Particle Swarm Optimization With a Balanceable Fitness Estimation for Many-Objective Optimization Problems

Qiuzhen Lin, Songbai Liu, Qingling Zhu, Chaoyu Tang, Ruizhen Song, Jianyong Chen, *Member, IEEE*, Carlos A. Coello Coello, *Fellow, IEEE*, Ka-Chun Wong, and Jun Zhang, *Senior Member, IEEE*

***Abstract*—Recently, it was found that most multiobjective particle swarm optimizers (MOPSOs) perform poorly when tackling many-objective optimization problems (MaOPs). This is mainly because the loss of selection pressure that occurs when updating the swarm. The number of nondominated individuals is substantially increased and the diversity maintenance mechanisms in MOPSOs always guide the particles to explore sparse regions of the search space. This behavior results in the final solutions being distributed loosely in objective space, but far away from the true Pareto-optimal front. To avoid the above scenario, this paper presents a balanceable fitness estimation method and a novel velocity update equation, to compose a novel MOPSO (NMPSO), which is shown to be more effective to tackle MaOPs. Moreover, an evolutionary search is further run on the external archive in order to provide another search pattern for evolution. The DTLZ and WFG test suites with 4–10 objectives are used to assess the performance of NMPSO. Our experiments indicate that NMPSO has superior performance over four current MOPSOs, and over four competitive multiobjective evolutionary algorithms (SPEA2-SDE, NSGA-III, MOEA/DD, and SRA), when solving most of the test problems adopted.**

***Index Terms*—Fitness estimation method, many-objective optimization problems (MaOPs), particle swarm optimization (PSO).**

I. Introduction

In some real-world applications, it is common to face optimization problems having several (often conflicting) objectives [1], [2]. They are termed multiobjective optimization problems (MOPs) and attempt to search a Pareto-optimal set (PS) consisting of the best possible tradeoffs among the objectives. The mapping of PS in objective space is termed Pareto-optimal front (PF). Over the past 20 years, a number of nature-inspired heuristic algorithms, e.g., multiobjective evolutionary algorithms (MOEAs) [3], [4] and multiobjective particle swarm optimizers (MOPSOs) [5], [6], have been reported as an alternative to tackle various kinds of MOPs. Some early reported MOEAs, such as NSGA-II [3] and SPEA2 [7], usually adopted two criteria for population selection. Pareto dominance is first used to guide the search, and then a density estimator is employed to diversify the set of solutions obtained. Such operations in MOEAs are very effective in tackling MOPs with two or three objectives. However, when solving many-objective optimization problems (MaOPs, i.e., MOPs with more than three objectives), the performance of these MOEAs severely deteriorates [8], mainly due to the loss of selection pressure toward the true PF [9]–[11] and the weakened search capabilities of their evolutionary operators [12]. With the increase of objectives in MaOPs, most of the generated solutions are mutually nondominated.

Most MOEAs designed for MOPs with two or three objectives, are no longer suitable for tackling MaOPs. Thus, some research efforts were made to reduce a high number of objectives [9] or to redesign MOEAs for MaOPs [8]. Based on their selection mechanisms, these redesigned MOEAs can be mainly divided into three categories, i.e., Pareto-based MOEAs, decomposition-based MOEAs and indicator-based MOEAs.

Pareto-based MOEAs often face difficulties in offering sufficient selection pressure to approach the true PFs of MaOPs, which is mainly induced by the exponential increase of nondominated solutions at each generation. In order to solve this problem, many Pareto-based MOEAs were redesigned in different ways, such as using relaxed forms of Pareto dominance, including preference order ranking [13], grid dominance [14], and even a new dominance relation [15] (i.e., θ dominance that is based on predefined reference points and penalty-based boundary intersection approach). These improved approaches of dominance ranking can significantly strengthen the selection pressure, thus making more effective the evolutionary search toward the true PF. Another research direction has consisted in embedding effective diversity maintenance

Manuscript received March 14, 2016; revised July 4, 2016 and October 5, 2016; accepted November 7, 2016. Date of publication December 14, 2016; date of current version January 26, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61402291, in part by CONACyT under Grant 221551, and in part by the Early Career Scheme Grant from Research Grant Council under CityU Project 9048072 and RGC Project 21200816.

Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, and J. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: jychen@szu.edu.cn).

C. A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), Mexico City 07300, Mexico.

K.-C. Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

J. Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2016.2631279

mechanisms into Pareto-based MOEAs, including two diversity management mechanisms in [16], a shift-based density estimation (SDE) method in [17], and the use of reference points in NSGA-III [18].

The decomposition-based MOEAs transform MaOPs into a set of aggregated subproblems by decomposition approaches [19], [20]. This way, a set of well-distributed weight vectors helps to maintain the population diversity [21]. These MOEAs have shown their effectiveness in solving complicated MOPs with two or three objectives [22]–[24]. Recently, some promising approaches for this kind of MOEAs have been reported to make them more suitable for tackling MaOPs, such as MaOEA-R&D [25], I-DBEA [26], and MOEA/DD [19].

The indicator-based MOEAs, e.g., IBEA [27], SMS-EMOA [28], and MO-CMA-ES [29], integrate the convergence and diversity performance into a single indicator (e.g., hypervolume (HV) [30]) for population selection. These HV-based MOEAs were once a popular alternative for MOPs. However, the computational cost of this indicator grows exponentially with the number of objectives, which has severely prevented the use of HV-based MOEAs for MaOPs. Thus, some efforts have been made in [31]–[33] to reduce this computational cost. Moreover, two recently proposed MOEAs, i.e., SRA [34] and Two_Arch2 [35], were designed by combining two performance indicators, in order to synchronously consider the status of convergence and diversity when tackling MaOPs.

More recently, some reference-based MOEAs [36]–[38] were designed for tackling MOPs or MaOPs. These approaches exploit a set of predefined reference points on the environment selection to perform a multiple targeted search, rather than decomposing MOPs into a set of subproblems in decomposition-based MOEAs. This kind of MOEAs for MaOPs includes PICEA-g [38] and RVEA [36]. In some sense, NSGA-III and MOEA/DD also fall into this class due to the use of reference points in their approaches. The above mentioned MOEAs are mostly focused on small-scale MaOPs with up to 30 decision variables. When encountering MaOPs also with a large number of decision variables (i.e., large-scale MaOPs), more challenges are brought for MOEAs as pointed out in two recent studies [39], [40]. However, the study of such large-scale problems is beyond the scope of this paper.

On the other hand, many MOPSOs were designed to tackle MOPs with two or three objectives, showing a promising performance [41], [42]. By witnessing the poor performance of traditional MOEAs on MaOPs, a natural question that arises is whether traditional MOPSOs also suffer from the curse of dimensionality and perform poorly on MaOPs. Unfortunately, based on our preliminary results (presented in Section II-B), the answer is "yes" and the performance of most MOPSOs severely deteriorates on MaOPs. To overcome this problem, a balanceable fitness estimation (BFE) method is proposed in this paper as a diversity maintenance mechanism, and then a novel MOPSO (NMPSO) algorithm is accordingly implemented. This BFE approach integrates the information of convergence and diversity for each individual, in order to strengthen the selection pressure toward the true PF. Then, based on the parents with high quality on convergence and diversity, a hybrid search, i.e., a PSO-based search using a new velocity update equation and an evolutionary search [42] with simulated binary crossover (SBX) and polynomial-based mutation (PM), is conducted to effectively generate the nondominated solutions.

The performance of NMPSO was evaluated using the DTLZ [43] and WFG [44] test suites with a number of objectives that goes from 4 to 10. As we will see later on, when compared to four current MOPSOs (dMOPSO [45], SMPSO [46], D^{2}MOPSO [47], and MMOPSO [42]) and four competitive MOEAs (SPEA2-SDE [17], NSGA-III [18], MOEA/DD [19], and SRA [34]), NMPSO performed better on most of the test problems adopted. The effectiveness of the three main components (i.e., the BFE method, the novel velocity update equation and the embedded evolutionary search) of NMPSO, are also experimentally analyzed. Finally, the performance of the BFE method in other state-of-the-art multiobjective algorithms and the parameter sensitivity of the BFE method are also experimentally analyzed in detail.

The rest of this paper is organized as follows. Section II introduces the related background of MOPSOs and the motivations of this paper. The details of NMPSO are given in Section III. Several experimental studies are conducted in Section IV, in order to investigate the performance of NMPSO. Finally, our conclusions and some possible paths for future work are provided in Section V.

II. Related Background and Motivations

A. Some Current MOPSOs

Particle swarm optimization (PSO) is a population-based heuristic method inspired on social behavior (e.g., bird flocking and fish schooling) [48]. Due to its easy implementation and high search efficiency, PSO has been widely applied to solve single-objective optimization problems (SOPs) [49], [50], MOPs [51], and other real-life applications [52]. Regarding MOPs, a number of MOPSOs have been reported recently [6], [41], [42]. Generally, two common approaches (Pareto ranking and decomposition methods) are used in MOPSOs to identify their swarm leaders. Thus, most of the existing MOPSOs can be classified into two main categories.

The first category is formed by Pareto-based MOPSOs, which incorporate Pareto ranking into the standard PSO. In this way, the personal- and global-best particles can be easily made to guide the swarm. Some representative MOPSOs in this category are OMOPSO [53], SMPSO [46], CMPSO [41], and pccsAMOPSO [6]. The second category consists of decomposition-based MOPSOs. These MOPSOs adopt a decomposition approach to transform MOPs into a set of SOPs and then PSO can be applied directly to solve all of these SOPs. Some representative MOPSOs belonging to this category include MOPSO/D [54], SDMOPSO [55], dMOPSO [45], D^{2}MOPSO [47], and MMOPSO [42].

In the above MOPSOs, it was experimentally validated that some recently proposed MOPSOs, such as SMPSO [46], pccsAMOPSO [6], D^{2}MOPSO [47], and MMOPSO [42],

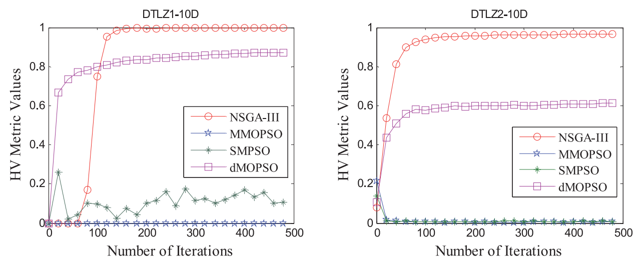


Fig. 1. HV curves of dMOPSO, SMPSO, MMOPSO, and NSGA-III for DTLZ1 and DTLZ2 with ten objectives.

have shown a promising performance in solving MOPs with two or three objectives. However, their performance on MaOPs are seldom investigated.

B. Motivations of Our Approach

In this section, a group of experiments were conducted to further study the performance of some MOPSOs on MaOPs. The DTLZ1 and DTLZ2 test problems [43] with ten objectives are adopted as the benchmark functions. The convergence behaviors of dMOPSO, SMPSO, and MMOPSO were studied as the representatives for two kinds of MOPSOs. NSGA-III [18] was also included in this paper. All the parameters of these algorithms were set as suggested in [18], [42], [45], and [46], which also can be found in Section IV-C. The experiments are run 30 times with a population size of 275 and a maximum number of iterations of 500. The average values of HV [30] in 30 runs were plotted in Fig. 1 for DTLZ1 and DTLZ2. It is noted that the HV values in Fig. 1 were normalized using the approach in [34], and a larger HV value indicates a better approximation of the true PF regarding both convergence and diversity.

As observed from Fig. 1, all the used MOPSOs face some difficulties in providing a sufficient selection pressure toward the true PFs of DTLZ1 and DTLZ2. Particularly for SMPSO and MMOPSO, their HV values are close to 0, which means that they cannot even find solutions that dominate the reference point. This poor performance is mainly induced by the use of Pareto ranking and crowding distance for the archive updates. Such approach was also found to perform poorly in MOEAs [17], as it favors poorly converged individuals in less-crowded regions of the objective space. dMOPSO is purely dependent on the decomposition approach adopted for the swarm update, thus it performs much better than SMPSO and MMOPSO. However, when compared to NSGA-III using HV, the performance of dMOPSO still has lots of room for improvement.

Based on the above study, it is clear that the performance of two kinds of MOPSOs significantly deteriorates on MaOPs. This is mainly due to the loss of sufficient selection pressure to approach the true PFs of MaOPs using the Pareto-ranking approach or the decomposition approach in them [28], [45], [46]. Some reported approaches [13]–[18] originally developed for MOEAs may be embedded into MOPSOs for enhancing their performance on MaOPs. However, these approaches may not lead to the optimal performance of MOPSOs, as they were not originally designed for them and may not properly integrate with the other components of a MOPSO, e.g., the selection of global-best particles and the PSO-based search pattern. Thus, in this paper, a novel BFE method was presented to build a new MOPSO algorithm (i.e., NMPSO). This BFE approach combines the convergence distance and the diversity distance of each particle, aiming to strengthen the selection pressure in approaching the true PFs of MaOPs. Moreover, a novel velocity update equation was also designed to provide another search direction and induce more disturbances, while an evolutionary search was further applied to the external archive in order to provide an extra search pattern. All of these new features enable NMPSO to effectively tackle MaOPs. When compared to the existing MOPSOs, the main contributions of this paper are summarized as follows.

1) The Pareto-ranking approach and the decomposition approach in MOPSOs have been found to lack sufficient selection pressure toward the true PFs of MaOPs. Thus, a novel BFE method was designed in this paper, which is able to drive the particles so that they converge fast to the true PFs of MaOPs. Both convergence distance and diversity distance are considered in the BFE approach. In this way, diversity and convergence can be properly balanced during the evolutionary process, and the curse of dimensionality that occurs in MaOPs can be greatly relieved for MOPSOs.
2) For traditional MOPSOs, the velocity and the position of each particle are usually guided by the positional information of the personal- and global-best particles. In our scheme, a novel velocity update equation was designed to provide an extra search direction pointing from the personal-best particle to the global-best ones. Thus, more disturbances are performed during the PSO-based search, which helps to guide all the particles to search toward the global-best particles.
3) An evolutionary search strategy for MOPSOs was extended to solve MaOPs by running on the external archive after the PSO-based search. This approach enables the exchange of elitist information among the individuals of the external archive. It is expected that this exchange provides an extra search pattern and may circumvent the inefficiency of PSO-based search on some MaOPs with special features.

III. Proposed Framework and Implementation

In this section, the details of our proposed NMPSO are described. Its four main components, i.e., the proposed BFE method, the evolutionary search on the external archive, the novel velocity update equation, and the archive update, are, respectively, introduced in the following sections. At last, the complete algorithm of NMPSO is also provided.

A. Novel Fitness Estimation Method

Here, we propose a novel BFE method to overcome the limitations of both Pareto ranking and decomposition approaches. The proposed method combines a convergence distance and a diversity distance to balance the convergence ability and the population diversity for each solution in objective space.

Let us assume that the swarm $P = \{p_1, p_2, \ldots, p_N\}$ includes N particles. Each particle has the position x_i and the velocity v_i ($i = 1, 2, \ldots, N$). For each particle p_i, its BFE value $\mathrm{fit}(p_i, P)$ consists of two components: a diversity distance and a convergence distance, as follows:

$$\mathrm{fit}(p_i, P) = \alpha \times \mathrm{Cd}(p_i, P) + \beta \times \mathrm{Cv}(p_i, P) \tag{1}$$

where $\mathrm{Cd}(p_i, P)$ and $\mathrm{Cv}(p_i, P)$, respectively, denote the normalized diversity and convergence distances of p_i; α and β are two factors that are used to tune the impacts of the diversity and convergence distances, respectively. When computing this BFE value, each objective of particle p_i is first normalized using the maximum and minimum values of the corresponding objective. Such normalization approach helps to eliminate the impact of different amplitudes on multiple objectives [18]. The normalized objectives $f_k'(p_i)$ ($k = 1, 2, \ldots, m$, and m is the total number of objectives) of p_i are obtained using

$$f_k'(p_i) = \frac{f_k(p_i) - f_k \min}{f_k \max - f_k \min} \tag{2}$$

where $f_k \max$ and $f_k \min$ are, respectively, the maximum and minimum values of the kth objective obtained from the nondominated solutions available in the external archive. Thus, this approach does not require any information from the true PFs of the MOP being solved. It is noted that, by this approach, the normalized objectives $f_k'(p_i)$ are mostly located between [0, 1], as observed from the experimental results.

Then, the normalized diversity distance $\mathrm{Cd}(p_i, P)$ is assigned by the normalized SDE distance [17], as follows:

$$\mathrm{Cd}(p_i, P) = \frac{\mathrm{SDE}(p_i) - \mathrm{SDE}_{\min}}{\mathrm{SDE}_{\max} - \mathrm{SDE}_{\min}} \tag{3}$$

where $\mathrm{SDE}_{\max}$ and $\mathrm{SDE}_{\min}$ are, respectively, the maximum and minimum SDE distances in the swarm (i.e., $\mathrm{SDE}_{\max} = \max\{\mathrm{SDE}(p) | p \in P\}$ and $\mathrm{SDE}_{\min} = \min\{\mathrm{SDE}(p) | p \in P\}$). A larger value of $\mathrm{Cd}(p_i, P)$ means that the particle p_i is surrounded with a faraway neighbor. $\mathrm{SDE}(p_i)$ is the original SDE distance defined in [17] using the shifted Euclidian distance to the nearest neighbor, as follows:

$$\mathrm{SDE}(p_i) = \min_{p_j \in P, j \neq i} \sqrt{\sum_{k=1}^{m} \mathrm{sde}\big(f_k'(p_i), f_k'(p_j)\big)^2} \tag{4}$$

where

$$\mathrm{sde}\big(f_k'(p_i), f_k'(p_j)\big) = \begin{cases} f_k'(p_j) - f_k'(p_i) & \text{if } f_k'(p_j) > f_k'(p_i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $f_k'(p_i)$ is the kth normalized objective of p_i as defined in (2).

On the other hand, the convergence distance $\mathrm{Cv}(p_i, P)$ is designed to reflect the convergence ability of $f_k'(p_i)$ ($k = 1, 2, \ldots, m$) with respect to the ideal point $z^* = (0, 0, \ldots, 0)$. It can be computed as follows:

$$\mathrm{Cv}(p_i, P) = 1 - \frac{\mathrm{dis}(p_i)}{\sqrt{m}} \tag{6}$$

where $\mathrm{dis}(p_i)$ denotes the Euclidean distance from $f_k'(p_i)$ ($k = 1, 2, \ldots, m$) to the ideal point z^*, as follows:

$$\mathrm{dis}(p_i) = \mathrm{sqrt}\left(\sum_{k=1}^{m} f_k'(p_i)^2\right). \tag{7}$$

A larger value of $\mathrm{Cv}(p_i, P)$ indicates that $f_k'(p_i)$ ($k = 1, 2, \ldots, m$) is closer to the ideal point z^*. Thus, when updating the external archive, a priority of selecting some individuals with larger convergence distances will increase the selection pressure toward the ideal point z^*, so as to minimize all the objectives simultaneously. Moreover, the use of diversity distance encourages to diversify the solutions as evenly as possible.

The above diversity and convergence distances are dynamically balanced using two weight factors α and β. They are adaptively adjusted for different particles, based on their status on convergence and diversity. For this purpose, the average values of all $\mathrm{Cd}(p_i, P)$ and all $\mathrm{Cv}(p_i, P)$ in the particle swarm P are calculated by

$$\mathrm{mean}_{\mathrm{Cd}} = \sum_{i=1}^{N} \mathrm{Cd}(p_i, P)/N \text{ and } \mathrm{mean}_{\mathrm{Cv}} = \sum_{i=1}^{N} \mathrm{Cv}(p_i, P)/N.$$

With $\mathrm{mean}_{\mathrm{Cv}}$, it is easy to find out the particles that are close to the ideal point. If we take a 2-D MOP in Fig. 2 as an example, the objective space can be easily split into two parts: the close area and the distant area, by comparing the convergence distance $\mathrm{Cv}(p_i, P)$ with $\mathrm{mean}_{\mathrm{Cv}}$. The close area is highlighted with light gray background, while the distant area is outside the close area, as illustrated in Fig. 2(a). The solutions in the close area are generally approaching to the ideal point z^*. Moreover, when using $\mathrm{mean}_{\mathrm{Cd}}$, it can further distinguish some relatively less crowded solutions and some relatively more crowded solutions, as shown in Fig. 2(b).

However, the above approach may miss some boundary solutions in the true PFs with long tails, as illustrated in Fig. 2. This problem is aggravated in solving MaOPs, since the number of boundary solutions is exponentially increased. Thus, two distance metrics are further introduced to fix the above problem. Let us assume that L is a line connecting the nadir point $z^{\mathrm{nad}} = (1, 1, \ldots, 1)$ and the ideal point $z^* = (0, 0, \ldots, 0)$, and $f'(p_i^*)$ is the projection of $f'(p_i)$ on the L. Then, let $d_1(p_i)$ be the projection distance of $f'(p_i^*)$ and z^*, and $d_2(p_i)$ be the perpendicular distance of $f'(p_i^*)$ and L. They are defined as follows:

$$d_1(p_i) = \frac{\big(f'(p_i) - z^*\big) \cdot \big(z^{\mathrm{nad}} - z^*\big)}{\big\| z^{\mathrm{nad}} - z^* \big\|} \tag{8}$$

$$d_2(p_i) = \sqrt{(f'(p_i) - z^*) \cdot (f'(p_i) - z^*) - d_1(x)^2}. \tag{9}$$

To further clarify the computation of $d_1(p_i)$ and $d_2(p_i)$, an example for a 2-D MOP is further given in Fig. 3. Using the above distance information of $d_1(p_i)$ and $d_2(p_i)$, their average values (mean_{d1} and mean_{d2}) among the particle swarm P can be calculated, by

$$\mathrm{mean}_{d1} = \sum_{i=1}^{N} d_1(p_i) \Big/ N \text{ and } \mathrm{mean}_{d2} = \sum_{i=1}^{N} d_2(p_i) \Big/ N.$$

The core idea of BFE is to use a weighted sum approach for the convergence distance and the diversity distance, as defined in (1). However, different particles should be assigned with various weight factors (α and β), in order to promote some potentially superior solutions and remove some poorly converged ones with good diversity. Therefore, using the above distance information, i.e., mean_{Cv}, mean_{Cd}, mean_{d1}, and mean_{d2}, different situations for the particles are considered in order to properly adjust the setting of α and β for each particle, aiming to properly balance convergence and diversity when tackling MaOPs. Two main categories are first classified using mean_{Cv}.

Case 1 (The Particles p_i With $Cv(p_i, P) < mean_{Cv}$): In this category, all the normalized particles are closer to the ideal point z^*. To further distinguish their distances from the ideal point z^*, mean_{d1} is used to produce two cases.

Case 1.1 (The Particles p_i With $d_1(p_i) < mean_{d1}$): As the normalized particles are very close to the ideal point, β is all set to 1.0 without any punishment on the convergence distance. mean_{Cd} is further adopted to classify their crowded information.

Case 1.1.1 (The Particles p_i With $Cd(p_i, P) < mean_{Cd}$): These normalized particles are close to the ideal point, but they are more crowded. Some of them need to be promoted, while others are punished. Thus, α is set to a random value in (0.6, 1.3) to randomly promote or punish their diversity distances.

Case 1.1.2 (The Particles p_i With $Cd(p_i, P) > mean_{Cd}$): These normalized particles are both close to the ideal point and less crowded. Thus, α is set to 1.0 without any punishment on the diversity distance.

Case 1.2 (The Particles p_i With $d_1(p_i) \geq mean_{d1}$): These normalized particles are a little far away from the ideal point when compared to the ones in case 1.1. Thus, β is set to 0.9 to slightly punish the convergence distance. Moreover, mean_{Cd} is further adopted to classify their crowded information.

Case 1.2.1 (The Particles p_i With $Cd(p_i, P) < mean_{Cd}$): Due to the more crowded status, α is further reduced to 0.6 to punish the diversity distance in this case.

Case 1.2.2 (The Particles p_i With $Cd(p_i, P) > mean_{Cd}$): Due to the less crowded status, α is set to 0.9 to slightly punish the diversity distance in this case.

Case 2 (The Particles p_i With $Cv(p_i, P) \geq mean_{Cv}$): These particles are generally far away from the true PF; only some boundary solutions may probably approach the true PF. In such case, $d_1(p_i)$ and $d_2(p_i)$ are adopted to find out these boundary solutions. The following cases are considered.

Case 2.1 (The Particles p_i With $d_1(p_i) < mean_{Cv}$ and $d_2(p_i) \geq mean_{d2}$): Although these normalized particles are far away from the ideal point, they may be close to the boundaries of the normalized PF. Thus, most of them should be promoted. mean_{Cd} is further used to classify their crowded information.

Case 2.1.1 (The Particles p_i With $Cd(p_i, P) < mean_{Cd}$): Since these normalized particles are more crowded, some of them are promoted, while others are punished. Thus, α and β are all set to a random value within the range (0.6, 1.3) to randomly promote or punish their diversity and convergence distances.

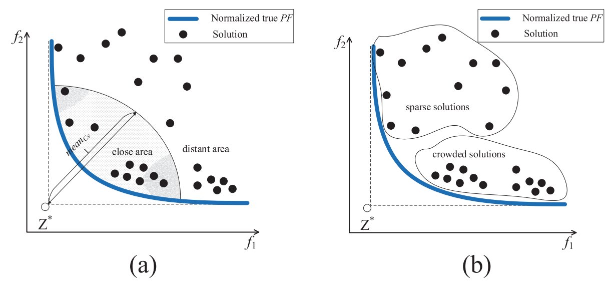


Fig. 2. Advantages of using the (a) convergence distance and (b) diversity distance in the proposed BFE approach.

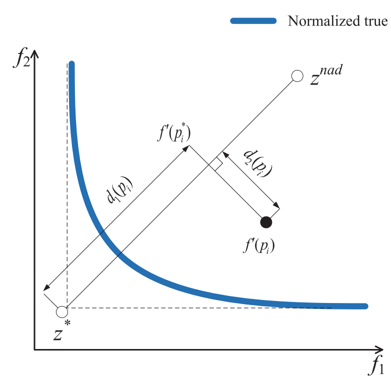


Fig. 3. Illustration of the projection distance and the perpendicular distance regarding the line L.

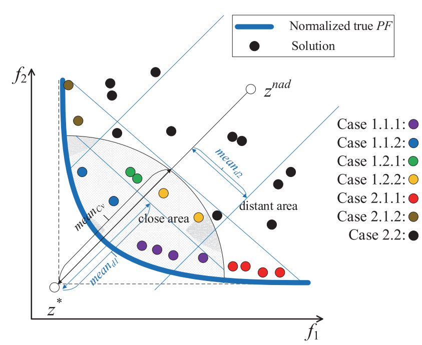


Fig. 4. Example with various cases for adjusting the values of α and β.

Case 2.1.2 (The Particles p_i With $Cd(p_i, P) \geq mean_{Cd}$): These particles are less crowded and they should be promoted. Thus, α and β are all set to 1.0 without any punishment on the diversity and convergence distances.

Case 2.2 (The Particles p_i With $d_1(p_i) \geq mean_{d1}$ or $d_2(p_i) < mean_{d2}$): These normalized particles are far away from the ideal point and they are all surrounded around the line L. Thus, they are generally worse than the solutions in case 1. These particles are restrained by setting α and β to 0.2.

An example with a 2-D MOP is shown in Fig. 4 to illustrate the above cases. It is worth noting that our approach also works for MaOPs and was experimentally validated in Section IV; however, due to the difficulty to illustrate a high dimensional MOP, only the simple examples with 2-D are plotted in Figs. 2–4.

B. Novel Velocity Update Equation

Originally, the velocity and the position of the particles are usually updated using the positional information of the personal- and global-best particles. To provide another search direction (i.e., the evolutionary direction from the personal-best particle pointing to the global-best one) and make more disturbances, a novel velocity update equation was designed in this paper, as follows:

$$v_i(t+1) = wv_i(t) + c_1r_1(x_{p\mathrm{best}_i} - x_i(t)) + c_2r_2(x_{g\mathrm{best}_i} - x_i(t)) + c_3r_3(x_{g\mathrm{best}_i} - x_{p\mathrm{best}_i}) \quad (10)$$

where t is the iteration number; w is the inertial weight; c_1, c_2, and c_3 are three learning factors; r_1, r_2, and r_3 are three uniformly distributed random numbers in [0, 1]; and $x_{p\mathrm{best}_i}$ and $x_{g\mathrm{best}_i}$ are the positional information of the personal-best particle and the global-best particle for p_i, respectively. It is noted that $x_{g\mathrm{best}_i}$ for p_i is randomly selected from the top 10% individuals in the external archive with better BFE values (this top 10% was determined after numerous of experiments). Then, the positional information of p_i is updated using its velocity, as follows:

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (11)$$

This embedded evolutionary direction helps to guide the particles to search toward the global-best particles. Thus, it is expected to enhance the convergence speed of NMPSO.

C. Evolutionary Search on the External Archive

After the PSO-based search takes place, some generated nondominated solutions are maintained in the external archive A using the proposed BFE method. They generally have better quality on convergence and diversity, and are regarded as swarm leaders to guide the PSO-based search. To further enhance the solution quality in the external archive, another search pattern, i.e., evolutionary search, is further conducted on these solutions. This approach is expected to repair the potential insufficiency of PSO-based search on some MaOPs, so as to effectively generate the nondominated solutions [42]. Recent studies indicate that this sort of hybrid scheme can enhance the search ability and make an algorithm more robust to tackle various kinds of MOPs [22], [56]. In NMPSO, the evolutionary operators, i.e., SBX and PM, are used, as they are widely adopted to solve MOPs and provide a promising search capability [3], [7], [41]. This evolutionary search strategy was reported by us in MMOPSO [42] for MOPs, and is extended for MaOPs in this paper. Due to page limitations, please refer to [42] for implementing this operator.

D. Archive Update

After performing the PSO-based search or the evolutionary search on the external archive, a population of new solutions is produced. In order to keep a number of elitist solutions in the external archive, an appropriate selection mechanism is required to update the external archive, so that the search direction can be effectively guided to approximate the true PF. Different from traditional MOPSOs, this paper exploits the proposed BFE method to further select the nondominated solutions in the external archive. Assuming that the set of new solutions is $\boldsymbol{S}$ and the external archive is $\boldsymbol{A}$, the pseudo-code for updating the archive is shown in Algorithm 1, where the input is $\boldsymbol{A}$ and $\boldsymbol{S}$, N_e is the maximum size of $\boldsymbol{A}$. In Algorithm 1, the function **CheckDominance**(A, B) returns the Pareto dominance relationship between A and B. If A dominates B, this function returns 1. Otherwise, it returns -1 when B dominates or is equal to A. At last, this operator will return the updated external archive $\boldsymbol{A}$.

Algorithm 1 Archive_Update (A, $\boldsymbol{S}$)

```
for i=1 to |S|
    for j=1 to |A|
        flag = CheckDominance(S_i, A_j);
        if flag == 1
            mark A_j as a dominated solution;
        else if flag == −1
            mark S_i as a dominated solution; break;
        end if
    end for
    delete the marked dominated solutions from A;
    if S_i is not marked as a dominated solution
        add S_i to A;
        if |A| > N_e
            compute the fitness values using (1)-(7);
            delete the one with the worst fitness value;
        end if
    end if
end for
return A
```

E. Complete Algorithm of NMPSO

The above sections have introduced the main components of NMPSO, which include the assignment of fitness values, velocity update equation, evolutionary search on the external archive, and the archive update operator. In order to describe the remaining operations and to facilitate the implementation of NMPSO, the pseudo-code of its complete algorithm is provided in Algorithm 2. The initialization procedure is first activated in lines 1–6 of Algorithm 2. The external archive $\boldsymbol{A}$ is initialized to be null and the iteration number t is set to 0. For each particle in $\boldsymbol{P}$, its positional information is randomly generated and its velocity is set to 0. After that, the objectives of each particle are evaluated. In line 7, all nondominated solutions in P are added into $\boldsymbol{A}$ and their fitness values are computed using (1)–(7) in line 8. Then, NMPSO steps into the main loop of the evolutionary process until the maximum number of iterations is reached.

In the main loop, the PSO-based search is performed first in lines 10–17. For each particle in P, its velocity and position are, respectively, updated by (10) and (11). After that, its objective values are evaluated, and the personal-best particle $p\mathrm{best}_i$ will be replaced when it does not dominate the new particle p_i. After the PSO-based search, the archive update procedure is executed in line 18, with the input $\boldsymbol{A}$ and $\boldsymbol{P}$. The pseudo-code of this operation can be found in Algorithm 1. Then, in line 19, the evolutionary search

Algorithm 2 Complete Algorithm NMPSO

1: initialize $t = 0$, $\boldsymbol{A}$ = null, and $P = \{p_1, p_2, \ldots p_N\}$;
2: **for** i=1 to N
3: randomly initialize position x_i and set $v_i = 0$ for p_i;
4: evaluate the objective values of p_i;
5: set $pbest_i = p_i$ as the personal-best position for p_i;
6: **end for**
7: add the non-dominated solutions from $\boldsymbol{P}$ into $\boldsymbol{A}$;
8: compute the fitness values for the solutions of $\boldsymbol{A}$ using (1)-(7);
9: **while** $t<T$
10: **for** i=1 to N
11: update the velocity v_i of p_i by (10);
12: update the position x_i of p_i by (11);
13: evaluate the objective values for p_i;
14: **if** $pbest_i$ cannot dominate p_i
15: set $pbest_i = p_i$;
16: **end if**
17: **end for**
18: $\boldsymbol{A}$=**archive_update(*A*,*P*)**;
19: evolutionary search strategy is applied on $\boldsymbol{A}$ to get a new swarm $\boldsymbol{S}$;
20: evaluate the objectives of new solutions in $\boldsymbol{S}$;
21: $\boldsymbol{A}$=**archive_update(*A*,*S*)**;
22: $t = t$+2;
23: **end while**
24: **output A**;

strategy [42] is further run on $\boldsymbol{A}$ to evolve the swarm leaders, providing another search pattern to cooperate with the PSO-based search. At last, the objectives of the newly generated solutions are computed in line 20, and then the archive update process (Algorithm 1) is applied again in line 21. The iteration counter t is increased by 2 in line 22 as both PSO-based search and evolutionary search are run. The above evolutionary phase is repeated until the preset maximum number of iterations is achieved. At last, the final solutions in $\boldsymbol{A}$ are reported as the final approximation of PF.

IV. Experimental Results

A. Benchmark Problems

In this paper, the DTLZ [43] and WFG [44] test problems were used, including DTLZ1–DTLZ6 and WFG1–WFG9. For each problem, the number of objectives was varied from 4 to 10, i.e., $m \in \{4, 6, 8, 10\}$. For DTLZ1–DTLZ6, the number of decision variables was set as $n = m + k - 1$, where n and m are, respectively, the number of decision variables and the number of objectives. As suggested in [43], the values of k were set to 5 for DTLZ1 and 10 for DTLZ2–DTLZ6. Regarding WFG1–WFG9, the decision variables are composed by k position-related parameters and l distance-related parameters. As recommended in [34], k is set to $2 \times (m-1)$ and l is set to 20. The main characteristics of the DTLZ and WFG test problems are summarized in Table A.I, in the supplementary file.[1]

[1]Due to space limitations, it is provided in the supplementary file.

TABLE I
Parameters Settings of All the Algorithms Compared

Algorithm	Parameters settings
SMPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$, $p_m = 1/n$, $\eta_m = 20$
dMOPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$
D²MOPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$
MMOPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$, $p_c = 0.9$, $p_m = 1/n$, $\eta_c = \eta_m = 20$
SPEA2-SDE	$p_c = 0.9$, $p_m = 1/n$, $\eta_c = 20$, $\eta_m = 20$
NSGA-III	$p_c = 1.0$, $p_m = 1/n$, $\eta_c = 30$, $\eta_m = 30$
MOEA/DD	$p_m = 1/n$, $\eta_c = 20$, $\eta_m = 20$, T=20, $\delta = 0.9$, $n_r = 2$
SRA	$p_c = 1.0$, $p_m = 0.1$, $\eta_c = 15$, $\eta_m = 15$, $p'_c = 0.4$
NMPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2, c_3 \in [1.5, 2.5]$, $p_m = 1/n$, $\eta_m = 20$

B. Performance Measures

The goals of MaOPs include the minimization of the distance of the solutions to the true PF (i.e., convergence) and the maximization of the uniform and spread distribution of solutions over the true PF (i.e., diversity). In this paper, inverted generational distance (IGD) [57] and HV [30] are adopted as the performance measures to assess convergence and diversity simultaneously. They are widely used in performance assessment for MaOPs [17]–[19]. Due to page limitations, please refer to [57] and [30] for details of IGD and HV. A smaller value of IGD and a larger value of HV indicate a better approximation of the true PF. It is noted that these two measures show high consistencies on convex PFs and certain contradictions on concave PFs as pointed out in [58].

As it is difficult to find uniformly distributed points in the true PFs of the WFG test problems, their IGD results were not provided in this paper. For the DTLZ test problems with m objectives, using the systematic approach developed in [59], the smallest possible value of H (H is the number of divisions in each objective) was found to make sure that $C_{H+m-1}^{m-1} \geq 200\,000$. Thus, at least 200 000 reference points were uniformly sampled from their true PFs and were used to compute IGD.

For computing the HV indicator, as suggested in [34], all the objectives are normalized using the vector $1.1 \times (f_1 \max, f_2 \max, \ldots, f_m \max)$, where $f_k \max$ $(k = 1, 2, \ldots, m)$ is the maximum value of the kth objective in the true PFs. Then, the reference point is set to $(1.0, 1.0, \ldots, 1.0)$ in order to compute HV. The recently proposed WFG algorithm [60] was used to compute the exact HV values. Note that the solutions that cannot dominate the reference point are not considered for computing HV.

C. Experimental Settings of All Compared Algorithms

In this paper, four MOPSOs (dMOPSO [45], SMPSO [46], D²MOPSO [47], and MMOPSO [42]), and four competitive MOEAs (SPEA2-SDE [17], NSGA-III [18], MOEA/DD [19], and SRA [34]), were used for performance comparison.

Due to the use of weight vectors, the population sizes of dMOPSO, D²MOPSO, MMOPSO, MOEA/DD, and SRA have to be same as the number of weight vectors. The population size of NSGA-III is slightly larger than the number of weight vectors as it has to keep the population size as a multiple of four. For the test problems with 4, 6, 8, and 10 objectives, the number of weight vectors are, respectively, set to 165, 252,

TABLE II
COMPARISON OF RESULTS OF NMPSO AND FOUR CURRENT MOPSOS ON DTLZ1–DTLZ6 USING HV

Problem	Obj	SMPSO	dMOPSO	D^{2}MOPSO	MMOPSO	NMPSO
DTLZ1	4	0.87112(8.35E-03)–	0.85891(1.59E-02)–	0.33849(3.82E-01)–	0.52503(4.07E-01)–	**0.93395(2.46E-03)**
	6	0.80783(1.88E-01)–	0.79468(7.26E-02)–	0.33610(3.36E-01)–	0.03923(9.88E-02)–	**0.98546(1.42E-03)**
	8	0.38105(3.62E-01)–	0.80518(6.53E-02)–	0.34003(3.53E-01)–	0.00001(8.20E-05)–	**0.99550(6.49E-04)**
	10	0.15327(2.13E-01)–	0.86621(2.63E-02)–	0.07738(1.45E-01)–	0.00000(0.00E+00)–	**0.99730(5.67E-04)**
DTLZ2	4	0.53785(2.24E-02)–	0.62180(1.71E-03)–	0.64826(4.98E-03)–	0.63686(5.37E-03)–	**0.71559(1.10E-03)**
	6	0.21131(4.38E-02)–	0.78214(1.26E-02)–	0.62044(2.86E-02)–	0.44369(6.90E-02)–	**0.87599(1.71E-03)**
	8	0.07322(2.70E-02)–	0.72540(1.45E-02)–	0.50662(3.63E-02)–	0.05344(4.27E-02)–	**0.93969(1.33E-03)**
	10	0.00591(8.50E-03)–	0.60867(1.39E-02)–	0.35002(5.09E-02)–	0.00264(9.10E-03)–	**0.96595(2.13E-03)**
DTLZ3	4	0.44853(6.14E-02)–	0.57035(3.61E-02)–	0.00000(0.00E+00)–	0.04925(1.51E-01)–	**0.71553(1.31E-03)**
	6	0.01338(3.61E-02)–	0.35392(9.39E-02)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.87539(1.46E-03)**
	8	0.00196(1.07E-02)–	0.31699(8.75E-02)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.94089(1.38E-03)**
	10	0.00195(1.07E-02)–	0.41953(5.56E-02)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.96347(2.05E-03)**
DTLZ4	4	0.63271(1.45E-02)–	0.55920(1.72E-02)–	0.66312(4.16E-03)–	0.63915(4.39E-03)–	**0.71615(1.30E-03)**
	6	0.60743(4.12E-02)–	0.61739(3.43E-02)–	0.76677(1.27E-02)–	0.51928(6.00E-02)–	**0.87529(1.13E-03)**
	8	0.35145(5.84E-02)–	0.58234(3.31E-02)–	0.81564(1.57E-02)–	0.10210(8.36E-02)–	**0.94394(9.80E-04)**
	10	0.10324(5.99E-02)–	0.49283(3.91E-02)–	0.82999(1.61E-02)–	0.01433(2.81E-02)–	**0.96842(1.29E-03)**
DTLZ5	4	0.14590(7.02E-04)–	0.14494(5.44E-05)–	0.14833(2.23E-03)–	0.14629(8.12E-04)–	**0.15062(2.03E-04)**
	6	0.11366(4.97E-04)+	0.10572(6.14E-05)–	**0.11768(1.15E-04)+**	0.11392(3.95E-04)+	0.11075(3.26E-03)
	8	0.10347(3.57E-04)+	0.09424(1.86E-05)+	**0.10620(7.72E-05)+**	0.10391(2.53E-04)+	0.08840(2.16E-02)
	10	0.09784(3.41E-04)+	0.09814(5.86E-05)+	**0.10076(3.26E-05)+**	0.09820(2.62E-04)+	0.09046(1.49E-02)
DTLZ6	4	0.14248(1.30E-03)–	0.14494(5.49E-06)+	0.14483(2.54E-03)–	0.01362(2.20E-02)–	**0.15038(3.54E-04)**
	6	0.10529(1.33E-03)–	0.10570(3.51E-06)–	0.11291(1.32E-03)~	0.00059(3.23E-03)–	**0.11306(1.07E-03)**
	8	0.09735(8.36E-04)–	0.09423(1.96E-06)–	**0.10197(8.43E-04)+**	0.00260(1.42E-02)–	0.09805(1.37E-03)
	10	0.09279(6.89E-04)~	0.09818(5.64E-06)+	**0.09834(6.55E-04)+**	0.00303(1.66E-02)–	0.09249(9.47E-04)
Best/All		0/24	0/24	5/24	0/24	19/24
Better/Worse/Similar		3/20/1	4/20/0	5/18/1	3/21/0	——

330, and 275, using the approach introduced in [19]. Thus, NSGA-III adopts population sizes of 166, 254, 332, and 276 for 4-, 6-, 8-, and 10-objective problems, respectively, while the other algorithms adopt population sizes and archive sizes which are the same as the number of weight vectors.

To allow a fair comparison, the related parameters of all the compared algorithms were set as suggested in their references, as summarized in Table I. p_c and p_m are, respectively, the crossover probability and the mutation probability; η_c and η_m are, respectively, the distribution indexes of SBX and PM. For dMOPSO, SMPSO, D^{2}MOPSO, MMOPSO, and NMPSO, their control parameters c_1, c_2, and c_3 were randomly sampled in [1.5, 2.5] and the inertia weight ω was randomly produced from [0.1, 0.5]. Regarding MOEA/DD, T defines the neighborhood size among the weight coefficients; δ indicates the probability to select parent solutions from T neighbors; and n_r is the maximum number of parent solutions that are replaced by each child solution. p_c' in SRA is an inherent parameter of stochastic ranking, which controls the balance between two indicators. Moreover, the parameter to control the execution of two velocity update equations is set to 0.9 for MMOPSO. The maximum number of function evaluations (FEs) was set to 10^5 for all the test problems adopted. All the experiments were run 30 times with different random seeds, and their mean HV and IGD values and the standard deviations (included in brackets after the mean results) in 30 runs were collected for comparison.

Moreover, to obtain a statistically sound conclusion, Wilcoxon's rank sum test was further performed in order to show the statistically significant differences between the results obtained by NMPSO and those obtained by other algorithms with a significance level $\alpha = 0.05$. In the following tables, the symbols "+," "$-$," and "$\sim$" indicate that the results obtained by other algorithms are significantly better than, worse than, and similar to the ones obtained by NMPSO using this statistical test, respectively.

D. Comparisons of NMPSO With Four Current MOPSOs

1) Comparison Results on DTLZ1–DTLZ6: Table II provides the HV comparison results of NMPSO with respect to four current MOPSOs (SMPSO, dMOPSO, D^{2}MOPSO, and MMOPSO) on DTLZ1–DTLZ6 with 4–10 objectives. As observed from Table II, NMPSO obtained the best results on 19 out of 24 comparisons, which validated the superior performance of NMPSO. Only D^{2}MOPSO performed best on five comparisons, while SMPSO, dMOPSO, and MMOPSO were not best on any DTLZ test problem. These results are summarized in the second last row of Table II.

It was also found that SMPSO, D^{2}MOPSO, and MMOPSO all performed significantly worse on DTLZ1–DTLZ4 regarding HV, especially when using a high number of objectives (i.e., ten objectives). This is mainly because they all use Pareto-ranking and crowding distance to update the external archive, leading them to favor the solutions that are less crowded but poorly converged. The pure decomposition approach adopted in dMOPSO can help to mitigate the above problem, but it cannot provide sufficient selection pressure toward the true PFs. NMPSO performed much better on these problems as the BFE method strongly prefers the solutions that are well converged and less crowded. For DTLZ5 and DTLZ6 with degenerate PFs, only MMOPSO performed significantly worse on DTLZ6 with different numbers of objectives, while other MOPSOs performed competitively on these two problems. To make a one-to-one comparison, NMPSO performed better than SMPSO, dMOPSO, D^{2}MOPSO, and MMOPSO on 20, 20, 18, and 21 out of 24 comparisons, but it was only, respectively, worse than SMPSO, dMOPSO,

TABLE III
COMPARISON OF RESULTS OF NMPSO AND FOUR CURRENT MOPSOS ON ALL THE WFG TEST PROBLEMS USING HV

Problem	Obj	SMPSO	dMOPSO	D^{2}MOPSO	MMOPSO	NMPSO
WFG1	4	0.31576(1.14E-03)–	0.26635(6.13E-03)–	0.31950(4.30E-03)–	0.39336(1.62E-02)–	**0.52338(4.68E-02)**
	6	0.28075(1.13E-03)–	0.12624(7.25E-02)–	0.28017(1.28E-03)–	0.31132(8.27E-03)~	**0.31310(2.87E-02)**
	8	0.25085(1.28E-03)–	0.17239(5.50E-02)–	0.25083(4.46E-03)–	0.27036(5.16E-03)–	**0.30105(2.48E-02)**
	10	0.22844(8.30E-04)–	0.23433(6.93E-03)–	0.22868(2.14E-03)–	0.25032(5.16E-03)–	**0.35450(2.85E-02)**
WFG2	4	0.88570(9.30E-03)–	0.89872(1.25E-02)–	0.93764(8.38E-03)+	**0.94506(7.75E-03)+**	0.90990(2.09E-02)
	6	0.85438(1.26E-02)–	0.89341(2.73E-02)–	**0.98044(5.73E-03)+**	0.97353(8.34E-03)+	0.95129(1.36E-02)
	8	0.82910(1.27E-02)–	0.88682(3.48E-02)–	**0.98490(6.23E-03)+**	0.97006(1.29E-02)+	0.95685(1.10E-02)
	10	0.81552(1.62E-02)–	0.91891(2.30E-02)–	**0.98364(1.10E-02)+**	0.96302(8.65E-03)–	0.97190(7.46E-03)
WFG3	4	0.57295(9.03E-03)–	0.61074(5.78E-03)–	0.58840(2.06E-02)–	0.63232(7.58E-03)–	**0.64159(7.46E-03)**
	6	0.57858(4.33E-03)–	0.51596(8.86E-03)–	0.44868(1.17E-02)–	0.61953(1.29E-02)–	**0.62875(1.93E-02)**
	8	0.58232(7.05E-03)–	0.50064(1.16E-02)–	0.44462(1.06E-02)–	**0.61338(1.57E-02)+**	0.59836(2.04E-02)
	10	0.57938(7.22E-03)–	0.59233(1.65E-02)~	0.43639(1.06E-02)–	**0.61220(1.85E-02)~**	0.59713(3.15E-02)
WFG4	4	0.49905(1.23E-02)–	0.50238(1.50E-02)–	0.54863(1.70E-02)–	0.60441(1.04E-02)–	**0.68391(4.57E-03)**
	6	0.51041(1.67E-02)–	0.48342(7.19E-02)–	0.60493(2.16E-02)–	0.56324(2.63E-02)–	**0.80582(5.81E-03)**
	8	0.54610(1.37E-02)–	0.43862(5.73E-02)–	0.59826(4.34E-02)–	0.53171(2.01E-02)–	**0.84608(6.52E-03)**
	10	0.54348(1.54E-02)–	0.52419(5.89E-02)–	0.57271(4.10E-02)–	0.51800(2.03E-02)–	**0.86091(1.12E-02)**
WFG5	4	0.42291(2.08E-02)–	0.44871(1.17E-02)–	0.58159(1.71E-02)–	0.55815(1.12E-02)–	**0.65076(3.18E-03)**
	6	0.44352(8.98E-03)–	0.44550(3.61E-02)–	0.63662(3.52E-02)–	0.52345(1.94E-02)–	**0.78738(3.95E-03)**
	8	0.47011(9.58E-03)–	0.41874(6.60E-02)–	0.62917(3.37E-02)–	0.48677(2.07E-02)–	**0.82642(2.12E-02)**
	10	0.46125(1.02E-02)–	0.42312(5.53E-02)–	0.56291(4.55E-02)–	0.46894(2.24E-02)–	**0.81452(7.04E-02)**
WFG6	4	0.57315(1.55E-02)–	0.48437(6.66E-03)–	0.52866(2.89E-02)–	0.59429(1.34E-02)–	**0.67962(1.93E-02)**
	6	0.58982(7.32E-03)–	0.58591(3.19E-02)–	0.54956(3.70E-02)–	0.62134(4.42E-02)–	**0.77341(1.33E-03)**
	8	0.66852(7.46E-03)–	0.52409(2.82E-02)–	0.52397(5.00E-02)–	0.53279(6.66E-02)–	**0.83212(1.45E-03)**
	10	0.70282(5.49E-03)–	0.46106(6.07E-02)–	0.48522(5.83E-02)–	0.47034(2.84E-02)–	**0.84099(3.52E-03)**
WFG7	4	0.43846(1.28E-02)–	0.44268(1.85E-02)–	0.57321(1.02E-02)–	0.62853(1.38E-02)–	**0.70792(1.84E-03)**
	6	0.44976(1.07E-02)–	0.33475(2.26E-02)–	0.62033(2.06E-02)–	0.57569(4.24E-02)–	**0.85530(2.93E-03)**
	8	0.47750(9.80E-03)–	0.32102(3.14E-02)–	0.64829(2.52E-02)–	0.51140(2.29E-02)–	**0.90840(3.18E-03)**
	10	0.48140(1.30E-02)–	0.36077(3.43E-02)–	0.59008(7.33E-02)–	0.48357(1.76E-02)–	**0.91978(3.73E-02)**
WFG8	4	0.35209(1.13E-02)–	0.30493(1.49E-02)–	0.41031(2.23E-02)–	0.48334(9.22E-03)–	**0.60192(2.72E-03)**
	6	0.38456(9.40E-03)–	0.25108(3.16E-02)–	0.32909(3.22E-02)–	0.47464(1.77E-02)–	**0.71651(6.05E-03)**
	8	0.41921(1.03E-02)–	0.26339(2.69E-02)–	0.34397(2.12E-02)–	0.49009(1.44E-02)–	**0.77505(1.35E-02)**
	10	0.42315(1.06E-02)–	0.30059(2.50E-02)–	0.32889(2.43E-02)–	0.48133(1.89E-02)–	**0.82269(2.01E-02)**
WFG9	4	0.47574(1.21E-02)–	0.42512(1.69E-02)–	0.56727(4.03E-02)–	0.53991(1.99E-02)–	**0.66352(5.74E-03)**
	6	0.44639(1.31E-02)–	0.44239(2.84E-02)–	0.56911(5.04E-02)–	0.53473(2.23E-02)–	**0.74228(3.09E-02)**
	8	0.47213(1.15E-02)–	0.42943(4.64E-02)–	0.52285(4.14E-02)–	0.48925(3.60E-02)–	**0.76002(1.01E-02)**
	10	0.47187(1.37E-02)–	0.45585(4.47E-02)–	0.46274(4.79E-02)–	0.44355(1.58E-02)–	**0.81563(2.04E-02)**
Best/All		0/36	0/36	3/36	3/36	30/36
Better/Worse/Similar		0/36/0	0/35/1	4/32/0	4/30/2	——

D^{2}MOPSO, and MMOPSO on 3, 4, 5, and 3 comparisons. These results were summarized in the last row of Table II, where "*Better/Worse/Similar*" indicates the number of test problems, in which the performance of the compared algorithm was better than, worse than or similar to that of NMPSO. Thus, when considering DTLZ1–DTLZ6 with 4–10 objectives, NMPSO showed a superior performance over the four current MOPSOs regarding HV.

In Table A.II,[2] in the supplementary file, the comparative results of NMPSO and four current MOPSOs are listed for DTLZ1–DTLZ6 with 4–10 objectives using IGD. Similar to the conclusions that are observed from the HV comparison results, NMPSO also performed best on most of 24 comparisons and significantly outperformed other compared MOPSOs on DTLZ1–DTLZ4. Based on the summary on the second last row and the last row of Table A.II, in the supplementary file, it was further confirmed using IGD that NMPSO performed best on most of the DTLZ test problems when compared to four current MOPSOs.

2) Comparison of Results on WFG1–WFG9: Table III provides the comparison of results of NMPSO and four current MOPSOs on all the WFG test problems with 4–10 objectives, using HV. From the second last row of Table III, it was found that NMPSO performed best on 30 out of 36 comparisons and this was very evident to justify the superior performance of NMPSO over other compared MOPSOs on the WFG test problems. Only D^{2}MOPSO and MMOPSO gave the best results on 3 out of 36 comparisons, while SMPSO and dMOPSO could not perform best on any WFG problem.

Regarding WFG1, which has a convex and mixed PF, NMPSO showed a better performance than the other compared MOPSOs. For WFG2, which has a disconnected and mixed PF, NMPSO only obtained the third rank, as it was outperformed by D^{2}MOPSO and MMOPSO, while it outperformed SMPSO and dMOPSO. Concerning WFG3 with linear and uni-modal PF, NMPSO was best on WFG3 with four and six objectives, while MMOPSO performed best on WFG3 with eight and ten objectives. For WFG4–WFG9 with concave PFs, NMPSO showed a superior performance over the other compared MOPSOs. As observed from the one-to-one comparisons in the last row of Table III, SMPSO and dMOPSO were unable to beat NMPSO on any of the WFG test problems; only D^{2}MOPSO and MMOPSO gave a better performance than NMPSO in 4 out of 36 comparisons. For the rest of the comparisons on the WFG test problems with different numbers of objectives, NMPSO performed best or at least similarly regarding HV. Due to the more difficult features and different scaled values in the true PFs of the WFG

[2]Due to space limitations, it is provided in the supplementary file.

TABLE IV
COMPARISON OF RESULTS OF NMPSO AND FOUR COMPETITIVE MOEAS ON DTLZ1–DTLZ6 USING HV

Problem	Obj	SPEA2/SDE	MOEA/DD	NSGA-III	SRA	NMPSO
DTLZ1	4	0.92637(3.77E-03)–	**0.94479(1.40E-04)+**	0.94447(3.94E-04)+	0.92765(3.63E-03)~	0.93395(2.46E-03)
	6	0.97587(2.68E-03)–	**0.99218(6.83E-05)+**	0.99130(2.72E-03)+	0.98165(2.24E-03)~	0.98546(1.42E-03)
	8	0.98804(1.46E-03)–	0.99836(1.36E-04)+	**0.99837(5.44E-04)+**	0.99143(1.52E-03)–	0.99550(6.49E-04)
	10	0.98828(1.75E-03)–	0.99698(4.11E-04)–	**0.99944(1.32E-04)+**	0.99165(1.59E-03)–	0.99730(5.67E-04)
DTLZ2	4	0.71072(1.25E-03)–	0.71525(1.12E-05)–	0.71498(1.84E-04)–	0.67848(3.46E-03)–	**0.71559(1.10E-03)**
	6	0.86799(1.98E-03)–	0.87524(4.90E-05)–	0.87219(3.48E-04)–	0.82319(3.93E-03)–	**0.87599(1.71E-03)**
	8	0.93509(1.50E-03)–	**0.94542(8.29E-05)+**	0.93900(8.59E-04)–	0.88231(4.33E-03)–	0.93969(1.33E-03)
	10	0.96429(1.52E-03)–	**0.96887(1.40E-04)+**	0.96341(7.45E-04)–	0.89041(6.78E-03)–	0.96595(2.13E-03)
DTLZ3	4	0.70774(3.65E-03)–	0.71025(4.17E-03)–	0.69637(1.43E-02)–	0.66303(2.32E-02)–	**0.71553(1.31E-03)**
	6	0.86115(4.08E-03)–	0.86617(5.03E-03)–	0.69106(2.84E-01)–	0.80647(1.89E-02)–	**0.87539(1.46E-03)**
	8	0.92084(7.76E-03)–	0.91744(6.42E-02)–	0.51569(3.58E-01)–	0.87154(2.07E-02)–	**0.94089(1.38E-03)**
	10	0.95344(5.66E-03)–	0.96100(3.87E-03)–	0.78669(2.90E-01)–	0.87668(1.87E-02)–	**0.96347(2.05E-03)**
DTLZ4	4	0.68919(5.27E-02)–	0.71528(8.01E-06)~	0.71518(8.29E-05)–	0.65811(7.33E-02)–	**0.71615(1.30E-03)**
	6	0.87179(1.78E-03)–	**0.87554(5.72E-05)+**	0.87553(1.87E-04)+	0.84474(2.65E-03)–	0.87529(1.13E-03)
	8	0.93972(1.12E-03)–	**0.94610(6.22E-05)+**	0.94358(5.06E-04)+	0.91777(3.09E-03)–	0.94394(9.80E-04)
	10	0.96388(2.38E-03)–	0.96977(4.77E-05)+	**0.96985(2.24E-04)+**	0.94006(2.99E-03)–	0.96842(1.29E-03)
DTLZ5	4	0.14205(1.39E-03)–	0.12704(3.33E-03)–	0.13552(4.53E-03)–	0.12453(4.03E-03)–	**0.15062(2.03E-04)**
	6	0.09711(3.38E-03)–	0.09524(6.93E-03)–	0.01462(1.35E-02)–	0.03780(1.57E-02)–	**0.11075(3.26E-03)**
	8	0.08108(4.44E-03)–	0.06662(1.58E-02)–	0.00447(1.50E-02)–	0.01085(1.25E-02)–	**0.08840(2.16E-02)**
	10	0.08158(5.91E-03)–	0.06660(1.21E-02)–	0.00410(1.13E-02)–	0.00608(1.10E-02)–	**0.09046(1.49E-02)**
DTLZ6	4	0.13295(9.98E-03)–	0.00000(0.00E+00)–	0.03224(1.89E-02)–	0.02239(5.91E-03)–	**0.15038(3.54E-04)**
	6	0.00810(3.62E-03)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.11306(1.07E-03)**
	8	0.00004(3.98E-05)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.09805(1.37E-03)**
	10	0.00001(2.75E-05)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	0.00000(0.00E+00)–	**0.09249(9.47E-04)**
Best/All		0/24	6/24	3/24	0/24	15/24
Better/Worse/Similar		0/24/0	8/15/1	7/17/0	0/22/2	——

test problems, the superior performance of NMPSO on these problems further confirmed its advantages over other MOPSOs and the effectiveness of the three proposed operators used in NMPSO

E. Comparisons of NMPSO With Four Competitive MOEAs

In the above sections, it was experimentally validated that NMPSO performed better than four current MOPSOs on most of the DTLZ and WFG test problems. However, these compared MOPSOs were not specifically designed to tackle MaOPs, and it may be not so challenging for NMPSO to outperform them. To further justify the advantages of NMPSO, it was compared to four competitive MOEAs, i.e., SPEA2-SDE [17], MOEA/DD [19], NSGA-III [18], and SRA [34].

1) Comparison Results on DTLZ1–DTLZ6: Table IV lists the comparison of results of NMPSO with four competitive MOEAs on DTLZ1–DTLZ6 with 4–10 objectives, using HV. As observed from the second last row of Table IV, NMPSO obtained the best results on 15 out of 24 comparisons, while MOEA/DD and NSGA-III, respectively, performed best on six and three comparisons. SPEA2-SDE and SRA could not perform best on any of the DTLZ test problems. Although these compared MOEAs are all redesigned to tackle MaOPs, NMPSO still achieved the best performance on more than half of 24 comparisons. These results further justified the advantages of NMPSO.

For DTLZ1 and DTLZ4, NMPSO performed better than SPEA2/SDE and SRA, but worse than MOEA/DD and NSGA-III. Regarding DTLZ2, NMPSO had a similar performance as MOEA/DD, and it was better than the other algorithms. For the rest of the DTLZ test problems, NMPSO showed a significantly better performance. Particularly, for DTLZ5 and DTLZ6 with degenerate PFs, MOEA/DD, NSGA-III, and SRA all performed poorly, as they used reference points to guide the evolutionary search and these reference points cannot uniformly match the degenerate PFs of DTLZ5 and DTLZ6. SPEA2/SDE had a better performance on DTLZ5 and DTLZ6, due to the use of the SDE method, but it was still worse than NMPSO, as our proposed BFE approach can provide a stronger selection pressure than the SDE method. As indicated by the one-to-one comparisons in the last row of Table IV, NMPSO, respectively, performed better than SPEA2/SDE, MOEA/DD, NSGA-III, and SRA on 24, 15, 17, and 22 out of 24 comparisons; whereas, NMPSO was only beaten by SPEA2/SDE, MOEA/DD, NSGA-III, and SRA on 0, 8, 7, and 0 comparisons, respectively. Therefore, when compared to these four MOEAs, NMPSO still showed a superior performance on most of the DTLZ test problems, regarding HV.

In Table A.III,[3] in the supplementary file, the comparison of results of NMPSO and four competitive MOEAs are listed for DTLZ1–DTLZ6 using IGD. The advantages of NMPSO were not so evident as observed from the HV comparison of results. SPEA2/SDE and MOEA/DD, respectively, gave the best performance on 5 and 8 out of 24 comparisons, while NMPSO only performed best on 9 out of 24 comparisons. From the one-to-one comparisons in the last row of Table A.III, in the supplementary file, NMPSO still performed better than the compared MOEAs as it performed better on most of the comparisons using DTLZ1–DTLZ6 with 4–10 objectives.

2) Comparison of Results on WFG1–WFG9: Table V provided the comparison of results of NMPSO and four competitive MOEAs on WFG1–WFG9 with 4–10 objectives using HV. As SPEA2/SDE, MOEA/DD, NSGA-III, and SRA are specifically redesigned to solve MaOPs, they

[3]Due to space limitations, it is provided in the supplementary file.

TABLE V
COMPARISON OF RESULTS OF NMPSO AND FOUR COMPETITIVE MOEAS ON WFG1–WFG9 USING HV

Problem	Obj	SPEA2/SDE	MOEA/DD	NSGA-III	SRA	NMPSO
WFG1	4	**0.65143(2.74E-02)+**	0.51368(2.79E-02)~	0.43286(1.78E-02)–	0.50900(3.93E-02)~	0.52338(4.68E-02)
	6	**0.60434(1.77E-02)+**	0.42223(3.17E-02)+	0.31471(1.28E-02)~	0.55975(2.32E-02)+	0.31310(2.87E-02)
	8	0.55220(1.86E-02)+	0.46695(2.83E-02)+	0.26840(3.31E-03)–	**0.55713(1.88E-02)+**	0.30105(2.48E-02)
	10	0.65586(1.71E-02)+	0.21728(1.83E-02)–	0.25086(9.36E-03)–	**0.66445(3.02E-02)+**	0.35450(2.85E-02)
WFG2	4	0.94238(6.35E-02)+	0.93301(5.65E-02)+	**0.94581(6.42E-02)+**	0.92642(4.14E-02)+	0.90990(2.09E-02)
	6	**0.95979(5.23E-02)+**	0.93687(2.84E-02)–	0.95272(5.13E-02)+	0.93878(4.18E-02)~	0.95129(1.36E-02)
	8	**0.97331(3.17E-02)+**	0.91781(2.91E-02)–	0.96453(3.21E-02)+	0.95742(7.38E-03)~	0.95685(1.10E-02)
	10	**0.97351(3.15E-02)+**	0.89845(2.05E-02)–	0.94013(6.91E-02)~	0.94468(4.10E-02)–	0.97190(7.46E-03)
WFG3	4	0.59096(1.38E-02)–	0.58279(9.17E-03)–	0.61316(7.48E-03)–	0.57766(1.20E-02)–	**0.64159(7.46E-03)**
	6	0.55960(1.99E-02)–	0.53002(1.34E-02)–	0.59198(8.80E-03)–	0.55007(1.52E-02)–	**0.62875(1.93E-02)**
	8	0.55431(1.80E-02)–	0.48280(1.06E-02)–	**0.62912(5.21E-03)+**	0.54755(1.56E-02)–	0.59836(2.04E-02)
	10	0.54560(2.09E-02)–	0.44924(1.00E-02)–	**0.62146(1.35E-02)+**	0.51375(1.51E-02)–	0.59713(3.15E-02)
WFG4	4	0.66718(3.60E-03)–	0.67761(2.67E-03)–	0.67839(3.87E-03)–	0.59355(7.34E-03)–	**0.68391(4.57E-03)**
	6	0.75318(6.93E-03)–	0.78002(6.26E-03)–	0.77915(6.23E-03)–	0.68045(1.25E-02)–	**0.80582(5.81E-03)**
	8	0.77802(8.74E-03)–	0.75855(1.90E-02)–	0.80935(7.16E-03)–	0.69552(1.13E-02)–	**0.84608(6.52E-03)**
	10	0.77130(6.32E-03)–	0.58753(3.00E-02)–	0.77864(1.87E-02)–	0.66495(1.65E-02)–	**0.86091(1.12E-02)**
WFG5	4	0.63077(3.29E-03)–	0.63450(1.96E-03)–	0.64966(2.80E-03)~	0.55280(7.34E-03)–	**0.65076(3.18E-03)**
	6	0.72647(5.31E-03)–	0.74107(2.78E-03)–	0.76937(3.52E-03)–	0.62870(1.31E-02)–	**0.78738(3.95E-03)**
	8	0.74001(8.62E-03)–	0.72923(1.17E-02)–	0.80113(4.70E-03)–	0.63074(1.84E-02)–	**0.82642(2.12E-02)**
	10	0.73887(9.34E-03)–	0.57475(2.74E-02)–	0.79803(1.06E-02)–	0.64139(2.58E-02)–	**0.81452(7.04E-02)**
WFG6	4	0.64677(5.12E-03)–	0.64310(6.40E-03)–	0.65311(6.37E-03)–	0.55670(1.46E-02)–	**0.67962(1.93E-02)**
	6	0.74256(5.87E-03)–	0.76130(9.75E-03)–	**0.77516(6.62E-03)~**	0.61882(1.20E-02)–	0.77341(1.33E-03)
	8	0.75288(8.22E-03)–	0.77939(1.26E-02)–	0.81660(1.20E-02)–	0.62233(1.79E-02)–	**0.83212(1.45E-03)**
	10	0.75606(1.52E-02)–	0.65932(2.11E-02)–	0.82710(1.00E-02)–	0.60421(2.66E-02)–	**0.84099(3.52E-03)**
WFG7	4	0.68503(3.37E-03)–	0.68890(1.06E-03)–	0.69819(1.66E-03)–	0.60671(9.88E-03)–	**0.70792(1.84E-03)**
	6	0.79354(4.80E-03)–	0.81015(4.11E-03)–	0.82118(6.88E-03)–	0.69191(8.73E-03)–	**0.85530(2.93E-03)**
	8	0.82817(6.03E-03)–	0.82636(9.19E-03)–	0.85661(8.27E-03)–	0.71407(1.74E-02)–	**0.90840(3.18E-03)**
	10	0.82361(9.47E-03)–	0.68170(2.59E-02)–	0.84829(9.74E-03)–	0.67854(2.61E-02)–	**0.91978(3.73E-02)**
WFG8	4	0.59254(2.80E-03)–	0.58559(2.85E-03)–	0.59304(3.77E-03)–	0.49760(8.89E-03)–	**0.60192(2.72E-03)**
	6	0.68863(4.54E-03)–	0.67691(1.39E-02)–	0.67374(7.31E-03)–	0.54864(1.06E-02)–	**0.71651(6.05E-03)**
	8	0.72513(6.52E-03)–	0.69091(2.59E-02)–	0.68192(1.51E-02)–	0.55550(1.49E-02)–	**0.77505(1.35E-02)**
	10	0.73895(9.09E-03)–	0.57145(2.45E-02)–	0.70996(1.82E-02)–	0.53447(1.96E-02)–	**0.82269(2.01E-02)**
WFG9	4	0.61530(2.59E-02)–	0.59102(2.04E-02)–	0.59086(1.42E-02)–	0.57022(1.16E-02)–	**0.66352(5.74E-03)**
	6	0.71834(2.22E-02)–	0.66347(1.76E-02)–	0.69495(1.32E-02)–	0.64949(1.16E-02)–	**0.74228(3.09E-02)**
	8	0.74802(9.44E-03)–	0.65233(2.45E-02)–	0.73441(1.73E-02)–	0.66342(1.71E-02)–	**0.76002(1.01E-02)**
	10	0.73542(1.78E-02)–	0.47674(4.16E-02)–	0.69967(2.07E-02)–	0.64793(2.15E-02)–	**0.81563(2.04E-02)**
Best/All		5/36	0/36	4/36	2/36	25/36
Better/Worse/Similar		8/28/0	3/32/1	5/27/4	4/29/3	——

showed a better performance than the compared MOPSOs in Table III. However, NMPSO still outperformed these competitive MOEAs on most comparisons using the WFG test problems. As observed from the last second row of Table V, NMPSO obtained the best results on 25 out of 36 comparisons, while SPEA2/SDE, NSGA-III, and SRA, respectively, performed best on 5, 4, and 2 comparisons. MOEA/DD was not able to perform best on any of the WFG test problems. Since these compared MOEAs were validated to perform well on MaOPs, the superior performance of NMPSO over these algorithms further confirmed the advantages of NMPSO.

For WFG1, SPEA2/SDE and SRA seemed more advantageous; NMPSO obtained the third rank on this problem as it performed better than MOEA/DD and NSGA-III. Regarding WFG2, SPEA2/SDE and NSGA-III showed the best performance, while NMPSO still obtained the third rank on DTLZ2, as it outperformed MOEA/DD and SRA. For the rest of the comparisons on WFG3–WFG9, it was evident that NMPSO performed best on most of the comparisons. Moreover, from the last row of Table V, NMPSO was found to outperform SPEA2/SDE, MOEA/DD, NSGA-III, and SRA on 28, 32, 27, and 29 out of 36 comparisons, respectively; whereas, NMPSO was only beaten by SPEA2/SDE, MOEA/DD, NSGA-III, and SRA on 8, 3, 5, and 4 comparisons, respectively. Therefore, it is reasonable to conclude that NMPSO performed better than SPEA2/SDE, MOEA/DD, NSGA-III, and SRA, on most of the HV comparisons for the WFG problems. This superiority of NMPSO was mainly produced by the proposed BFE method, which strengthens the selection pressure toward the true PFs, by properly balancing the convergence and diversity distances.

To further study the evolutionary behaviors of all compared algorithms, Fig. A.1[3] plotted their evolutionary curves using their average HV values on all the 10-objective WFG test problems. The average HV results are recorded periodically at each 1000-FEs in all the 10^5 FEs. The subfigures in Fig. A.1, in the supplementary file, confirm the advantages of NMPSO in providing a strong selection pressure toward the true PFs for 10-objective WFG4–WFG9 problems. Only for 10-objective WFG1 problem, NMPSO was shown to perform significantly worse than SPEA2/SDE and SRA.

F. Further Discussion and Analysis on the Performance

Based on the above comparison of results on the DTLZ and WFG test problems, it is interesting to find out that, the decomposition and reference point based MOEAs, such as MOEA/DD and NSGA-III, performed competitively on DTLZ1–DTLZ4. This observation is consistent with that observed from [18] and [19]. It is mainly because a set of distributed weight vectors or reference points used in decomposition or reference points based MOEAs can properly match the true PFs of DTLZ1–DTLZ4 due to the regularity of their PFs (being either a hyperplane or a hypersphere), which can correctly guide the search and diversify the obtained solutions evenly over the true PFs. Moreover, these MOEAs also adopt specific mechanisms to keep a good balance between diversity and convergence. For example, the PBI approach used in MOEA/DD simultaneously measures diversity and convergence of a solution. The nondominated sorting method and a well-designed niching procedure in NSGA-III work collaboratively to strengthen the selection pressure. However, regarding the other test problems, such as DTLZ5, DTLZ6, and WFG, their results are not so promising since the predefined weight vectors or reference points cannot properly match their PFs. Therefore, the performance of decomposition or reference point based MOEAs strongly depends on the shapes of PFs as pointed out in [61]. It is worth emphasizing that our proposed NMPSO is not based on decomposition nor on reference points, but it is mainly guided by the proposed BFE method to strengthen the selection pressure toward the true PFs. Thus, the performance of NMPSO is not significantly affected by the shapes of the PFs, and it also performs relatively well on the DTLZ and WFG test problems.

To further support the above discussion, a recent reference-based MOEA (RVEA [36]) was also included to compare with NMPSO. Its parameters settings follow those indicated in [36], also with the same population size and maximum FEs as introduced in Section IV-C. The source code of RVEA is provided by the authors and the experimental results of RVEA and NMPSO are listed in Table A.IV, in the supplementary file[3]. It is noted that, due to the larger number of decision variables used in the WFG test problems, the results of RVEA on these problems are different from those presented in [36]. From Table A.IV, in the supplementary file, it is clear that RVEA also performed competitively with NMPSO on DTLZ1–DTLZ4, as their true PFs can be properly matched by the reference points in RVEA. However, on the WFG test problems with true PFs that cannot be properly matched by the reference points, RVEA was outperformed by NMPSO on most comparisons.

TABLE VI
COMPARISON OF RESULTS OF NMPSO AND THREE NMPSO VARIANTS ON DTLZ1–DTLZ6 AND WFG1–WFG9 USING HV

Problem	Obj	NMPSO-I	NMPSO-II	NMPSO-III	NMPSO
DTLZ1	4	0.90135– 1.13E-04	**0.93404~** **1.48E-03**	0.93074 – 1.28E-03	0.93395 2.46E-03
	10	0.99739~ 5.99E-05	0.98182– 9.74E-03	**0.99763+** **3.75E-04**	0.99730 5.67E-04
DTLZ2	4	0.60044– 4.08E-03	0.69318– 4.02E-03	0.71435– 1.96E-03	**0.71559** **1.10E-03**
	10	0.87332– 3.69E-03	0.90822– 7.28E-03	0.96051– 5.65E-03	**0.96595** **2.13E-03**
DTLZ3	4	0.60013– 5.38E-03	0.69169– 4.97E-02	0.71444– 1.45E-03	**0.71553** **1.31E-03**
	10	0.86390– 1.08E-02	0.75762– 6.28E-02	0.96188~ 3.69E-03	**0.96347** **2.05E-03**
DTLZ4	4	0.60076– 4.33E-03	0.65808– 2.71E-02	0.71531– 1.46E-03	**0.71615** **1.30E-03**
	10	0.87900– 3.30E-03	0.63043– 2.16E-02	0.96267– 4.87E-03	**0.96842** **1.29E-03**
DTLZ5	4	0.09426– 2.94E-03	0.14951– 5.85E-04	0.14825– 3.96E-03	**0.15062** **2.03E-04**
	10	0.09091+ 3.79E-10	**0.09095+** **2.31E-04**	0.04915– 3.32E-02	0.09046 1.49E-02
DTLZ6	4	0.09326– 1.76E-03	0.14546~ 2.75E-02	**0.15054+** **2.59E-04**	0.15038 3.54E-04
	10	0.09091– 5.51E-06	0.09091– 2.09E-06	**0.09334+** **8.75E-04**	0.09249 9.47E-04
WFG1	4	**0.67899+** **3.16E-02**	0.31155– 4.02E-03	0.51099~ 3.26E-02	0.52338 4.68E-02
	10	**0.57343+** **2.51E-02**	0.22949– 2.33E-03	0.35623~ 2.00E-02	0.35450 2.85E-02
WFG2	4	**0.96124+** **7.25E-03**	0.85073– 1.33E-02	0.91376~ 1.14E-02	0.90990 2.09E-02
	10	**0.98563+** **3.91E-03**	0.88240– 1.90E-02	0.97773+ 7.52E-03	0.97190 7.46E-03
WFG3	4	**0.65397+** **6.54E-03**	0.60975– 1.02E-02	0.65005+ 6.12E-03	0.64159 7.46E-03
	10	**0.68056+** **1.63E-02**	0.60291~ 2.75E-02	0.57778~ 4.42E-02	0.59713 3.15E-02
WFG4	4	0.42772– 4.04E-02	0.64645– 4.72E-03	**0.68852+** **4.78E-03**	0.68391 4.57E-03
	10	0.68630– 3.81E-02	0.76935– 2.62E-02	0.85612– 1.25E-02	**0.86091** **1.12E-02**
WFG5	4	0.47973– 2.26E-02	0.60830– 9.28E-03	**0.65245~** **4.00E-03**	0.65076 3.18E-03
	10	0.58443– 2.42E-02	0.61798– 2.87E-02	**0.82647~** **3.94E-02**	0.81452 7.04E-02
WFG6	4	0.49185– 2.61E-02	0.63589– 1.01E-02	0.66352– 2.83E-02	**0.67962** **1.93E-02**
	10	0.70638– 2.62E-02	**0.84583+** **3.71E-03**	0.84075~ 3.46E-03	0.84099 3.52E-03
WFG7	4	0.48193– 3.22E-02	0.56678– 8.00E-03	0.70707~ 1.46E-03	**0.70792** **1.84E-03**
	10	0.75482– 3.80E-02	0.63185– 4.16E-02	0.91600~ 3.17E-02	**0.91978** **3.73E-02**
WFG8	4	0.43538– 2.87E-02	0.48388– 1.13E-02	**0.60300~** **2.51E-03**	0.60192 2.72E-03
	10	0.67815– 3.26E-02	0.60757– 3.11E-02	0.81522~ 2.73E-02	**0.82269** **2.01E-02**
WFG9	4	0.39874– 3.15E-02	0.60874– 1.45E-02	0.65780– 5.53E-03	**0.66352** **5.74E-03**
	10	0.49258– 2.51E-02	0.75873– 2.42E-02	0.78337~ 2.64E-02	**0.81563** **2.04E-02**
Best/All		6/30	3/30	7/30	14/30
Better/Worse/Similar		7/22/1	2/25/3	6/11/13	——

G. *Effectiveness of the Proposed Operators*

The above comparisons have validated the superior performance of NMPSO over four current MOPSOs and four competitive MOEAs, on the DTLZ and WFG test problems with 4–10 objectives. As introduced in Section I, three novel operators, i.e., the BFE method, evolutionary search on the external archive, and a novel velocity update equation, were employed in NMPSO to enhance its performance. To study their respective contributions, NMPSO was compared to three variants of NMPSO, named NMPSO-I, NMPSO-II, and NMPSO-III. In NMPSO-I, the proposed BFE method was replaced by the SDE method proposed in [17]. Regarding NMPSO-II, evolutionary search on the external archive was removed from NMPSO, only performing the pure PSO-based search. For NMPSO-III, the proposed velocity update equation in (10) was replaced by the original one in [46]. In order to allow a fair comparison, all the parameters settings were the same in NMPSO and its three variants (NMPSO-I, NMPSO-II, and NMPSO-III).

Table VI shows the comparison of results of NMPSO and three NMPSO variants on DTLZ1–DTLZ6 and WFG1–WFG9 with 4 and 10 objectives, using HV. In this table, the mean results are listed above the standard deviations obtained in 30 runs. Based on the second last row of Table VI, NMPSO performed best on 14 out of 30 comparisons, while NMPSO-I, NMPSO-II, and NMPSO-III, respectively, obtained the best performance on 6, 3, and 7 comparisons. These experimental results validate that all the three operators contribute to enhancing the performance of NMPSO. To study their respective contributions, each variant was separately compared to NMPSO below.

Regarding the comparison of NMPSO with NMPSO-I, NMPSO performed better on 22 comparisons, similarly on one comparison, and worse on seven comparisons as revealed in the last row of Table VI. The effectiveness of the BFE method was justified experimentally, as it helps to enhance the performance of NMPSO on most of the 30 comparisons performed. The BFE method was very effective in improving the performance of NMPSO on DTLZ1–DTLZ6 and WFG4–WFG9 with 4 and 10 objectives. These experimental results validated that the proposed BFE method can provide a stronger selection pressure than the SDE method, when solving most of the test problems adopted. Also, the BFE method was the main contribution to the enhancement of NMPSO, as NMPSO-II and NMPSO-III with the BFE method all showed a better performance than NMPSO-I on most of the comparisons shown in Table VI.

When compared to NMPSO-II, NMPSO was better on 25 comparisons, similar on three comparisons, and worse on two comparisons. These results corroborated the advantages of performing evolutionary search on the external archive. In other words, the absence of evolutionary search on the external archive will lower the performance of NMPSO on most of the test problems adopted, which justifies the statement that the evolutionary search is able to cooperate with the PSO-based search and to overcome the inefficiency of PSO-based search on some MaOPs. By further comparing NMPSO-II to NMPSO-I and NMPSO-III, it was also found that this operator was the second main contribution to the enhanced performance of NMPSO.

Finally, the effectiveness of our novel velocity update equation was also verified, as NMPSO, respectively, performed better than, worse than, and similarly to NMPSO-III on 11, 6, and 13 out of 30 comparisons. This modified velocity update equation can slightly enhance the performance of NMPSO by introducing one evolutionary direction pointing to the global-best particles. Since the global-best particles are randomly selected from the solutions with better BFE values in the external archive, the embedded evolutionary direction provides more disturbances and guides the particles to search toward the global-best particles, so as to enhance the convergence speed of NMPSO.

Therefore, based on the above experimental results, it is reasonable to conclude that all three novel operators contribute to enhancing the overall performance of NMPSO in solving the DTLZ and WFG test problems with 4 and 10 objectives.

H. More Discussions About the BFE Method

Due to page limitations, further discussions were provided in the supplementary file of this paper, in order to evaluate the effectiveness of the BFE method in other state-of-the-art algorithms and to analyze the impact of α and β in the BFE method.

V. Conclusion

In this paper, NMPSO, an algorithm with a BFE method was designed to tackle MaOPs. This BFE method combines the convergence and diversity distances, which helps to relieve the curse of dimensionality in MaOPs and guides all the particles to approach the true PFs of MaOPs. Two more operators were also used in NMPSO. One is the evolutionary search on the external archive, which can provide another search pattern and is aimed to overcome the ineffectiveness of PSO-based search on certain types of MaOPs; the other is a novel velocity update equation used to provide another search direction for PSO-based search and to induce more diversity. These proposed novel operators cooperate to enhance the overall performance of NMPSO in tackling MaOPs. The DTLZ and WFG test problems with 4–10 objectives were adopted in our experimental comparisons to validate the performance of NMPSO. When compared to four current MOPSOs (i.e., dMOPSO [45], SMPSO [46], D^2MOPSO [47], and MMOPSO [42]), and four competitive MOEAs (i.e., SPEA2-SDE [17], NSGA-III [18], MOEA/DD [19], and SRA [34]), the experimental results clearly showed the superior performance of NMPSO in obtaining a well-approximated and well-distributed solution set for MaOPs. The effectiveness of our three novel operators (the BFE method, evolutionary search on the external archive and a novel velocity update equation) was also experimentally verified in NMPSO. More discussions about the BFE method were provided in the supplementary file. The BFE method was embedded into other multiobjective algorithms (i.e., NSGA-II [3] and NNIA [62]), which revealed that the BFE method was more effective on MaOPs with a higher number of objectives and more suitable to cooperate with the PSO-based search. The parameter sensitivity analysis of BFE was also conducted by comparing our dynamic settings with different predefined fixed settings of α and β.

This behavior of the BFE method will be further studied in our future work. Its performance on NMPSO will be further studied to tackle MaOPs with more than ten objectives, and the BFE method will be investigated by embedding it into other kinds of MOEAs. Moreover, the application of NMPSO to some real-world problems is also considered as part of our future work.

References

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

[2] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY, USA: Springer, 2007.

[3] 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, Apr. 2002.

[4] V. A. Shim, K. C. Tan, and H. J. Tang, "Adaptive memetic computing for evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 610–621, Apr. 2015.

[5] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[6] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, Feb. 2015.

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

[8] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. 2nd Int. Conf. Evol. Multi Criterion Optim. (EMO)*, Faro, Portugal, 2003, pp. 376–390.

[9] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 400–413, Jun. 2015.

[10] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 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, Apr. 2015.

[12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[13] F. di Pierro, S.-T. Khu, and D. A. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, Feb. 2007.

[14] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[15] 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, Feb. 2016.

[16] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.

[17] M. Li, S. Yang, and 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, Jun. 2014.

[18] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[19] 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, Oct. 2015.

[20] 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, Dec. 2007.

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

[22] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.

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

[24] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, Feb. 2016.

[25] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 145–160, Feb. 2016.

[26] 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, Jun. 2015.

[27] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving From Nature—PPSN VIII* (LNCS 3242). Heidelberg, Germany: Springer, 2004, pp. 832–842.

[28] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[29] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, Mar. 2007.

[30] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

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

[32] R. 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, Feb. 2012.

[33] K. Bringmann and T. Friedrich, "An efficient algorithm for computing hypervolume contributions," *Evol. Comput.*, vol. 18, no. 3, pp. 383–402, Sep. 2010.

[34] 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, Mar. 2016.

[35] 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, Aug. 2015.

[36] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016, doi: 10.1109/TEVC.2016.2519378.

[37] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.

[38] 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, Aug. 2013.

[39] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, to be published, doi: 10.1109/TEVC.2016.2600642.

[40] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2600577.

[41] Z. Zhan *et al.*, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[42] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming, "A novel multi-objective particle swarm optimization with multiple search strategies," *Eur. J. Oper. Res.*, vol. 247, no. 3, pp. 732–744, 2015.

[43] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization* (Advanced Information and Knowledge Processing), A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[44] S. Huband, L. Barone, R. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. 3rd Conf. Evol. Multi Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 280–295.

[45] S. Z. Martínez and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proc. 13th Annu. Conf. Genet. Evol. Comput.*, Dublin, Ireland, 2011, pp. 69–76.

[46] A. J. Nebro *et al.*, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis. Making*, Nashville, TN, USA, 2009, pp. 66–73.

[47] N. Al Moubayed, A. Pertovski, and J. McCall, "D^2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, Mar. 2014.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[49] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, Jan. 2015.

[50] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.

[51] J. Moore, R. Chapman, and G. Dozier, "Multiobjective particle swarm optimization," in *Proc. 38th Annu. ACM Southeast Conf.*, Clemson, SC, USA, 2000, pp. 56–57.

[52] R. C. Eberhart and Y. H. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Congr. Evol. Comput.*, vol. 1. Seoul, South Korea, 2001, pp. 81–86.

[53] M. R. Sierra and C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and $\in$-dominance," in *Evolutionary Multi-Criterion Optimization* (LNCS 3410). Heidelberg, Germany: Springer, 2005, pp. 505–519.

[54] W. Peng and Q. Zhang, "A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems," in *Proc. Conf. Granular Comput.*, Hangzhou, China, 2008, pp. 534–537.

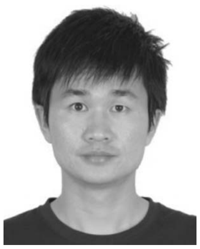
[55] N. Al Moubayed, A. Petrovski, and J. McCall, "A novel smart multi-objective particle swarm optimisation using decomposition," in *Parallel Problem Solving From Nature, PPSN XI* (LNCS 6239). Heidelberg, Germany: Springer, 2010, pp. 1–10.

[56] Q. Z. Lin *et al.*, "A hybrid evolutionary immune algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 711–729, Oct. 2016.

[57] 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, Apr. 2003.

[58] S. W. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.

[59] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
[60] 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, Feb. 2012.
[61] 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.*, to be published, doi: 10.1109/TEVC.2016.2587749.
[62] M. G. Gong, L. C. Jiao, H. F. Du, and L. F. Bo, "Multi-objective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, 2008.



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

He is currently a Lecturer with the College of Computer Science and Software Engineering, Shenzhen University. He has published over 20 research papers since 2008. His current research interests include artificial immune system, multiobjective optimization, and dynamic system.

Songbai Liu received the B.S. degree from Changsha University, Changsha, China, in 2012. He is currently pursuing the M.S. degree with Shenzhen University, Shenzhen, China.

He was a Software Testing Engineer with Shenzhen TVT Digital Technology Company Ltd., Shenzhen, from 2013 to 2015. His current research interests include nature-inspired computation, many-objective optimization, and machine learning.

Qingling Zhu received the B.S. degree from the Nanchang Institution of Technology, Nanchang, China, in 2013, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2016.

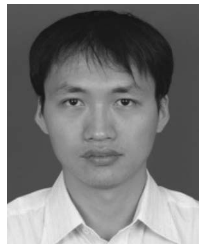
He is currently a Research Assistant with Shenzhen University. His current research interest includes evolutionary multiobjective optimization.

Chaoyu Tang received the B.S. degree from Pingdingshan University, Pingdingshan, China, and the M.S. degree from Shenzhen University, China, in 2013 and 2016, respectively.

Her current research interests include evolutionary computation, many-objective optimization and artificial intelligence.

Ruizhen Song received the B.S. degree from Gannan Normal University, Ganzhou, China, in 2012, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2015.

Her current research interests include evolutionary computation, many-objective optimization, and artificial intelligence.



Jianyong Chen (M'11) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2003.

He is a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He worked for ZTE Corporation as senior engineer of network technology from 2003 to 2006. After that, he joined the Shenzhen University. He was the Vice-Chairman of the International Telecommunication Union-Telecommunication (ITU-T) SG17 from 2004 to 2012, and an Editor of three recommendations developed in ITU-T SG17. His current research interests include artificial intelligence and information security. He has published over 30 papers and holds over 30 patents in the above areas.

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

He is a Professor (CINVESTAV-3F Researcher) with the Department of Computer Science of CINVESTAV-IPN, Mexico City, Mexico. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book entitled *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second edition, Springer, 2007). His publications currently report over 33 000 citations in Google Scholar with an H-index of 70. His current research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Dr. 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 an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves on the editorial board of 12 other international journals. He is a member of ACM, Sigma Xi, and the Mexican Academy of Science.

Ka-Chun Wong received the B.Eng. degree in computer engineering and the M.Phil. degree from the Chinese University of Hong Kong, Hong Kong, in 2008 and 2010, respectively, and the Ph.D. degree from the Department of Computer Science, University of Toronto, Toronto, ON, Canada, in 2015.

He is an Assistant Professor with the City University of Hong Kong. His current research interests include bioinformatics, computational biology, evolutionary computation, data mining, machine learning, and interdisciplinary research.

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

He joined Sun Yat-sen University since 2004 and became a Changjiang Chair Professor since 2012. Since 2016, he is a Changjiang Chair Professor with the South China University of Technology, Guangzhou, China. His current research interests include computational intelligence, cloud computing, wireless sensor networks, operations research, and power electronic circuits. He has authored seven research books and book chapters, and over 50 IEEE TRANSACTIONS papers in his research areas.

Prof. Zhang was a recipient of 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 INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and Current Chair of the IEEE Guangzhou Subsection and the IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapters, and ACM Guangzhou Chapter.