A Self-guided Reference Vector Strategy for Many-objective Optimization

Songbai Liu, Qiuzhen Lin, Ka-Chun Wong, Carlos A. Coello Coello, *Fellow, IEEE*, Jianqiang Li, Zhong Ming, and Jun Zhang, *Fellow*, *IEEE*

***Abstract*—Generally, decomposition-based evolutionary algorithms in many-objective optimization (MaOEA/Ds) have widely used reference vectors to provide search directions and maintain diversity. However, their performance is highly affected by the matching degree on the shapes of the reference vectors and the Pareto front. To address this problem, this paper proposes a self-guided reference vector (SRV) strategy for MaOEA/Ds, aiming to extract reference vectors from the population using a modified *k*-means clustering method. To give a promising clustering result, an angle-based density measurement strategy is used to initialize the centroids, which are then adjusted to get the final clusters, aiming to properly reflect the population's distribution. Afterwards, these centroids are extracted to get adaptive reference vectors for self-guiding the search process. To verify the effectiveness of this SRV strategy, it is embedded into three well-known MaOEA/Ds that originally use the fixed reference vectors. Moreover, a new strategy of embedding SRV into MaOEA/Ds is discussed when the reference vectors are adjusted at each generation. Simulation results validate the superiority of our SRV strategy, when tackling numerous many-objective optimization problems with regular and irregular Pareto fronts.**

***Index Terms*—Many-objective optimization, Self-guided reference vector, Evolutionary algorithm.**

I. Introduction

MANY real-life applications often involve the optimization problems [1] [2], some of which concern the simultaneous optimization of m (often conflicting) objectives [3] [4], giving rise to the so-called multi-objective optimization problems (MOPs) with m = 2 and 3, or many-objective optimization problems (MaOPs) with $m > 3$, as modeled by

$$\min_{x\in\Omega} F(x) = (f_1(x),..., f_m(x)), \quad (1)$$

where $x = (x_1,\ldots,x_n)$ is an n dimensional decision vector in the decision space Ω, and $F(x)$ defines m objective functions in the objective space. In (1), there usually does not exist a single optimal solution for all the objectives, but a set of trade-off solutions termed Pareto optimal set (PS), with its mapping into the objective space termed Pareto front (PF) [5].

During the past two decades, multi-objective evolutionary algorithms (MOEAs) have become a popular and effective approach for solving MOPs [6]-[10]. However, when tackling MaOPs, there have been four main challenges for the conventional MOEAs [11]-[13]: 1) the inefficiency of the Pareto dominance relationship; 2) the difficulty of diversity maintenance; 3) the inefficiency of variation operators; and 4) the high computational cost required. To address the above challenges, numerous research studies have been made to design many-objective evolutionary algorithms (MaOEAs) [14]-[29]. Especially, decomposition-based MOEAs (MOEA/Ds) have shown a great potential [21]-[29], by using reference vectors (RVs) to provide search directions and maintain diversity. In traditional MOEA/Ds [7]-[8], a MOP is decomposed into a set of subproblems, each of which will be associated with an individual using a collaborative search [30]. However, in most MOEA/Ds for solving MaOPs (MaOEA/Ds), a more generalized use of RVs for decomposition of MaOPs is presented, in which the association relation of individuals to subproblems is not so strict, i.e., each subproblem may be associated to none, one or several individuals, and vice versa [21]-[29], [31].

Generally, two main procedures for environmental selection are adopted in most MaOEA/Ds [21]-[29], which will reserve the promising individuals based on their RVs. Assuming that we have N RVs, a candidate union population (marked by P_c) with no less than N individuals will be associated to RVs (N is also the population size). The first procedure (an association-based partition process, APP) will assign each individual of P_c to the closest RV based on a distance metric, aiming to divide P_c into N subsets in objective space. Please note that there are three cases (none, one or several individuals) in each subset. Then, the population's diversity can be maintained by selecting individuals from different subsets, and the search direction of each individual will be guided by its associated RV. Based on the above APP result, the second procedure (an effective elitism selection strategy, ESS) will select N individuals from P_c as the next population, aiming to balance convergence and diversity for tackling MaOPs. Generally, most ESSs are realized by considering convergence and diversity simultaneously [26]-[27] or separately [28]-[29]. In Section II.A, more details of the two above procedures in some existing MaOEA/Ds will be discussed.

As revealed by the studies in [32]-[33], most MaOEA/Ds can properly maintain diversity on regular MaOPs such as the DTLZ [34] and WFG [35] test problems, as their adopted APP

This work was supported by the National Natural Science Foundation of China under Grants 61876110, 61836005, 61672358, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, the Natural Science Foundation of Guangdong Province under Grant 2017A030313338, and the Fundamental Research Project in the Science and Technology Plan of Shenzhen under Grant JCYJ20170817102218122. The work of C. A. Coello Coello was supported in part by CONACyT Project 1920 (Fronteras de la Ciencia) and in part by SEP-Cinvestav 2018 Project (application no. 4). (Corresponding authors: Qiuzhen Lin)

S.B. Liu, Q.Z. Lin, J.Q. Li, and Z. Ming are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (email of Qiuzhen Lin: qiuzhlin@szu.edu.cn).

S.B. Liu and K.-C. Wong are with the Department of Computer Science, City University of Hong Kong, Hong Kong.

C.A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), México, D.F. 07300, MÉXICO.

J. Zhang is with Victoria University, Melbourne, VIC 8001, Australia.

and ESS can perform well to track their PFs. However, they may confront great challenges when tackling irregular and complex MaOPs such as the MaF [36] and mDTLZ [37] test problems, as their specified RVs cannot properly match the PF shapes of such problems. This observation is also pointed out in [38]-[39] in which the authors provide evidence showing that the performance of MaOEA/Ds will be highly affected by the matching degree on the shapes of RVs and PFs. To alleviate this problem, some recent MaOEAs have been designed to adaptively adjust their used RVs [40]-[51]. A-NSGA-III [40] proposes the use of addition and deletion strategies, by removing the non-useful RVs (i.e., they are not associated with any individual) and adding new RVs near to the crowded RVs (i.e., they are associated with a larger number of individuals). When generating new RVs, a (m-1) dimensional simplex of m points is inserted by treating the crowded RV as the centroid (here m is the number of objectives). At last, each individual will be associated to only a unique RV. To further enhance the adaptation capability of A-NSGA-III, A^2-NSGA-III [41] uses a more appropriate simplex size and runs the addition strategy only on the RV that has been found to be crowded during the past τ generations (τ is set to 10 in [41]). Similarly, RVEA* [27] also replaces the non-useful RVs, but its addition strategy is much simpler by randomly generating some new unit RVs based on the minimum and maximum values of each objective. A-IM-MOEA [42] designs two replacement strategies for RVs, by dividing the evolutionary process into the exploration and the exploitation phases. A new RV is randomly generated to renew the most crowded RV (i.e., it is associated with the largest number of individuals) in the exploration phase and to update the least crowded RV (i.e., it is associated with the minimal number of individuals) in the exploitation phase. In g-DBEA [43], two sets of RVs are constructed, namely the active and inactive sets. At first, the active set will reserve N evenly distributed RVs. Once a RV is not associated with any individual over a period of τ generations (τ is a preset integer), it will be moved to the inactive set and then a new RV will be generated around the most crowded RVs in the active set. At the same time, RVs in the inactive set can be moved back to the active set once they are associated with individuals again. Moreover, RVs can be adaptively adjusted by searching a number of well-distributed local ideal points from the current non-dominated individuals in RPEA [44], by co-evolving the preferences and the candidate individuals in PICEA-g [45], by the guidance of an enhanced inverted generational distance indicator in AR-MOEA [46], by employing a self-organizing map in MOEA/D-SOM [47], by using an incremental support vector machine in CLIA [48], by running a linear interpolation of the non-dominated solutions in W-MOEA/D [49] and an equidistant interpolation in its variant [50], and by analyzing the geometric relationship of RVs in MOEA/D-AWA [51]. In MaOEA/D-2ADV [52], m boundary RVs are first generated to achieve fast convergence and then extended to get another N-m RVs by using a systematic method [17], [53]. In the evolutionary process, all the non-useful RVs are removed and new RVs are inserted between two RVs with the maximal distance to their nearest RVs. In MOEA/D-AM2M [54], the individuals are treated as RVs, which are found by iteratively searching the individual with the maximal angle to the selected RVs.

To summarize, MaOEA/Ds show promising performance by using fixed RVs in handling MaOPs with regular PFs and by adaptively tuning RVs to balance convergence and diversity in tackling MaOPs with irregular PFs [48]. Different from the existing adjustment methods of RVs in most MaOEA/Ds [40]-[45], which may not track the evolutionary process as quickly as needed to adjust the RVs according to the individuals' distribution, this paper proposes a self-guided RV (SRV) strategy, aiming to extract RVs from the population to guide its evolutionary process. Our SRV strategy is a generalized production method for RVs and can be embedded into any MaOEA/D for further enhancing their performance, and can perform well on various MaOPs with regular or irregular PFs. Here, the three main contributions of this paper are clarified as follows:

1) We propose the SRV strategy for MaOEA/Ds, by using a k-means clustering method modified from [55] to extract N RVs from the population. In order to improve the clustering result, an angle-based density measurement (ADM) strategy is designed for initialization, which obtains N original centroids to better represent the population's distribution.

2) We integrate the SRV strategy into three well-known MaOEA/Ds, i.e., NSGA-III [21], θ-DEA [23] and EFR-RR [22], forming three new MaOEA/Ds named as NSGA-III/S, θ-DEA/S, and EFR-RR/S, respectively. Their performance on some MaOPs with irregular and complex PFs (i.e., the MaF [36] and mDTLZ [37] problems) is highly improved, which confirms the effectiveness of our SRV strategy.

3) We present a new way to embed SRV into MaOEA/Ds. To avoid the poor convergence induced by frequently changed RVs at each generation, N RVs extracted by our SRV are only used in APP to maintain diversity, while a shared and fixed RV is employed in ESS to balance convergence and diversity by joining the ideal and nadir points. This way, a novel competitive MaOEA named MaOEA/SRV is presented in this paper.

The rest of this paper is organized as follows. Section II introduces the previous related work on MaOEA/Ds and the motivations to design SRV. Section III presents the details of SRV and the way of embedding SRV into some well-known MaOEAs. The experimental results and some discussions are provided in Section IV, which motivated us to present MaOEA/SRV. At last, our conclusions and future work are presented in Section V.

II. Related Work and Motivations

A. Summary of Some Existing MaOEA/Ds

MaOEA/Ds run the environmental selection, as guided by N RVs, which are represented by $r^1, r^2, ..., r^N$ (N is the number of RVs and population size). As introduced in Section I, there are two procedures (APP and ESS) in this operator. An APP is used to divide the candidate union population P_c into N subsets and then an ESS is adopted to select N promising individuals from P_c as the next population. Generally, in most MaOEA/Ds with fixed RVs, N RVs are uniformly distributed on the unit

TABLE I
A SUMMARY OF SOME EXISTING MAOEA/DS

MaOEA/Ds	D-Metrics	P_c	Normalization in P_c	I_c (Convergence)	I_d (Diversity)	The way to run ESS
NSGA-III [21]	$d_2(x,r^i)$ in (4)	P_c=S_t	By (2), estimate z^{nadir} with an adaptive procedure on all $x \in P_c$	Non-dominated fronts	$d_2(x,r^i)$ in (4)	Processing I_c and I_d separately
EFR-RR [22]	$d_2(x,r^i)$ in (4)	P_c=U	By (2), estimate z^{nadir} with an adaptive procedure on all $x \in F_1$	Modified Tchebycheff function	$d_2(x,r^i)$ in (4)	Processing I_c and I_d separately
θ -DEA [23]	$d_2(x,r^i)$ in (4)	P_c=S_t	By (2), estimate z^{nadir} with an adaptive procedure on all $x \in F_1$	$d_1(x,r^i)$ in (5)	$d_2(x,r^i)$ in (4)	Processing I_c and I_d simultaneously
RPD-NSGA-II [24]	$d_2(x,r^i)$ in (4)	P_c=U	By (2), set z^{nadir} as the maximal values of all objectives from all $x \in P_c$	$d_1(x,r^i)$ in (5)	RV-density	Processing I_c and I_d separately
MOEA/D-LWS [25]	$\theta(x,r^i)$ in (6)	P_c=U	By (2), set z^{nadir} as the maximal values of all objectives from all $x \in F_1$	Localized weighted sum function	$\theta(x,r^i)$ in (6)	Processing I_c and I_d separately
SPEA/R [26]	$\theta(x,r^i)$ in (6)	P_c=U	By (2), set z^{nadir} as the maximal values of all objectives from all $x \in P_c$	Local strength value	Local density	Processing I_c and I_d simultaneously
RVEA [27]	$\theta(x,r^i)$ in (6)	P_c=U	By (3), translate each objective to the coordinate origin	$EdI(x)$ by (7)	$\theta(x,r^i)$ in (6)	Processing I_c and I_d simultaneously
MOEA/D-SAS [28]	$\theta(x,r^i)$ in (6)	P_c=U	No normalization	Modified Tchebycheff function	$\theta(x,r^i)$ in (8)	Processing I_c and I_d separately
ASEA [29]	$\theta(x,r^i)$ in (6)	P_c=U	By (2), estimate z^{nadir} with an adaptive procedure on all $x \in F_1$	$EdI(x)$ by (7)	$\theta(x,r^i)$ in (6)	Processing I_c and I_d separately

hyperplane or unit hypersphere by a systematic method [17], [53]. Moreover, to match the used RVs, both APP and ESS are processed in a normalized objective space. In order to interpret the differences of various MaOEA/Ds, a summary of some existing MaOEA/Ds is provided in Table I, which lists the adopted distance metrics (D-metrics) in APP, the selected candidate population P_c, the normalization procedures, two indicators in ESS (i.e., I_c and I_d that are respectively used to reflect convergence and diversity of each individual), and the way to run ESS. The details of these components are respectively introduced below.

1) Candidate population P_c: in the traditional fast non-dominated sorting procedure [6], the union population U with $2N$ individuals (i.e., N parents plus N offspring) is divided into multiple fronts according to the non-dominated ranks (i.e., F_1, F_2, and so on). Then, a subset S_t can be constructed by including one front each time, starting from F_1, then F_2 and so on, until the size of S_t exceeds N for the first time. Let F_l be the last front included into S_t (i.e., $S_t = F_1 + F_2 + ... + F_l$), and the individuals in the remainder fronts of U are all rejected by S_t. Generally, there are two settings of P_c in the existing MaOEA/Ds from Table I, e.g., P_c=S_t in [21], [23] that give priority to the individuals with good convergence, and P_c=U in [22], [24]-[29] that emphasize the population's distribution.

2) Normalization in P_c: in most MaOEA/Ds, normalization is necessary to tackle MaOPs with different scales on different objectives, which helps to fit the used RVs. Generally, the i-th objective $f_i(x)$ of a solution $x \in P_c$ is normalized as follows:

$$f_i'(x) = \frac{f_i(x) - z_i^*}{z_i^{nadir} - z_i^*}, \quad (2)$$

where z^* and z^{nadir} are respectively the ideal and nadir points, m is the number of objectives and $i = 1,2,...,m$. However, in RVEA [27], $f_i(x)$ is only translated to the coordinate origin by

$$f_i'(x) = f_i(x) - z_i^*, \quad (3)$$

Please note that z^* can be easily determined by setting z_i^* as the minimal value of $f_i(x)$ from all $x \in P_c$, while the estimation of z^{nadir} is very challenging as it requires the true information of PF [56]. There are two common approaches to estimate the nadir point: 1) set z_i^{nadir} as the maximal value of $f_i(x)$ from all $x \in P_c$ [24] or all $x \in F_1$ [25], [29], and 2) set z_i^{nadir} by using the adaptive procedures, e.g., the calculation of the intercepts along each objective axis for all $x \in P_c$ [21] or all $x \in F_1$ [22]-[23]. Note that, for some degenerate cases like no intercepts in certain directions or negative intercepts, z_i^{nadir} is set as the maximal value of $f_i(x)$ from all $x \in P_c$ in this paper [23].

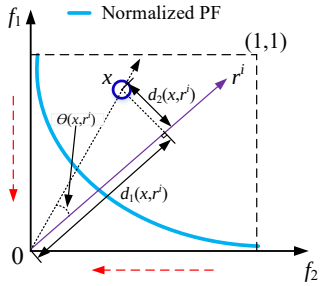

Fig. 1. Illustration of two distance metrics used in APP

3) The distance metrics: when associating a solution $x \in P_c$ to a RV r^i ($i = 1,2,...,N$), there are two distance metrics used in most MaOEA/Ds [21]-[29]. The first metric is the perpendicular distance from $f'(x)$ in (2) to r^i, termed $d_2(x,r^i)$, which is computed by

$$d_2(x,r^i) = \sqrt{\left\| f'(x) - z^* \right\|^2 - d_1(x,r^i)^2}, \quad (4)$$

where z^* is an ideal point and $d_1(x,r^i)$ is the distance from the projection of $f'(x)$ on r^i to the origin point, as calculated by

$$d_1(x,r^i) = \frac{(f'(x) - z^*) \cdot (r^i - z^*)}{\|r^i - z^*\|}, \quad (5)$$

The second metric termed $\theta(x,r^i)$ is a cosine distance to indicate the acute angle from $f'(x)$ to r^i, as computed by

$$\theta(x,r^i) \triangleq \arccos\left| \frac{(f'(x) - z^*) \cdot (r^i - z^*)}{\left\| f'(x) - z^* \right\| \times \left\| r^i - z^* \right\|} \right|, \quad (6)$$

To clarify the meaning of $d_1(x,r^i)$, $d_2(x,r^i)$, and $\theta(x,r^i)$, a simple example is provided in Fig. 1 for the 2-dimensional normalized objective space.

4) The elitism selection strategy (ESS): after running APP, each individual $x \in P_c$ is associated to its closest RV and P_c is divided into N subsets $S_1^R, S_2^R, ..., S_N^R$, where S_i^R includes all the individuals associated to the i-th RV r^i. For example, each $x \in P_c$ can be associated to its K closest RVs in EFR-RR [22], while each RV in MOEA/D-SAS [28] is associated with its L closest individuals (here K and L are the integers preset by the users). ESS endeavors to select N promising individuals based on the subsets $S_1^R, S_2^R, ..., S_N^R$ from APP. In each S_i^R associated to r^i ($i = 1,2,...,N$), two indicators I_c and I_d are designed to

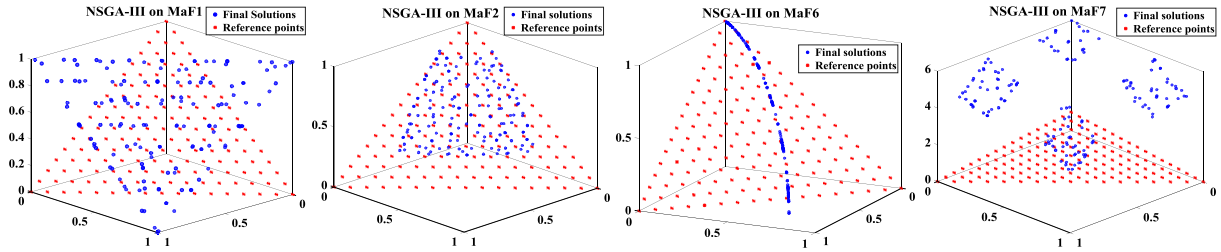

Fig. 2 The poor distributions of final solutions obtained by NSGA-III on MaF1 with an inverted PF, MaF2 with a partial concave PF, MaF6 with a degenerated PF and MaF7 with a disconnected PF, mainly due to the mismatch of the adopted RVs and the above PFs.

respectively reflect the convergence and diversity information for each individual x ($x \in S_i^R$). Then, the ranking and selection of individuals in S_i^R are run based on their I_c and I_d values. Based on the processing principle, ESS can be classified into two categories. The first strategy is to process I_c and I_d separately, which can be found in NSGA-III [21], RPD-NSGA-II [24], MOEA/D-LWS [25], ASEA [29], MOEA/D-SAS [28] and EFR-RR [22]. The second strategy is to process I_c and I_d simultaneously, which is used in θ-DEA [23], RVEA [27] and SPEA/R [26]. In Table I, the I_c based on the Euclidean distance from an individual x to the ideal point z^*, termed $EdI(x)$, is computed by

$$EdI(x)=\sqrt{\sum_{k=1,2,\ldots,m}(f_k'(x)-z_k^*)\times(f_k'(x)-z_k^*)}, \qquad (7)$$

where $f_k'(x)$ is the normalized objective value in (2). The I_d represents the angle of two individuals (x, y), as computed by

$$\theta(x,y) \triangleq \arccos\left|\frac{f'(x)\cdot f'(y)}{\|f'(x)\|\times\|f'(y)\|}\right|, \qquad (8)$$

More details of executing APP and ESS in these MaOEA/Ds can be found in Section 2 of the supplementary file.

B. Our Motivations

As summarized in Table I, MaOEA/Ds design the specific APPs and ESSs in their environmental selection, which can settle regular MaOPs well, but faces enormous challenges when dealing with irregular MaOPs. Their APPs often use uniformly distributed RVs, which also determine I_c and I_d in their ESSs. Here, the final solutions obtained by NSGA-III in a typical run are plotted in Fig. 2 when solving the 3-objective inverted MaF1, partial concave MaF2, degenerated MaF6 and disconnected MaF7 [36], showing poor distributions mainly due to the mismatch of the used RVs and the target problems' PF. Thus, an appropriate set of RVs plays an essential role in their performance when solving these irregular MaOPs. Many research studies [40]-[51] have pointed out that the adaptive adjustment of RVs is an effective way for MaOEA/Ds to tackle irregular MaOPs. However, most of these adjustment strategies introduced in **Section I** have been designed to remove the non-useful vectors and then to generate new RVs around the original RVs, which may also face some challenges, e.g., the difficulty of identifying the timing and frequency of RV adaptation [27] [45], the hypersensitivity of additional parameters [42] [51], the high computational cost required [47]-[48] and the limited extension to solve MaOPs with complex PFs [45] or constrained PFs [41] [43].

Recently, a number of MaOEAs have been proposed to use acute angles of individuals in their environmental selection for tackling MaOPs, which actually treat individuals as RVs, e.g., VaEA [57], MaOEA-DDFC [58], 1by1EA [59], MaOEA-CSS [60], and DDEA [61]. Naturally, instead of setting fixed RVs in MaOEA/Ds, the individuals selected from P_c, if properly manipulated, can be treated as RVs [54]. However, as an RV should well represent its associated region in the objective space, the individuals in P_c are insufficient to reflect the entire distribution, especially for P_c with a poor quality in terms of distribution at the early stage of the evolutionary process. Similar to *k*-means, it is also insufficient and ineffective to complete the entire clustering process by only getting *k* initial points as centroids [62], whereas the running of the self-adjustment process is necessary and effective to get a better clustering result.

To sum up, when the distribution of the current population is detected with a lower matching degree to the fixed RVs, a set of self-guided RVs extracted from the population by a clustering method may be more helpful in guiding the evolutionary process, even when the quality of the population is poor for clustering. Motivated by the above discussions, this paper proposes a SRV strategy, trying to extract *N* RVs from P_c using a modified *k*-means clustering method. In order to improve the clustering result, an ADM strategy is designed to initialize centroids, so that these initial centroids can properly reflect the distribution of P_c. Then, the centroids are self-adjusted to classify P_c into *N* clusters. Finally, a suitable set of *N* RVs can be extracted from the *N* centroids. Therefore, there is no need to use systematic methods to generate RVs [17] [53] or to adjust RVs during the evolutionary process [40]-[46], but RVs can be extracted directly from the population to quickly track and guide the evolutionary process. In our previous work (MaOEA/C [63]), a two-step clustering strategy was used to divide the union population into *N* clusters. However, its used hierarchical clustering method may induce high computational cost when handling extremely imbalanced population and its final cluster results are still not good enough to extract good RVs according to the experiments. Therefore, this paper modifies the state-of-the-art *k*-means method [55] to classify the population as *N* clusters, which can be more appropriate to extract good RVs as introduced in the next section.

III. The Proposed SRV Strategy

In this section, the proposed SRV strategy is introduced. Without using a systematic method, RVs are extracted from the population in SRV to self-guide the evolutionary process. To achieve this purpose, the population is classified into a number of clusters using a clustering method and then an equal number of RVs is obtained by extracting the centroids of these

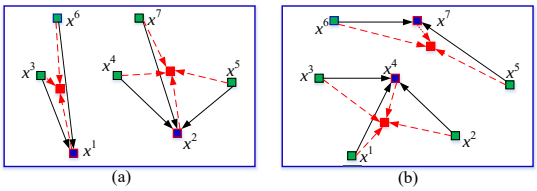

Fig. 3 The influence of different initial centroids in *k*-means.

clusters. Thus, a candidate union population P_c for clustering and an effective clustering method should be needed. Generally, N RVs are used in MaOEA/Ds (N is the population size), i.e., N clusters are obtained from P_c, which should contain at least N individuals. Here, we set $P_c = S_t$ as introduced in **Section II. A,** where the outlier solutions may have been excluded to improve the quality of the data for clustering. A classic *k*-means method is modified by setting $k = N$ due to its popularity and easy implementation when compared to other clustering methods like spectral clustering [64] and agglomerative clustering [65], which includes two main procedures (an ADM-based initialization of centroids and the self-adjustment for centroids). Note that, k=N is a sufficiently large number of clusters when compared to the number of solutions in P_c, which can alleviate the impact by the quality of the clustering data [55]. To illustrate how it works, the general framework of SRV is provided in **Algorithm 1** with the inputs: P_c (a candidate population), N (the population size), m (the number of objectives) and θ_c (a parameter used in ADM). In line 1, N initial centroids $(c^1, c^2, ..., c^N)$ are obtained by running the ADM-based initialization in **Algorithm 2**. In line 2, the centroids c= $(c^1, c^2, ..., c^N)$ are updated by the self-adjustment method in **Algorithm 3**. At last, in lines 3-5, N RVs $(rv^1, rv^2, ..., rv^N)$ are extracted by mapping each c^i on the unit hypersphere ($i = 1, 2, ..., N$).

To clarify the implementation of SRV and its use, the details of the ADM-based initialization of centroids (**Algorithm 2**) and the self-adjustment method for centroids (**Algorithm 3**) are introduced in Section II.A and Section II.B, respectively. At last, the embedding of the SRV strategy into MaOEA/Ds is described in Section III. C.

A. ADM-based Initialization of Centroids

In *k*-means, two important factors (the value of k and the selection of initial centroids) will highly affect the final clustering result [62]. As k is set to N in SRV, the initialization of centroids becomes important. For example, as shown in Fig. 3 with k=2, there are seven individuals $x^1, x^2, ..., x^7$. One case is shown in Fig. 3(a) in which two initial centroids are selected as x^1 and x^2, and the clustering result will produce one cluster with three individuals (x^1, x^3, x^6) and another cluster with four individuals (x^2, x^4, x^5, x^7), with their centroids marked in red color. For another case in Fig. 3(b) with two initial centroids x^4 and x^7, two clusters (x^1, x^2, x^3, x^4) and (x^5, x^6, x^7) can be obtained in the same manner. Obviously, the selection of initial centroids has a significant impact on the final clustering result. Therefore, to enhance the clustering result, an ADM strategy is designed for initializing centroids in SRV.

As pointed out in [66], centroids should be selected based on two assumptions: 1) they are surrounded by neighbors with lower local densities and 2) they have a relatively large distance from any point with a higher local density. Following the two assumptions, our ADM strategy endeavors to find out N initial centroids. Two quantities for each individual $x \in P_c$, i.e., the local density $\rho(x)$ (i.e., an angle-based crowding degree of x) and the separation distance $\delta(x)$ (i.e., the angle between x to the individual with a higher local density), are considered in ADM. Thus, each individual $x \in P_c$ is firstly normalized by (2), and then $\rho(x)$ is computed by (9), where $\theta_c \in (0, \pi/2]$ is a cutoff angle value used to control the local density.

$$\begin{cases} \rho(x) = \sum_{y \in P_c} e^{-\left(\frac{\theta(y,x)}{\theta_c}\right)^2} \\ s.t. \quad y \in P_c, y \neq x \text{ and } \theta(y,x) \leq \theta_c \end{cases}, \tag{9}$$

Another quantity $\delta(x)$ is considered by computing the minimum angle from x to any other solution y in P_c with a higher local density, as calculated by

$$\delta(x) = \min_{y \in P_c: \rho(y) > \rho(x)} \left(\theta(x,y)\right), \tag{10}$$

Specially, for an individual $x \in P_c$ with the highest local density, its separation distance is set as $\delta(x)$=$\pi/2$. In [66], the two above quantities are considered simultaneously to avoid interference from noise data. However, in our ADM strategy, only the separation distance is considered as all individuals in P_c have to be involved in the evolutionary process.

To clarify this initialization process, **Algorithm 2** is provided to show the process of ADM-based initialization in SRV, with the inputs: P_c, N, m and θ_c (a cutoff angle). In line 1, all individuals in P_c are normalized by (2) in order to provide individuals with normalized objectives. Then, the local density $\rho(x)$ and separation distance $\delta(x)$ of each individual $x \in P_c$ are computed in line 2 and line 3, respectively. In line 4, for each $x \in P_c$, a flag ($flag(x)$) indicating whether it is a centroid is initialized to be **false**. In order to reserve boundary information, m extreme individuals $x^{e(1)}, x^{e(2)}, ..., x^{e(m)}$ are firstly selected as m initial centroids ($c^1, c^2, ..., c^m$) in lines 5-8, by resetting their flags as **true** in line 7. Here, the individual with the minimal

Algorithm 1 *SRV* (P_c, N, m, θ_c)

1: $(c^1, c^2, ..., c^N)$ = *Initialization_ADM* (P_c, N, m, θ_c)
2: $(c^1, c^2, ..., c^N)$ = *Self-guided_Adjustment* (P_c, N, m, c)
3: **for** i := 1 to N
4: get rv^i by mapping c^i on the unit hypersphere
5: **end for**
6: **return** $(rv^1, rv^2, ..., rv^N)$

Algorithm 2 *Initialization_ADM* (P_c, N, m, θ_c)

1: normalize all individuals of P_c by (2)
2: compute $\rho(x)$ of each individual $x \in P_c$ by (9) with θ_c
3: compute $\delta(x)$ of each individual $x \in P_c$ with (10)
4: set $flag(x)$ = **false** for each $x \in P_c$
5: **for** i := 1 to m
6: find the nearest individual $x^{e(i)}$ to the ith axis e^i by (11)
7: set $c^i = x^{e(i)}$ and $flag(x^{e(i)})$ = **true**
8: **end for**
9: sort all individuals with a descending order by $\delta(x)$
10: set j = 1 and i = m+1
11: **while** $i \leq N$
12: **if** $flag(x^j)$ = **false**
13: set c^i=x^j and i++
14: **end if**
15: j++
16: **end while**
17: **return** $c^1, c^2, ..., c^N$

Algorithm 3 *Self-guided Adjustment* (P_c, N, m, c)

1: N empty clusters $(C^1, C^2, ..., C^N)$ and $cl(i) = 0$ of $x \in P_c$
2: **for** each $x \in P_c$
3: assign x to the centroid c^T, which satisfies:
4: $T = \{j : \arg\min \theta(x, c^j), j = 1, 2, ..., N\}$
5: add x into cluster C^T and **set** $cl(x) = T$
6: **end for**
7: **set** $t = 0$, and *flag* = **false**
8: **while** $t \le 2m$ && *flag*=**false**
9: **set** $C^j = \varnothing$, $j = 1, 2, ..., m$ and **set** $l = 0$
10: **for** i:=m+1 to N
11: adjust centroid c^i based on (13), and set $C^i = \varnothing$
12: **end for**
13: **for** each $x \in P_c$
14: assign x to c^T like lines 3-4 and add it to C^T
15: **if** $cl(x)$!= T
16: **set** $cl(x) = T$ and l++
17: **end if**
18: **end for**
19: **set** *flag*=**true** if l=0
20: t++
21:**end while**
22:**return** $(c^1, c^2, ..., c^N)$

angle to the ith axis vector e^i, $i = 1, 2, ..., m$, is selected as the ith extreme individual $x^{e(i)}$, where $x^{e(i)}$ can be computed by

$$x^{e(i)} = \arg\min_{x \in P_c} \theta(x, e^i), \tag{11}$$

where m axis vectors are (1, 0,…, 0), (0, 1, 0,…, 0),…, (0,.., 0, 1), respectively, and $\theta(x, e^i)$ can be computed by (6). Next, all individuals are sorted on a descending order by only considering their separation distances in line 9. Finally, the rest N-m individuals with the higher separation distances are preferentially selected as the centroids in lines 10-16 and the output of this algorithm is N initial centroids $c^1, c^2, ..., c^N$. Obviously, these initial centroids obtained by the ADM-based initialization can more properly reflect the distribution of P_c.

B. Self-Adjustment method for Centroids

The pseudo-code of the self-adjustment method for centroids in SRV is given in **Algorithm 3** with the inputs: P_c, N, m, and c=$(c^1, c^2, ..., c^N)$ (a set of N initial centroids from **Algorithm 2**). In line 1, N clusters $(C^1, C^2, ..., C^N)$ are initialized as empty sets, and each individual $x \in P_c$ will be assigned a cluster label (termed $cl(x)$ and initialized as 0) at first, which indicates the cluster it belongs to. For example, $cl(x) = j$ means that x belongs to the jth cluster. Thus, in lines 3-4, each individual $x \in P_c$ will find its nearest centroid C^T by considering all the angles $\theta(x, c^j)$ from x to c^j in (8), as follows:

$$T = \{j : \arg\min \theta(x, c^j), j = 1, 2, ..., N\}. \tag{12}$$

Then, x is included into cluster C^T and we set its cluster label $cl(x) = T$ in line 5. Next, in lines 7-21, the self-adjustment method is used to generate N new centroids $(c^1, c^2, ..., c^N)$. Let t count the iteration number of the self-adjustment and let *flag* denote whether all N clusters have reached a stable state (i.e., the cluster label $cl(x)$ keeps unchanged for each individual $x \in P_c$). In line 8, $t = 0$ and *flag* = **false** are set at first. When t is smaller than $2m$ (this condition aims to control the complexity of adjustment) and *flag* = **false** in line 8, the procedures in lines 9-20 will be run iteratively. To maintain boundary information, the first m extreme centroids are kept by directly clearing m clusters $(C^1, ..., C^m)$ in line 9, while the remaining centroids are adjusted in lines 10-12 by (13)-(14), where $i = m+1, ..., N$,

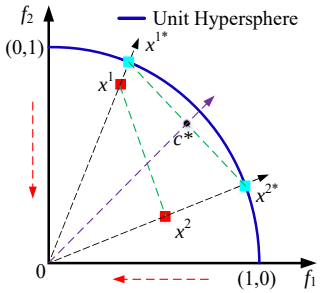

Fig. 4 An example of the centroid update by (13)-(14)

$j = 1, 2, ..., m$, and $f_j^*(x)$ indicates the mapping value of $f_j'(x)$ on the unit hypersphere, and $EdI(x)$ can be calculated in (7).

$$f_j'(c^i) = \frac{\sum_{x \in S^i} f_j^*(x)}{|S^i|}, \tag{13}$$

$$f_j^*(x) = f_j'(x) / EdI(x), \tag{14}$$

Figure 4 shows the centroid update by (13)-(14), in which x^{1*} and x^{2*} respectively indicate the projections of x^1 and x^2 by (14) on the unit hypersphere, and c^* is their centroid by (13). In addition, the counter l, initialized as 0 for each iteration in line 9, is used to mark the number of individuals whose cluster label is changed. Then, cluster C^i is set to be empty after centroid c^i has been updated in line 11. N new clusters $(C^1, C^2, ..., C^N)$ are obtained in lines 13-18 by reassigning each individual $x \in P_c$ to its nearest centroid (line 14), followed by updating its cluster label $cl(x)$ in lines 15-17. If the cluster label $cl(x)$ of an individual $x \in P_c$ is changed to a new label T as determined in lines 3-4, set $cl(x)$=T and the counter l is increased by 1 in line 16. However, if there is no individual $x \in P_c$ whose cluster label is changed (i.e., l is still 0), *flag*=**true** is set in line 19. After finishing the procedures in lines 9-19, t is increased by 1 in line 20 if $l != 0$, and the algorithm will go back to check the termination conditions in line 8. If it is terminated, this algorithm will output N centroids $(c^1, c^2, ..., c^N)$ that can properly represent the distribution of P_c. As introduced in **Algorithm 1** of SRV, N self-guided RVs $(rv^1, rv^2, ..., rv^N)$ are obtained by mapping each centroid c^i onto the unit hypersphere by

$$rv_j^i = f_j'(c^i) / EdI(c^i), \tag{15}$$

where $i = 1, ..., N$ and $j = 1, 2, ..., m$.

C. Embedding SRV into MaOEA/Ds

As discussed in **Section II**, one of the most important factors affecting the performance of MaOEA/Ds is the used RVs. In this section, we show some examples in which we embed SRV into three well-known MaOEA/Ds (NSGA-III [21], θ-DEA [23], and EFR-RR [22]), which will help these MaOEA/Ds to tackle complicated MaOPs with irregular PFs. Without changing other procedures in these MaOEA/Ds, their original N uniformly distributed RVs $(r^1, r^2, ..., r^N)$ are replaced by N RVs $(rv^1, rv^2, ..., rv^N)$ from SRV, forming three new algorithms (i.e., NSGA-III/S, θ-DEA/S, and EFR-RR/S). In addition, the cutoff angle θ_c in SRV is designed as an adaptive quantity, which is increased linearly with the evolutionary process, and its minimum value $\theta_c^{\min}$ is determined by getting the minimum angle between two neighboring RVs among N original RVs $(r^1, r^2, ..., r^N)$, as follows:

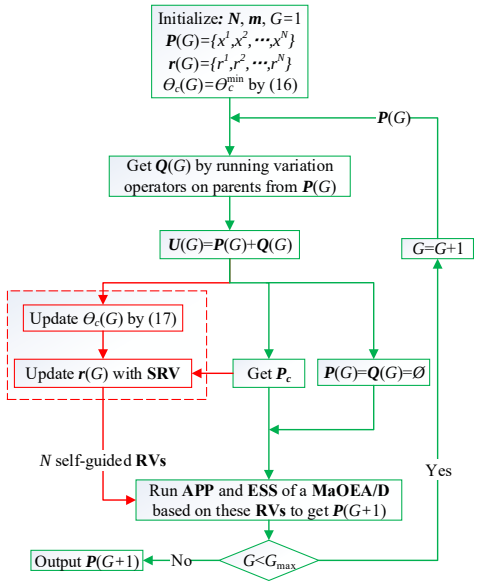

Fig. 5 Process of embedding SRV into MaOEA/RVs

$$\begin{cases} \theta_c^{\min} = \min\limits_{i=1,2,\ldots,N} \theta(r^i, r^u) \\ s.t.\ \ u = \{j : \arg\min\limits_{j=1,2,\ldots,N} \theta(r^i, r^j)\} \end{cases}, \tag{16}$$

where $\theta(r^i, r^j)$ computes the angle of r^i and r^j. The maximum value of θ_c is $\pi/2$, thus θ_c is adaptively set to

$$\theta_c = \frac{(\pi - \theta_c^{\min}) \times (G-1)}{2 \times G_{\max}} + \frac{\theta_c^{\min}}{2}, \tag{17}$$

where G is the generation counter and $G_{\max}$ is a predefined maximum number of generations. At the initial stages of the evolutionary process, a small value of θ_c is used to measure the local densities of individuals in ADM, as the early population shows a poor distribution. At a later stage, the population tends to converge and a nearly global density is considered in ADM by assigning a relatively larger value for θ_c. Thus, the adaptive θ_c can properly control the generation of N initial centroids in **Algorithm 2** and N more suitable adaptive RVs can be obtained in **Algorithm 1**.

To clarify the embedding process of SRV, the whole process of NSGA-III/S, θ-DEA/S and EFR-RR/S is illustrated in Fig. 5, which embeds our SRV strategy (marked in red color) into the original procedures of NSGA-III, θ-DEA and EFR-RR (as identified by the green color). In Fig. 5, with the generation counter (i.e., G), $P(G)$, $Q(G)$ and $r(G)$ represent the parent population, the offspring population and the set of RVs at the G-th generation, respectively; N and m are the population size and number of objectives, respectively. At first, G is initialized as 1, while $P(G)$ and $r(G)$ are initialized by N randomly generated individuals $x^1, x^2, ..., x^N$ and N uniformly distributed RVs $r^1, r^2, ..., r^N$, respectively. Then, $Q(G)$ with N offspring can be obtained by running variation operators (i.e., simulated binary crossover (SBX) and polynomial-based mutation [6]) on $P(G)$, followed by combining $P(G)$ and $Q(G)$ as a union population $U(G)$ (i.e., $U(G)=P(G)+Q(G)$). Then, the candidate population P_c can be obtained by the strategies introduced in **Section II. A** (Here, $P_c = S_t$ in both NSGA-III and θ-DEA, while $P_c = U(G)$ in EFR-RR). Meanwhile, with the initial value of $\theta_c^{\min}/2$ computed by (16), the cutoff angle θ_c used in SRV is updated by (17), while $P(G)$ and $Q(G)$ are reset as empty sets. Then, based on θ_c and P_c, $r(G)$ is updated as N self-guided RVs generated from **Algorithm 1**. Finally, the environmental seletion by running APP and ESS in a MaOEA/D (i.e., NSGA-III, θ-DEA or EFR-RR) can be guided by these N self-guided RVs to produce the next population $P(G+1)$ from P_c. Therefore, without using the SRV strategy as marked by the red color, the environmental selection of NSGA-III, θ-DEA and EFR-RR is always guided by N initial RVs $(r^1, r^2, ..., r^N)$ at each generation. However, in NSGA-III/S, θ-DEA/S and EFR-RR/S, their environmental selection is directed by N self-guided RVs extracted from the population by **Algorithm 1**, which is more able to track the evolutionary process and fit the PF shape. Please note that this embedding process of SRV in Fig. 5 can also be applied to other MaOEA/Ds [24]-[29].

IV. Experimental Studies

A. Benchmark Problems and Performance Indicator

In this study, the MaF1-MaF13 test problems [36] with irregular PFs and the mDTLZ1-mDTLZ4 [37] test problems with hardly-dominated boundaries in PFs were used. For each problem, the number of objectives m was varied from 3 to 10, i.e., $m \in \{3,4,5,7,10\}$. The number of decision variables n was set by $n = m+k-1$ for MaF1-MaF7 (here, k was set to 10 for MaF1-MaF6 and to 20 for MaF7 as suggested in [36]). n was set to 2 for MaF8 and MaF9 and to 5 for MaF13. Especially, the decision variables have k position-related parameters and l distance-related parameters for MaF10-MaF12 (here, k was set to $2\times(m-1)$ and l was set to 20 as recommended in [63]). For mDTLZ1-mDTLZ4, n was set by $n = 2m+k-1$ and k=5 was used in [37]. Due to page limitations, the main characteristics of the MaF and mDTLZ test problems are summarized in Table A.I of the supplementary file.

The hypervolume (HV) and inverted generation distance (IGD) [67] were considered as our performance indicators in this paper. These two indicators assesse both convergence and maximum spread of the final solution set. A larger HV value indicates a better approximation to the true PF, while IGD is the opposite. Here, the recently proposed walking fish group method [68] was used to compute the exact HV values for all test problems, where a reference point z that is dominated by all points of the true PF should be carefully set for MaOPs. To ensure a fair comparison, the z in our experiments was set as suggested in [57], [63], which normalize all the objective values in the final solution sets by using $1.1\times z^{nad}$ (here z^{nad} is the nadir point of the true PF) and we set the z as $(1.0,1.0,...,1.0)$. When computing IGD, a set of reference points uniformly sampled from the true PF was required. Here, 20,000 reference points were sampled for all the test problems. Due to page limitations, please refer to [67] and [68] for details on how to compute HV and IGD.

B. Parameters Settings for the Compared Algorithms

In our experiments, SRV was embedded into three competitive MaOEA/Ds, i.e., NSGA-III [21], EFR-RR [22], θ-DEA [23] for performance verification. Moreover, four competitive MaOEA/Ds adaptively adjusting RVs (MOEA/D-AWA [51],

A-NSGA-III [40], RVEA* [27], and AR-MOEA [46]), one promising MaOEA treating individuals as RVs (VaEA [57]), and one recently proposed MaOEA based on clustering (MaOEA/C [63]) were also included for comparison. The parameters settings of these MaOEAs are provided in Table A. II of the supplementary file, as suggested in their references. All the compared MaOEAs use the same evolutionary operators, i.e., SBX and polynomial-based mutation [21].

The settings of population size for different numbers of objectives are listed in Table A. III of the supplementary file. For test problems with 3, 4, 5, 7 and 10 objectives, the numbers of RVs were respectively set to 153, 165, 210, 238 and 275, using the two-layer generation method with the simplex-lattice design factor H in [27]. According to [63], the population size in MaOEA/C should be set as a multiple of m. Thus, its population sizes were set to 153, 164, 210, 238, and 280, respectively for 3-, 4-, 5-, 7-, and 10-objective test problems. For other compared MaOEAs, their population sizes were set the same with the number of RVs. All the compared MaOEAs[1] were run 30 times independently on each test problem. The mean HV values and their standard deviations (included in brackets after the mean HV results) in 30 runs were collected for comparison. All the compared MaOEAs were terminated when a predefined maximum number of generations $G_{\max}$ was reached. The values of $G_{\max}$ were set to 600, 700, 800, 900, and 1000, respectively for 3-, 4-, 5-, 7-, and 10-objective test problems. Their maximum function evaluations (*MFE*) can be easily obtained by computing $MFE = N \times G_{\max}$.

C. Results of embedding SRV into Three MaOEA/Ds

Here, SRV was embedded into three competitive MaOEA/Ds (NSGA-III [21], EFR-RR [22], θ-DEA [23]), forming three enhanced algorithms (NSGA-III/S, EFR-RR/S and θ-DEA/S). Thus, three comparisons of NSGA-III/S vs NSGA-III, θ-DEA/S vs θ-DEA, and EFR-RR/S vs EFR-RR are made on MaF1-MaF13 and mDTLZ1-mDTLZ14 with 3 , 4, 5, 7 and 10 objectives. Due to page limitations, the detailed HV results of all the compared algorithms are presented in Table A. IV of the supplementary file, on tackling all the adopted test problems. Furthermore, the summary of significance test on HV for each comparison is given in Table II. To ensure a statistically sound conclusion, Wilcoxon rank sum test with a 0.05 significance level and Wilcoxon signed ranks test from the KEEL tool [69] were run, showing statistically significant differences on the HV results. In Table II, the numbers under the columns "+", "–", and "~" indicate the comparison times that the results of the enhanced MaOEA (NSGA-III/S, EFR-RR/S, or θ-DEA/S) are respectively better than, worse than, and similar to the original MaOEA (NSGA-III, EFR-RR or θ-DEA) on tackling each problem with different objectives. Moreover, an asymptotic p-value obtained by Wilcoxon signed ranks test on the KEEL tool is also provided in the column "p-value", where a p-value closer to 0 means that there are more significant differences on the HV results.

TABLE II
SUMMARY OF SIGNIFICANCE TEST BETWEEN THREE MAOEA/RVS AND THEIR ENHANCED VERSIONS WITH SRV ON HV

Comparisons based on		NSGA-III/S vs. NSGA-III				θ-DEA/S vs. θ-DEA				EFR-RR/S vs. EFR-RR			
		+	–	~	p-value	+	–	~	p-value	+	–	~	p-value
Test Problems	MaF1	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**
	MaF2	0	4	1	1.000000	0	5	0	1.000000	**4**	**0**	**1**	**0.055748**
	MaF3	1	1	3	0.787406	0	3	2	0.957384	**3**	**1**	**1**	**0.077353**
	MaF4	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**
	MaF5	0	5	0	1.000000	0	5	0	1.000000	0	5	0	1.000000
	MaF6	**5**	**0**	**0**	**0.030971**	**4**	**1**	**0**	**0.077353**	**4**	**1**	**0**	**0.077353**
	MaF7	1	2	2	0.787406	2	1	2	0.787406	**5**	**0**	**0**	**0.030971**
	MaF8	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**	**4**	**1**	**0**	**0.077353**
	MaF9	**4**	**1**	**0**	**0.059385**	**3**	**1**	**1**	0.589639	1	4	0	0.957384
	MaF10	**5**	**0**	**0**	**0.030971**	2	1	2	0.589639	0	5	0	1.000000
	MaF11	1	1	3	0.787406	2	2	1	0.787406	0	3	2	0.957384
	MaF12	2	3	0	0.957384	1	2	2	0.957384	2	1	2	0.589639
	MaF13	**4**	**0**	**1**	**0.055748**	**5**	**0**	**0**	**0.030971**	**4**	**0**	**1**	**0.055748**
	mDTLZ1	3	2	0	0.787406	1	3	1	1.000000	**5**	**0**	**0**	**0.030971**
	mDTLZ2	**4**	**0**	**1**	**0.030971**	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**
	mDTLZ3	**4**	**0**	**1**	**0.030971**	**4**	**1**	**0**	**0.077353**	**5**	**0**	**0**	**0.030971**
	mDTLZ4	**5**	**0**	**0**	**0.030971**	**5**	**0**	**0**	**0.030971**	**4**	**1**	**0**	**0.077353**
No. (m)	m=3	**10**	**3**	**4**	**0.021797**	10	6	1	0.740367	**10**	**4**	**3**	**0.075641**
	m=4	**12**	**2**	**3**	**0.008027**	**10**	**3**	**4**	**0.030257**	**11**	**4**	**2**	**0.050021**
	m=5	**11**	**4**	**2**	**0.040889**	**11**	**3**	**3**	**0.021797**	**13**	**4**	**0**	**0.012279**
	m=7	10	5	2	0.294907	**9**	**5**	**3**	0.320174	**12**	**5**	**0**	**0.079002**
	m=10	**11**	**5**	**1**	**0.089487**	9	8	0	0.999077	10	5	2	0.300436
	All	**54**	**19**	**12**	**0.042814**	49	25	11	0.097011	**56**	**12**	**7**	**0.040159**

Regarding the summary in Table II for various test problems with different objectives, it is clear that the perfomance of NSGA-III/S can be obviously enhanced on MaF1, MaF4, MaF6, MaF8-MaF10, MaF13 and mDTLZ2-mDTLZ4 with all considered objectives. Similarly, θ-DEA/S is significantly better than θ-DEA on MaF1, MaF4, MaF6, MaF8, MaF13 and mDTLZ2-mDTLZ4, and EFR-RR/S also achieves remarkable improvement over EFR-RR on MaF1-MaF4, MaF6-MaF8, MaF13, and mDTLZ1-mDTLZ4. From these results, it is reasonable to conclude that the embedding of SRV into these MaOEA/Ds can highly improve their performance in solving MaOPs with irregular PFs, such as MaF1, MaF4 and mDTLZ1 -mDTLZ4 with inverted PFs, MaF6, MaF8-MaF9 and MaF13 with degenerated PFs, and MaF7 with disconnected PF. However, for MaOPs with regular PFs, like MaF5 and MaF12 with concave PFs, the effect is not obvious or even becomes poor. It is worth noting that MaF2 is modified from DTLZ2 [34] by increasing the difficulty of convergence, and the performance of SRV on this problem becomes poor when it is embedded into NSGA-III and θ-DEA.

As observed from the comparisons in the last row of Table II, the numbers of comparisons that show the superiority of NSGA-III/S over NSGA-III, θ-DEA/S over θ-DEA, and EFR-RR/S over EFR-RR, are respectively 54, 49, and 56 out of a total of 85 comparisons, while the numbers of comparisons showing the superiority of NSGA-III over NSGA-III/S, θ-DEA over θ-DEA/S, and EFR-RR over EFR-RR/S, are only 19, 25 and 12, respectively. Moreover, the p-values for their comparisons are all nearly close to 0 on all 85 comparisons. Therefore, it is reasonable to conclude that the embedding of SRV into NSGA-III, θ-DEA and EFR-RR can bring significant improvements on these MaOPs with different objectives.

[1] The codes of NSGA-III, EFR-RR and θ -DEA are downloaded from http://www.cs.bham.ac.uk/~xin/papers/TEVC2016FebManyEAs.zip.
The codes of MOEA/D-AWA, A-NSGA-III, RVEA*and AR-MOEA are downloaded from https://github.com/BIMK/PlatEMO.
The codes of VaEA are downloaded from https://www.researchgate.net/profile/Xiang_Yi9/publications.
The codes of MaOEA/C and the SRV proposed in this paper are downloaded from https://github.com/songbai-liu/ManyobjectiveOptimization.

D. Further Studies on Embedding SRV into MaOEA/Ds

As discussed in **Section IV.C**, the embedding of SRV into NSGA-III, θ-DEA and EFR-RR can obviously improve their performance in solving irregular MaOPs, but it may perform worse on regular MaOPs. This is reasonable as the original fixed RVs have already fitted the PF shapes of regular MaOPs, and the frequent change of RVs in MaOEA/Ds may affect their convergence, which also introduces an extra computational cost [43] [51]. In addition, the quality of the data (i.e., the distribution of P_c) for clustering in SRV may also impact the efficiency on guiding the evolution by the extracted RVs. In most MaOEAs with adaptive adjustment of RVs, there are two common strategies to alleviate this problem. The first strategy evolves the population as guided by N fixed RVs during a period of previous generations to get a population with relative high quality (good convergence and distribution), and then triggers the adaptive adjustment of RVs at the following generations to generate N new RVs [45] [51]. The second strategy runs the adaptive adjustment of RVs at a periodic interval of generations [41], [42], [43], [54]. In this section, the performance of these two strategies of embedding SRV into MaOEA/Ds is studied below.

1) Triggering SRV after $\rho \times G_{\max}$ generations

Here, the three above MaOEAs enhanced with SRV (i.e., NSGA-III/S, θ-DEA/S and EFR-RR/S) were also investigated, by further using a parameter ρ to control the running of SRV. Let's assume that G is the counter of generations and $G_{\max}$ is the maximum number of generations. When $G \le \rho \times G_{\max}$, their evolutionary processes are all guided by the pre-set N uniformly distributed RVs $(r^1, r^2, ..., r^N)$, where $\rho \in [0,1]$ and N is the number of RVs and the population size. When $G > \rho \times G_{\max}$, the pre-set RVs $(r^1, r^2, ..., r^N)$ will be replaced by the self-guided RVs $(rv^1, rv^2, ..., rv^N)$ from SRV. Seven values of ρ, i.e., ρ= 0.0, 0.1, 0.3, 0.5, 0.7, 0.9 and 1.0, are included to study the impacts of SRV that is triggered after $\rho \times G_{\max}$ generations in each of these enhanced MaOEAs. Please note that the comparison results of ρ= 0.0 (i.e., the original MaOEAs) and ρ= 1.0 (i.e., the above enhanced MaOEAs) have been discussed in **Section IV. C**, which validate the effectiveness of SRV. In this experiment, all the test problems (i.e., MaF1-MaF13 and mDTLZ1-mDTLZ4 with 3, 4, 5 and 10 objectives) were used. To clearly quantify how well each of these enhanced MaOEAs performs under different values of ρ, Friedman's test from the KEEL tool [69] was used, which will give the rank of each algorithm under different values of ρ on all 68 test cases. In Table III, their Friedman's average performance ranks are provided for each algorithm under different values of ρ. It is difficult to find a regular trend for the ranks' change in each of NSGA-III/S, θ-DEA/S and EFR-RR/S, as they all experience the rise and the decline on seven ranks. For example, θ-DEA/S obtains a rank as 4.133 at ρ=0.0, and gradually gets better at ρ=0.3 with a rank as 3.816, but it becomes worse at ρ=0.5 with a rank as 4.032. However, θ-DEA/S gets the best rank with 3.613 at ρ=0.7 and then gets the worst rank with 4.426 at ρ=1.0. For NSGA-III/S, it obtains the best rank with 2.937 at ρ=0.0 and gets the worst rank with 5.031 at ρ=0.9. Meanwhile, EFR-RR obtains the best rank with 3.369 at ρ=0.5 and gets the worst rank with 5.463 at ρ=1.0. Therefore, it is hard to determine which value of ρ is optimal for embedding SRV into these MaOEA/Ds.

TABLE III
ρ- BASED AVERAGE PERFORMANCE RANKS OF THREE MaOEA/RVS

NSGA-III/S		θ-DEA/S		EFR-RR/S	
ρ = 0.0	**2.937**	ρ = 0.0	4.133	ρ = 0.0	3.675
ρ = 0.1	3.094	ρ = 0.1	4.125	ρ = 0.1	3.712
ρ = 0.3	3.438	ρ = 0.3	3.816	ρ = 0.3	3.943
ρ = 0.5	4.406	ρ = 0.5	4.032	ρ = 0.5	**3.369**
ρ = 0.7	4.269	ρ = 0.7	**3.613**	ρ = 0.7	3.687
ρ = 0.9	**5.031**	ρ = 0.9	3.882	ρ = 0.9	4.151
ρ = 1.0	4.825	ρ = 1.0	**4.426**	ρ = 1.0	**5.463**

TABLE IV
τ- BASED AVERAGE PERFORMANCE RANKS OF THREE MaOEA/RVS

NSGA-III/S		θ-DEA/S		EFR-RR/S	
τ = 1	**1.437**	τ = 1	**1.813**	τ = 1	**2.621**
τ = 10	2.953	τ = 10	3.372	τ = 10	3.188
τ = 30	**4.227**	τ = 30	**4.239**	τ = 30	**4.189**
τ = 50	4.211	τ = 50	3.879	τ = 50	3.783
τ = 80	4.195	τ = 80	3.708	τ = 80	3.313
τ = 100	3.977	τ = 100	3.989	τ = 100	3.906

2) Triggering SRV at every interval of τ generations

Here, the three above MaOEAs enhanced with SRV (i.e., NSGA-III/S, θ-DEA/S and EFR-RR/S) were also studied, by updating their RVs at a regular interval of generations. This strategy aims to reduce the computational cost and avoids affecting the convergence speed due to the frequent change of RVs. In this experiment, a parameter τ is used to control the interval of generations. By this way, SRV is triggered at every interval of τ generations to obtain N self-guided RVs $(rv^1, rv^2, ..., rv^N)$ in NSGA-III/S, θ-DEA/S and EFR-RR/S. Six typical values of τ (i.e., 1, 10, 30, 50, 80 and 100) are chosen to study the impact of τ. Similarly, Friedman's test from the KEEL tool [69] was adopted, which will give the rank for each algorithm under different values of τ on all 68 test cases. Thus, each of NSGA-III/S, θ-DEA/S and EFR-RR/S has six average performance ranks, which are reported in Table IV. Obviously, NSGA-III/S, θ-DEA/S and EFR-RR/S all show the best rank at τ=1, obtaining the best performance ranks as 1.437, 1.813 and 2.621, respectively, which are much smaller than those of their ranks with other τ values. Moreover, they all perform worst at τ= 30, obtaining the worst performance ranks as 4.227, 4.239 and 4.189, respectively for NSGA-III/S, θ-DEA /S, and EFR-RR/S. Besides that, there is no clear trend for the impact of τ in these MaOEA/Ds, as the ranks of EFR-RR/S and θ-DEA/S get better from τ= 30 to τ= 80, and then become worse at τ= 100, while the ranks of NSGA-III/S become better from τ= 30 to τ= 100.

Although the setting of τ= 50 has been used in many other MaOEAs [52] [54], our experimental results suggest that τ= 1 is the best choice of embedding SRV into these MaOEA/Ds. However, as reflected by the results in **Section IV. C**, the convergence speed still may be affected due to the frequent change of RVs when τ= 1 is adopted.

E. A new strategy of embedding SRV into MaOEA/Ds

As pointed out in [27] and [43], it is difficult to identify the timing and frequency to adapt RVs in MaOEA/Ds, which is discussed in **Section IV. D** to show that these challenges also exist for our SRV strategy. In this subsection, a new strategy of

Algorithm 4 Framework of MaOEA/SRV

1: initialize: N, m, $G_{\max}$, $Q=\varnothing$, G=1, P=$\{x^1, x^2, ..., x^N\}$, $\theta_c^{\min}$
2: **while** $G \le G_{\max}$
3: perform SBX and PM on P to get Q with N individuals
4: $U = P+Q$ to get S_t by non-dominated sort and set $P=\varnothing$
5: set $P_c = S_t$ and normalize all individuals of P_c by (2)
6: update θ_c by (17)
7: get N RVs $(rv^1, rv^2, ..., rv^N) = SRV(P_c, N, m, \theta_c)$
8: get N subsets $S_1^{RV}, S_2^{RV}, ..., S_N^{RV}$ by each $x \in P_c$ associated with the closest RV in $(rv^1, rv^2, ..., rv^N)$ by $\theta(x, rv^i)$ of (6)
9: get r^* by jointing the ideal point and the nadir point
10: **for** i:=1 to N
11: **for** each $x \in S_i^{RV}$
12: compute $d_1(x, r^*)$ by (5) to as the I_c value of x
13: **end for**
14: sort individuals of S_i^{RV} by their I_c values to give a rank
15: **end for**
16: all $x \in P_c$ are divided into multiple subsets: $S_1^{I_c}, S_2^{I_c}, ..., S_L^{I_c}$
17: set P= $S_1^{I_c}+S_2^{I_c}+...+S_{l-1}^{I_c}$, where $l \le L$, $|P|<N$, $|P+S_l^{I_c}| \ge N$
18: add N-$|P|$ solutions (randomly selected from $S_l^{I_c}$) into P
19: $G = G+1$ and $Q=\varnothing$
20: **end while**
21: **return** P

embedding SRV into MaOEA/Ds is proposed to balance convergence and diversity for tackling MaOPs, forming a new algorithm named MaOEA/SRV. In MaOEA/SRV, its APP is run based on N self-guided RVs from our SRV strategy, which are selected from different subsets to maintain diversity, while its ESS only considers the convergence information (I_c) for all individuals in the same subset, aiming to accelerate the convergence speed. Furthermore, all individuals in these N subsets share the same RV (r^*) to compute their I_c values. Here, r^* is a fixed RV connecting the ideal point and the nadir point in the normalized objective space. This way, MaOEA/SRV performs well in maintaining the balance of convergence and diversity when tackling different MaOPs.

The pseudo-code of MaOEA/SRV is given in **Algorithm 4**. In line 1, an initialization process is run to get the values of N (the population size), m (the number of objectives), and $G_{\max}$ (the maximum number of generations). An initial population P is randomly generated with N individuals, an offspring population Q is set as an empty set, and the generation counter G is set to 1. While $G \le G_{\max}$, the evolutionary loop in lines 3-19 is run. In line 3, N new individuals in Q are generated by running the variation operators (SBX and polynomial-based mutation) on P. In line 4, P and Q are combined to get the union population U, P is reset as an empty set, and then a temporary population S_t is obtained by reserving no less than N individuals with the best non-dominated ranks [6] from U. After that, we set the candidate population $P_c = S_t$ and normalize all individuals of P_c by (2) in line 5. Next, θ_c is computed by (17) in line 6 and our SRV strategy (**Algorithm 1**) is run in line 7 (details are provided in **Section III. B**), which gets N RVs. In line 8, N subsets $S_1^{RV}, S_2^{RV}, ..., S_N^{RV}$ are obtained by associating each individual $x \in P_c$ to its closest RV using the distance metric in (6). The shared RV r^* is obtained in line 9 by joining the ideal point and the nadir point. For each $x \in S_i^{RV}$ ($i = 1, ..., N$), $d_1(x, r^*)$ in (5) is used as its I_c indicator, as computed in lines 11-13. Then, in line 14, all individuals of S_i^{RV} are sorted based on their I_c values, which gives an I_c rank for each individual, and then all individuals of $S_1^{RV}, S_2^{RV}, ..., S_N^{RV}$ can be further divided into multiple L subsets ($S_1^{I_c}, S_2^{I_c}, ..., S_L^{I_c}$) based on their I_c ranks. Thereafter, as shown in lines 17-18, all individuals in $S_1^{I_c}+S_2^{I_c}+...+S_{l-1}^{I_c}$ are first added into P, while N-$|P|$ individuals in the last subset $S_l^{I_c}$ are randomly selected into P. At last, Q is reset as an empty set and the counter G is increased by 1 in line 19. If G is still smaller than $G_{\max}$, the above procedures in lines 3-19 will be run iteratively. Otherwise, P is the output produced in the last generation in line 21, as the final approximation set.

TABLE V
HV RESULTS OF FOUR MAOEAS WITH SRV ON CONCAVE PROBLEMS

Problem	m	NSGA-III/S	θ-DEA/S	EFR-RR/S	MaOEA/SRV
MaF2	3	2.439E-01	2.380E-01	2.425E-01	**2.509E-01**
	5	2.122E-01	2.180E-01	2.411E-01	**2.542E-01**
	7	1.786E-01	1.886E-01	2.269E-01	**2.346E-01**
	10	1.565E-01	1.759E-01	2.171E-01	**2.220E-01**
MaF5	3	5.497E-01	5.638E-01	5.223E-01	**5.663E-01**
	5	7.879E-01	7.887E-01	7.708E-01	**7.986E-01**
	7	8.756E-01	8.803E-01	8.745E-01	**8.993E-01**
	10	9.347E-01	9.381E-01	9.449E-01	**9.640E-01**
MaF12	3	5.116E-01	5.194E-01	4.865E-01	**5.213E-01**
	5	6.577E-01	6.639E-01	6.565E-01	**7.089E-01**
	7	6.882E-01	6.945E-01	7.195E-01	**7.744E-01**
	10	6.869E-01	6.813E-01	7.503E-01	**8.028E-01**

Table V provides the HV results of four MaOEAs with SRV on three concave problems, i.e., MaF2, MaF5 and MaF12 with 3 to 10 objectives. From the results in Table V, it is observed that the performance of MaOEA/SRV were improved significantly when compared to NSGA-III/S, EFR-RR/S and θ-DEA/S. In order to further study the performance of MaOEA/SRV, six competitive MaOEAs (MOEA/D-AWA [51], A-NSGA-III [40], RVEA* [27], AR-MOEA [46], VaEA [57] and MaOEA/C [63]) were also included for comparison.

Table A.V and Table A.VI in the supplementary file respectively list their HV and IGD comparison results on MaF1-MaF13 with 3 to 10 objectives. As observed from these results, MaOEA/SRV also shows a superior performance, as it is best in about half of the comparisons, i.e., in 29 out of 65 HV cases and 28 out of 65 IGD cases, while MOEA/D-AWA, A-NSGA-III, RVEA*, AR-MOEA, VaEA and MaOEA/C are respectively best in 3, 2, 2, 11, 4, 14 HV cases and best in 6, 0, 4, 9, 12, 6 IGD cases. From the one-by-one comparisons in the last row of Table A.V and Table A.VI, MaOEA/SRV performs better than MOEA/D-AWA, A-NSGA-III, RVEA*, AR-MOEA, VaEA and MaOEA/C in 54, 53, 53, 40, 39, 34 out of 65 HV cases and in 44, 50, 48, 35, 29, 37 out of 65 IGD cases, respectively, while it is only outperformed by them in 6, 8, 9, 18, 13, 13 HV cases and in 15, 11, 12, 22, 22, 13 IGD cases, respectively. Thus, MaOEA/SRV is validated to present a superior performance over these six competitors in most cases.

To visually show and support the above discussion results, some final solution sets with the 15th better HV values from 30 runs are plotted in Figs.A1-A6 of the supplementary file, respectively for 3-objective MaF7 with a disconnected PF to have a better visualization, for 4-objective MaF8 with a degenerated PF, for 5-objective MaF4 with an inverted PF, for 7-objective MaF2 with a concave PF, and for 10-objective MaF6 with a degenerated PF, which show the solutions' distribution for different test problems with various objectives in the high dimensional objective space. Obviously, all the final solution sets obtained by MaOEA/SRV are distributed evenly on these five representative problems with different

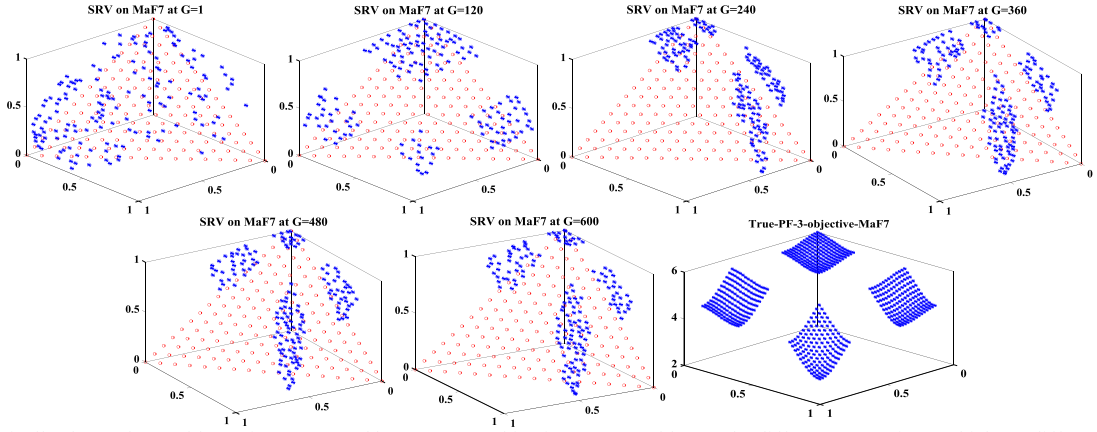

Fig. 6 The dynamically changed RVs (blue points) extracted by SRV on 3-objective MaF7 problem at the different generations, which are different from the preset evenly distributed RVs (red points) and can well trace the shapes of the true PF.

kinds of PFs. However, AR-MOEA fails to find the leftmost segment of the PF on MaF7 with 3 objectives, while MOEA/D-AWA, A-NSGA-III and RVEA* find most of the final solution sets with poor distributions.

F. More Discussions about SRV

1) Effectiveness of the Two Main Components in SRV

The effectiveness of our proposed SRV is validated by the comparison results of embedding SRV into three MaOEA/Ds in Section IV. C and the superior performance of MaOEA /SRV is also studied in Section IV. E. As introduced in Section III, N RVs are obtained by running SRV on the candidate population P_c, which has two main procedures (ADM-based initialization of centroids and self-guided adjustment for centroids). To study their respective contributions, two variants of MaOEA/SRV, respectively named MSRV-I and MSRV-II, are designed here for comparison. In MSR-I, the self-guided adjustment is removed from MaOEA/SRV, i.e., only the ADM-based initialization is considered in MSRV-I to get the RVs. Regarding MSRV-II, the ADM-based initialization for centroids is replaced by a random-based initialization, i.e., solutions in P_c are randomly selected as the initial centroids in MSRV-II. All the parameter settings in MaOEA/SRV, MSRV-I, and MSRV-II are the same.

The HV comparison results of MaOEA/SRV and its two variants on MaF1-Ma13 with 3 to 10 objectives are provided in Table. A. VII of the supplementary file due to page limitations. From the second to last row of this table, MSRV-I obtained the best results on 6 out of 65 comparisons, while MSRV-II could not perform best on any MaF problem. Considering the one-by-one comparisons in the last row of this table, MaOEA/SRV is only outperformed by MSRV-I on 4 cases. Therefore, it is reasonable to conclude that these two main procedures in SRV contribute to enhancing its overall performance, which can help to get appropriate RVs for these adopted test problems. Especially, the ADM-based initialization of centroids seems to be the main contribution as the performance of MSRV-II without this process deteriorates greatly, which indicates that the selected initial centroids have a significant impact on the final clustering results.

2) Observation of RVs obtained by SRV

The experimental studies given above have validated the effectiveness of the SRV strategy, where RVs are automatically extracted from the population to properly guide the evolutionary process. To observe the change of RVs under populations with different qualities for clustering during the evolutionary process, N RVs extracted by our SRV (**Algorithm 1**) in MaOEA/SRV (**Algorithm 4**) are examined for each test problem at some specified generations. Considering the visualization, only the RVs on 3-objective problems are plotted here. Thus, N is set to 153, the generation counter G is specified as $\{1,120,240,360,480,600\}$, and the extracted RVs at these specified generations are plotted. Due to page limitations, Figs. A7-A10 are provided in the supplementary file, respectively for MaF1 with an inverted PF, MaF2 with a partial concave PF, MaF3 with a convex PF, and MaF6 with a degenerated PF, while Fig. 6 is given here as an example to show the dynamic change of RVs at the different generations for solving MaF7 with a disconnected PF. In these figures, the RVs extracted by SRV and the preset evenly distributed RVs are respectively marked by blue points and red points to show the dynamic change of these extracted RVs, while the true PF is plotted in the last figure to show whether these extracted RVs can fit the shape of the true PF. As observed from these figures, in the initial stage (G =1 and 120), the RVs extracted by SRV may mismatch the shape of the true PF, which is mainly induced by the poor quality (i.e., poor distribution) of the initial population, but these self-guided RVs are still more suitable to guide the evolutionary process than the preset fixed RVs. With the running of generations, in the mid and later stages (G=240, 360, 480, and 600), the extracted RVs can gradually match the shape of the true PF, i.e., the extracted RVs by SRV can self-guide the population to approach the PF.

G. Computational Complexity Analysis of MaOEA/SRV

The process of MaOEA/SRV is introduced in **Algorithm 4.** Here, the computational complexities for its main procedures are analyzed as follows. To get the candidate population P_c, the non-dominated sort is run on the union population U with $2N$ solutions in line 4, which requires a time complexity of $O(mN^2)$, where m is the objective number and N is the population size. Then, in line 5, the normalization of solutions in P_c requires $O(m^2|P_c|)$, $N \le |P_c| \le 2N$. In line 7, to get N RVs from P_c by **Algorithm 1**, $O(m|P_c|^2)$ is required to run the ADM -based initialization of centroids in **Algorithm 2** and $O(m|P_c|^2 \eta)$ is needed to finish the self-guided adjustment of centroids

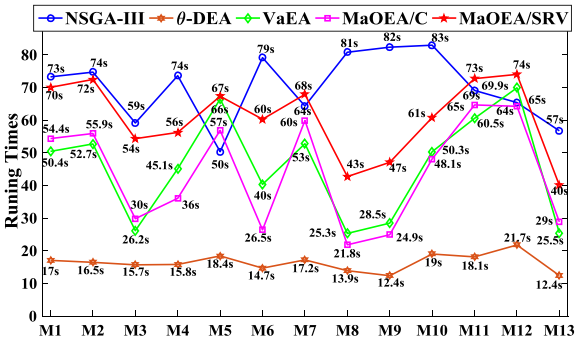

Fig. 7 The average running times of MaOEA/SRV and other four MaOEAs on 10-objective MaF1-MaF13 (abbreviated to M1-M13) problems.

in **Algorithm 3,** where $\eta = \min\{2m, \sigma\}$, and $\sigma \geq 1$ indicates the number of times that the RVs need to be self-adjusted in order to reach a stable state, i.e., *flag*=**false** in **Algorithm 3**. In line 8, N subsets $S_1^{RV}, S_2^{RV}, ..., S_N^{RV}$ of P_c are obtained by an associating procedure, which requires a time complexity of $O(m|P_c|^2)$. Moreover, to get the final population P in lines 10-18, the selection process on these N subsets needs a total time complexity of $O(mNlogN)$. As indicated above, since $|P_c|$ and η are respectively equal to $2N$ and $2m$ in the worst case, the overall worst time complexity of MaOEA/SRV is $O(m^2N^2)$ in one generation. Note that the times σ of calling the self-guided adjustment method for centroids in MaOEA/SRV are often smaller than $2m$ (i.e., $\sigma < 2m$) in solving MaF1-MaF13, which can be observed from the 10-objective cases in Fig. A11 of the supplementary file due to page limitations. Thus, the actual time complexity will be smaller than the worst one.

To evaluate the actual runtime of MaOEA/SRV, its average running times (in seconds: *s*) from 30 runs are plotted in Fig. 7, for MaF1-MaF13 with 10 objectives. Additionally, the average running times of NSGA-III, θ-DEA, MaOEA/C and VaEA on these 10-objective problems are also plotted in Fig. 7 for comparison. Obviously, θ-DEA shows the fastest speed due to its simple implementation. Although MaOEA/SRV has a slightly slower running speed than VaEA and MaOEA/C, it still runs faster than NSGA-III on most problems.

V. Conclusions and Future Work

This paper proposes a novel SRV strategy to extract reference vectors from the population, which can quickly track the evolutionary process and fit the PF shapes. To achieve this purpose, individuals in the candidate population P_c are classified into N clusters using a modified k-means with k=N (N is the population size), which includes two main procedures: initialization of centroids using an angle-based density measurement strategy and the self-adjustment method for centroids. Then, N reference vectors can be extracted by mapping the centroids of these clusters on the unit hypersphere. When integrating SRV into three competitive MaOEA/Ds (NSGA-III, EFR-RR and θ-DEA), their performance can be highly improved, especially on some problems with irregular PFs. Moreover, our experiments also investigated the impact of the two common strategies to embed SRV into MaOEA/Ds, and this motivated us to present a new strategy of embedding SRV into MaOEA/Ds, under the use of frequently changed reference vectors. Thus, N reference vectors generated from SRV are only included in the association-based partition process to maintain diversity, while the use of shared and fixed reference vectors by joining the ideal point and nadir point is considered in the elitism selection strategy to ensure convergence. This way, a new algorithm named MaOEA/SRV was presented, which can provide a good balance between convergence and diversity. When compared to other MaOEAs with adaptive reference vectors (MOEA/D-AWA, A-NSGA-III, RVEA* and AR- MOEA) and two competitive MaOEAs (VaEA and MaOEA/C), MaOEA/SRV showed superiority in solving most of the test problems adopted. Furthermore, the experiments have validated the effectiveness and efficiency of SRV in MaOEA/SRV, which has a comparable time complexity with most of the compared MaOEAs.

In our future work, other machine learning methods will be studied to generate reference vectors and this SRV strategy will be extended to solve more difficult MaOPs with constraints [70]-[72]. Since our SRV strategy can adaptively extract reference vectors, it may be more suitable for real-life applications as their PF shapes are generally irregular, which is considered as part of our future work.

References

[1] L. Liu, Y. Song, H. Zhang, H. Ma, A.V. Vasilakos, "Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 818-831, 2015.

[2] Y. Song, L. Liu, H. Ma, A.V. Vasilakos, "A Biology-Based Algorithm to Minimal Exposure Problem of Wireless Sensor Networks," *IEEE Trans. Network and Service Management*, vol. 11, no. 3, pp. 417-430, 2014.

[3] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, "Multi-objective trajectory optimization of Space Manoeuvre Vehicle using adaptive differential evolution and modified game theory," *Acta Astronautica*, vol. 136, pp. 273-280, 2017.

[4] L. Wu, C. Zuo, H. Zhang, "A cloud model based fruit fly optimization algorithm," *Knowledge based Systems*, vol. 89, pp. 603-617, 2015.

[5] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, Ma, USA: Kluwer Academic, 1999.

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

[7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des., Optimisation Control.*, pp. 95–100, 2002.

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

[9] M. Wu, K. Li, S. Kwong, Q. Zhang, J. Zhang, "Learning to Decompose: A Paradigm for Decomposition-Based Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376–390, 2019.

[10] S. Jiang, J. Zhang, "A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm," *IEEE Trans. Cybernetics.*, vol. 45, no. 10, pp. 2202–2213, 2015.

[11] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective Knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, 2015.

[12] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 13:1–13:35, 2015.

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

[14] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, 2014.

[15] H. Wang, Y. Yao, "Corner Sort for Pareto-Based Many-Objective Optimization," *IEEE Trans. Evol. Cybernetics.*, vol. 44, no. 1, pp. 92–102, 2014.

[16] M. Li, S. Yang, X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, 2014.

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

[18] H. Wang, L. Jiao and X. Yao, "Two_Arch2: An Improved Two-Archive

Algorithm for Many-Objective Optimization", *IEEE Trans. Evol. Comput.*, vol. 19, no. 4,pp. 524–541, 2015.
[19] B.D. Li, K. Tang, J.L. Li, and X. Yao, "Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators", *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924–938, 2016.
[20] X. Cai, H. Sun, Q. Zhang, Y. Huang, "A Grid Weighted Sum Pareto Local Search for Combinatorial Multi and Many-Objective Optimization," *IEEE Trans. Cybern.*, vol. 49, no. 9. pp. 3586–3598, 2019.
[21] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.
[22] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, 2016
[23] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, 2016.
[24] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, Y. S. Ong, "A New Decomposition-Based NSGA-II for Many-Objective Optimization," IEEE Trans. Systems, Man, And Cybernetics: Systems, vol. 48, no. 7, pp. 1191–1210, 2018.
[25] R. Wang, Z. B. Zhou, and H. ishibuchi, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3–18, 2018.
[26] S. Y. Jiang, S. X. Yang, "A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multi-objective and Many-objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, 2017.
[27] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.
[28] X. Cai, Z. X. Yang, Z. Fan, Q. F. Zhang, "Decomposition-Based-Sorting and Angle-Based-Selection for Evolutionary Multiobjective and Many-Objective Optimization," *IEEE Trans. Evol. Cybernetics.*, vol. 47, no. 9, pp. 2824–2837, 2017.
[29] C. Liu, Q. Zhao, B. Yan, et all, "Adaptive Sorting-based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 247–257, 2019.
[30] M. Asafuddoula, T. Ray, and R. Sarker "A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no.3, pp.445–460, 2015.
[31] M. Y. Wu, K. Li, S. Kwong, Q. Zhang, "Evolutionary Many-Objective Optimization Based on Adversarial Decomposition," *IEEE Trans. Evol. Cybernetics.*, in press, doi:10.1109/TEVC.2018.2872803, 2018.
[32] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2017.
[33] M. Li, S. Yang, and X. Liu, "Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, 2016.
[34] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," *Evolutionary Multiobjective Optimization, ser. Advanced Information and Knowledge Processing*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer London, pp. 105–145, 2005.
[35] S. Huband, L. Barone, R. while, and P. Hingston, "A scalable multi-objective test problem tookit," in *Proc. 3rd Conf. Evol .Multi-Criterion Optimiz.*, *Lecture Notes in Computer Science*, vol. 3410, pp. 280–295, 2005.
[36] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.
[37] Z. Wang, Y. S. Ong, H. Ishibuchi, "On Scalable Multiobjective Test Problems with Hardly-dominated Boundaries," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 217–231, 2019.
[38] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. "Generalized decomposition." *International Conference on Evolutionary Multi-Criterion Optimization. Springer*, Berlin, Heidelberg, 2013.
[39] H. Ishibuchi, R. Imada, et al. "Reference point specification in inverted generational distance for triangular linear Pareto front." *IEEE Transactions on Evolutionary Computation*. Vol. 22, no. 6 pp. 961–975, 2018.
[40] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014..
[41] H. Jain and K. Deb, "An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization," *(LNCS 7811). Heidelberg, Germany: Springer*, pp. 307–321. 2013.
[42] 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 Proc. Int. Conf. Evol. Multi Criterion Optim.*, pp. 127–140, 2015.
[43] M. Asafuddoula, H. K. Singh, T. Ray, "An Enhanced Decomposition-Based Evolutionary Algorithm With Adaptive Reference Vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2321–2334, 2018.
[44] Y. Liu, D. Gong, X. Sun, Y. Zhang, "Many-objective evolutionary optimization based on reference points," *Applied Soft Computing*, vol. 50, pp. 344–355, 2017.
[45] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, 2013.
[46] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, "An Indicator Based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, 2018.
[47] F. Q. Gu, Y. M. Cheung, "Self-organizing Map-based Weight Design for Decomposition-based Many-objective Evolutionary Algorithm," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211–225, 2018.
[48] H. Ge, M. Zhao, L. Sun, et all, "A Many-Objective Evolutionary Algorithm with Two Interacting Processes: Cascade Clustering and Reference Point Incremental Learning," *IEEE Trans. Evol. Comput.*, in press, doi:10.1109/TEVC.2018.2874465, 2018.
[49] F. Gu, H. Liu. "A novel weight design in multi-objective evolutionary algorithm." *2010 International Conference on Computational Intelligence and Security*. *IEEE*, 2010.
[50] F. Gu, H. Liu, and K. C. Tan. "A multiobjective evolutionary algorithm using dynamic weight design method." *International Journal of Innovative Computing*, *Information and Control,* vol. 8, no. 5(B), pp. 3677-3688, 2012.
[51] Y. Qi, X. Ma, et al. "MOEA/D with adaptive weight adjustment." *Evolutionary computation*. vol. 22, no. 2, pp. 231-264, 2014.
[52] X. Cai, Z. Mei, Z. Fan, "A Decomposition-Based Many-Objective Evolutionary Algorithm With Two Types of Adjustments for Direction Vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2335–2348, 2018.
[53] I. Das and J. E. Dennis, "Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems," SIAM J. Optim., vol. 8, no. 3, pp. 631–657, 1998.
[54] H. Liu, L. Chen, Q. Zhang, K. Deb, "Adaptively Allocating Search Effort in Challenging Many-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433–448, 2018.
[55] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, no. 14, pp. 281–297,1967.
[56] K. Deb, K. Miettinen, et al., "Toward an Estimation of Nadir Objective Vector Using a Hybrid of Evolutionary and Local Search Approaches," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 821–841, 2010.
[57] Y. Xiang, Y. R. Zhou, M. Q. Li, "A Vector Angle based Evolutionary Algorithm for Unconstrained Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, 2017
[58] J. Cheng, G. Yen, G. Zhang, "A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 592–605, 2015
[59] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, 2017.
[60] Z. N. He, G. G. Yen, "Many-Objective Evolutionary Algorithm Based on Coordinated Selection Strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220–233, 2017.
[61] X. He, Y. Zhou, Z. Chen, Q. Zhang, "Evolutionary Many-objective Optimization based on Dynamical Decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 361–375, 2019.
[62] H. Liu, J. Lu, "Brief Survey of K-Means Clustering Algorithms," *Applied Mechanics and Materials*, vol. 740, pp. 624–628, 2015.
[63] Q. Lin, S. Liu, K. Wong, et all, "A Clustering-Based Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391–405, 2019
[64] U. V. Luxburg. "A tutorial on spectral clustering." *Statistics and computing*, vol. 17, no. 4, pp. 395-416, 2007.
[65] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function." *Journal of the American statistical association,* vol. 58, no. 301, pp. 236-244, 1963.
[66] A. Rodriguez, A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[67] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174-188, 2003.
[68] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, 2012.
[69] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrel, J.Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, pp. 307–318, 2009.
[70] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman. "Difficulty adjustable and scalable constrained multi-objective test problem toolkit." *Evolutionary Computation*, in press, doi: 10.1162/evco_a_00259, 2019.
[71] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, S. Chai, "Solving Multiobjective Constrained Trajectory Optimization Problem by an Extended Evolutionary Algorithm," *IEEE Trans. Cybern.*, in press, doi: 10.1109/TCYB.2018.2881190, 2018.
[72] Z. Ma, Y. Wang. "Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons." *IEEE Trans. Evol. Comput.*, in press, 2019.

Songbai Liu received the B.S. degree from Changsha University and the M.S. degree from Shenzhen University, China, in 2012 and 2018, respectively. He worked for ShenZhen TVT Digital Technology Co., Ltd as a software QA engineer from 2013 to 2015, and he worked for Shenzhen University as a research assistance from 2018 to 2019.

He is currently a PhD student in Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include nature-inspired computation, evolutionary many-objective optimization, and machine learning.

Qiuzhen Lin received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over twenty research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, and dynamic system

Ka-Chun Wong received his B.Eng. in Computer Engineering from The Chinese University of Hong Kong in 2008. He has also received his M.Phil. degree at the same university in 2010. He received his PhD degree from the Department of Computer Science, University of Toronto in 2015. After that, he assumed his duty as assistant professor at City University of Hong Kong. His research interests include Bioinformatics, Computational Biology, Evolutionary Computation, Data Mining, Machine Learning, and Interdisciplinary Research.

Carlos A. Coello Coello (M'98-SM'04-F'11) received Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996.

He is currently a Professor (CINVESTAV-3F Researcher) with the Computer Science Department, CINVESTAV-IPN, Mexico City, México. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His has over 45000 Google Scholar citations with an h-index of 83. His current research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Prof. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of *Exact Sciences*, the 2013 IEEE Kiyo Tomiyasu Award and the 2012 National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences. He is currently an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves on the editorial board of 12 other international journals. He is a member of ACM and the Mexican Academy of Science.

Jianqiang Li received his B.S. and Ph.D. Degree in automation major from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is a professor at the College of Computer and Software Engineering of Shenzhen University. He led a project of the National Natural Science Foundation, and a project of the Natural Science Foundation of Guangdong Province, China. His major research interests include embedded systems and Internet of Things.

Zhong Ming is a professor at College of Computer and Software Engineering of Shenzhen University. He is a member of a council and senior member of China Computer Federation. His major research interests are AI and cloud computing. He led two projects of National Natural Science Foundation, including one key project (61836005), one normal project (61672358)

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Visiting Scholar with Victoria University, Melbourne, VIC, Australia. His current research interests include computational intelligence, cloud computing, high-performance computing, operations research, and power-electronic circuits.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.