^{-4}$; ϕ_2 generations	No parameter/archive required, easily extended to MaOPs	Slow adjustment on degenerated PFs
NSGA-MPBI[71]	Add the new reference points close to the valid reference points with a threshold value δ	$M_r G_{max}$ generations; f_r generations	Help to obtain the good diversity of population	Nontrivial to apply it for MaOPs and degenerate MOPs
PSA[26]	Modify weight vector of each individual mildly to make the individual away from its neighbor	Begin of evolution; every generation	Help to keep the good diversity of population	Slow convergence speed, a parameter δ introduced [75]
AWD-MOEA/D [72]	Reset weight vectors whose optimal solutions are the non-dominated solutions in P_t	Begin of evolution; 100 generations	No parameter/archive required, easily extended to MaOPs	Slow update frequency of weight vector
AWA[73]	Add the midpoint of two weight vectors into $\mathbf{W}$, adjust them to search the desired solutions	Begin of evolution; every generation	Robust to MOPs with complex, discontinuous and degenerate PFs	Round of weight vector, slow convergence, and early convergence[74]
AWA-SSCAW[74]	AWA [73] with a step size control	Begin of evolution; every generation	Robust to MOPs with degenerate, discontinuous, and complex PFs	Slow convergence

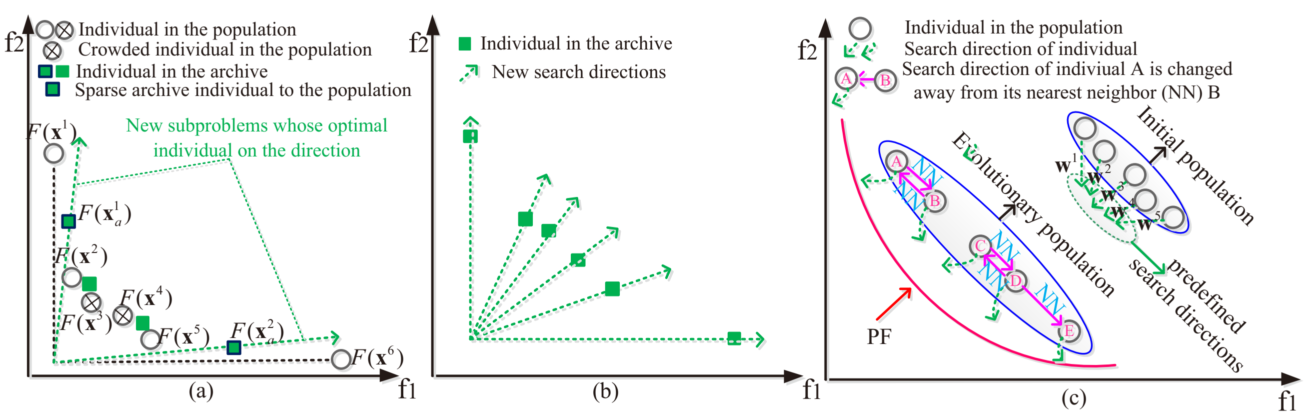


Fig. 6. Local archive-guided adjustment. (a) Deleting weight vectors searching the overcrowded area of the current population and select sparse archive individuals to the population for adding weight vectors in MOEA/D-AWA [11], (b) resetting the search directions, whose optimal individuals are the non-dominated individuals in archive [61], (c) predefined-then-adaptive weight vector adjustment making the population individual away from its nearest neighbor in EMOSA [10, 75].

TABLE V
SUMMARY OF STUDIES ON LOCAL-ARCHIVE-GUIDED ADJUSTMENT.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
MOEA/D-AWA [11, 78]	Delete weight vectors seeking crowded areas, add weight vectors searching sparse areas	$\geq 0.8G_{max}$; wag generations	Suit to MOPs with irregular PFs	Multiple extra parameters for weight adjustment[48], high computation cost for individual sparsity[49]
MOEA/D-URAW [76]	Delete weight vectors seeking crowded areas, add weight vectors seeking sparse areas	$[0.05G_{max}, 0.9G_{max}]$; every generation	Suit to MOPs with irregular PFs	Multiple extra parameters for weight adjustment[48], high computation cost for individual sparsity[49]
E-IM-MOEA[77]	Select N uniform non-dominated solutions from the archive as the reference vectors	Begin of evolution; every generation	Improve IM-MOEA in weight adaptation	High computation cost of the inverse modeling
FV-MOEA/D[61]	Reset weight vectors whose optimal solutions are the non-dominated solutions in archive	Begin of evolution; 300 generations	Robust to MOPs with various PFs	Hard to extend to MaOPs due to the expensive computation of the HV metric
AdaW [15]	Add weight vectors searching promising regions, delete poor weight vectors	$[0.05G_{max}, 0.9G_{max}]$; $0.05G_{max}$ generations	Robust to MOPs with various PFs	Effectiveness has not been verified on MOPs and MaOPs adequately
AR-MOEA[21]	Use reference points close to contributed solutions as a new $\mathbf{R}$, add the solutions in archive with the maximal angle to $\mathbf{R}$ into $\mathbf{R}$	Begin of evolution; every generation	Robust to both MOPs with regular PFs and MOPs with irregular PFs	Difficult to specify reference points to calculate the IGD-like metric [79]

TABLE VI
SUMMARY OF THE STUDIES ON NEIGHBOR-WEIGHT-VECTOR-BASED ADJUSTMENTS.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
EMOSA[75]	Modify the weight vector slightly making each individual away from its nearest neighbor	Begin of evolution; $(T_0 - T_c)/\alpha$ generations	Help to obtain a population of individuals of good convergence	The whole uniformity of the obtained solutions are not considered well [10]
EMOSA[10]	Adjust the weight vectors making each individual away from its nearest neighbor	$\lceil log_{\alpha_1} \frac{T_{max}}{T_{min}} \rceil$ generations; $\lceil log_{\alpha_2} \frac{T_{rehart}}{T_{min}} \rceil$ generations	Suit to multi-objective combination optimization	Hard to extend it to MaOPs [48]
MOEA/D-AWVAM [82]	Assign more weight vectors to search subregions with more search difficulty	Begin of evolution; 50 generations	More subproblems assigned for searching undeveloped regions	Not verified on MaOPs, not good for discontinuous PFs
MOEA-ABD[81]	Delete invalid weights, group rest weights into subsets, add new weights uniformly	T_1 generation; T_2 generations	Suit to bi-objective problems with discontinuous PFs	Nontrivial to determine the thresholds of the algorithm [18]
[80]	Each weight vector is adjusted to minimize the subproblem fitness and maximize the distance to its nearest weight vector	Begin of evolution; every generation	Not require the weight vectors in advance	Introduce the weight vector as optimization variable adding the optimization difficulty
CLIA[83]	Reduce invalid reference vectors, generate new reference vectors near valid ones, select the potential ones	Begin of evolution; every generation	Help to maintain more valid reference vectors than A-NSGA-III	Slow incremental learning, extra parameter δ required to set
MOEA/HD[84]	Reset the search direction of a lower-hierarchy subproblem as the bisector between the current solutions of two superior subproblems	Begin of evolution; 20 generations	Suit to MOPs with degenerate, disconnected PFs	Hard to extend it to MaOPs

F. Preference-based adjustment

The majority of the above-mentioned methods aim at approximating the entire PF, so that the decision-maker (DM) can select the most desirable solution(s). However, if only the region of interest (ROI) on the PF is considered [85–88], the preference information can be used in weight vector adjustment to reduce the computational overheads, especially in MaOPs, whose PFs are difficult to approximate with limited population size [37].

The reference point is the most commonly used preference information in weight vector adjustment to guide the population to search for the preference region. For example, the uniformly distributed reference vectors in NUMS-MOEA/D-STM [89] are shrunk into non-uniform ones close to the aspiration point specified by the DM. Besides the aspiration point, there are also a few ways [30, 90, 91] to generate new weight vectors around the base weight vector $\mathbf{w}_b$, decided by the nearest population individual to the reference point $\mathbf{r}$. Rather than using only a few reference points like the above methods [30, 89], PICEA-g [31] co-evolves the population and a set of preference points by constantly adding random preference points. Each individual survives depending on the number of preference points dominated by this individual, and a preference point survives depending on the number of survival individuals that can dominate this preference point. PICEA-g might not maintain the population diversity very well [92]. To address this issue, a-PICEA-g [92] introduces the crowding distance to measure the density of the candidate individuals.

Another popular preference information used for weight vector adjustment is the reference direction or the relative importance of the objectives. For instance, cwMOEA/D [93] co-evolves the weight vectors and the population to meet the user preferences, specified by the relative importance of the objectives. More specifically, new weight vectors are generated by mutation, and the parent weight vectors are replaced by their offspring if the latter has a higher preference degree than the former. In IEMO/D [94], the DM is asked to compare some pairs of solutions to obtain the preference search directions towards the corresponding preferred regions of PF.

The preference region is also a widely used information in weight vector adjustment. The work [95] uses a preference cone, defined by the desired direction and a tolerated angle, to guide the weight vector adjustment. Weight vectors are updated by small perturbation based on a preference-first-uniformity-second criterion. pMOEA/D [96] defines the preference region with an aspiration point, a reservation point, and a preference radius to adjust the weight vectors. In particular, the aspiration point and the reservation point are used to specify the preference direction of the DM and select the most preferred solution from the population. The preference region is defined by the most preferred solution and the preference radius. Similarly, RR-MOEA/D [97] also uses the aspiration point, reservation point, and preference radius to construct a preference region. The difference is that RR-MOEA/D adjusts weight vectors to search the preference region based on the update direction, i.e., from the farthest to the nearest weight vector of the desired direction. MOEA/D-PRE [98] uses a simplex as the preference region. Particularly, to obtain a uniform weight vector set $\mathbf{W}$, $\mathbf{W}$ is initialized as the corner weight vectors of the simplex, and the midpoints of every two weight vectors in $\mathbf{W}$ are added iteratively until $|\mathbf{W}| = N$. In [99], constrained optimization was modeled as a bi-objective optimization problem and solved by progressively biased weight vectors to search the preference region of better objective values and lower constraint violations.

The preference information for weight vector adjustment can also be based on value function or utility function. For example, IEM [100] progressively constructs a linear value function, i.e., a weighted sum of objectives, to guide the search of the population by asking the DM to rank pairs of solutions. However, a single weight vector may bias the search to some solutions not necessarily preferred by the DM. To handle this issue, interactive PMA [101] uses a set of weighted sum of objectives to focus the search on the preference region. To find the preferred solutions on a nonconvex part of the PF, I-MOEA/D-PLVF [102] applies an interactive way based on approximated value function (AVF) to lead MOEA/D to search the preferred solutions. In the interactive session, the DM is asked to score a few candidate solutions for learning an AVF. The learned AVF is then translated into a set of reference points biased toward the ROI. iMOEA/D [103] uses the utility function to select the most preferred individual in each interactive stage, and dynamically adjusts the weight vectors into the preferred region specified by a hypersphere.

In comparison to other weight vector adjustment strategies, preference-based adjustment tends to consume fewer computational overheads, and handle MaOPs with a smaller population size [37]. However, in such methods, the entire population may be misled by the reference point close to the PF and get trapped in local optima [98]. A summary of the studies on preference-based adjustments is presented in Table VII.

IV. Test problems and real-world applications

Both benchmark and real-world problems are widely used to investigate the advantages and weaknesses of the proposed weight vector adjustment methods. This section provides a brief overview of the benchmark problems and real-world applications used in the literature.

It is necessary and important to use controllable yet challenging test problems to investigate the performance of weight vector adjustment methods. Identifying what problems the methods work well enables researchers to get a better insight into the underlying principle of different methods and designs more efficient algorithms [105]. The current weight vector adjustment methods mainly aim at obtaining uniformly distributed solutions on MOPs with irregular PFs, including disconnect, long-tailed/peaked, degenerate, inverted, badly-scaled, incomplete, strongly convex, and hybrid PFs. The commonly used MOPs with irregular PFs to test weight vector adjustment methods are summarized in Table VIII. On MOPs with disconnect, degenerate, inverted, and/or incomplete PFs, the search direction of an MOEA/D may have no interaction with the PF as shown in Fig. 2 (b)-(d), which leads to the waste of computing effort and performance deterioration. Such problems are widely used to test the ability of an algorithm to fit the distribution of weight vectors to the shape of PF. If the PF of an MOP is badly-scaled, strongly convex, and/or long-tailed/peaked, MOEA/D with uniformly distributed weight vectors hardly can attain evenly distributed optimal solutions over the PF as shown in Fig. 2 (e). When the PF has multiple irregular features, the performance of an MOEA/D with uniformly distributed weight vectors degrades significantly and weight vector adjustment becomes more necessary and important. MOPs with irregular PFs are often selected as the test problems to show the advantage of weight vector adjustment methods over MOEA/Ds with uniformly distributed weight vectors. Weight vector adjustment methods have also gone through test in various real-world applications as summarized in Table IX.

V. Future directions

Although there are many publications on weight vector adjustment in the last two decades, many important problems are still open and new application domains are emerging constantly. Some important directions for potential future research are discussed as follows:

- Compared with the development of MOEA/Ds, weight vector adjustment methods are still in its infancy and require more attention for further growth. Most methods are based on empirical knowledge or heuristic information. Rigorous theoretical analyses are extremely scarce in this field.
- The performance of fitting-based weight vector adjustment relies on the assumed prior model and continuity of the PFs. Self-adaptive techniques could be introduced to address this problem.
- The local density of an individual/reference vector in its neighborhood is used as the main metric for weight vector adjustment. However, the local density tends to suffer from outliers. A potential research direction is to combine both local and global distributions of the population to guide the weight vector adjustment.
- Population/archive-guided adaptation might not work if the target MOP is hard to solve, i.e., the population/archive does not show good convergence and/or uniformity. It is interesting to have an adaptive adjustment frequency based on the estimated problem difficulty and the evolutionary stage.
- Most weight vector adjustment methods have demonstrated their effectiveness on the limited types of irregular PFs [15, 15, 21]. More comprehensive investigations should be carried out to cover more irregular PFs to fully reveal the advantages, disadvantages, and features of the adjustment methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2020.2978158, IEEE Transactions on Evolutionary Computation

TABLE VII
SUMMARY OF STUDIES ON PREFERENCE-BASED ADJUSTMENTS.

Algorithm	Weight/reference vector adjustment	Adjustment criterion and frequency	Advantage	Disadvantage
R-MEAD[30], IRMEAD[91], R-MEAD2[90]	Generate the weight vector $\mathbf{w}_b$ from $\mathbf{r}$ to its closest individual, sample new uniform weight vectors around $\mathbf{w}_b$	Preference is given; every generation	Simple and easy to implement	Hard to apply R-MEAD to MaOPs[90], the uniformity of the obtained solutions is dependent on preferred region shape
NUMS [89]	Shrink the uniformly distributed reference points into non-uniform ones around the pivot point	Preference is given; DM needs to interact	Obtain more reference points closer to the aspiration level	Too many parameters, i.e., $\mathbf{z}^r$, H, τ, in the preference model
RVPA[104]	Select K uniform reference vectors as the preference of DM from the population	Preference is given; DM needs to interact	Can maintain the uniform of the population	Not easy to estimate the ideal point $\mathbf{z}^*$ due to focusing on the preference region
PICEA-g[31]	Integrate random preferences $\mathbf{P}_r$ into the current preferences $\mathbf{P}$ to select the elite solutions, which chooses the new preferences from $\mathbf{P}_r \cup \mathbf{P}$	Preference is given; every generation	Coevolve preferences with the population [22]	Lack a powerful diversity maintain [92], hard to deal with MOPs whose PF with long peak and tail [22]
a-PICEA-g[92]	Similar to PICEA-g[31] but crowding distance used	Preference is given; every generation	Coevolve preferences with the population [22]	Hard to deal with MaOPs due to the use of crowding distance
cwMOEA/D[93]	Replace parent weight vector by its offspring if the later has better fitness than the former	Preference is given; every generation	Can obtain a set of solutions in the preference region	Slow weight adjustment, hard to maintain the population uniformity along PF [55]
IEMO/D[94]	DM compares some pair of solutions to obtain the preference search directions	Preference is given; $G_{max}/10$ generations	Can obtain a set of diverse solutions in the preference region	Uniformity of the obtained solutions can be improved
pMOEA/D[96]	Gradually adjust weight vectors to make the population focus the search on preference regions	Archive size$> N$; $G_{max}/30$ generations	Can find a set of solutions in the preference region	Too many parameters need to be provided by DM
[95]	A new weight vector competes with the worst parent weight vector	Preference is given; every generation	Can find uniform individuals in the preference region	Easy to miss the boundary weight vectors
RR-MOEA/D [97]	Decide the current best weight vectors, reset weights uniformly to search preference regions	$\Delta < \delta$; every generation	Can find a set of individuals in the preference region	Lack of diversity maintain mechanism for the preferred solutions
MOEA/D-PRE [98]	Initialize $\mathbf{W}$ as the pivot direction and m corner preference weight vectors, add the midpoints of any two weight vectors in $\mathbf{W}$ into $\mathbf{W}$	Preference is given; every generation	Simple and easy to implement, good convergence to the PF, controlled the size of desired regions	Is not an interactive method and do not consider multiple reference points
[99]	Adjust weight vectors to search preference region with better objective values and constraint violations	Begin of evolution; every generation	Keep good convergence and diversity of the population	Work not well on problems with rotated constraints [99]
IEM[100]	DM compares some pair of solutions to obtain a search direction	Preference is given; DM needs to interact	Suit to MOPs with convex PF	Can't find the non-convex preference part of the PF [102]
Interactive PMA [101]	Randomly sample several weight vectors compatible with DM's preference	Preference is given; from high to low	Suit to MOPs with convex PF	Can't find the non-convex preference part of the PF [102]
I-MOEA/D-PLVF [102]	Select the preferred reference vectors based on AVFs learnt from DM's preference	Preference is given; τ generations	Suit to linear and nonlinear value functions	A monotonic value function assumed [102]
iMOEA/D[103]	Select the most preferred one by a utility function and renew weight vectors into the preferred region	Preference is given; DM needs to interact	Work well on bi-objective ZDT and tri-objective DTLZ problems	Hard to model the DM's preference by utility function and a radius parameter

TABLE VIII
SUMMARY OF VARIOUS KINDS OF IRREGULAR PFS USED FOR WEIGHT VECTOR ADJUSTMENT METHODS.

Impact(s) on MOEA/D	Feature of PF	MOPs
The search direction may have no interaction with the PF using predefined weight vectors	Disconnect	ZDT3 [106], DTLZ7 [107], disconnected DTLZ1 [84], WFG2 [108], WFG47-WFG48 [109], UF5, UF6, UF9, CF1-CF3, CF8-CF10 [110], GL2-GL3 [28], GLT1,GLT4,GLT6 [51], LZDTZ2, LZDTZ5 [27], TDY1-TDY6 [111], MOP4 [35], CTP2,CTP7,CTP8,TNK [81], JY2-JY3 [112],MaF7, MaF11 [113], MaOP1-MaOP7 [52], KUR, LAU, LIS, POL, SCH2, OSY2, TAN [114], and LLZD1-LLZD7 [52].
	Degenerate [12]	DTLZ5, DTLZ6, DTLZ9 [107], DTLZ5(I,m), DTLZ4(I,m) [11], WFG3 [108], VIE3 [114], CPFT1 [115], VNT2-VNT3 [78], LLZD1 [52], MaF6, MaF8, MaF9, MaF13 [113], TOY1-TOY7 [53], and MaOP1-MaOP7 [52].
	Inverted [13]	inverted DTLZ2 [37], MaF1, MaF4 [113], DTLZ1^{-1}-DTLZ4^{-1} [17], and WFG1^{-1}-WFG9^{-1} [13].
	Constrained	C2-DTLZ2, C3-DTLZ1 [3], C1-DTLZ1, C3-DTLZ4 [22], CF1-CF10, CTP2, CTP7, CTP8, TNK [81], and CPFT1 [115].
	Incomplete [58]	ZDT3 [106], DTLZ5-DTLZ7,DTLZ9 [107], disconnected DTLZ1 [84],inverted DTLZ2 [37], WFG2-WFG3 [108],UF5-UF6, UF9 [110],GLT1, GLT6 [51], LZDTZ2,LZDTZ5 [27],MaF6-MaF11 [113],MaOP1-MaOP7 [52],TDY1-TDY7 [111],inverted DTLZ [17], and inverted WFG [13]
Uniformly distributed weight vectors cannot guarantee evenly distributed optimal solutions over the PF	Long tail and peak [11]	convex DTLZ2 [37], C2-DTLZ2, C3-DTLZ1 [3], LZDTZ3-LZDTZ4 [27], POL, REN1, SCH1-SCH2, VIE1, BIN2 [114], GLT3, GLT5 [51], GL2-GL4 [28], F1-F2 [11], JY1 [112], F1, F3, mF4, F5-F6, MP1-MP3 [71], and MaF3 [113].
	Badly-scaled	WFG1-WFG9 [108], variants of ZDT1-ZDT4, ZDT6 [116], scaled DTLZ1-DTLZ2 [37], REN2, SCH2 [114], GL1 [28], GLT2 [51], and LZDTZ1 [27].
	Strongly convex	F4 [28], F1,F3 [51], LZDTZ3-LZDTZ4 [27], REN1,SCH1-SCH2 [114], F1-F2 [11], C3-DTLZ1 [3], JY1,JY3,JY5-JY6, mF4, convex DTLZ2 [112], F1,F4-F6[71]
May have the above two imparts	Hybrid	WFG1-WFG3 [108], C2-DTLZ2, C3-DTLZ1 [3], GL4 [28],GLT3,GLT6 [51], TDY2-TDY6 [111], JY2-JY3 [112], MaF2, MaF4, MaF6-MaF11,MaF15 [113], POL,REN1,VIE2,VIE4,SCH1,BIN2,BIN4 [114],DTLZ8 [107], P^* problems, modified P^*[53], multiline distance[117], CPFT2-CPFT8[115], MaOP1,MaOP5 [52].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2020.2978158, IEEE Transactions on Evolutionary Computation

TABLE IX
SUMMARY OF REAL-WORLD APPLICATIONS OF WEIGHT VECTOR ADJUSTMENT METHODS.

Areas	Applied algorithms
fNIRS channel selection	Adaptive MOEA/D [82]
Multi-objective portfolio	DW-ACO$_R$ [118]
Energy consumption scheduling	W-N [116]
Weapon target assignment	MOEA/D-AWA [78]
Water problem	A^2-NSGA-III [17]
Aerodynamic design optimization	EDWA [16]
carbon fiber drawing	CA-MOEA [18]
Crash worthiness design	DDEA [49],A-NSGA-III [3], [17], [64]
Welded beam design	[103]
Car side impact	A-NSGA-III[3],A^2-NSGA-III[17],[64], [14]
Machine performance optimization	A^2-NSGA-III [3]
Multiobjective 0-1 knapsack	PSA [26], [82], EMOSA [10], IEM [100]
Multiobjective TSP	EMOSA [10], PMA [101]
Multiobjective flowshop scheduling	MOGLS [44], PMRPEA [119]
Job shop scheduling	GP-A-NSGA-III [47]
Multiobjective spanning tree	IEM [100]
Reservoir flood control	p-MOEA/D [96]
Three-bar truss optimization	EDWA [16], [120]

- Weight adjustment methods have shown commendable performance on MOPs with irregular PF. However, their performance tends to deteriorate on MOPs with regular PFs. To pursue a better compromise between the two types of problems is a challenging task. Some early attempts have been made along this line [21, 22, 83].
- It is difficult to distinguish between the discontinuous and sparse regions on the PF [11]. The former needs few search efforts, whereas the latter requires more subproblems to search. Using an external archive of non-dominated solutions could be the solution to distinguish them as the archive can provide an overview of the PF.
- For the preference-based weight vector adjustment, an external archive storing the non-dominated solutions also could be imposed to outline the whole PF and direct the search toward the ROI more accurately.
- The existing weight vector adjustment methods usually associate a weight vector with an individual. This "one-reference-one-individual" tends to suffer from high computation cost and low comprehensibility. Associating a reference vector with multiple individuals could be a candidate solution of this issue [35, 52].
- The effectiveness of the current adjustment methods has been verified mainly on the simple ZDT-like, DTLZ-like, and WFG-like problems, on which different adaptation methods obtain small differences in metrics. Overfitting the weight vector adjustment strategy on widely used test problems limits the generalization ability of the proposed algorithms. It is desired to develop more controllable yet challenging test problems to investigate the weight vector adjustment methods.
- There are few weight vector adjustment methods for constraint optimization, dynamic optimization, robust optimization, large-scale optimization, and expensive optimization. Metrics well designed to evaluate the performance of the obtained solutions on the ROI, especially on a real-world problem without prior knowledge of the PF, are also highly required.

VI. CONCLUSIONS

MOEA/Ds, as one kind of mainstream MOEAs, have attracted increasing attention and achieved great success. Using predefined and uniformly distributed weight vectors, MOEA/Ds have shown good performance on MOPs with regular PFs but not on MOPs with irregular PFs. With the growing complexity of real-world MOPs whose PFs tend to be irregular, weight vector adjustment strategies are proposed to enhance the existing MOEA/Ds on solving such MOPs. This paper focuses on weight vector adjustment on a simplex and presents a comprehensive survey of MOEA/Ds with various weight vector adjustment strategies. In the last twenty years, many weight vector adjustment strategies have been proposed for MOEA/Ds. This work provides an overview of the historical development of the weight vector adjustment strategies. A systematic classification of the adjustment strategies into six groups is presented for the sake of algorithm comparison and analysis. In particular, each adjustment strategy is discussed in detail to reveal their advantages and disadvantages. The survey also covers the benchmark and real-world problems used to test the existing weight vector adjustment methods. Finally, open problems and potential future directions are discussed to inspire more in-depth studies in this field.

REFERENCES

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

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

[3] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[4] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[5] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[6] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.

[7] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[8] 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.

[9] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 82–95.

[10] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[11] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[12] D. K. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 772–787.

[13] H. Ishibuchi, S. Yu, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.

[14] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Use of two reference points in hypervolume-based evolutionary multiobjective optimization algorithms," in *International Conference on Parallel Problem Solving from Nature*, 2018, pp. 384–396.

[15] M. Li and X. Yao, "What weights work for you? Adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation," *arXiv*, pp. 1–2, 2017, arXiv:1709.02679.

[16] Y. Jin, "Effectiveness of weighted aggregation of objectives for evolutionary multiobjective optimization: methods, analysis and applications," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2002, pp. 1–32.

[17] H. Jain and K. Deb, "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2013, pp. 307–321.

[18] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular pareto fronts," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2758–2770, 2019.

[19] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 807–813, 2016.

[20] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Two-layered weight vector specification in decomposition-based multi-objective algorithms for many-objective optimization problems," in *IEEE Congress on Evolutionary Computation*, 2019, pp. 2434–2441.

[21] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2018.

[22] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors." *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 1–14, 2018.

[23] Y. Jin, T. Okabe, and B. Sendhoff, "Adapting weighted aggregation for multiobjective evolution strategies," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 96–110.

[24] ——, "Dynamic weighted aggregation of evolutionary multi-objective optimization: Why does it work and how?" in *International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 1042–1049.

[25] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[26] P. Czyzak and A. Jaszkiewicz, "Pareto simulated annealinga metaheuristic technique for multiobjective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–37, 1998.

[27] H. L. Liu, F. Gu, and Y. Cheung, "T-MOEA/D: MOEA/D with objective transform in multi-objective problems," in *International Conference of Information Science and Management Engineering*, 2010, pp. 282–285.

[28] F. Q. Gu and H. L. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *International Conference on Computational Intelligence and Security*, 2010, pp. 137–141.

[29] S. Jiang, J. Zhang, and Y. S. Ong, "Asymmetric Pareto-adaptive scheme for multiobjective optimization," in *International Conference on Advances in Artificial Intelligence*, 2011, pp. 351–360.

[30] A. Mohammadi, M. N. Omidvar, and X. Li, "Reference point based multi-objective optimization through decomposition," in *IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[31] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[32] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[33] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[34] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 226–244, 2018.

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

[36] I. Das and J. Dennis, "Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[38] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.

[39] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Computers and Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.

[40] X. Ma, Y. Qi, L. Li, F. Liu, L. Jiao, and J. Wu, "MOEA/D with uniform decomposition measurement for many-objective problems," *Soft Computing*, vol. 18, no. 12, pp. 2541–2564, 2014.

[41] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *IEEE Transaction on Cybernetics*, vol. 50, no. 2, pp. 753–764, 2020.

[42] P. Serafini, "Simulated annealing for multi objective optimization problems," in *International Conference on Multiple Criteria Decision Making*, 1994, pp. 283–292.

[43] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.

[44] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.

[45] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected Pareto fronts," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2015, pp. 127–140.

[46] H. Li, M. Ding, J. Deng, and Q. Zhang, "On the use of random weights in moea/d," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 978–985.

[47] A. Masood, Y. Mei, G. Chen, and M. Zhang, "A PSO-based reference point adaption method for genetic programming hyper-heuristic in many-objective job shop scheduling," in *Australasian Conference on Artificial Life and Computational Intelligence*, 2017, pp. 326–338.

[48] X. Cai, Z. Mei, and Z. Fan, "A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 1–14, 2017.

[49] X. He, Y. Zhou, Z. Chen, and Q. Zhang, "Evolutionary many-objective optimization based on dynamical decomposition," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–15, 2018.

[50] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *International Conference on Natural Computation*, 2011, pp. 1260–1264.

[51] F. Gu, H. Liu, and K. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *International Journal of Innovative Computing*, vol. 8, no. 5, pp. 3677–3688, 2012.

[52] H. L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[53] F. Gu and Y. M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2018.

[54] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 376–390, 2019.

[55] M. Pilt and R. Neruda, "General tuning of weights in MOEA/D," in *IEEE Congress on Evolutionary Computation*, 2016, pp. 965–972.

[56] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *International Conference on Natural Computation*, 2011, pp. 1260–1264.

[57] M. Wu, S. Kwong, Y. Jia, K. Li, and Q. Zhang, "Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression," in *Genetic and Evolutionary Computation Conference*, 2017, pp. 641–648.

[58] 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 Transactions on Evolutionary Computation*, 2019, doi: 10.1109/TEVC.2019.2926151.

[59] K. S. Bhattacharjee, H. K. Singh, T. Ray, and Q. Zhang, "Decomposition based evolutionary algorithm with a dual set of reference vectors," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 105–112.

[60] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[61] S. Jiang, L. Feng, D. Yang, K. H. Chen, Y. S. Ong, N. S. Zhang, P. S. Tan, and Z. Cai, "Towards adaptive weight vectors for multiobjective evolutionary algorithm based on decomposition," in *IEEE Congress on Evolutionary Computation*, 2016, pp. 500–507.

[62] M. Wang, Y. Wang, and X. Wang, "A space division multiobjective evolutionary algorithm based on adaptive multiple fitness functions," *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 30, no. 3, pp. 1–29, 2016.

[63] H. L. Liu, L. Chen, Q. Zhang, and K. Deb, "An evolutionary many-objective optimisation algorithm with adaptive region decomposition," in *IEEE Congress on Evolutionary Computation*, 2016, pp. 4763–4769.

[64] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.

[65] H. Zhao, C. Zhang, B. Zhang, P. Duan, and Y. Yang, "Decomposition-based sub-problem optimal solution updating direction-guided evolutionary many-objective algorithm," *Information Sciences*, vol. 448-449, pp. 91–111, 2018.

[66] C. Zhou, G. Dai, C. Zhang, X. Li, and K. Ma, "Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems," *Information Sciences*, vol. 465, pp. 232–247, 2018.

[67] A. Camacho, G. Toscano, R. Landa, and H. Ishibuchi, "Indicator-based weight adaptation for solving many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization (EMO)*, 2019, pp. 216–228.

[68] Q. Liu, Y. Jin, M. Heiderich, and T. Rodemann, "Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular pareto fronts," in *IEEE Congress on Evolutionary Computation*, 2019, pp. 1726–1733.

[69] A. R. Snchez, A. Ponsich, and A. L. Jaimes, "Generation techniques and a novel on-line adaptation strategy for weight vectors within decomposition-based moeas," in *Genetic and Evolutionary Computation Conference Companion (GECCO'19)*, 2019, pp. 229–230.

[70] C. Zhu, X. Cai, Z. Fan, and M. Sulaman, "A two-phase many-objective evolutionary algorithm with penalty based adjustment for reference lines," in *IEEE Congress on Evolutionary Computation*, 2016, pp. 2161–2168.

[71] Q. Zhang, W. Zhu, B. Liao, X. Chen, and L. Cai, "A modified PBI approach for multi-objective optimization with complex Pareto fronts," *Swarm and Evolutionary Computation*, vol. 40, pp. 216–237, 2018.

[72] X. Guo, X. Wang, and Z. Wei, "MOEA/D with adaptive weight vector design," in *International Conference on Computational Intelligence and Security*, 2015, pp. 291–294.

[73] N. Hamada, Y. Nagata, S. Kobayashi, and I. Ono, "On scalability of adaptive weighted aggregation for multiobjective function optimization," in *IEEE Congress on Evolutionary Computation*, 2011, pp. 669–678.

[74] T. Shioda and I. Ono, "Adaptive weighted aggregation with step size control weight adaptation for multiobjective continuous function optimization," *SICE Journal of Control, Measurement, and System Integration*, vol. 8, no. 5, pp. 303–311, 2015.

[75] H. Li and D. Landa-Silva, "Evolutionary multi-objective simulated annealing with adaptive and competitive search direction," in *IEEE Conference on Evolutionary Computation*, 2008, pp. 3311–3318.

[76] L. R. C. de Farias, P. H. M. Braga, H. F. Bassani, and A. F. R. Arajo, "MOEA/D with uniformly randomly adaptive weights," in *Genetic and Evolutionary Computation Conference*, 2018, pp. 641–648.

[77] Y. Lin, H. Liu, and Q. Jiang, "Dynamic reference vectors and biased crossover use for inverse model based evolutionary multi-objective optimization with irregular pareto fronts," *Applied Intelligence*, vol. 48, no. 9, pp. 3116–3142, 2018.

[78] J. Li, J. Chen, B. Xin, L. Dou, and Z. Peng, "Solving the uncertain multi-objective multi-stage weapon target assignment problem via MOEA/D-AWA," in *IEEE Congress on Evolutionary Computation*, 2016, pp. 4934–4941.

[79] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," in *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 2014, pp. 170–177.

[80] F. Gu, Y.-m. Cheung, H.-L. Liu, and Z. Lin, "A parameterless decomposition-based evolutionary multi-objective algorithm," in *International Conference on Advanced Computational Intelligence*, 2018, pp. 842–845.

[81] C. Zhang, K. C. Tan, L. H. Lee, and L. Gao, "Adjust weight vectors in MOEA/D for bi-objective optimization problems with discontinuous pareto fronts," *Soft Computing*, vol. 22, no. 12, pp. 3997–4012, 2018.

[82] K. Harada, S. Hiwa, and T. Hiroyasu, "Adaptive weight vector assignment method for MOEA/D," in *IEEE Symposium Series on Computational Intelligence*, 2017, pp. 1–9.

[83] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. L. Philip Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 572–586, 2019.

[84] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 517–526, 2019.

[85] C. A. Coello Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," in *IEEE Congress on Evolutionary Computation*, 2000, pp. 30–37.

[86] J. Branke, "Consideration of partial user preferences in evolutionary multiobjective optimization," in *Multiobjective Optimization-Interactive and Evolutionary Approaches*, 2008, pp. 157–178.

[87] K. Miettinen, F. Ruiz, and A. Wierzbicki, "Introduction to multiobjective optimization: Interactive approaches," in *Multiobjective Optimization- Interactive and Evolutionary Approaches*, 2008, pp. 27–57.

[88] K. Li, K. Deb, and X. Yao, "R-metric: Evaluating the performance of preference-based evolutionary multi-objective optimization using reference points," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 821–835, 2018.

[89] ——, "Integration of preferences in decomposition multi-objective optimization," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2019.

[90] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *IEEE Congress on Evolutionary Computation*, 2014, pp. 421–428.

[91] H. Zhu, Z. He, and Y. Jia, "An improved reference point based multi-objective optimization by decomposition," *International Journal of Machine Learning & Cybernetics*, vol. 7, no. 4, pp. 581–595, 2016.

[92] R. Wang, R. C. Purshouse, and P. J. Fleming, "On finding well-spread Pareto optimal solutions by preference-inspired co-evolutionary algorithm," in *Genetic and Evolutionary Computation Conference*, 2013, pp. 695–702.

[93] M. Pilat and R. Neruda, "Incorporating user preferences in moea/d through the coevolution of weights," in *Genetic and Evolutionary Computation Conference*, 2015, pp. 727–734.

[94] M. K. Tomczyk and M. Kadzinski, "Decomposition-based interactive evolutionary algorithm for multiple objective optimization," *IEEE Transactions on Evolutionary Computation*, 2019, doi: 10.1109/TEVC.2019.2915767.

[95] I. R. Meneghini and F. G. Guimares, "Evolutionary method for weight vector generation in multi-objective evolutionary algorithms based on decomposition and aggregation," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 1900–1907.

[96] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, and Y. Zhang, "MOEA/D with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem," *Soft Computing*, vol. 20, no. 12, pp. 4999–5023, 2016.

[97] R. Liu, R. Wang, W. Feng, J. Huang, and L. Jiao, "Interactive reference region based multi-objective evolutionary algorithm through decomposition," *IEEE Access*, vol. 4, pp. 7331–7346, 2016.

[98] G. Yu, J. Zheng, R. Shen, and M. Li, "Decomposing the user-preference in multiobjective optimization," *Soft Computing*, vol. 20, no. 10, pp.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2020.2978158, IEEE Transactions on Evolutionary Computation

4005–4021, 2016.

[99] C. Peng, H.-L. Liu, and F. Gu, "A novel constraint-handling technique based on dynamic weights for constrained optimization problems," *Soft Computing*, vol. 22, pp. 3919–3935, 2018.

[100] S. P. Phelps and M. Koksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Management Science*, vol. 49, no. 12, pp. 1726–1738, 2003.

[101] A. Jaszkiewicz, "Interactive multi-objective optimization with pareto memetic algorithm," in *the 4th EU/ME Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics*, 2004, pp. 15–20.

[102] K. Li, R. Chen, D. Savic, and X. Yao, "Interactive decomposition multiobjective optimization via progressively learned value functions," *IEEE Transactions on fuzzy systems*, vol. 27, no. 5, pp. 849–860, 2019.

[103] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang, "Interactive MOEA/D for multi-objective decision making," in *Genetic and Evolutionary Computation Conference*, 2011, pp. 721–728.

[104] R. Cheng, M. Olhofer, and Y. Jin, "Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 939–946.

[105] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Genetic and Evolutionary Computation Conference*, 2006, pp. 1141–1148.

[106] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[107] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *International Conference on Evolutionary Multiobjective Optimization*, 2005, pp. 105–145.

[108] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transaction on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[109] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[110] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *IEEE Congress on Evolutionary Computation*, 2009, pp. 203–208.

[111] C. K. Chow and S. Y. Yuen, "A multiobjective evolutionary algorithm that diversifies population by its density," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 149–172, 2012.

[112] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts." *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.

[113] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[114] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Air Force Institute of Technology, Wright Patterson AFB, OH, USA, 1999.

[115] H. Li, Q. Zhang, and J. Deng, "Multiobjective test problems with complicated pareto fronts: Difficulties in degeneracy," in *IEEE Congress on Evolutionary Computation*, 2014, pp. 2156–2163.

[116] H. Lu, M. Zhang, Z. Fei, and K. Mao, "Multi-objective energy consumption scheduling based on decomposition algorithm with the non-uniform weight vector," *Applied Soft Computing*, vol. 39, pp. 223–239, 2016.

[117] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao, "Multiline distance minimization: A visualized many-objective test problem suite," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 61–78, 2018.

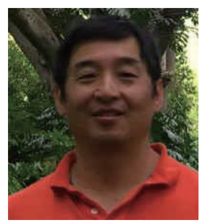
[118] M. Khalidji, M. Zeiaee, A. Taei, M. R. Jahed-Motlagh, and H. Khaloozadeh, "Dynamically weighted continuous ant colony optimization for bi-objective portfolio selection using value-at-risk," in *Asia International Conference on Modelling & Simulation*, 2009, pp. 1–5.

[119] J. R. Figueira, A. Liefooghe, E. Talbi, and A. Wierzbicki, "A parallel multiple reference point approach for multi-objective optimization," *European Journal of Operational Research*, vol. 205, no. 2, pp. 390–400, 2010.

[120] I. Y. Kim and O. d. Weck, "Adaptive weighted sum method for multiobjective optimization," *Structural & Multidisciplinary Optimization*, vol. 31, no. 2, pp. 105–116, 2006.

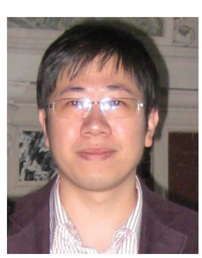
Xiaoliang Ma received the B.S. degree in computing computer science and technology from Zhejiang Normal University, China in 2006, and the PH.D. degree at the school of computing of Xidian University, China in 2014. He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include evolutionary computation, multiobjective optimization, and cooperative coevolution.

Yanan Yu received the B.S. degree in computing computer science and technology from Chengdu University of Information Technology, Chengdu, China, in 2018. She is currently pursuing her M.S. degree in the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include evolutionary computation, multiobjective optimization, and multitask optimization.



Xiaodong Li (M'03-SM'07-F'20) received his B.Sc. degree from Xidian University, Xi'an, China, and Ph.D. degree in information science from University of Otago, Dunedin, New Zealand, respectively. He is a Professor with the School of Science (Computer Science and Software Engineering), RMIT University, Melbourne, Australia. His research interests include machine learning, evolutionary computation, neural networks, data analytics, multiobjective optimization, multimodal optimization, and swarm intelligence. He serves as an Associate Editor of the IEEE Transactions on Evolutionary Computation and Swarm Intelligence (Springer). He is a vice-chair of IEEE Task Force on Multi-modal Optimization and a former chair of IEEE CIS Task Force on Large Scale Global Optimization. He is the recipient of 2013 ACM SIGEVO Impact Award and 2017 IEEE CIS "IEEE Transactions on Evolutionary Computation Outstanding Paper Award". He is an IEEE Fellow.

Yutao Qi was born in 1981, in China. He received the B.Sc. degree in software engineering, the M.Sc. degree in computer science and technology and the PH.D degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2003, 2006 and 2008, respectively. His main research interests include multi-objective optimization, evolutionary computation, machine learning and big data.



Zexuan Zhu (M'12) received the B.S. degree in computer science and technology from Fudan University, China, in 2003 and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include computational intelligence, machine learning, and bioinformatics. Dr. Zhu is an Associate Editor of IEEE Transactions on Evolutionary Computation and IEEE Transactions on Emerging Topics in Computational Intelligence. He is also the Chair of the IEEE CIS Emergent Technologies Task Force on Memetic Computing.