A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization

Xingyi Zhang, Ye Tian, and Yaochu Jin, *Senior Member, IEEE*

***Abstract*—Evolutionary algorithms (EAs) have shown to be promising in solving many-objective optimization problems (MaOPs), where the performance of these algorithms heavily depends on whether solutions that can accelerate convergence toward the Pareto front and maintaining a high degree of diversity will be selected from a set of nondominated solutions. In this paper, we propose a knee point-driven EA to solve MaOPs. Our basic idea is that knee points are naturally most preferred among nondominated solutions if no explicit user preferences are given. A bias toward the knee points in the nondominated solutions in the current population is shown to be an approximation of a bias toward a large hypervolume, thereby enhancing the convergence performance in many-objective optimization. In addition, as at most one solution will be identified as a knee point inside the neighborhood of each solution in the nondominated front, no additional diversity maintenance mechanisms need to be introduced in the proposed algorithm, considerably reducing the computational complexity compared to many existing multiobjective EAs for many-objective optimization. Experimental results on 16 test problems demonstrate the competitiveness of the proposed algorithm in terms of both solution quality and computational efficiency.**

***Index Terms*—Convergence, diversity, evolutionary multiobjective optimization, hypervolume (HV), knee point, many-objective optimization.**

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs) are commonly seen in real-world applications, especially in the areas of engineering, biology, and economics [1]–[5]. Such optimization problems are characterized by multiple objectives that conflict with each other. Due to the conflicting nature of the objectives, usually no single optimal solution exists; instead, a set of trade-off solutions, known as Pareto optimal solutions can be found for MOPs. Over the past two decades, evolutionary algorithms (EAs) and other population-based meta-heuristics have been demonstrated to be a powerful framework for solving MOPs, since they can find a set of Pareto optimal solutions in a single run. A large number of multiobjective evolutionary algorithms (MOEAs) have been developed, e.g., NSGA-II [6], SPEA2 [7], IBEA [8], MOEA/D [9], PESA-II [10], and M-PAES [11], just to name a few. In all these MOEAs, a variety of selection strategies have been proposed to achieve fast convergence and high diversity, which play the most important role in determining the effectiveness and efficiency of the MOEA in obtaining the Pareto optimal solutions.

Among various selection strategies, the Pareto-based nondominated sorting approaches are the most popular, where solutions having a better Pareto rank in the parent population or a combination of the parent and offspring populations are selected. In addition to the dominance-based criterion, a secondary criterion, often a diversity-related strategy, will be adopted to achieve an even distribution of the Pareto optimal solutions. NSGA-II [6] and SPEA2 [7] are two representative Pareto-based MOEAs, which have been shown to be very effective in solving MOPs having two or three objectives. However, the efficiency of such Pareto-based MOEAs will seriously degrade when the number of objectives is more than three, which are often known as many-objective optimization problems (MaOPs).

MaOPs are widely seen in real-world applications (see [12] and [13]). Increasing research attention has therefore been paid to tackling MaOPs in recent years, as it has been shown that MaOPs cannot be solved efficiently using MOEAs developed for solving MOPs with two or three objectives [14]–[17]. For example, NSGA-II performs very well on MOPs with two or three objectives; however, its performance will dramatically deteriorate when the MOPs have more than three objectives [18]. The main reason for this performance deterioration is that the selection criterion based on the standard dominance relationship fails to distinguish solutions in a population already in the early stage of the search, since most of the solutions in the population are nondominated, although some of them may have a better ability to help the population to converge to the Pareto optimal front [19]. Once the dominance-based selection criterion is not able to distinguish solutions, MOEAs will often rely on a secondary criterion, usually a metric for population diversity. As a result, MOEAs may end up with a set of well-distributed nondominated solutions, which are unfortunately far from Pareto optimal.

To enhance the ability of MOEAs to converge to the Pareto front, a variety of ideas have been proposed, which can be

Manuscript received June 15, 2014; revised September 30, 2014; accepted November 28, 2014. Date of publication December 4, 2014; date of current version November 25, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61272152, 61033003, 61202011, and 61472002; in part by the Natural Science Foundation of Anhui Higher Education Institutions of China under Grant KJ2012A010; in part by the Joint Research Fund for Overseas Chinese, Hong Kong; and in part by the Macao Scholars of the National Natural Science Foundation of China under Grant 61428302.

X. Zhang and Y. Tian are with Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China.

Y. Jin is with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, U.K., and also with the College of Information Sciences and Technology, Donghua University, Shanghai 201620, China (e-mail: yaochu.jin@surrey.ac.uk).

Digital Object Identifier 10.1109/TEVC.2014.2378512

largely divided into three categories [20], [21]. The first group of ideas is to modify the traditional Pareto dominance definition to increase the selection pressure toward the Pareto front. This type of ideas has been widely adopted for solving MaOPs, such as ϵ-dominance [22], [23], L-optimality [24], fuzzy dominance [25], and preference order ranking [26]. Compared with MOEAs using the traditional Pareto dominance relationship, these strategies have been shown to considerably improve the performance of MOEAs for solving MaOPs, although they are very likely to converge into a sub-region of the Pareto front.

The second category of the ideas aims to combine the traditional Pareto dominance-based criterion with additional convergence-related metrics. Based on these ideas, solutions are selected first based on the dominance relationship, and then on the convergence-related metric. For example, some substitute distances based on the degree to which a solution is nearly dominated by any other solutions were proposed by Köppen and Yoshida [27] to improve the performance of NSGA-II. In [28], a binary ϵ-indicator based preference is combined with dominance to speed up convergence of NSGA-II for solving MaOPs. A grid dominance-based metric was also defined by Yang *et al.* [29] based on which an effective MOEA, termed GrEA, for MaOPs has been proposed.

The third type of ideas is to develop new selection criteria based on some performance indicators. Three widely used performance indicator based MOEAs are IBEA [8], SMS-EMOA [30], and HypE [31]. IBEA uses a predefined optimization goal to measure the contribution of each solution, while SMS-EMOA and HypE are based on the hypervolume (HV) value.

There are also a large number of other many-objective optimization algorithms, which adopt different ideas from those discussed above. For example, some researchers attempted to solve MaOPs by using a reduced set of objectives [32], [33], while others suggested to use interactive user preferences [34] or reference points [35] during the search. An interesting MOEA for solving MaOPs, called NSGA-III, was also based on a set of reference points [36], where nondominated solutions close to the reference points are prioritized. Note also that some MOEAs have shown to perform fairly well for some MaOP test problems [37], such as the decomposition-based multiobjective evolutionary algorithm, termed MOEA/D [9], although they are not specifically designed for solving MaOPs.

In multiobjective optimization, knee points are a sub-set of Pareto optimal solutions for which an improvement in one objective will result in a severe degradation in at least another one. For MOPs, solutions in the knee region of the Pareto front will be naturally preferred if no other user-specific or problem-specific preferences are available. As previously discussed, most existing MOEAs do not work efficiently for MaOPs mainly due to the loss of selection pressure because most or all solutions in the population are nondominated already in a very early search stage. In this paper, we propose a knee point-driven evolutionary algorithm (KnEA), in which preferences are given to knee points among the nondominated solutions in selection. In other words, knee points are used as the secondary criterion for selecting parents for the next generation in addition to the nondominance selection criterion. Therefore, the proposed KnEA belongs to the second class of MOEAs discussed above for solving MaOPs. Note, however, that by knee points, we do not mean the knee points of the theoretical (true) Pareto front; instead, we mean the knee points of the nondominated fronts in the current population during the search process. Since at most one knee point is identified in the neighborhood of each solution, a preference over the knee points also promotes diversity of the population, and consequently no additional measures need to be taken in KnEA in environmental selection. Note that calculating a diversity measure such as the crowding distance in NSGA-II can be highly time-consuming for MaOPs.

New contributions of this paper can be summarized as follows.

1) A knee point-driven MOEA has been suggested, where knee points of the nondominated fronts in the current population are preferred in selection. We show that preference over knee points can approximately be seen as a bias toward larger HV, which is therefore very effective in both accelerating the convergence of the population to the Pareto optimal front and maintaining diversity of the solutions. We should stress that a large body of research work has been performed on identifying knee points in solving MOPs, most of which, however, concentrated on how to accurately find the knee points or local knee regions of the true Pareto front. To the best of our knowledge, no work has been reported on using knee points as the secondary criterion to enhance the search performance of MOEAs for MaOPs.
2) Within the KnEA, an adaptive strategy for identifying knee points in a small neighborhood, i.e., local knee regions, has been proposed without using prior knowledge about the number of knee points in the true Pareto front. The purpose of the adaptive strategy is not to find precisely the knee points of the true Pareto front; instead, it is meant to locate those local knee points of the nondominated fronts in the population combining the parent and offspring populations at the present generation to accelerate the convergence and promote diversity.
3) Extensive experimental results have been conducted to verify the performance of KnEA for solving MaOPs by comparing it with several state-of-the-art MOEAs for MaOPs on two suites of widely used test problems. Our results demonstrate that KnEA outperforms the compared MOEAs for MaOPs in terms of two widely used performance indicators. Moreover, KnEA is computationally much more efficient than two of the three compared Pareto-based MOEAs and comparable to the rest one, although it is slightly inferior to MOEA/D, which is known for its high computational efficiency for MaOPs.

The rest of this paper is organized as follows. In Section II, existing work related to the identification of knee points in multiobjective optimization is discussed and the motivation of using knee points as a selection criterion is justified. The details of the proposed KnEA for MaOPs are described in

Section III. Simulation results are presented in Section IV to empirically compare search performance and runtime of the KnEA with four state-of-the-art methods for MaOPs. Finally, the conclusion is given in Section V.

II. Related Work and Motivation

In this section, we first review the related work on finding knee points in evolutionary multiobjective optimization. Then, we elaborate the motivation of using knee points detected during the search for driving the population toward the Pareto optimal front and maintaining population diversity.

A. Related Work

A large number of MOEAs have been proposed to find local regions or points of interest in the Pareto optimal front [38]–[40]. Among various preferences, knee points are often considered to be of interest in the Pareto optimal front and much research work has been dedicated to finding knee points or knee regions (neighboring regions of knee points) using MOEAs.

Intuitively, a knee point in the Pareto optimal front refers to the solution with the maximum marginal rates of return, which means that a small improvement in one objective of such a solution is accompanied by a severe degradation in at least another. As knee points are naturally preferred, several multiobjective optimization algorithms have been developed to find the knee points or knee regions in the Pareto optimal front instead of approximating the whole front. Das [41] suggested a method to find the knee points in the Pareto front based on normal boundary intersection, which has been shown to be very efficient for characterizing knee points. Branke *et al.* [42] proposed two variants of NSGA-II for finding knee regions, where the crowding distance in NSGA-II was substituted by two new measures: 1) an angle-based measure and 2) a utility measure. The variant with the utility measure can be used for problems with any number of objectives, while the one with the angle-based measure works only for bi-objective problems. These two variants of NSGA-II have been shown to perform very well on finding knee regions in the Pareto front, which, however, are not able to control the spread of these regions.

To control the spread of knee regions, Rachmawati and Srinivasan [43], [44] developed an MOEA based on a weighted sum niching method, where the extent and the density of convergence of the knee regions were controlled by the niche strength and the total number of solutions in the region, known as pool size. However, such control on the extent and the density of the knee regions is very rough. Schütze *et al.* [45] presented two strategies for finding knee points and knee regions that can be integrated into any stochastic search algorithms. Experimental results illustrated that these two strategies were very efficient in finding the knee points and knee regions of bi-objective optimization problems. However, these methods are not extendable to MOPs with more than two objectives.

Bechikh *et al.* [46] proposed an MOEA for finding knee regions, termed KR-NSGA-II by extending the reference point based MOEA, called R-NSGA-II [47]. In KR-NSGA-II, the knee points were used as mobile reference points and the search of the algorithm was guided toward these points. KR-NSGA-II has been shown to perform well in controlling the extent of knee regions for MOPs with two or more objectives, assuming that some prior information on the number of knee points of the MOP is known. Deb and Gupta [48] suggested several new definitions for identifying knee points and knee regions for bi-objective optimization problems. The possibility of applying such methods to solve bi-objective engineering problems has also been discussed. Branke *et al.* [42] presented two test problems with knee points, one bi-objective and one three-objective, to evaluate the ability of MOEAs for finding knee points. Tušar and Filipič [49] extended these two test problems to four-objective and five-objective optimization problems.

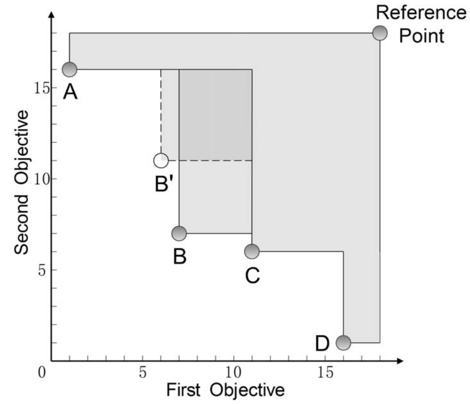

Fig. 1. Illustration of the motivation of KnEA. B can be seen as a knee point among the five nondominated solutions A, B', B, C, and D. Selecting solution B, the knee point can be more beneficial than B' in terms of HV.

Various definitions for characterizing knee points and knee regions have been suggested (see [41]–[44], [48]). In this paper, we adopt the definition presented by Das [41] and Bechikh *et al.* [46], which will be further discussed in Section III-C.

B. Motivation of This Paper

As can be seen in the discussions above, the importance of knee points and knee regions has long been recognized in evolutionary multiobjective optimization. Nevertheless, the use of knee points to improve the search ability of MOEAs, especially for solving MaOPs, has not been reported so far. In this paper, we hypothesize that the search ability of MOEAs for solving MaOPs can be significantly enhanced by giving preferences to the knee points among nondominated solutions.

To elaborate this hypothesis, consider five nondominated solutions of a bi-objective optimization problem, $A(1, 16)$, $B'(6, 11)$, $B(7, 7)$, $C(11, 6)$, and $D(16, 1)$, where the two elements of a solution indicate the values of the two objectives, as shown in Fig. 1. From Fig. 1, we can see that solution B can be considered as a knee point of the nondominated front consisting of the five nondominated solutions. Assume that four solutions are to be selected from the five nondominated solutions for next population. Since these five solutions are all nondominated, a secondary criterion must be used for selecting four out of the five solutions. If a diversity-based criterion,

TABLE I
RELATIONSHIP BETWEEN DISTANCE OF B TO THE EXTREME LINE AD WITH THE HV OF SOLUTION SET CONSISTING OF A, B, C, AND D

Position of B	B(6,7)	B(7,7)	B(74/9,7)	B(10,7)	B(11,7)
Distance to AD	2.83	2.12	1.26	0	−0.71
Hypervolume	159	150	139	123	114

for example, the crowding distance defined in [6] is used for selection, then solutions A, B', C, and D will be selected. If we replace solution B' with the knee point B, the selected solution set will be A, B, C, and D.

Let us now compare the quality of the above two solution sets using the HV, which is one of the most widely used performance indicators in multiobjective optimization [50]. For calculating the HV of the two sets, we assume that the reference point is $(18, 18)$. In this case, the HV of the solution set consisting of A, B', C, and D is 139, while the HV of the solution set consisting of A, B, C, and D is 150.

From the above illustrative example, we can observe that selecting knee points can be more beneficial than selecting more diverse solutions in terms of the HV. To take a closer look at the relationship between the position of point B and the HV of the solution set consisting of A, B, C, and D, we move the position of B from $B(6, 7)$, which is the leftmost possible position, to $B(11, 7)$, which is rightmost possible position to maintain the nondominated relationships between B and the other four solutions. Now, we examine the relationship between the distance of B to the extreme line AD, which is described by $f_1 + f_2 = 17$, and the HV of the solution set consisting of A, B, C, and D on five different positions. The results are listed in Table I.

From Table I, we can see that when B moves from $B(6, 7)$ to $B(7, 7)$, the HV of the solution set consisting of A, B, C, and D decreases from 159 to 150, while the distance to the extreme line decreases from 2.83 to 2.12. When point B further moves to the right to $B(74/9, 7)$, the HV drops to 139, which is equal to the HV of the solution set consisting of A, B', C, and D. In this case, the distance of point B to the extreme line is further reduced to 1.26 and B is no longer a typical knee point. If point B continues to move to $B(10, 7)$, B is exactly located on the extreme line, and the HV of solution set consisting of A, B, C, and D becomes 123, which is even smaller than that of solution set consisting of A, B', C, and D. Therefore, we can conclude that the more typical B is a knee point, the more likely it will contribute to a large HV.

From the above example, we can hypothesize that a preference over knee points can be considered as an approximation of the preference over larger HVs. Compared with the HV-based selection, however, knee point-based selection offers the following two important advantages. First, the identification of knee points is computationally much more efficient than calculating the HV, in particular when the number of objectives is large. To be more specific, the computational time for calculating the HV increases exponentially as the number of objectives increases, while the time for identifying knee points increases only linearly. Second, although the HV implicitly takes diversity into account, it cannot guarantee a good diversity. By contrast, diversity is explicitly embedded in the knee point identification process proposed in this paper, since at most one solution will be labeled as a knee point in the neighborhood of a solution. The above hypothesis has been verified by our empirical results comparing the proposed method with HypE, a HV-based method. Refer to Section IV for more details.

Algorithm 1 General Framework of KnEA

Require: P (population), N (population size), K (set of knee points), T (rate of knee points in population)

$r \leftarrow 1$, $t \leftarrow 0$ /*adaptive parameters for finding knee points*/
$K \leftarrow \emptyset$
$P \leftarrow Initialize(N)$
while termination criterion not fulfilled **do**
  $P' \leftarrow Mating\_selection(P, K, N)$
  $P \leftarrow P \bigcup Variation(P', N)$
  $F \leftarrow Nondominated\_sort(P)$ /*find the solutions in the first i fronts F_j, $1 \le j \le i$, where i is the minimal value such that $|F_1 \cup \ldots \cup F_i| >= N$ */
  $[K, r, t] \leftarrow Finding\_knee\_point(F, T, r, t)$
  $P \leftarrow Environmental\_selection(F, K, N)$
end while
return P

III. PROPOSED ALGORITHM FOR MANY-OBJECTIVE OPTIMIZATION

KnEA is in principle an elitist Pareto-based MOEA. The main difference between KnEA and other Pareto-based MOEAs such as NSGA-II is that knee points are used as a secondary selection criterion in addition to the dominance relationship. During the environmental selection, KnEA does not use any explicit diversity measure to promote the diversity of the selected solution set. In the following, we describe the main components of KnEA.

A. General Framework of the Proposed Algorithm

The general framework of KnEA is similar to that of NSGA-II [6], which consists of the following main components. First, an initial parent population of size N is randomly generated. Second, a binary tournament strategy is applied to select individuals from the parent population to generate N offspring individuals using a variation method. In the binary tournament selection, three tournament metrics are adopted, namely, the dominance relationship, the knee point criterion, and a weighted distance measure. Third, nondominated sorting is performed on the combination of the parent and offspring population, followed by an adaptive strategy to identify solutions located in the knee regions of each nondominated front in the combined population. Fourth, an environmental selection is conducted to select N individuals as the parent population of the next generation. This procedure repeats until a termination condition is met. The above main components of KnEA are presented in Algorithm 1.

Algorithm 2 *Mating_Selection*(P, K, N)

Require: P (population), K (set of knee points), N (population size)

1: $Q \leftarrow \emptyset$
2: **while** $|Q| < N$ **do**
3: randomly choose a and b from P
4: **if** $a \prec b$ **then**
5: $Q \leftarrow Q \bigcup \{a\}$
6: **else if** $b \prec a$ **then**
7: $Q \leftarrow Q \bigcup \{b\}$
8: **else**
9: **if** $a \in K$ and $b \notin K$ **then**
10: $Q \leftarrow Q \bigcup \{a\}$
11: **else if** $a \notin K$ and $b \in K$ **then**
12: $Q \leftarrow Q \bigcup \{b\}$
13: **else**
14: **if** $DW(a) > DW(b)$ **then**
15: $Q \leftarrow Q \bigcup \{a\}$
16: **else if** $DW(a) < DW(b)$ **then**
17: $Q \leftarrow Q \bigcup \{b\}$
18: **else**
19: **if** $rand() < 0.5$ **then**
20: $Q \leftarrow Q \bigcup \{a\}$
21: **else**
22: $Q \leftarrow Q \bigcup \{b\}$
23: **end if**
24: **end if**
25: **end if**
26: **end if**
27: **end while**
28: **return** Q

In the following, we describe in detail the binary tournament mating selection, the adaptive knee point detection method, and the environmental selection, which are three important components in KnEA.

B. Binary Tournament Mating Selection

The mating selection in KnEA is a binary tournament selection strategy using three tournament strategies, namely, dominance comparison, knee point criterion, and a weighted distance. Algorithm 2 describes the detailed procedure of the mating selection strategy in KnEA.

In the binary tournament mating selection in KnEA, two individuals are randomly chosen from the parent population. If one solution dominates the other solution, then the former solution is chosen, referring to lines 4–7 in Algorithm 2. If the two solutions are nondominated with each other, then the algorithm will check whether they are both knee points. If only one of them is a knee point, then the knee point is chosen, seeing lines 9–12 in Algorithm 2. If both of them are knee points or neither of them is a knee point, then a weighted distance will be used for comparing the two solutions, as described in lines 14–17 in Algorithm 2. The solution with the larger weighted distance wins the tournament. If both solutions have an equal weighted distance, then one of them will be randomly chosen for reproduction.

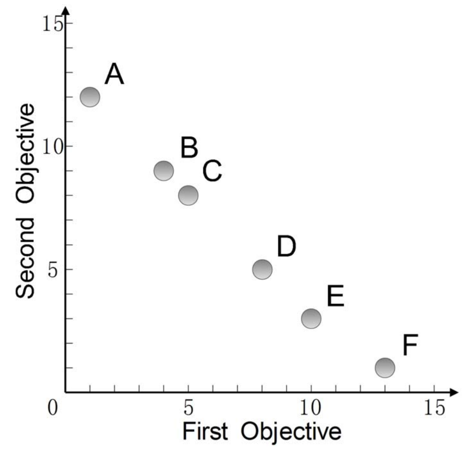

Fig. 2. Illustrative example where the proposed weighted distance may be advantageous over the crowding distance. In this example, neither solution B nor C will have the chance to win against other solutions if the crowding distance is adopted. Both B and C have a chance to win according to the defined weighted distance.

A weighted distance is designed for choosing a winning solution in the binary tournament mating selection if neither the dominance comparison nor the knee point criterion can distinguish the two solutions involved in the tournament. We adopt here the weighted distance measure to address some potential weakness of the crowding distance metric proposed in NSGA-II [6]. Fig. 2 illustrates a situation, where if the crowding distance is used, neither solution B nor solution C will have the chance to win against other solutions. However, from the diversity point of view, it would be helpful if either B or C can have a chance to win in the tournament for reproduction.

The weighted distance of a solution p in a population based on the k-nearest neighbors is defined as

$$DW(p) = \sum_{i=1}^{k} w_{p_i} \mathrm{dis}_{pp_i} \tag{1}$$

$$w_{p_i} = \frac{r_{p_i}}{\sum_{i=1}^{k} r_{p_i}} \tag{2}$$

$$r_{p_i} = \frac{1}{\left| \mathrm{dis}_{pp_i} - \frac{1}{k} \sum_{i=1}^{k} \mathrm{dis}_{pp_i} \right|} \tag{3}$$

where p_i represents the ith nearest neighbor of p in the population, w_{p_i} represents the weight of p_i, dis_{pp_i} represents the Euclidean distance between p and p_i, and r_{p_i} represents the rank of distance dis_{pp_i} among all the distances dis_{pp_j}, $1 \leq j \leq k$. From (3), it can be seen that a neighbor of p will have a larger rank if it is nearer to the center of all considered neighbors of p. By using the above weighted distance, we can verify that both solutions B and C have a certain probability to be selected in tournament selection. Note that some existing distance metrics can also address the above weakness of the crowding distance, such as the grid crowding distance (GCD) proposed in GrEA [29]. Compared to GCD, the weighted distance presented above is easier to calculate.

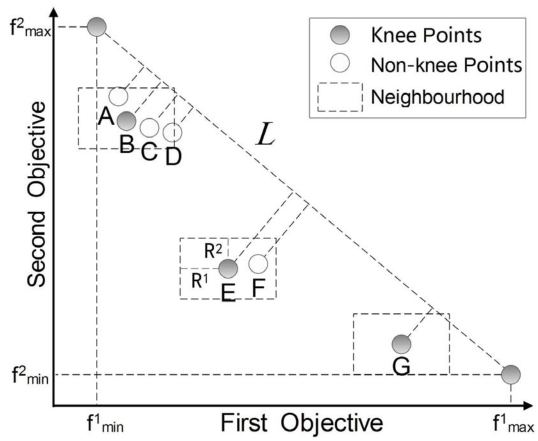

Fig. 3. Illustration for determining knee points in KnEA for a bi-objective minimization problem. In this example, solutions B, E, and G are identified as knee points for the given neighborhood denoted by the rectangles in dashed lines.

C. Adaptive Strategy for Identifying Knee Points

Knee points play a central role in KnEA. The knee points are used as a criterion only next to the dominance criterion in both mating and environmental selection. Therefore, an effective strategy for identifying solutions in the knee regions of the nondominated fronts in the combined population is critical for the performance of KnEA. To this end, an adaptive strategy is proposed for finding knee points in the population combining the parent and offspring populations at the present generation.

Fig. 3 presents an example for illustrating the main idea for determining knee points in the proposed strategy, where the nondominated front of a bi-objective minimization problem in consideration consists of nine solutions. First of all, an extreme line L is defined by the two extreme solutions, one having the maximum of f_1 and the other having the maximum of f_2 among all the solutions in the nondominated front. Then, we calculate the distance of each solution to L. A solution is identified as a knee point if its distance to the extreme line is the maximum in its neighborhood.

By looking at Fig. 3, we can see that solution B is a knee point in its neighborhood denoted by the rectangle in dashed lines, as it has the maximum distance to L among A, B, C, and D inside its neighborhood. Intuitively, solution E is also a knee point compared with solution F in its neighborhood. Note that, if there is only one solution in its neighborhood, e.g., solution G in Fig. 3, this solution will also be considered as a knee point. The above knee point definition leads to the benefit that the diversity of the population is implicitly taken into account.

The use of distance to the extreme line L to characterize knee points was first proposed by Das [41]. For a bi-objective minimization problem, L can be defined by $ax + by + c = 0$, where the parameters can be determined by the two extreme solutions. Then, the distance from a solution $A(x_A, y_A)$ to L can be calculated as

$$d(A, L) = \frac{|ax_A + by_A + c|}{\sqrt{a^2 + b^2}}. \quad (4)$$

For minimization problems, only solutions in the convex knee regions are of interest. Therefore, the distance measure in (4) can be modified as follows to identify knee points:

$$d(A, L) = \begin{cases} \frac{|ax_A+by_A+c|}{\sqrt{a^2+b^2}} & \text{if } ax_A + by_A + c < 0 \\ -\frac{|ax_A+by_A+c|}{\sqrt{a^2+b^2}} & \text{otherwise.} \end{cases} \quad (5)$$

The above distance measure for identifying knee points can be easily extended to optimization problems with more than two objectives, where the extreme line will become a hyperplane.

The example in Fig. 3 indicates that the size of neighborhood of the solutions will heavily influence the results of the identified knee points. Given the size of the neighborhood defined in Fig. 3, solutions B, E, and G are identified as knee points. Imagine, however, that if all solutions are included in the same neighborhood of a solution, then only solution E will be identified as knee point. For this reason, a strategy to tune the size of the neighborhood of solutions is proposed, which will be described in the following.

Assume the combined population at generation g contains N_F nondominated fronts, each of which has a set of nondominated solutions denoted by F_i, $1 \leq i \leq N_F$. The neighborhood of a solution is defined by a hyper cube of size $R_g^1 \times R_g^2 \times \cdots \times R_g^j \times \cdots \times R_g^M$, where $1 \leq j \leq M$, M is the number of objectives. Specifically, the size of the neighborhood regarding the jth objective, R_g^j, is determined as

$$R_g^j = \left(f\max_g^j - f\min_g^j\right) \cdot r_g \quad (6)$$

where $f\max_g^j$ and $f\min_g^j$ denote the maximal and the minimal values of the jth objective at the gth generation in set F_i, and r_g is the ratio of the size of the neighborhood to the span of the jth objective in nondominated front F_i at the gth generation, which is updated as

$$r_g = r_{g-1} * e^{-\frac{1-t_{g-1}/T}{M}} \quad (7)$$

where r_{g-1} is the ratio of the size of the neighborhood to the span of the jth objective of the solutions in F_i at the $(g-1)$th generation, M is the number of objectives, t_{g-1} is the ratio of knee points to the number of nondominated solutions in front F_i at the $(g-1)$th generation, and $0 < T < 1$ is a threshold that controls the ratio of knee points in the solution set F_i. Equation (7) ensures that r_g will significantly decrease when t_{g-1} is much smaller than the specified threshold T, and the decrease of r_g will become slower as the value of t_{g-1} becomes larger. r_g will remain unchanged when t_{g-1} reaches the given threshold T. t_g and r_g are initialized to 0 and 1, respectively, i.e., $t_0 = 0$ and $r_0 = 1$.

Fig. 4 presents the change of parameters R_g^1, r_g, and t_g on DTLZ2 with three objectives as the evolution proceeds, where T is set to $T = 0.5$. The size of the neighborhoods is adapted according to the ratio of the identified knee points to the total number of nondominated solutions. In the early stage of the evolutionary optimization, the size of neighborhoods will decrease quickly, and thus the number of found knee points will significantly increase. The ratio of knee points to all nondominated solutions (t_g) will increase as the evolution

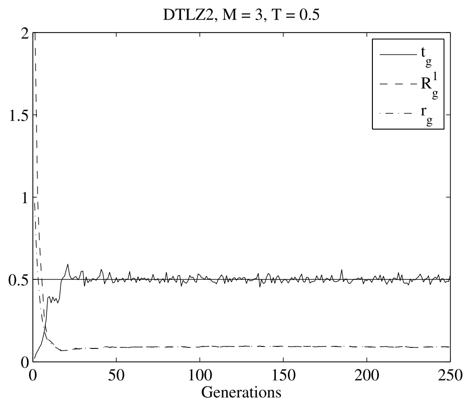

Fig. 4. Example of the changes of the parameters R_g^1, r_g, and t_g of the first front over the number of generations on DTLZ2 with three objectives.

Algorithm 3 *Finding_Knee_Point*(F, T, r, t)

Require: F (sorted population), T (rate of knee points in population), r, t (adaptive parameters)

$K \leftarrow \emptyset$ /* knee points */
for all $F_i \in F$ **do**
  $E \leftarrow$ *Find_extreme_solution*(F_i) /* F_i denotes the set of solutions in the ith front */
  $L \leftarrow$ *Calculate_extreme_hyperplane*(E)
  update r by formula (7)
  $f\max \leftarrow$ maximum value of each objective in F_i
  $f\min \leftarrow$ minimum value of each objective in F_i
  calculate R by formula (6)
  calculate the distance between each solution in F_i and L by formula (5)
  sort F_i in a descending order according to the distances
  $Size_{F_i} \leftarrow |F_i|$
  for all $p \in F_i$ **do**
    $NB \leftarrow \{a | a \in F_i \rightarrow |f_a^j - f_p^j| \leq R^j, 1 \leq j \leq M\}$
    $K \leftarrow K \bigcup \{p\}$
    $F_i \leftarrow F_i \backslash NB$
  end for
  $t = |K| / Size_{F_i}$
end for
return K, r and t

proceeds, which, in the meantime, will gradually decrease the size of the neighborhoods. When t_g is close to the threshold T, the size of the neighborhoods will remain constant.

The main steps of the adaptive strategy for detecting knee points are presented in Algorithm 3. The same procedure can be repeated for all nondominated fronts in the combined population until knee points are identified for all nondominated fronts. Note, however, that in the late search stage of MOPs, and actually already in the early stages of MaOPs, we only need to find the knee points in the first front due to the large number of nondominated solutions present in this front.

From the above descriptions, we can find that the proposed adaptive knee point identification algorithm differs considerably from the existing methods for finding knee points. Whereas, most existing MOEAs for finding the knee points are to accurately locate the knee points in the true Pareto front, the proposed adaptive strategy aims to find out the knee solutions in the neighborhoods, which will be preferred in the mating and environmental selection. Note again that by knee points here, we do not mean the knee points of the true Pareto front; instead, we mean the knee points of the nondominated fronts in the combined population at the current generation. In addition, some of the solutions identified as knee points may not be true knee points, which, however, can speed up the convergence performance and enhance the diversity of the population.

Algorithm 4 *Environmental_Selection*(F, K, N)

Require: F (sorted population), K (set of knee points), N (population size)

1: $Q \leftarrow \emptyset$ /*next population*/
2: $Q \leftarrow F_1 \bigcup \ldots \bigcup F_{i-1}$
3: $Q \leftarrow Q \bigcup (K \bigcap F_i)$
4: **if** $|Q| > N$ **then**
5:   delete $|Q| - N$ solutions from Q which belong to $K \bigcap F_i$ and have the minimum distances to the hyperplane
6: **else if** $|Q| < N$ **then**
7:   add $N - |Q|$ solutions from $F_i \backslash (K \bigcap F_i)$ to Q which have the maximum distances to the hyperplane
8: **end if**
9: **return** Q

D. Environmental Selection

Environmental selection is to select fitter solutions as parents for the next generation. Similar to NSGA-II, KnEA selects parents for the next generation from a combination of the parent and offspring populations of this generation, which therefore is an elitist approach. Whereas both NSGA-II and KnEA adopt the Pareto dominance as the primary criterion in environmental selection, KnEA prefers knee points instead of the nondominated solutions with a larger crowding distance as NSGA-II does. Algorithm 4 presents the main steps of environmental selection in KnEA.

Before environmental selection, KnEA performs nondominated sorting using the efficient nondominated sorting (ENS) algorithm reported in [51], forming N_F nondominated fronts, F_i, $1 \leq i \leq N_F$. Similar to NSGA-II, KnEA starts to select the nondominated solutions in the first nondominated front (F_1). If the number of solutions in F_1 is larger than the population size N, which is very likely already in the early generations in many-objective optimization, then knee points in F_1 are selected first as parents for the next population. Let the number of knee points in F_1 be NP_1. In case NP_1 is larger than N, then N knee points having a larger distance to the hyperplane are selected, referring to line 5 in Algorithm 4. Otherwise, NP_1 knee points are selected together with $(N - NP_1)$ other solutions in F_1 that have a larger distance to the hyperplane of F_1.

If the number of solutions in F_1 is smaller than N, KnEA turns to the second nondominated front for selecting the remaining $(N - |F_1|)$ parent solutions. If $|F_2|$ is larger than $N - |F_1|$, then the same procedure described above

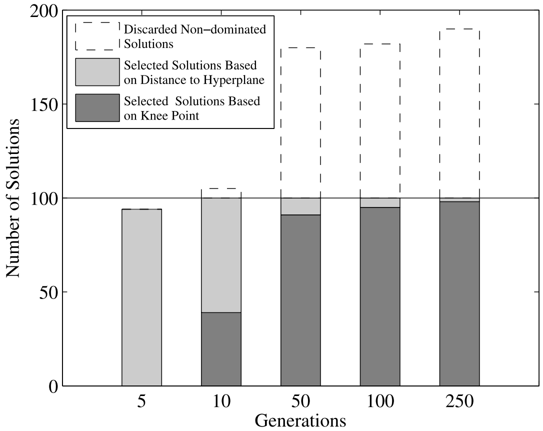

Fig. 5. Example showing the number of solutions in the first nondominated front, together with the number of solutions selected based on the knee point criterion and the distance to the hyperplane criterion. The results are obtained on the three-objective DTLZ2 using a population size of 100, i.e., the combined population size is 200.

will applied to F_2. This process is repeated until the parent population for the next generation is filled up.

It would be of interest to know how many solutions in the combined population are nondominated, how many are identified as knee points, and how many will be selected based on the distance to the hyperplane as the evolution proceeds. Take the three-objective DTLZ2 as an illustrative example and assume the population size is 100 and T is set to $T = 0.5$. Fig. 5 presents the number of solutions in the first nondominated front in the combined population at generations 5, 10, 50, 100, and 250, where the number of identified knee points and the number of solutions selected based on the distance to the hyperplane are indicated in black and gray, respectively. From the figure, we can see that the number of nondominated solutions is slightly less than 100 at generation 5 and thus all solutions in the first nondominated front will be selected. At generation 50, by contrast, almost all solutions (180 out of 200) are nondominated and a majority of the selected solutions (91 out of 100) are knee points. We can imagine that as the number of objectives increases, most selected solutions will be knee points even in early generations. These results indicate that the proposed method is different from the nondominance based selection and the distance based selection, and therefore the identified knee points play an essential role in determining the performance of the algorithm.

E. Empirical Computational Complexity Analysis

In this section, we provide an upper bound of the runtime of KnEA. Within one generation, KnEA mainly performs the following five operations: 1) mating selection; 2) genetic variations; 3) nondominated sorting; 4) knee-point identification; and 5) environmental selection. For a population size N and an optimization problem of M objectives, mating selection needs a runtime of $O(MN^2)$ to form a mating pool of size N, as the calculation of the weighted distances involves calculating the distance between pairs of solutions in the population. Genetic variations, here the simulated binary crossover (SBX) [52] and polynomial mutation [53], are performed on each decision variable of the parent solutions, which needs a runtime of $O(DN)$ to generate N offspring, where D is the number of decision variables. Nondominated sorting needs a runtime of $O(MN^2)$ in the worst case for the combined population of size $2N$ for optimization problems with M objectives. Knee point identification consists of the following two operations. First, obtaining the hyperplane and calculating the distance between each nondominated solution and the hyperplane, which at most needs a runtime of $O(MN)$. Second, checking whether the nondominated solutions are knee points in their neighborhoods, which costs a runtime of $O(MN^2)$. Therefore, knee point identification takes at most a runtime of $O(MN^2)$ in total. For environmental selection, a runtime of $O(N \log N)$ is needed, since the most time-consuming step is to sort the nondominated solutions according to their distances to the hyperplane. Therefore, KnEA needs at most a total runtime of $O(GMN^2)$, where G is the number of generations.

Compared with most popular MOEAs for MaOPs, KnEA is computationally very efficient. A theoretical comparison of the computational time of KnEA with these algorithms is beyond the scope of this paper; however, we will empirically compare the runtime performance of KnEA with four state-of-the-art MOEAs for MaOPs, details of which will be presented in the next section.

IV. Experimental Results and Discussion

In this section, we verify the performance of KnEA by empirically comparing it with four popular MOEAs for MaOPs, namely, GrEA [29], HypE [31], MOEA/D [9], and NSGA-III [36]. The experiments are conducted on 16 test problems taken from two widely used test suites, DTLZ [54] and WFG [55]. For each test problem, 2, 4, 6, 8, and 10 objectives will be considered, respectively. We compare both the quality of the obtained nondominated solution sets in terms of widely used performance indicators and the computational efficiency with respect to runtime. Note that the ENS-SS reported in [51] has been adopted as the nondominated sorting approach in all compared MOEAs.

A. Experimental Setting

For a fair comparison, we adopt the recommended parameter values for the compared algorithms that have achieved the best performance. Specifically, the parameter setting for all conducted experiments are as follows.

1) Crossover and Mutation: The SBX [52] and polynomial mutation [53] have been adopted to create offspring. The distribution index of crossover is set to $n_c = 20$ and the distribution index of mutation is set to $n_m = 20$, as recommended in [56]. The crossover probability $p_c = 1.0$ and the mutation probability $p_m = 1/D$ are used, where D denotes the number of decision variables.

2) Population Sizing: To avoid that the generated reference points are all located along the boundary of Pareto fronts for problems with a large number of objectives, the strategy of two-layered reference points recommended in NSGA-III [36] was adopted to generate uniformly distributed weight vectors

TABLE II
SETTING OF POPULATION SIZE IN NSGA-III AND MOEA/D, WHERE p_1 AND p_2 ARE PARAMETERS CONTROLLING THE NUMBERS OF REFERENCE POINTS ALONG THE BOUNDARY OF THE PARETO FRONT AND INSIDE IT, RESPECTIVELY

Number of objectives	Parameter (p_1, p_2)	Population size
2	(99, 0)	100
4	(7, 0)	120
6	(4, 1)	132
8	(3, 2)	156
10	(3, 2)	275

TABLE III
PARAMETER SETTING OF *div* IN GREA ON DTLZ AND WFG TEST SUITES

Problem	Obj. 2	Obj. 4	Obj. 6	Obj. 8	Obj. 10
DTLZ1	55	10	10	10	11
DTLZ2	45	10	10	8	12
DTLZ3	45	11	11	10	11
DTLZ4	55	10	8	8	12
DTLZ5	55	35	14	11	11
DTLZ6	55	36	20	20	20
DTLZ7	16	9	6	6	4
WFG1	45	8	9	7	10
WFG2	45	11	11	11	11
WFG3	55	18	18	16	22
WFG4–9	45	10	9	8	12

TABLE IV
PARAMETER SETTING OF *T* IN KNEA ON DTLZ AND WFG TEST SUITES

Problem	Obj. 2	Obj. 4	Obj. 6	Obj. 8	Obj. 10
DTLZ1	0.6	0.6	0.2	0.1	0.1
DTLZ3	0.6	0.4	0.2	0.1	0.1
DTLZ5	0.6	0.5	0.5	0.3	0.3
DTLZ6	0.6	0.5	0.4	0.3	0.3
DTLZ7	0.6	0.5	0.5	0.5	0.4
WFG4	0.6	0.5	0.5	0.3	0.3
WFG9	0.6	0.5	0.5	0.3	0.3
others	0.6	0.5	0.5	0.5	0.5

TABLE V
SETTING OF TEST PROBLEMS DTLZ1 TO DTLZ7

Problem	Number of Objectives (M)	Number of Variables (n)	Parameter (k)
DTLZ1	2, 4, 6, 8, 10	$M-1+k$	5
DTLZ2	2, 4, 6, 8, 10	$M-1+k$	10
DTLZ3	2, 4, 6, 8, 10	$M-1+k$	10
DTLZ4	2, 4, 6, 8, 10	$M-1+k$	10
DTLZ5	2, 4, 6, 8, 10	$M-1+k$	10
DTLZ6	2, 4, 6, 8, 10	$M-1+k$	10
DTLZ7	2, 4, 6, 8, 10	$M-1+k$	20

in NSGA-III and MOEA/D. Table II presents the setting of population size in NSGA-III and MOEA/D, where p_1 and p_2 are parameters controlling the numbers of reference points along the boundary of the Pareto front and inside it, respectively. For each test problem, the population size of HypE, GrEA, and KnEA is set to the same as that of NSGA-III and MOEA/D.

3) Number of Runs and Stopping Condition: We perform 20 independent runs for each algorithm on each test instance on a PC with a 3.16 GHz Intel Core 2 Duo CPU E8500 and the Windows 7 SP1 64 bit operating system. The number of iterations is adopted as the termination criterion for all considered algorithms. For DTLZ1 and WFG2, the maximum number of iterations is set to 700, and to 1000 for DTLZ3 and WFG1. For DTLZ2, DTLZ4, DTLZ5, DTLZ6, DTLZ7, and WFG 3 to WFG9, we set the maximum number of iterations to 250.

4) Other Parameters: The parameter setting for *div* in GrEA is taken from [29], which stands for the number of divisions in each dimension in GrEA. The method for calculating HV suggested in [50] is adopted in HypE: the exact method suggested in [50] is used to calculate the indicator value for test instances with two objectives, and otherwise the Monte Carlo sampling described in [31] is adopted to approximately calculate the indicator, where 10 000 samples are used in our experiments. For MOEA/D, the range of neighborhood is set to $N/10$ for all test problems, and Tchebycheff approach is employed as the aggregation function, where N is the population size. Three-nearest neighbors are used for calculating the weighted distance in KnEA, unless otherwise specified. Table III lists the parameter setting of *div* in GrEA on DTLZ and WFG test suites. To get the optimal setting for *div*, we tested many values for *div* for each of the test instances based on the recommendation in [29] and chose the one that produced the best performance for GrEA. Table IV lists the setting of T in KnEA on DTLZ and WFG test suites. As shown in the table, for DTLZ2, DTLZ4, and all test problems in the WFG suite except for WFG4 and WFG9, T is set to 0.6 for problems with two objectives and 0.5 otherwise.

5) Quality Metrics: Two widely used performance indicators, the HV [50] and the inverted generational distance (IGD) [57], [58] are used to evaluate the performance of the compared algorithms. It is believed that these two performance indicators can not only account for convergence (closeness to the true Pareto front), but also the distribution of the achieved nondominated solutions. However, as we discussed in Section II-B, HV may have a bias toward typical knee points. Note that the larger the HV value is, the better the performance of the algorithm. By contrast, a smaller IGD value indicates better performance of the MOEA. In this paper, $(1, 1, \ldots, 1)$ is chosen as the reference point in HV calculation. For the objective values of WFG test problems to have the same scale, each of the objective values has been normalized between the interval [0, 1] before calculating the HV. In addition, since the exact calculation of HV is computationally extremely intensive for MaOPs, the Monte Carlo method is adopted for estimating the HV when the test problem has more than four objectives, where 1 000 000 sampling points are used. We should stress that most recently a computationally efficient method was also suggested to calculate the exact HV [59]. On the other hand, IGD requires a reference set of Pareto optimal solutions, which are uniformly chosen from the true Pareto fronts of test problems. However, for different test

TABLE VI
IGD RESULTS OF THE FIVE COMPARED ALGORITHMS ON DTLZ1 TO DTLZ7, WHERE THE BEST MEAN FOR EACH TEST INSTANCE IS SHOWN WITH A GRAY BACKGROUND

Problem	Obj.	HypE	MOEA/D	GrEA	NSGA-III	KnEA
DTLZ1	2	1.9146E-3 (7.47E-6)	1.8057E-3 (1.81E-5)	3.2201E-3 (2.02E-5)	1.8512E-3 (8.52E-5)	2.2770E-3 (1.23E-4)
	4	1.2845E-1 (7.70E-3)	9.2918E-2 (2.95E-4)	4.8789E-2 (6.01E-3)	4.0182E-2 (1.83E-4)	5.1140E-2 (6.79E-3)
	6	2.3463E-1 (2.23E-2)	2.0355E-1 (4.55E-2)	1.0536E-1 (6.72E-2)	8.0213E-2 (1.17E-3)	1.6217E-1 (2.23E-2)
	8	3.2690E-1 (1.96E-2)	1.9820E-1 (6.66E-3)	1.2305E-1 (3.37E-2)	1.3814E-1 (4.11E-2)	2.6544E-1 (2.18E-2)
	10	3.2591E-1 (1.92E-2)	2.2471E-1 (1.37E-2)	1.7756E-1 (3.69E-2)	1.3406E-1 (3.66E-2)	2.4424E-1 (3.23E-2)
DTLZ2	2	5.5610E-3 (8.45E-5)	3.9634E-3 (2.44E-6)	1.0559E-2 (3.97E-5)	3.9811E-3 (1.16E-5)	5.7892E-3 (1.00E-3)
	4	2.4772E-1 (2.68E-3)	2.3719E-1 (1.79E-3)	1.2487E-1 (7.78E-4)	1.1604E-1 (1.33E-4)	1.2451E-1 (2.10E-3)
	6	3.8253E-1 (1.17E-2)	4.7756E-1 (6.73E-2)	2.5591E-1 (2.00E-3)	2.5871E-1 (1.47E-3)	2.5499E-1 (1.93E-3)
	8	5.9295E-1 (2.63E-2)	7.6487E-1 (7.53E-2)	3.4959E-1 (2.65E-3)	3.8780E-1 (4.35E-3)	3.4812E-1 (8.40E-3)
	10	7.1588E-1 (3.13E-2)	8.9000E-1 (5.45E-2)	3.4384E-1 (2.63E-3)	4.2250E-1 (7.50E-2)	3.2818E-1 (4.67E-2)
DTLZ3	2	6.1526E-3 (1.98E-4)	4.3390E-3 (2.14E-4)	1.0747E-2 (2.98E-4)	4.3476E-3 (3.46E-4)	7.1996E-2 (1.31E-2)
	4	4.9340E-1 (4.66E-2)	2.3896E-1 (7.80E-4)	1.4101E-1 (3.10E-2)	1.1725E-1 (1.90E-3)	1.9241E-1 (2.74E-2)
	6	7.5556E-1 (5.18E-2)	7.4559E-1 (1.91E-1)	3.3091E-1 (1.92E-1)	2.8209E-1 (6.65E-2)	5.5634E-1 (9.83E-2)
	8	9.0628E-1 (3.88E-2)	9.5772E-1 (9.21E-2)	4.2811E-1 (2.52E-1)	4.9400E-1 (1.79E-1)	8.8696E-1 (6.85E-2)
	10	9.6857E-1 (4.83E-2)	1.0364E+0 (6.94E-2)	4.9388E-1 (2.91E-1)	5.0141E-1 (1.28E-1)	8.7947E-1 (1.64E-1)
DTLZ4	2	5.8710E-3 (2.10E-4)	5.5751E-1 (3.28E-1)	8.8575E-3 (5.04E-4)	3.9883E-3 (2.70E-5)	5.9260E-3 (6.91E-4)
	4	4.5634E-1 (4.40E-3)	5.1153E-1 (2.02E-1)	1.4473E-1 (7.24E-2)	1.3346E-1 (7.32E-2)	1.2611E-1 (3.14E-3)
	6	5.9335E-1 (1.27E-1)	6.3958E-1 (9.95E-2)	2.5515E-1 (1.93E-3)	2.7069E-1 (3.99E-3)	2.5392E-1 (1.42E-4)
	8	5.7719E-1 (2.60E-2)	7.4405E-1 (7.85E-2)	3.4667E-1 (1.96E-3)	3.9306E-1 (3.02E-3)	3.3896E-1 (2.62E-3)
	10	6.5036E-1 (1.49E-2)	8.3080E-1 (3.64E-2)	3.4736E-1 (1.38E-3)	4.1024E-1 (2.19E-2)	3.2591E-1 (2.04E-3)
DTLZ5	2	5.1750E-3 (1.38E-4)	4.1369E-3 (1.76E-6)	8.2565E-3 (2.69E-4)	4.1603E-3 (1.39E-5)	6.7929E-3 (1.03E-3)
	4	2.4911E-2 (2.82E-3)	2.8469E-2 (2.26E-3)	1.5966E-2 (1.04E-3)	4.5300E-2 (1.17E-2)	8.3933E-2 (2.52E-2)
	6	2.5307E-2 (3.54E-3)	7.6702E-2 (1.44E-2)	1.3084E-1 (1.54E-2)	3.1645E-1 (7.08E-2)	2.0106E-1 (4.13E-2)
	8	3.2089E-2 (3.78E-3)	6.9405E-2 (1.75E-2)	2.2361E-1 (3.95E-2)	3.1737E-1 (9.53E-2)	2.5071E-1 (4.24E-2)
	10	3.3685E-2 (2.67E-3)	8.1111E-2 (2.34E-2)	3.1038E-1 (6.47E-2)	4.1988E-1 (8.12E-2)	2.5135E-1 (4.31E-2)
DTLZ6	2	4.8444E-3 (2.24E-4)	4.1320E-3 (2.15E-7)	8.5374E-3 (5.23E-6)	4.1321E-3 (7.72E-7)	3.2900E-2 (8.05E-3)
	4	2.6170E-1 (9.75E-2)	4.6896E-2 (5.00E-3)	2.9803E-2 (5.24E-3)	1.8616E-1 (5.29E-2)	2.2044E-1 (5.13E-2)
	6	2.1229E-1 (7.22E-2)	1.4971E-1 (4.91E-2)	2.3027E-1 (1.68E-1)	1.4700E+0 (4.05E-1)	3.9050E-1 (7.46E-2)
	8	2.1145E-1 (6.32E-2)	1.3321E-1 (3.69E-2)	4.1556E-1 (1.79E-1)	2.8652E+0 (6.39E-1)	3.8130E-1 (5.66E-2)
	10	1.2351E-1 (1.67E-2)	2.4177E-1 (5.51E-2)	4.7387E-1 (1.08E-1)	3.7696E+0 (4.27E-1)	3.7462E-1 (4.38E-2)
DTLZ7	2	4.3180E-3 (1.78E-5)	7.1417E-2 (1.61E-1)	2.4814E-2 (2.23E-3)	5.9160E-3 (1.89E-4)	5.5982E-3 (4.79E-4)
	4	1.1427E+0 (3.13E-2)	8.2375E-1 (4.49E-1)	1.7933E-1 (5.68E-3)	1.8538E-1 (7.96E-3)	1.4060E-1 (4.82E-2)
	6	1.7579E+0 (9.27E-2)	7.7783E-1 (2.05E-1)	3.8856E-1 (2.83E-2)	6.1746E-1 (2.28E-2)	3.8160E-1 (1.86E-3)
	8	2.8847E+0 (1.44E-1)	1.6876E+0 (2.92E-1)	1.0671E+0 (2.03E-2)	9.7790E-1 (6.04E-2)	8.6947E-1 (6.49E-2)
	10	3.5870E+0 (1.01E-1)	1.7550E+0 (5.08E-1)	1.4079E+0 (1.08E-1)	1.2284E+0 (9.01E-2)	1.1988E+0 (5.41E-2)

problems with different numbers of objectives, it is impossible to use exactly the same number of reference points. In this paper, we set the number of reference points to the integer that is closest to 500.

B. Results on the DTLZ Suite

The DTLZ test suite [54] is a class of widely used benchmark problems for testing the performance of MOEAs. Seven test functions, from DTLZ1 to DTLZ7 are used in the experiments here and their parameters are set as suggested in [54], which are presented in Table V.

The results on the seven DTLZ test problems are given in Table VI, with both the mean and standard deviation of the IGD values averaged over 20 independent runs being listed for the five compared MOEAs, where the best mean among the five compared algorithms is highlighted. From the table, we can find that both MOEA/D, HypE and NSGA-III performed well on DTLZ test problems with two objectives. Among the seven DTLZ test problems, MOEA/D achieved the smallest IGD values on five bi-objective test problems, while HypE and NSGA-III achieved an IGD value very close to the smallest one on all bi-objective DTLZ test problems. Note that

MOEA/D obtained a worse IGD value on the bi-objective DTLZ4. It appeared that MOEA/D does not work well on DTLZ4 with any number of objectives. The main reason is that DTLZ4 is a nonuniform MOP, which means that a set of evenly distributed weight combinations will lead to nonuniformly distributed Pareto optimal solutions. This is a known weakness of weighted aggregation methods for nonuniform MOPs.

For DTLZ test problems with more than three objectives, GrEA and NSGA-III performed better than MOEA/D and HypE on all test problems except for DTLZ5 and DTLZ6. MOEA/D and HypE worked very well on DTLZ5 and DTLZ6 with more than three objectives. HypE obtained the smallest IGD value among the five MOEAs under comparison on all DTLZ5 test instances with more than four objectives and DTLZ6 with ten objectives, while MOEA/D obtained the smallest IGD value on DTLZ6 with six and eight objectives and obtained the second smallest IGD value on the remaining test instances of DTLZ5 and DTLZ6 with more than three objectives except for DTLZ5 with four objectives (on this instance, MOEA/D achieved a value very close to the second smallest IGD value). These empirical results may illustrate that HypE and MOEA/D are well suited for dealing with MaOPs whose Pareto front is a degenerated curve.

Similar to GrEA and NSGA-III, the performance of KnEA is also very promising on the seven DTLZ test problems with more than three objectives. For DTLZ2, DTLZ4, and DTLZ7 with more than three objectives, KnEA achieved a slightly better IGD value than GrEA and NSGA-III on all test instances except for DTLZ2 with four objectives. For DTLZ5 and DTLZ6 with more than three objectives, KnEA achieved a similar IGD value as GrEA and NSGA-III on DTLZ5, but it achieved a much better IGD value than GrEA and NSGA-III on DTLZ6, although these IGD values obtained by KnEA are still slightly worse than those obtained by HypE and MOEA/D. Note, however, that GrEA and NSGA-III outperformed KnEA on DTLZ1 and DTLZ3 with more than three objectives. This may be attributed to the fact that DTLZ1 and DTLZ3 are multimodal test problems containing a large number of local Pareto optimal fronts and preference over knee points in the neighborhood easily results in the preference over local Pareto optimal solutions. This could be partly alleviated by using a smaller threshold T, which is the predefined maximal ratio of knee points to the nondominated solutions in a nondominated front. Therefore, for multimodal MOPs, T needs to be chosen more carefully to balance exploration and exploitation. More detailed discussions on the influence of T on the performance of KnEA will be presented in Section IV-D.

From the 35 test instances of the DTLZ test suite presented in Table VI, we can find that KnEA wins in 11 instances in terms of IGD, while GrEA wins 5, HypE 5, MOEA/D 7, and NSGA-III 7. From these results, we can conclude that KnEA outperforms HypE, MOEA/D, GrEA, and NSGA-III on DTLZ test problems in terms of IGD, especially for problems with more than three objectives.

Fig. 6 illustrates the runtime of the five algorithms on all DTLZ test problems, where the runtime of an algorithm on M objectives is obtained by averaging over the runtimes consumed by the algorithm for one run on all M-objective DTLZ problems. Note that the runtimes are displayed in logarithm in the figure. As shown in the figure, MOEA/D outperforms the four compared MOEAs on all instances in terms of runtime, which are much less than HypE, GrEA, NSGA-III, and KnEA. Note, however, that although KnEA consumed more time than MOEA/D did, it used much less time than GrEA and HypE and consumed comparable runtime with NSGA-III. We see that KnEA took roughly only one-third of the runtime of GrEA on bi-objective instances. As the number of objectives increases, the runtime of KnEA increased only very slightly. For ten-objective test problems, the runtime of KnEA is only about one seventh of that of GrEA. Among the five algorithms under comparison, HypE consumes the highest amount of runtime on all numbers of objectives, which is due to its very intensive computational complexity for repeatedly calculating the HV.

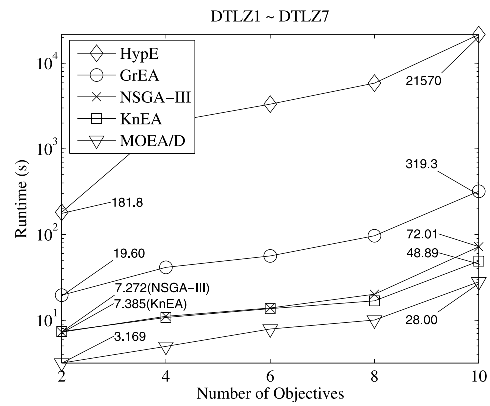

Fig. 6. Runtime(s) of the five algorithms on all DTLZ test problems, where the runtime of an algorithm on M objectives is obtained by averaging over the runtimes consumed by the algorithm for one run on all DTLZ problems with M objectives.

The runtime of MOEA/D should remain roughly the same as the number of objectives increases. The main reason is that MOEA/D decomposes an MOP into a number of single-objective optimization subproblems, where the number of subproblems is determined by the predefined population size, regardless of the number of objectives of the MOP. However, from Fig. 6, we can see that the runtime of MOEA/D on DTLZ test problems increased as the number of objectives increases, which is attributed to the larger population size on problems with an increased number of objectives. The runtime consumed by KnEA, NSGA-III, GrEA, and HypE is expected to increase as the number of objectives increases, since GrEA, NSGA-III, and KnEA are all based on nondominated sorting and the number of nondominated solutions will increase significantly as the number of objectives increases, while the computational time for calculating the HV suffers from a dramatic increase when the number of objectives increases.

The rapid increase in runtime of GrEA can be attributed to its environmental selection, where only one solution is selected at a time from solutions that cannot be distinguished using dominance comparison, which is quite time-consuming when the number of nondominated solutions becomes large. In KnEA, by contrast, all other nondominated solutions apart

TABLE VII
PARAMETER SETTING FOR TEST PROBLEMS WFG1 TO WFG9

Number of Objectives (M)	Position Parameter (K)	Distance Parameter (L)	Number of Variables
2	4	10	$K+L$
4	6	10	$K+L$
6	10	10	$K+L$
8	7	10	$K+L$
10	9	10	$K+L$

from the knee points can be selected at once according to their distance to the hyperplane. This saves much time for KnEA compared to GrEA and HypE, particularly when the number of objectives is large.

To summarize, we can conclude from Table VI and Fig. 6 that KnEA performs the best among the five compared algorithms. KnEA is computationally also much more efficient than many Pareto-based or performance indicator based popular MOEAs such as GrEA and HypE, and comparable with NSGA-III and MOEA/D, which are computationally very efficient MOEAs.

C. Results on the WFG Suite

The WFG test suite was first introduced in [60] and systematically reviewed and analyzed in [55], which was designed with the aim to introduce a class of difficult benchmark problems for evaluating the performance of MOEAs. In this paper, we used nine test problems, from WFG1 to WFG9. The parameters of these problems are set as suggested in [55], which are listed in Table VII.

Like in previous work, we compare the quality of the solution sets obtained by the compared algorithms on the nine WFG test problems in terms of HV, which is another very popular performance indicator that takes both accuracy (closeness to the true Pareto front) and the diversity of the solution set into account. Table VIII presents the mean and standard deviation of the HVs of the five algorithms on WFG1 to WFG9, averaging over 20 independent runs, where the best mean among the five algorithms is highlighted. From this table, the following observations can be made. First, MOEA/D, HypE, and NSGA-III still achieved a good performance on WFG test problems with two objectives in terms of HV. MOEA/D and NSGA-III obtained a HV close to the best one on all WFG test problems with two objectives, while HypE obtained the best HV on WFG4, WFG5, and WFG9 with two objectives among the five algorithms under comparison. These empirical results confirm that MOEA/D, HypE, and NSGA-III are promising algorithms for MOPs with a small number of objectives. For WFG problems with two objectives, the performance of GrEA and KnEA is also encouraging, since they were able to produce comparable results with those of MOEA/D, HypE, and NSGA-III on all WFG problems with two objectives.

By contrast, KnEA, NSGA-III, and GrEA performed consistently much better than MOEA/D and HypE in terms of HV on WFG problems with more than three objectives. The best HV or close to the best HV was obtained by KnEA, NSGA-III, and GrEA on all WFG problems with more than three objectives, especially for WFG5, WFG6, WFG8, and WFG9. On these four WFG problems, KnEA, NSGA-III, and GrEA obtained a HV that is at least two times of that obtained by HypE and MOEA/D. HypE and MOEA/D achieved a good performance on some WFG test instances with more than three objectives. Among the five compared algorithms, HypE obtained the best HV on WFG2 with eight and ten objectives, while MOEA/D achieved the best HV on WFG3 with four, eight, and ten objectives.

KnEA performed comparably well with NSGA-III and GrEA on WFG test problems with more than three objectives, and often better on most WFG test instances when the number of objectives is larger than six. For all 18 WFG test instances with eight and ten objectives, KnEA only obtained a slightly worse HV than NSGA-III and GrEA on WFG2 with eight and ten objectives, WFG3 with eight and ten objectives and WFG8 with eight objectives. These results indicate that KnEA is more suited to deal with MaOPs with more than six objectives than GrEA and NSGA-III.

Overall, KnEA performed better than MOEA/D, HypE, NSGA-III, and GrEA on the WFG test suite in terms of HV. KnEA achieved the best HV on 22 test instances out of 45 WFG test instances considered in this paper, while GrEA, NSGA-III, HypE, and MOEA/D achieved the best HV on ten instances, three instances, five instances, and five instances, respectively. Therefore, we can conclude that KnEA is very competitive for solving the WFG test functions, especially for problems with more than three objectives. Note that KnEA performed very well for all WFG test functions even on the multimodal problems WFG4 and WFG9, since a small value of T has been adopted in KnEA on WFG4 and WFG9 with a large number of objectives, which confirms that a carefully selected small value of T is helpful for KnEA to achieve a good performance on multimodal problems.

Fig. 7 illustrates the runtime of the five algorithms on all WFG test problems, where the runtime of an algorithm on M objectives is obtained by averaging over the runtimes consumed by the algorithm for one run on all WFG test problems with M objectives. Note that in the figure the runtimes are displayed in logarithm. As can be seen from Fig. 7, we can find that the average runtime of KnEA is much less than that of GrEA and HypE, comparable to NSGA-III, however, is still slightly more than that of MOEA/D. This demonstrates that the performance of KnEA is very promising in terms of runtime.

From Table VIII and Fig. 7, we can conclude that overall, KnEA showed the most competitive performance on the WFG test problems. In addition, KnEA is computationally much more efficient than GrEA and HypE, and comparable to NSGA-III and MOEA/D, which is very encouraging.

D. Sensitivity of Parameter T in KnEA

KnEA has one algorithm specific parameter T, which is used to control the ratio of knee points to the nondominated

TABLE VIII
HVS OF THE FIVE ALGORITHMS ON WFG1 TO WFG9, WHERE THE BEST MEAN FOR EACH TEST INSTANCE IS SHOWN WITH A GRAY BACKGROUND

Problem	Obj.	HypE	MOEA/D	GrEA	NSGA-III	KnEA
WFG1	2	4.2990E-1 (2.62E-3)	6.3175E-1 (4.79E–3)	6.3072E-1 (6.86E–4)	6.3033E-1 (1.07E-2)	6.2722E-1 (1.90E-2)
	4	8.0119E-1 (2.42E-3)	9.4650E-1 (1.57E-2)	9.4877E-1 (4.31E-3)	7.2716E-1 (3.57E-2)	9.7950E-1 (4.96E-3)
	6	9.0084E-1 (7.07E-3)	9.4059E-1 (6.64E-2)	9.7543E-1 (4.86E-3)	7.3024E-1 (5.24E-2)	9.8950E-1 (1.65E-2)
	8	9.6318E-1 (5.15E-4)	9.0191E-1 (7.96E-2)	9.8379E-1 (2.41E-3)	5.4589E-1 (5.88E–2)	9.9091E-1 (4.73E-3)
	10	9.8739E-1 (9.08E-3)	8.3757E-1 (1.29E-1)	9.8728E-1 (2.00E-3)	4.9141E-1 (6.73E-2)	9.9443E-1 (5.51E-3)
WFG2	2	1.9803E-1 (7.64E-5)	5.0407E-1 (3.13E-2)	5.4851E-1 (5.11E-4)	5.5176E-1 (1.85E-3)	5.4979E-1 (1.03E-3)
	4	5.9595E-1 (4.04E-3)	7.9745E-1 (5.20E-2)	9.4954E-1 (4.92E-3)	9.3457E-1 (7.44E-2)	9.7240E-1 (2.46E-3)
	6	5.0112E-1 (1.02E-1)	7.6779E-1 (8.63E-2)	9.3146E-1 (7.54E-2)	9.5678E-1 (7.08E-2)	9.8882E-1 (1.77E-3)
	8	9.9691E-1 (5.62E-4)	9.1592E-1 (1.12E-1)	9.6876E-1 (3.52E-3)	9.9228E-1 (3.04E-3)	9.9161E-1 (1.08E-3)
	10	9.9901E-1 (2.63E-4)	9.3037E-1 (4.78E-2)	9.7813E-1 (3.43E-3)	9.9660E-1 (2.28E-3)	9.9317E-1 (1.31E-3)
WFG3	2	4.3313E-1 (1.30E-3)	4.8868E-1 (1.96E-3)	4.8970E-1 (1.07E-3)	4.9073E-1 (9.04E-4)	4.9286E-1 (8.69E-4)
	4	4.8242E-1 (2.11E-2)	5.7038E-1 (7.53E-3)	5.6215E-1 (6.06E-3)	5.4981E-1 (6.34E-3)	5.4941E-1 (1.08E-2)
	6	3.6553E-1 (4.23E-4)	5.7269E-1 (1.48E-2)	5.8912E-1 (3.50E-3)	5.5618E-1 (1.56E-2)	5.4849E-1 (1.37E-2)
	8	4.5940E-1 (1.76E-2)	5.9451E-1 (6.48E-3)	5.9245E-1 (3.51E-3)	5.3840E-1 (2.63E-2)	5.5483E-1 (2.05E-2)
	10	4.4752E-1 (5.21E-3)	6.0178E-1 (4.76E-3)	6.0055E-1 (2.09E-3)	6.0049E-1 (1.80E-2)	5.5756E-1 (1.50E-2)
WFG4	2	2.0985E-1 (4.59E-5)	2.0563E-1 (9.26E-4)	2.0597E-1 (5.47E-4)	2.0727E-1 (6.71E-4)	2.0793E-1 (3.85E-4)
	4	5.1111E-1 (5.67E-4)	3.4408E-1 (2.14E-2)	5.1253E-1 (5.14E-3)	4.7834E-1 (8.03E-3)	5.0660E-1 (4.50E-3)
	6	5.2722E-1 (8.08E-3)	2.5191E-1 (2.55E-2)	6.2377E-1 (4.58E-3)	5.8534E-1 (2.76E-2)	6.2568E-1 (1.33E-2)
	8	6.5253E-1 (1.09E-3)	3.8362E-1 (4.64E-2)	6.7778E-1 (7.65E-3)	7.0102E-1 (9.95E-3)	7.5446E-1 (6.69E-3)
	10	6.4900E-1 (4.51E-2)	4.0333E-1 (7.03E-2)	8.1735E-1 (5.89E-3)	7.8515E-1 (1.65E-2)	8.3767E-1 (6.59E-3)
WFG5	2	1.7937E-1 (1.52E-4)	1.7821E-1 (6.95E-5)	1.7592E-1 (9.25E-5)	1.7834E-1 (4.55E-5)	1.7399E-1 (2.66E-3)
	4	2.9364E-1 (6.14E-3)	3.0592E-1 (1.90E-2)	4.9028E-1 (2.71E-3)	4.6965E-1 (3.98E-3)	4.8274E-1 (2.81E-3)
	6	3.0323E-1 (1.38E-2)	2.4446E-1 (2.89E-2)	6.0923E-1 (8.17E-3)	6.0141E-1 (6.78E-3)	6.0681E-1 (5.31E-3)
	8	4.7190E-1 (8.51E-3)	3.2769E-1 (1.94E-2)	6.4884E-1 (6.33E-3)	7.1140E-1 (5.45E-3)	7.1573E-1 (7.18E-3)
	10	4.8701E-1 (4.23E-3)	3.1971E-1 (2.73E-2)	7.8627E-1 (6.50E-3)	7.8370E-1 (5.18E-3)	8.1223E-1 (3.48E-3)
WFG6	2	9.7306E-2 (9.01E-4)	1.6798E-1 (1.41E-2)	1.6607E-1 (6.9096E-3)	1.7070E-1 (9.18E-3)	1.7043E-1 (8.75E-3)
	4	1.2887E-1 (3.70E-3)	2.9948E-1 (2.17E-2)	4.7686E-1 (1.75E-2)	4.5274E-1 (1.21E-2)	4.6385E-1 (1.74E-2)
	6	1.2832E-1 (1.72E-3)	3.1998E-1 (4.92E-2)	5.9722E-1 (2.38E-2)	5.9342E-1 (2.38E-2)	5.8903E-1 (1.92E-2)
	8	1.3260E-1 (8.53E-4)	3.4564E-1 (3.05E-2)	6.1993E-1 (1.60E-2)	6.8735E-1 (1.58E-2)	6.9360E-1 (1.89E-2)
	10	1.3391E-1 (1.91E-3)	3.6008E-1 (4.07E-2)	7.6846E-1 (1.60E-2)	7.7138E-1 (1.95E-2)	7.8831E-1 (1.53E-2)
WFG7	2	1.7148E-1 (7.26E-3)	2.0843E-1 (2.84E-4)	2.0627E-1 (2.28E-4)	2.0884E-1 (3.46E-4)	2.0896E-1 (2.40E-4)
	4	4.8545E-1 (1.02E-2)	3.8074E-1 (2.48E-2)	5.5277E-1 (2.14E-3)	5.2453E-1 (5.85E-3)	5.3764E-1 (3.35E-3)
	6	3.8823E-1 (8.75E-4)	3.6256E-1 (4.17E-2)	6.8524E-1 (6.43E-3)	6.4957E-1 (3.81E-2)	6.8807E-1 (6.74E-3)
	8	7.5233E-1 (3.53E-2)	3.8309E-1 (5.04E-2)	7.1584E-1 (7.08E-3)	7.6820E-1 (8.62E-3)	7.8708E-1 (1.20E-2)
	10	7.9526E-1 (2.91E-2)	3.7048E-1 (5.45E-2)	8.7285E-1 (6.48E-3)	8.5128E-1 (1.05E-2)	8.9484E-1 (3.18E-3)
WFG8	2	4.4350E-2 (1.68E-3)	1.5386E-1 (2.40E-3)	1.4837E-1 (1.30E-3)	1.4731E-1 (1.33E-3)	1.4170E-1 (5.94E-3)
	4	1.1250E-1 (9.82E-4)	2.1642E-1 (1.10E-2)	3.7820E-1 (6.32E-3)	3.3674E-1 (1.03E-2)	3.4842E-1 (9.69E-3)
	6	1.3361E-1 (1.38E-2)	1.9018E-1 (1.86E-2)	4.6842E-1 (3.58E-2)	4.3429E-1 (1.95E-2)	4.3532E-1 (3.25E-2)
	8	1.8083E-1 (2.24E-3)	3.1187E-1 (2.63E-2)	4.6036E-1 (2.07E-2)	5.6980E-1 (1.48E-2)	5.5710E-1 (2.08E-2)
	10	1.8045E-1 (4.23E-3)	3.1283E-1 (4.53E-2)	7.1165E-1 (4.59E-3)	6.6252E-1 (2.81E-2)	7.1503E-1 (5.14E-2)
WFG9	2	2.0467E-1 (5.57E-4)	1.7407E-1 (3.41E-2)	2.0275E-1 (1.14E-3)	1.9825E-1 (2.00E-2)	1.7393E-1 (6.23E-2)
	4	3.3698E-1 (2.14E-2)	2.6820E-1 (3.57E-2)	4.9720E-1 (5.19E-3)	4.1054E-1 (5.53E-2)	4.9287E-1 (5.36E-3)
	6	1.8562E-1 (3.74E-3)	1.6388E-1 (4.18E-2)	5.7679E-1 (3.88E-2)	4.8225E-1 (4.56E-2)	5.9820E-1 (4.28E-2)
	8	3.2783E-1 (6.45E-2)	3.0523E-1 (4.51E-2)	6.5642E-1 (1.44E-2)	6.7658E-1 (2.39E-2)	7.2698E-1 (8.28E-3)
	10	3.4602E-1 (2.05E-2)	3.0884E-1 (4.32E-2)	7.9937E-1 (4.68E-3)	7.5520E-1 (9.49E-3)	8.0769E-1 (8.06E-3)

solutions in the combined population. In the following, we investigate the influence of T on the performance of KnEA, which varies from 0.1 to 0.9. Note that $0 < T < 1$.

From the parameter settings in the previous experiments, we have already noted that T has been set to different values in KnEA depending on whether the optimization problem has a

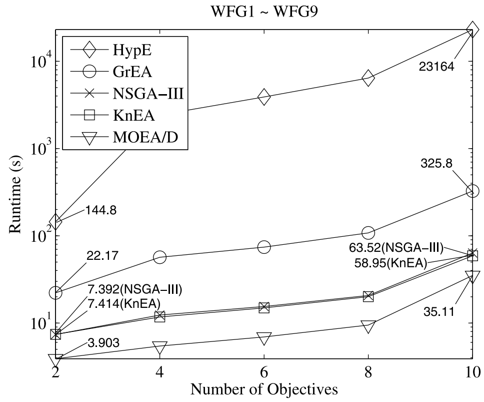

Fig. 7. Runtime(s) of the five algorithms on all WFG test problems, where the runtime of an algorithm on M objectives is obtained by averaging over the runtimes consumed by the algorithm for a run on all M-objective WFG problems.

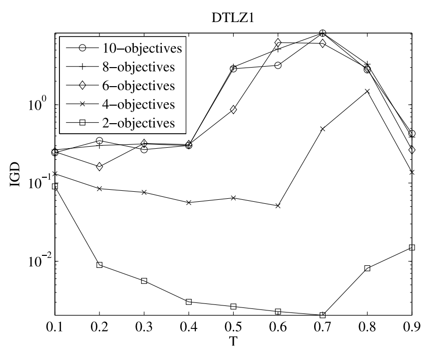

Fig. 8. IGD values on DTLZ1 of KnEA with different settings for parameter T, averaging over 20 independent runs.

large number of local Pareto optimal fronts. The main reason is that a relatively small T is helpful for KnEA to escape from local Pareto fronts. For this reason, we consider the setting of T on two DTLZ test problems, DTLZ1 and DTLZ2, with the former representing a class of optimization problems having a large number of local Pareto fronts, while the latter representing a class of test problems that do not have a large number of local Pareto optimal fronts. Note that similar results have been obtained on other test problems.

Fig. 8 shows the results of IGD values for different settings of parameter T on DTLZ1 with 2, 4, 6, 8, and 10 objectives, averaging over 20 independent runs. Note that in the figure the IGD values are displayed in logarithm. We can see that as T varies from 0.1 to 0.9, the IGD value of KnEA on DTLZ1 first decreases, and then will increase again. For DTLZ1 with two and four objectives, the best performance has been achieved when T is around 0.6, while for DTLZ1 with six objectives, the best performance is achieved when $T = 0.2$, and for DTLZ1 with eight and ten objectives, $T = 0.1$ produces the best performance. In general, the experimental results confirm that for multimodal MOPs, a relatively small T, e.g., between 0.1 and 0.4 may be more likely to lead to good performance, particularly when the number of objectives is larger than four. For multimodal MOPs having two to four objectives, T can be set to between 0.5 and 0.6.

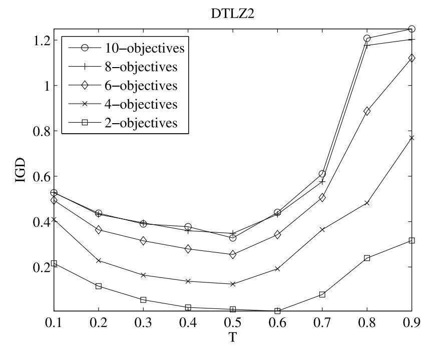

Fig. 9. IGD values on DTLZ2 of KnEA with different settings for parameter T, averaging over 20 independent runs.

The experimental results on DTLZ2 are summarized in Fig. 9, where the mean IGD values for different settings of parameter T on DTLZ2 with 2, 4, 6, 8, and 10 objectives averaging over 20 independent runs are presented. We can see from the figure that the IGD value will first become smaller as T increases up to 0.6 for the bi-objective DTLZ2 and up to 0.5 for DTLZ2 having more than two objectives. Compared to the T values that produce the best performance for DTLZ1, we can conclude that for MOPs that do not have a large number of local Pareto fronts, T can be set between 0.5 and 0.6, where a slightly larger T can be used for a smaller number of objectives.

To summarize the above results, we can conclude that although the performance of KnEA varies with the value of parameter T, there is a pattern that can be followed to guide the setting for T. For MOPs without a large number of local Pareto optimal fronts, T can be set to 0.6 for bi-objective problems, to a value around 0.5 for problems having more than two objectives. For MOPs with a large number of local Pareto optimal fronts, $T = 0.5$ is recommended for bi- or three-objective problems, while for problems with more than three objectives, a small value of T is recommended and the larger the number of objectives is, the smaller the value of T should be used.

V. Conclusion

In this paper, a novel MOEA for solving MaOPs, called KnEA, has been proposed. The main idea is to make use of knee points to enhance the search performance of MOEAs when the number of objectives becomes large. In KnEA, the knee points in the nondominated solutions are preferred to other nondominated solutions in mating selection and environmental selection. To the best of our knowledge, this is the first time that knee points have been used to increase the selection pressure in solving MaOPs, thereby improving the convergence performance of Pareto-based MOEAs.

In KnEA, a new adaptive algorithm for identifying knee points in the nondominated solutions has been developed. While most existing MOEAs for knee points aim to accurately locate the knee solutions in the true Pareto front, the proposed adaptive knee point identification algorithm intends to find knee points in the neighborhood of solutions in the nondominated fronts during the optimization, thereby distinguishing some of the nondominated solutions from others. To this end, the adaptive strategy attempts to maintain a proper ratio of the identified knee points to all nondominated solutions in each front by adjusting the size of the neighborhood of each solution in which the solution having the maximum distance to the hyperplane is identified as the knee point. In this way, the preference over knee points in selection will not only accelerate the convergence performance but also the diversity of the population.

Comparative experimental results with four popular MOEAs, namely, MOEA/D, HypE, GrEA, and NSGA-III demonstrate that the proposed KnEA significantly outperforms MOEA/D and HypE, and is comparable with GrEA and NSGA-III on MaOPs with more than three objectives. Most encouragingly, KnEA is computationally much more efficient compared with other Pareto-based MOEAs such as GrEA and performance indicator based MOEAs such as HypE. Therefore, the overall performance of KnEA is highly competitive compared to the state-of-the-art MOEAs for solving MaOPs.

This paper demonstrates that the idea of using knee points to increase the selection pressure for MaOPs is very promising. Further work on developing more effective and computationally more efficient algorithms for identifying knee solutions is highly desirable. In KnEA, nondominated solutions other than the knee points have been selected according to their distance to the hyperplane. This idea has been shown to be effective in KnEA, however, the performance of KnEA could be further improved by introducing criteria other than the distance to the hyperplane. Finally, the performance of KnEA remains to be verified on real-world MaOPs.

REFERENCES

[1] J. G. Herrero, A. Berlanga, and J. M. M. López, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 151–168, Feb. 2009.

[2] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.

[3] S. H. Yeung, K. F. Man, K. M. Luk, and C. H. Chan, "A trapeizform U-slot folded patch feed antenna design optimized with jumping genes evolutionary algorithm," *IEEE Trans. Antennas Propag.*, vol. 56, no. 2, pp. 571–577, Feb. 2008.

[4] J. Handl, D. B. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 4, no. 2, pp. 279–292, Apr. 2007.

[5] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.

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

[7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Design Optim. Control Appl. Ind. Probl.*, Athens, Greece, 2001, pp. 95–100.

[8] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Probl. Solving Nat.*, Birmingham, U.K., 2004, pp. 832–842.

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

[10] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multi-objective optimization," in *Proc. 2001 Genet. Evol. Comput. Conf.*, San Francisco, CA, USA, pp. 283–290.

[11] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, La Jolla, CA, USA, 2000, pp. 325–332.

[12] K. Narukawa and T. Rodemann, "Examining the performance of evolutionary many-objective optimization algorithms on a real-world application," in *Proc. 6th Int. Conf. Genet. Evol. Comput.*, Kitakyushu, Japan, 2012, pp. 316–319.

[13] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 641–655.

[14] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *Proc. IEEE Congr. Evol. Comput.*, Seoul, Korea, 2001, pp. 957–962.

[15] D. Brockhoff *et al.*, "On the effects of adding objectives to plateau functions," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 591–603, Jun. 2009.

[16] O. Schütze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 444–455, Aug. 2011.

[17] L. S. Batista, F. Campelo, F. G. Guimaraes, and J. A. Ramírez, "A comparison of dominance criteria in many-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, New Orleans, LA, USA, 2011, pp. 2359–2366.

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

[19] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. 9th Conf. Genet. Evol. Comput.*, London, U.K., 2007, pp. 773–780.

[20] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.

[21] Z. He and G. G. Yen, "Ranking many-objective evolutionary algorithms using performance metrics ensemble," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, 2013, pp. 2480–2487.

[22] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.

[23] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evol. Comput.*, vol. 21, no. 2, pp. 231–259, 2013.

[24] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1402–1412, Oct. 2008.

[25] G. Wang and H. Jiang, "Fuzzy-dominance and its application in evolutionary many objective optimization," in *Proc. Int. Conf. Comput. Intell. Security Workshops*, Harbin, China, 2007, pp. 195–198.

[26] 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.

[27] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim.*, Matsushima, Japan, 2007, pp. 727–741.

[28] A. López, C. A. C. Coello, A. Oyama, and K. Fujii, "An alternative preference relation to deal with many-objective optimization problems," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 291–306.

[29] 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.

[30] 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.

[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, 2011.

[32] D. K. Saxena and K. Deb, "Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 3204–3211.

[33] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.

[34] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multi-objective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, Oct. 2010.

[35] 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.

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

[37] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between neighborhood size and MOEA/D performance on many-objective problems," in *Proc. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 459–474.

[38] L. B. Said, S. Bechikh, and K. Ghédira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 801–818, Oct. 2010.

[39] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, and R. Caballero, "g-dominance: Reference point based dominance for multiobjective metaheuristics," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 685–692, 2009.

[40] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.

[41] I. Das, "On characterizing the 'knee' of the Pareto curve based on normal-boundary intersection," *Struct. Optim.*, vol. 18, nos. 2–3, pp. 107–115, 1999.

[42] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Proc. 8th Int. Conf. Parallel Probl. Solving Nat.*, Birmingham, U.K., 2004, pp. 722–731.

[43] L. Rachmawati and D. Srinivasan, "A multi-objective genetic algorithm with controllable convergence on knee regions," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1916–1923.

[44] L. Rachmawati and D. Srinivasan, "A multi-objective evolutionary algorithm with weighted-sum niching for convergence on knee regions," in *Proc. 8th Int. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2006, pp. 749–750.

[45] O. Schütze, M. Laumanns, and C. A. C. Coello, "Approximating the knee of an MOP with stochastic search algorithms," in *Proc. 10th Int. Conf. Parallel Probl. Solving Nat.*, Dortmund, Germany, 2008, pp. 795–804.

[46] S. Bechikh, L. B. Said, and K. Ghédira, "Searching for knee regions in multi-objective optimization using mobile reference points," in *Proc. ACM Symp. Appl. Comput.*, Sierre, Switzerland, 2010, pp. 1118–1125.

[47] K. Deb, J. Sundar, U. Bhaskara, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 635–642, 2006.

[48] K. Deb and S. Gupta, "Understanding knee points in bicriteria problems and their implications as preferred solution principles," *Eng. Optim.*, vol. 43, no. 11, pp. 1175–1204, 2011.

[49] T. Tušar and B. Filipič, "Scaling and visualizing multiobjective optimization test problems with knees," in *Proc. 15th Int. Multiconf. Inf. Soc.*, Ljubljana, Slovenia, 2012, pp. 155–158.

[50] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

[51] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, 2015.

[52] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Compl. Syst.*, vol. 9, no. 4, pp. 115–148, 1995.

[53] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.

[54] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Dept. Comput. Eng. Netw. Lab., ETH Zurich, Zurich, Switzerland, TIK-Tech. Rep. 112, 2001.

[55] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

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

[57] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[58] M. Li and J. Zheng, "Spread assessment for evolutionary multiobjective optimization," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 216–230.

[59] L. M. S. Russo and A. P. Francisco, "Quick hypervolume," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 481–502, Aug. 2014.

[60] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multiobjective test problem toolkit," in *Proc. 3th Int. Conf. Evol. Multi-Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 280–294.

Xingyi Zhang received the B.Sc. degree from Fuyang Normal College, Anhui, China, in 2003, and the M.Sc. and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

He is an Associate Professor with the School of Computer Science and Technology, Anhui University, Anhui. His research interests include unconventional models and algorithms of computation, multiobjective optimization, and membrane computing.

Ye Tian received the B.Sc. degree from Anhui University, Anhui, China, in 2012, where he is currently working toward the master's degree from the School of Computer Science and Technology.

His research interests include multiobjective optimization methods and their application.

Yaochu Jin (M'98–SM'02) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University, Bochum, Germany, in 2001.

He is a Professor of Computational Intelligence, Department of Computing, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes). His research interests include the interdisciplinary areas that bridge the gap between computational intelligence, computational neuroscience, computational systems biology, and nature-inspired, real-world-driven problem-solving. He has (co)edited five books and three conference proceedings, authored a monograph, and (co)authored over 200 peer-reviewed journal and conference papers. He holds eight U.S., EU, and Japan patents. His research is funded by EC FP7, U.K. Engineering and Physical Sciences Research Council, and industry. He has delivered 20 invited keynote speeches at international conferences.

Dr. Jin received the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology and the 2014 IEEE Computational Intelligence Magazine Outstanding Paper Award. He is an Associate Editor or an Editorial Board Member of IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NANOBIOSCIENCE, *IEEE Computational Intelligence Magazine*, *Evolutionary Computation*, *BioSystems*, *International Journal of Fuzzy Systems*, *Soft Computing*, *Memetic Computing*, and *Natural Computing*. He is an IEEE Distinguished Lecturer and a Vice President for Technical Activities of the IEEE Computational Intelligence Society. He is a fellow of the British Computer Society.