Differential Evolution Enhanced With Multiobjective Sorting-Based Mutation Operators

Jiahai Wang, *Member, IEEE*, Jianjun Liao, Ying Zhou, and Yiqiao Cai

***Abstract*—Differential evolution (DE) is a simple and powerful population-based evolutionary algorithm. The salient feature of DE lies in its mutation mechanism. Generally, the parents in the mutation operator of DE are randomly selected from the population. Hence, all vectors are equally likely to be selected as parents without selective pressure at all. Additionally, the diversity information is always ignored. In order to fully exploit the fitness and diversity information of the population, this paper presents a DE framework with multiobjective sorting-based mutation operator. In the proposed mutation operator, individuals in the current population are firstly sorted according to their fitness and diversity contribution by nondominated sorting. Then parents in the mutation operators are proportionally selected according to their rankings based on fitness and diversity, thus, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. Since fitness and diversity information is simultaneously considered for parent selection, a good balance between exploration and exploitation can be achieved. The proposed operator is applied to original DE algorithms, as well as several advanced DE variants. Experimental results on 48 benchmark functions and 12 real-world application problems show that the proposed operator is an effective approach to enhance the performance of most DE algorithms studied.**

***Index Terms*—Differential evolution, diversity, exploration and exploitation, mutation operator, nondominated sorting.**

I. Introduction

DIFFERENTIAL evolution (DE) [1], [2] is one of the most successful and popular evolutionary algorithms (EAs) for global optimization. The advantages of DE are its easiness of use, simple structure, efficacy, and robustness. Recently, DE is successfully applied to various scientific and engineering fields. Details can be found in recent two reviews of DE [3], [4].

The salient feature of DE lies in its mutation mechanism that distinguishes it from other EAs. In the mutation operator of DE, a mutant vector can be treated as a leading individual to explore the search space and generated by adding a difference vector to a base vector. We have observed, however, on the one hand, both vectors (i.e., base and difference vectors) in most of DE are selected randomly, which does not fully utilize the fitness information. Hence, all vectors are equally likely to be selected as parents without selective pressure at all [3]. On the other hand, the diversity information is always ignored. This will make the population lose diversity. The algorithm is trapped in local optimum more frequently, especially when the optimization problem is multimodal [3], [4]. Hence, the fitness and diversity information of the population is not fully exploited in the designing of DE.

In view of the limitations of previous works, this paper proposes an alternative to the uniform random selection of parents during mutation in original DE. During the selection of parents, the fitness and diversity information is introduced, with the aim of favoring those individuals with both high fitness values and high diversity contributions. In this way, two underlying objectives, fitness and diversity, are both considered simultaneously. The proposed scheme is named as multiobjective sorting-based mutation operator. More specifically, individuals in the current population are first sorted according to their fitness and diversity contribution by nondominated sorting. Then some of the parents in the mutation operators are proportionally selected according to their rankings, thus, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. DE enhanced with multiobjective sorting-based mutation operators is named as MS-DE framework. In order to evaluate the effectiveness of MS-DE, the proposed framework is applied to original DE algorithms, as well as several advanced DE variants. Extensive experiments have been carried out on two sets of benchmark functions and some real-world application problems. The results show that MS-DE is able to enhance the performance of DE for most of the considered problems.

The main contribution of this paper is to select promising or healthy individuals with better fitness and diversity simultaneously as parents in DE mutation. This leads to the major advantages of the proposed algorithm as follows.

1) Since fitness and diversity information is considered by multiobjective sorting to select the parents, a good balance between exploration and exploitation can be achieved. The balance between exploration and exploitation is a core of all kinds of EAs. The proposed

Manuscript received September 1, 2013; revised January 10, 2014, March 23, 2014, and March 29, 2014; accepted April 6, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 60805026, Grant 61070076, and Grant 61305085, and in part by the Zhujiang New Star of Science and Technology in Guangzhou City under Grant 2011J2200093. This paper was recommended by Associate Editor Y. Jin.

J. Wang, J. Liao, Y. Zhou are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wjiahai@hotmail.com; liaojianj@mail2.sysu.edu.cn; dyingdy@gmail.com).

Y. Cai is with the College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China (e-mail: yiqiao00@163.com).

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

Digital Object Identifier 10.1109/TCYB.2014.2316552

approach provides a simple yet efficient way to deal with the balance.
2) The proposed approach is very simple because it keeps the simple structure of the original DE algorithm, and no additional parameter is introduced.
3) The proposed approach can be easily applied to other advanced DE variants. Thus, it cooperates with different kinds of modifications in advanced DE variants to further improve the performance. This provides a promising approach for solving benchmark and real-world problems.
4) The multiobjective sorting-based selection scheme can be seen as a general and fundamental mating selection scheme to select promising or healthy individuals. Since mating selection operator plays an important role in most of EAs, the proposed approach can be used to further improve other EAs.

The rest of this paper is organized as follows. In Section II, DE and related work are reviewed. MS-DE framework is presented in Section III. In Section IV, experimental results are reported. Finally, conclusions are drawn in Section V.

II. Related Work

A. DE

The following global optimization problem is considered:

$$\text{Minimize} \quad f(X) \tag{1}$$

where $X = (x_1, x_2, \ldots, x_D) \in R^D$ is a vector in the D-dimensional decision (variable) space (solution space), and the feasible solution space is $x_i \in [L_i, U_i]$, where L_i and U_i represent the lower and upper bound of the ith parameter, respectively. DE evolves a population of NP candidate individuals (solutions) with D-dimensional parameter vectors [1]. Each individual is denoted as $X_{i,G} = [x_{i,1,G}, x_{i,2,G}, \ldots, x_{i,D,G}]$, where $i = 1, 2, \ldots, NP$, NP is the population size and G is the current generation. It has three main operators: mutation, crossover, and selection.

1) Mutation: This operator generates a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ (named as target vector). The originally proposed and most frequently used mutation strategies in the literature are [1] and [2].

1) DE/rand/1

$$V_{i,G} = X_{i_{1,G}} + F \cdot (X_{i_{2,G}} - X_{i_{3,G}}). \tag{2}$$

2) DE/rand/2

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{i_2,G} - X_{i_3,G}) + F \cdot (X_{i_4,G} - X_{i_5,G}). \tag{3}$$

3) DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{i_1,G} - X_{i_2,G}). \tag{4}$$

4) DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{i_1,G} - X_{i_2,G}) + F \cdot (X_{i_3,G} - X_{i_4,G}). \tag{5}$$

5) DE/current-to-best/1

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{i_1,G} - X_{i_2,G}). \tag{6}$$

6) DE/rand-to-best/1

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{best,G} - X_{i_1,G}) + F \cdot (X_{i_2,G} - X_{i_3,G}). \tag{7}$$

The indices i_1, i_2, i_3, i_4 and i_5 are mutually different random indices chosen from the set $\{1, 2, \ldots, NP\} \setminus \{i\}$. $X_{best,G}$ is the best vector in terms of fitness value at the current generation G. In the mutation defined by (2), the first-term at the right hand of the equation is called base vector, and remaining term is called difference vector. F, called mutation scaling factor, is a positive control parameter for scaling the difference vectors.

In general, we can distinguish between mutation operators that promote exploration (called explorative strategies) and operators that promote exploitation (called exploitative strategies). Operators that incorporate the best individual (see DE/best/1, DE/best/2, DE/current-to-best/1, DE/rand-to-best/1) favor exploitation, since the mutant individuals are attracted around the current best individual. DE/rand-to-best/1 has a stronger exploration capability by introducing more perturbation with the random individual. In contrast, operators that incorporate randomly selected individuals (see DE/rand/1, DE/rand/2) enhance the exploration ability, since a high degree of random variability is introduced.

2) Crossover: When the mutant vector is generated, crossover operator is applied to each pair of target vector $X_{i,j,G}$ and mutant vector $V_{i,j,G}$ to generate a trial vector $U_{i,G} = [u_{i,1,G}, u_{i,2,G}, \ldots, u_{i,D,G}]$. There are two kind of crossover operators, binomial and exponential. The most widely used is the binomial crossover as follows:

$$u_{i,j,G} = \begin{cases} v_{i,j,g}, & \text{if } rand_j(0,1) \le CR \quad \text{or} \quad j = j_{rand} \\ x_{i,j,g}, & \text{otherwise} \end{cases} \tag{8}$$

where $\text{rand}_j(0,1)$ is a uniform random number in (0,1) for jth dimension, $CR \in (0, 1)$ is a predefined crossover control parameter, and $j_{rand} \in (1, D)$ is an integer randomly chosen from 1 to D.

3) Selection: This operator uses a one-to-one greedy scheme to select the better one between target vector $X_{i,G}$ and trial vector $U_{i,G}$ to survive in the next generation as follows:

$$X_{G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \tag{9}$$

B. Improved Mutation Operators

DE has drawn the attention of many researchers [3], [4] due to the attractive characteristics. However, DE is not completely free from problems of stagnation and premature convergence due to the limited amount of exploratory moves [4]. To overcome these problems, a number of DE variants are proposed. According to [4], these modern DE variants can be classified into two categories: DE integrated with an extra component and DE with modified structures. Modifications on DE in these DE variants mainly focus on control parameters (i.e., NP, F and CR), operators (mutation, crossover, and selection), population (multipopulation or parallel population) and hybridization of DE with other operators or algorithms. Compared to many empirical works, few research has so far been devoted to theoretically analyze the search mechanism and convergence properties of DE. Recently, [40]

takes a first significant step toward the convergence analysis of DE.

In this section, we focus on the related work that how the mutation operators can be modified in DE to improve its performance. These modifications can be classified roughly into three categories: proposal of new mutation operator, ensemble of different mutation operators, and selection of vectors in the existing mutation operator.

Several new mutation operators are proposed, for example, rotation-invariant mutation operator [5], DE/rand-to-best/*p*best [6], Gaussian PBX-α [7], and Gaussian mutation [8]. Direction information of the best and worst near neighbors is incorporated into the mutation strategies [9].

Two representatives of algorithms with ensemble of different mutation operators are self-adaptive DE (SaDE) [10] and composite DE (CoDE) [11]. In SaDE, both mutation strategies and their associated control parameter values are gradually self-adapted by learning from their previous experiences in generating promising solutions. It is a learning-based ensemble method. In CoDE, three mutation strategies and three control parameter settings are randomly combined to generate new solutions. It is a random combination-based ensemble method. [41] also proposes a mutation ensemble method. It adopts different existing parameter control methods of DE to adaptively choose a suitable mutation strategy.

Selection of suitable vectors in the existing mutation operator is also an effective approach to improve DE. Tournament best base vector selection for DE is proposed in [12]. Several selection methods for the best individual in DE/current-to-best/1, DE/rand-to-best/1, and DE/best/1 are proposed in [13]–[16] and [42], respectively. These selection methods all aim to appropriately make use of the bias of the best individuals, but they introduce additional control parameters or sophisticated control mechanisms. Role differentiation and malleable mating for DE are proposed in [17]. Recently, two general frameworks for the selection of suitable vectors in the existing mutation operators, called proximity-based mutation in DE (ProDE) [18] and rank-based mutation in DE (rank-DE) [19], are proposed. ProDE advocates a stochastic selection in which the probability of selecting an individual to become a parent is inversely proportional to its distance from the individual undergoing mutation. In rank-DE, some of the parents in the mutation operators are proportionally selected according to their fitness rankings in the current population. The higher ranking a parent obtains, the more frequently it will be selected. Similar rank-based parent selection scheme is also proposed in earlier work [20]. Unlike the selection methods in [12]–[16], and [42] designed for a specific mutation strategy, ProDE and rank-DE are general frameworks and can be applied to all mutation strategies.

III. Multiobjective Sorting-Based Mutation Operators

A. Motivations

In order to be successful, an EA algorithm, for example, DE, needs to establish a good balance between exploration and exploitation [21]. There are some additional characteristics in DE. On the one hand, not all DE search operators have the same impact on the balance of exploration and exploitation [18]. DE search operators can be roughly classified into explorative and exploitative mutation operators as mentioned above. Hence, the choice of the most efficient mutation operator can be cumbersome and problem dependent. On the other hand, most of DE mutation strategies have clustering tendency [18], which easily leads to premature convergence.

Recent works, including rank-DE and ProDE, aim to promote the exploitation ability of DE. Rank-DE increases the selective pressure on fitter solutions in the population, and hence promotes efficient exploitation. But it may be over-exploitative and lead to premature convergence to the local optima in the multimodal problems especially for exploitative mutation operators. Earlier work [20] has pointed out that one inherent danger of rank-based parent selection with high selective pressure is the rapid loss of diversity within the population, and the increase of exploitative power comes at a cost to potential exploration. By favoring search in the vicinity of the mutated individual, ProDE promotes efficient exploitation, while diminishing the exploration abilities of the mutation operator. Therefore, the proximity scheme does not lead to great benefits for exploitative mutation strategies, and sometimes deteriorates their performance. This highlights the importance of a good balance between exploration and exploitation in DE search.

Exploration is associated with the distribution of individuals on a landscape, and can be estimated by a genotypic diversity measure. In contrast, exploitation is related to individual response (fitness), which can be described with phenotypic (fitness) convergence measure [21]–[23]. Since the exploration and exploitation are measured indirectly using diversity and fitness, respectively, balance between exploration and exploitation can be guided or achieved by diversity and fitness. High diversity does not necessarily mean that a diverse population is obtained by a good balance between exploration and exploitation [21]. We are interested in diversity that helps to find fit individuals, i.e., useful diversity or healthy diversity. In rank-DE and ProDE, the diversity information is always ignored. This will make the population lose diversity and thus be trapped in local optimum more frequently. Therefore, in this paper, the fitness and diversity are considered to select parents for mutation operator in DE, with the aim of favoring those promising or healthy individuals with both high fitness values and high diversity contributions. To achieve this, two key problems arise: how to measure the contribution of diversity of an individual to the whole population and how to consider fitness and diversity simultaneously to guide the evolution.

B. Multiobjective Sorting-Based Mutation Operators

1) Fitness and Diversity Measures: The aim of using two measures (objectives) is to select promising or healthy individuals with better fitness and diversity. The fitness is associated with exploitation and the diversity helps exploration. Since greedy convergence and maintenance of diversity (i.e., exploitation and exploration) are often conflicting objectives of search process, we may consider the diversity versus selective pressure (fitness) problem as a multiobjective problem.

Prior to discuss about the multiobjective sorting procedure, one more measure that would have a direct relation with the population diversity for each individual is defined. There are a lot of diversity measures for whole population (i.e., on population level) as reviewed in [23], but there is few diversity contribution measures for a single individual to whole population (i.e., on individual level). Here, an Euclidean distance-based diversity metric for each individual is defined:

$$\Psi_i = \sum_{j=1}^{NP} ||X_i - X_j|| \tag{10}$$

where $i \in (1, NP)$, NP is the population size, $||X_i - X_j|| = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2}$, and D denotes the dimensionality of the search space. Ψ_i equals to the sum of the distances of ith individual from other members in the population. High diversity value means the ith individual is far from other ones in the population. This measures the contribution of diversity of an individual to the whole population.

In fact, the worth of an individual is not considered to rely only on its fitness value. When mating with other individuals, the ability of an individual to produce offspring dispersed enough in the solution space is also very important, because it helps to explore the solution space efficiently and exhaustively. An individual that is distant from the others in the population has more chances, when mating, to produce offspring in the regions of the search space not covered by the current population. Thus, two measures or objectives associated with each individual in the population can be defined as

$$\text{Minimize} \quad f_{fitness}(X_i) = f(X_i) \tag{11}$$

$$\text{Minimize} \quad f_{diversity}(X_i) = -\Psi_i \tag{12}$$

where $f(X_i)$ is the fitness function value of ith individual and Ψ_i is the contribution of diversity of ith individual to the population. The diversity (12) is not the final objective of the single objective optimization problem, and thus can be seen as a helper objective.

2) Nondominated Sorting: To understand the nondominated sorting procedure, some definitions about multiobjective optimization are first briefly reviewed. The concept of dominance can be formally stated in the following way. In general, a multiobjective optimization problem can be state as follows:

$$\text{Minimize} \quad F(X) = \{f_1(X), f_2(X), \ldots, f_M(X)\} \tag{13}$$

where $X = (x_1, x_2, \ldots, x_D) \in R^D$ is a vector in the D-dimensional decision space as in single optimization problem (1), and $F = (f_1, f_2, \ldots, f_M) \in \Omega^M$ is the objective space with M minimization objectives. A solution X dominates Y iff X is at least as good as Y in all objectives, and better in at least one. Two solutions are incomparable or nondominated iff they neither dominate the other.

In this paper, the nondominated sorting procedure [24] is adopted to sort the individuals of a given population according to the level of nondomination. The nondominated sorting procedure mainly consists of two steps [24]. In the first step, for each individual, domination count c_p, the number of individuals which dominate the solution p, and a set of individuals S_p that solution p dominates, are calculated. All individuals in the first nondominated front will have their domination count as zero. In the second step, for each solution p with $c_p = 0$, each member q in its set S_p is visited and its domination count is reduced by one. In doing so, if the domination count becomes zero for any member q, it is added into a separate list Q. These members belong to the second nondominated front. The above procedure is continued with each member of Q and the third front is identified. This process continues until all fronts are identified. Details of the nondominated sorting procedure can be found in nondominated sorting genetic algorithm II (NSGA-II) [24].

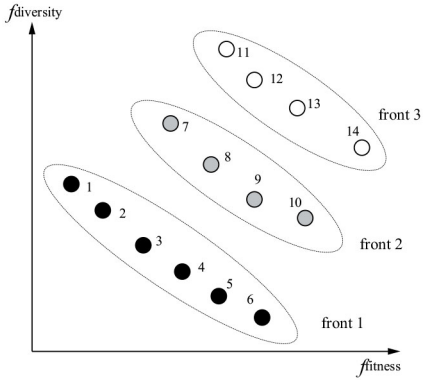


Fig. 1. Typical output of the nondominated sorting procedure.

The nondominated sorting procedure is mainly used to partition the population of individuals into different fronts based on the multiple objectives or measures considered. Fig. 1 illustrates the typical output of the nondominated sorting procedure. The individuals in the lowest layer belong to front 1. The second lowest layer of individuals form front 2, and remaining individuals belong to front 3.

The time complexity of the nondominated sorting is $O(M \cdot NP^2)$, where M is the number of objectives and NP is the population size. Since only two objectives are considered in this paper, the time complexity can be simplified as $O(NP^2)$.

3) Selection Procedure With Nondominated Sorting: Using nondominated sorting procedure, the population is sorted into several fronts, thus we can differentiate the individuals among different fronts. In order to further differentiate the individuals within the same front, an objective (fitness or diversity) is randomly selected to sort the individuals. This sorting mechanism within the same front is simple, and can automatically switch back and forth between fitness-based selection and diversity-based selection. Finally, the population can be fully sorted in ascending order based on the fitness and diversity. Fig. 1 shows the ascending order of 14 individuals in a typical population, where fitness is selected to sort the individuals within the same front. The number or index of an individual in the sorted population shown in Fig. 1 indicates its order.

Thus, the ranking of ith individual (vector) in the sorted population is assigned as follows:

$$R_i = NP + 1 - i, i = 1, 2, \ldots, NP \tag{14}$$

where NP is the population size, i is the index of ith individual in the sorted population. According to (14), better individual

Algorithm 1 MS-DE/rand/1: Multiobjective Sorting Based Mutation for DE/rand/1 /***Mutation Operator***/

1: Compute the fitness and diversity of each individual according to (11) and (12).
2: Sorting the population according to the fitness and diversity of each individual using the nondominated sorting procedure.
3: Calculate the probability vector based on (15).
4: Utilize a roulette wheel to select parents based on the probability vector.
5: For each target vector, generate the corresponding mutant vector using DE/rand/1 mutation (2).

in terms of bi-objective, defined by (11) and (12), in the sorted population will obtain better ranking.

Then, the selection probability of the ith individual for mutation operation is proportional to

$$P_i = \frac{R_i}{NP}, i = 1, 2, \ldots, NP. \tag{15}$$

Hence, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. Note that the simplest linear model is adopted to compute the selection probability vector based on the ranking, which is inspired by the will to make things as simple as possible, neither introducing arbitrary parameters nor using sophisticated heuristics. In the future, other sophisticated models, for example, those defined in [19] and [20], can be used.

Finally, we incorporate a stochastic selection procedure, in the form of a simple roulette wheel selection with replacement, to select parents for mutation operators. Note that all selection probabilities of individuals, in the roulette wheel selection process, are firstly normalized since all selection probabilities obtained by (15) do not sum to 1.

C. MS-DE Framework

The proposed multiobjective sorting-based mutation framework affects only the mutation step of the original DE, hence it can be directly applied to any DE mutation strategy. The application of this framework for DE/rand/1, named MS-DE/rand/1, is described in Algorithm 1.

In the application of this framework for operators that incorporate the best individual (see DE/best/1, DE/best/2, DE/current-to-best/1, DE/rand-to-best/1), an individual randomly selected from the first front is set as the best individual (Note that remaining parent individuals are still selected by roulette wheel selection process). Thus, the mutant individuals are attracted around the best individual in terms of both fitness and diversity values. Here, what is the best individual is redefined. This new definition of the best individual can be used in other EAs, for example, particle swarm optimization (PSO), to select global best individual.

Ong and Keane [25] stated that the simplicity of an algorithm is also very important. Simplicity means ease of implementation and a minimum number of control parameters of the algorithm [25]. The proposed MS-DE is very simple. MS-DE modifies the original DE only in the selection of parents in mutation parents, thus it keeps the simple structure of the original DE without introducing additional parameter. Further, it can be easily applied to other advanced DE variants.

Although MS-DE adopts the nondominated sorting procedure from multiobjective optimization, it is different from multiobjective optimization algorithm. Multiobjective optimization algorithms, for example, NSGA-II, aim to simultaneously optimize multiple objectives. The nondominated sorting is originally used in environmental selection in NSGA-II. Environmental selection in multiobjective optimization algorithms aims to obtain a well-approximated and well-distribution solution set, which is implemented by picking out the best solutions from the previous population and the newly created population. In NSGA-II, the nondominated sorting and crowding distance are used for approximation and distribution, respectively. MS-DE aims to optimize only a single objective to find a global optimum instead of a non-dominated solution set, and the nondominated sorting is introduced for mating selection instead of environmental selection.

The fitness and diversity are associated with exploitation and exploration, respectively. Thus, the classic trade-off or balance between exploration and exploitation is explicitly modeled using multiobjective optimization. Mating selection aims to make a good preparation for exchanging the information of individuals, which plays an important role in EAs. In MS-DE, the nondominated sorting for mating selection selects promising individuals in mutation operator for sharing the information among them, which is expected to improve the performance of DE. Further, the proposed nondominated sorting for mating selection also can be used in other EAs.

1) Complexity Analysis: Compared with the original DE, MS-DE has additional computation burden on calculation of diversity measure and nondominated sorting procedure. During one generation, the computation complexity of diversity measure mainly depends on the distance measure, which takes $\frac{NP\cdot(NP-2)}{2}$ times of calculating the pairwise distances between individuals (owing to the symmetric property of the distance measure). Therefore, the complexity of diversity measure is $O\left(D\cdot\frac{NP\cdot(NP-2)}{2}\right)$. As mentioned above, the complexity of nondominated sorting is $O(NP^2)$. Since the complexity of the original DE algorithm is $O(G_{max}\cdot NP\cdot D)$, where G_{max} is the maximal number of generation, the overall complexity of MS-DE is $O\left(G_{max}\cdot NP\cdot D + G_{max}\cdot\left(D\cdot\frac{NP\cdot(NP-2)}{2}+NP^2\right)\right)$, that is, $O(G_{max}\cdot NP\cdot NP\cdot D)$.

It should be noted that the recalculation of distance measure in MS-DE is performed only when at least one individual of every pair is changed at current generation. As a result, the complexity of MS-DE is far lower than $O(G_{max}\cdot NP\cdot NP\cdot D)$. In addition, under the circumstances of that the function evaluation is costly (see the CEC 2013 benchmark problems [26]), the computational burden of MS-DE is mostly determined by the function evaluation. According to the study in [18], the additional computational cost by computing the pairwise distance between individuals is shown to be negligible for the functions with costly evaluation. Therefore, the additional

overhead caused by diversity calculation and nondominated sorting is relatively small when the function evaluation is costly. The comparison of MS-DE with the original DE in terms of computational cost will be given in Section IV-F.

2) Comparison With ProDE and Rank-DE: MS-DE, ProDE and rank-DE are all general frameworks for the selection of suitable parent vectors in the existing mutation operators of DE. However, the ways to select the parent vectors in these frameworks are totally different. ProDE advocates a stochastic selection of some individuals according to their distances from the individual undergoing mutation. It is a neighborhood (distance) biased parent selection and favors search in the vicinity of the mutated individual. rank-DE selects parents according to their fitness ranking in the current population. It is a fitness biased parent selection and increases the selective pressure on fitter solutions. As a result, both ProDE and rank-DE promote efficient exploitation but diminish exploration abilities of the mutation operators. The proposed MS-DE selects parents according to the fitness and diversity information of individuals in the current population, and thus it is a fitness-and-diversity biased parent selection. In MS-DE, the exploitation is represented by fitness value, and the exploration is represented by a defined diversity measure. It is expected that balance of exploration and exploitation can be achieved by simultaneously considering fitness and diversity. The experimental comparisons of three frameworks will be given in Section IV-E.

IV. Experimental Results

In this section, comprehensive experiments are carried out to evaluate the effectiveness of MS-DE. Firstly, MS-DE is tested on 28 benchmark functions from CEC 2013 [26], and the comparisons of MS-DE with corresponding DE algorithms, advanced DE variants and related frameworks, respectively, are given in detail. Then, MS-DE is tested on 20 benchmark functions at 1000D from CEC 2012 special session on large scale global optimization [27] to study the scalability of MS-DE. Finally, MS-DE is applied to solve 12 real-world application problems from CEC 2011 [28]. The simulations are carried out on an Intel Core Dual-Core E5400 PC with 2.70 GHz CPU and 2GB RAM.

A. Experiment Setup

MS-DE is firstly tested on CEC 2013 test suite with 28 benchmark functions. These functions are divided into three groups: unimodal functions ($F1$-$F5$), basic multimodal functions ($F6$-$F20$), and composition functions ($F21$-$F28$). Elaborate descriptions of the 28 functions can be found in [26].

To evaluate the performances of original DE and MS-DE in each of the six original DE mutation strategies, we select parameters as follows.

1) Population size: $NP = 100$ [9], [29], and [30].
2) Scaling factor and crossover factor: $F = 0.5$, $CR = 0.9$ [9], [30].
3) Dimension of each function: $D = 30$ and $D = 50$ [26].
4) Maximum number of function evaluations (MaxFEs): $MaxFEs = 10^4 D$ [26].
5) Independent Number of runs (NumR): $NumR = 51$ [26].

Three evaluation criteria are adopted to measure the final performance of each algorithm.

1) *Solution error measure:* As an evaluation indicator, the function error value for solution X is defined as $f(X) - f(X^*)$, where X^* is the global optimum of the corresponding function. The best error value of each run is recorded when the maximal number of generation is reached. The average and standard deviation of the best error values are also calculated for comparison.

2) *Convergence:* The convergence graphs for some typical functions are plotted to illustrate the mean error performance of the best solution over the total run in the respective experiments.

3) *Statistics by Wilcoxon and Friedman tests [31]–[34]:* To show the significant differences between two algorithms on single problem, Wilcoxon signed-rank test [31]–[34] at 5% significance level is conducted. The result of the test is represented as "$w/t/l$," which means that one algorithm is significantly better than, equal to and worse than the corresponding competitor on w, t, and l functions, respectively. The significantly better values in terms of mean solution error between MS-DE and its competitor are highlighted in boldface in tables. To identify differences between pair of algorithms on all problems, the multiproblem Wilcoxon signed-rank test is carried out. The Friedman test, conducted by the KEEL software [34], is used to obtain the rankings of multiple algorithms (more than two algorithms) on all problems.

Due to the space limit, all numerical values (solution errors) of the simulations are presented in the supplemental file (available at http://ieeexplore.ieee.org). Statistics summarizing those numerical values are shown in Tables I–V.

B. Comparison With Original DE Algorithms

MS-DE is compared with the six original DE algorithms presented earlier (i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/current-to-best/1, and DE/rand-to-best/1) on the 28 benchmark functions at 30D and 50D. Table I provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms corresponding DE algorithm on most of functions. Then, multiproblem Wilcoxon signed-rank test is carried out to identify differences between each pair of algorithms on all problems [33]. The results are also summarized in Table I. It is clear that MS-DE obtains higher $R+$ values than $R-$ values in all cases. Further, the p values of all cases are less than 0.05, which indicates that MS-DE is significantly better than its corresponding DE algorithm.

The overall results of Table I clearly show that MS-DE can improve the performance of the original DE for both explorative and exploitative mutation strategies. The improvements on the mutation operators with the best vector, i.e., DE/best/1, DE/best/2, DE/current-to-best/1, and DE/rand-to-best/1, on different kinds of functions are more obvious. The reason is that the best individual is redefined. Additionally, the entire population is guided by the different best vectors randomly selected from the first front instead of a single globally best vector, which greatly improves the diversity of population.

TABLE I
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH ORIGINAL DE ALGORITHMS FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Alogrithm at 30D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	17/4/7	288.5	89.5	1.60E-02	Yes	Yes
MS-DE/rand/2 *vs* DE/rand/2	20/7/1	348	58	8.92E-04	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	22/2/4	306	72	4.76E-03	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	17/8/3	296	92	1.92E-02	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	21/5/2	322	56	1.34E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	20/4/4	308	70	4.09E-03	Yes	Yes
Alogrithm at 50D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	12/9/7	280	98	2.79E-02	Yes	Yes
MS-DE/rand/2 *vs* DE/rand/2	22/5/1	383.5	22.5	3.80E-05	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	23/2/3	327	51	8.76E-04	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	19/4/5	292.5	113.5	4.04E-02	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	22/3/3	321	57	1.46E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	20/5/3	334	44	4.73E-04	Yes	Yes

TABLE II
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH ADVANCE DE VARIANTS FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Alogrithm at 30D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-ODE *vs* ODE	14/7/7	262	116	7.74E-02	No	Yes
MS-jDE *vs* jDE	10/14/4	273	105	4.23E-02	Yes	Yes
MS-SaDE *vs* SaDE	18/5/5	338.5	39.5	3.14E-04	Yes	Yes
MS-CoDE *vs* CoDE	18/5/5	391	15	1.80E-05	Yes	Yes
MS-OXDE *vs* OXDE	17/4/7	304.5	101.5	2.02E-02	Yes	Yes
Alogrithm at 50D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-ODE *vs* ODE	11/11/6	264	114	6.89E-02	No	Yes
MS-jDE *vs* jDE	12/10/6	229.5	148.5	3.25E-01	No	No
MS-SaDE *vs* SaDE	21/2/5	346	32	1.54E-04	Yes	Yes
MS-CoDE *vs* CoDE	20/3/5	308.5	69.5	3.83E-03	Yes	Yes
MS-OXDE *vs* OXDE	14/8/6	280	126	7.76E-02	No	Yes

TABLE III
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH PRODE AND RANK-DE, RESPECTIVELY, FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Mutation	**Alogrithm at 30D**	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
DE/rand/1	ProDE	11/10/7	244	134	1.81E-01	No	No
	rank-DE	12/9/7	239	167	4.06E-01	No	No
DE/rand/2	ProDE	16/9/3	269	137	1.28E-01	No	No
	rank-DE	20/5/3	305.5	100.5	1.87E-02	Yes	Yes
DE/best/1	ProDE	22/3/3	329	49	7.37E-04	Yes	Yes
	rank-DE	14/8/6	285.5	120.5	5.88E-02	No	Yes
DE/best/2	ProDE	12/8/8	199.5	178.5	9.09E-01	No	No
	rank-DE	12/10/6	220	158	4.48E-01	No	No
DE/rand-to-best/1	Pro-DE	21/3/4	309	69	3.79E-03	Yes	Yes
	rank-DE	21/4/3	337.5	68.5	2.05E-03	Yes	Yes
DE/current-to-best/1	ProDE	19/4/5	292	86	1.29E-02	Yes	Yes
	rank-DE	20/2/6	304	102	2.08E-02	Yes	Yes
Mutation	**Alogrithm at 50D**	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
DE/rand/1	ProDE	8/15/5	222.5	155.5	4.14E-01	No	No
	rank-DE	10/11/7	288.5	117.5	5.02E-02	No	Yes
DE/rand/2	ProDE	21/6/1	348	58	9.22E-04	Yes	Yes
	rank-DE	20/8/0	324	54	1.13E-03	Yes	Yes
DE/best/1	ProDE	23/1/4	330	48	6.75E-04	Yes	Yes
	rank-DE	22/3/3	329	49	7.37E-04	Yes	Yes
DE/best/2	ProDE	14/6/8	226.5	179.5	5.85E-01	No	No
	rank-DE	15/3/10	247	131	1.60E-01	No	No
DE/rand-to-best/1	ProDE	22/3/3	309	69	3.79E-03	Yes	Yes
	rank-DE	22/3/3	337.5	68.5	2.05E-03	Yes	Yes
DE/current-to-best/1	ProDE	20/4/4	333	45	5.17E-04	Yes	Yes
	rank-DE	19/5/4	326	52	9.22E-04	Yes	Yes

With respect to the features of the benchmark functions, a close inspection of the numerical values of the simulations presented in the supplemental file indicates that MS-DE versions can find better results for most of cases in different kinds of functions. In the case of unimodal functions ($F_1 - F_5$), MS-DE versions greatly improve the performances of original DE/best/1, DE/current-to-best/1, DE/rand/2, and DE/rand-to-best/1. MS-DE versions obtain even one order of magnitude

TABLE IV
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH SELECTED DE ALGORITHMS AND ADVANCED DE VARIANTS, RESPECTIVELY, ON CEC 2012 FUNCTIONS AT 1000D

Alogrithm at 1000D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/2 *vs* DE/rand/2	17/2/1	194	16	8.34E-04	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	18/1/1	210	0	8.20E-05	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	16/4/0	159	31	9.09E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	18/1/1	207	3	1.30E-04	Yes	Yes
MS-jDE *vs* jDE	10/3/7	165	45	2.39E-02	Yes	Yes
MS-SaDE *vs* SaDE	12/3/5	158	52	4.58E-03	Yes	Yes

TABLE V
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH CORRESPONDING DE ALGORITHMS ON REAL-WORLD PROBLEMS

Alogrithm	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	5/7/0	42.5	23.5	3.74E-01	No	No
MS-DE/rand/2 *vs* DE/rand/2	4/8/0	57.5	8.5	2.62E-02	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	8/4/0	61.5	4.5	9.93E-03	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	8/1/3	39	27	5.63E-01	No	No
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	7/4/1	58.5	7.5	2.08E-02	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	4/4/4	40.5	37.5	8.75E-01	No	No
MS-ODE *vs* ODE	4/8/0	66	12	2.94E-02	Yes	Yes
MS-jDE *vs* jDE	4/6/2	55.5	22.5	1.82E-01	No	No
MS-SaDE *vs* SaDE	8/3/1	68.5	9.5	1.75E-02	Yes	Yes
MS-CoDE *vs* CoDE	10/2/0	76.5	1.5	2.87E-03	Yes	Yes
MS-OXDE *vs* OXDE	6/5/1	59	19	1.08E-01	No	No

improvement for $F_1 - F_3$. In the cases of multimodal functions and hybrid composition functions ($F_6 - F_{28}$), MS-DE versions also obtain better performance on most of functions. MS-DE versions obtain one order of magnitude improvement on F_6, F_7, F_{10}, F_{11}, F_{19}, and F_{28}. Since MS-DE versions consider both fitness and diversity information, a good balance between exploration and exploitation can be achieved. Thus, MS-DE versions can obtain good results for most of cases in unimodal problems, multimodal problems, and composition problems. Detailed discussion about the search behavior of MS-DE will be given in Section IV-G.

C. Comparison With Advanced DE Variants

To further evaluate the effectiveness of proposed framework, MS-DE is then applied to five state-of-the-art advanced DE variants, including ODE [30], jDE [29], SaDE [10], CoDE [11], and OXDE [35]. These advanced DE variants introduce different kinds of modifications on initialization, control parameters, operators (including ensemble of mutation, and orthogonal crossover), and their hybrid (see ensemble of mutation and parameter adaptation in SaDE) into DE, respectively. They have different characteristics and all obtain very promising results. The comparisons of MS-DE with the corresponding advanced DE variants are carried out on 28 benchmark functions with 30D and 50D. All parameter settings of the advanced DE variants are kept the same as the original papers. The results are shown in the supplemental file.

Table II provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding advanced DE variants on most of functions. From the multiproblem Wilcoxon signed-rank test, it is clear that MS-DE obtains higher $R+$ values than $R-$ values in all cases. It means that MS-DE is better than its corresponding advanced DE variant for all functions. Additionally, for the functions at 30D, the p values are less than 0.05 in four cases (except in ODE case), while the p values are less than 0.1 in all cases. For the functions at 50D, the p values of two cases (MS-SaDE *vs* SaDE and MS-CoDE *vs* CoDE) are less than 0.05, while the p values in all cases, except for MS-jDE *vs* jDE, are less than 0.1. These results indicate that MS-DE is significantly better than most of its corresponding DE variants according to the multiple-problem statistical analysis.

With respect to the features of the benchmark functions, a close inspection of the results in the supplemental file also indicates that MS-DE versions can find better results for most of unimodal, multimodal, and composition functions. For solving unimodal functions, more exploitation is beneficial, while more exploration is beneficial for solving multimodal functions. MS-DE versions consider both fitness and diversity information for parent selection, where the fitness is associated with exploitation and the diversity helps exploration. Thus, MS-DE versions can simultaneously deal with most of unimodal, multimodal, and composition functions.

In summary, MS-DE can improve the performance of the advanced DE variants. It means that the proposed scheme in mutation operator can cooperate with other different kinds of modifications to further improve the performance of DE. This provides a promising approach for solving benchmark and real-world problems.

D. More Comparison of MS-DE Versions

In this section, we firstly compare different MS-DE versions. Then the best overall performing MS-DE versions are further compared with other state-of-the-art algorithms.

The Friedman test is used to obtain the rankings of different MS-DE versions on all problems. The final rankings of all MS-DE versions for all functions are shown in the supplemental file. Overall, MS-SaDE gets the first rank, followed

by MS-OXDE and MS-jDE for all functions at $D = 30$, while MS-jDE gets the first rank, followed by MS-SaDE and MS-OXDE for all function at $D = 50$.

Note that the main contribution of this paper is to propose a multiobjective sorting-based parent selection scheme, and not to propose a "Best" algorithm or competitor to defeat other state-of-the-art algorithms. However, to provide additional comparison for reference, the best overall performing algorithms, MS-SaDE for $D = 30$ and MS-jDE for $D = 50$ are selected to further compare with five selected state-of-the-art algorithms from CEC 2013 competition [26]. The selected algorithms include: two memetic algorithms, named DRMA-LSCh-CMA [43] and MVMO-SH [44], a hybrid CMA-ES, named CMA-ES-RIS [45], a hybrid PSO and artificial bee colony (ABC) algorithm, named ABC-SPSO [46], and a advanced genetic algorithm, named GA-TPC [47]. These algorithms are very powerful algorithms with different features, and thus can be seen as representatives of the state-of-the-art algorithms for comparison.

The results for the functions at $D = 30$ and $D = 50$ are shown in the supplemental file. The Friedman test is used to obtain the ranking of different algorithms on all problems [33], [34]. Overall, MS-SaDE gets the third rank for the functions at $D = 30$. It outperforms CMA-ES-RIS, ABC-SPSO and GA-TPC, but is outperformed by two memetic algorithms, DRMA-LSCh-CMA and MVMO-SH. MS-jDE also gets the third rank for the functions at $D = 50$. It is also outperformed by two memetic algorithms and outperforms remaining algorithms. Note that DRMA-LSCh-CMA and MVMO-SH are advanced memetic algorithms combing advanced evolutionary algorithm and sophisticated local search. It is not surprising that they obtain superior results. MS-DE versions are pure EAs, and their performance can be greatly enhanced if the sophisticated local search is introduced.

E. Comparison With ProDE and Rank-DE

In this section, the proposed algorithm is compared with rank-DE and ProDE. Comparison of MS-DE with rank-DE also helps to show the contribution or effect of diversity information introduced in MS-DE.

The experimental comparisons of MS-DE with ProDE and rank-DE are carried out on 28 benchmark functions with 30D and 50D. All six DE mutation strategies are used in three frameworks, and all statistics are summarized in Table III. From Table III, it is clear that MS-DE obtains the significantly better results than ProDE and rank-DE in most of the cases. In some cases, the differences are more significant. For example, with DE/rand/2, MS-DE significantly outperforms rank-DE on 20 functions without losing any function; and with DE/best/1, MS-DE significantly outperforms rank-DE on 22 functions, and is outperformed by it on only three functions.

In order to further show the significant differences between MS-DE and two related frameworks, Friedman test is carried out and the results are shown in the supplemental file. With respect to the average rankings of different algorithms by Friedman test, MS-DE consistently obtains the best rankings in all six cases for the functions at 30D and 50D.

Moreover, the multiproblem Wilcoxon signed-rank test is also conducted between MS-DE and the two related frameworks on all problems for each case. The results in Table III show that MS-DE obtains higher $R+$ values than $R-$ values in all cases. For the functions at 30D, according to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in three cases between MS-DE and ProDE, and in three cases between MS-DE and rank-DE. According to the Wilcoxon test at $\alpha = 0.1$, significant differences between MS-DE and ProDE are observed in three cases, while in four cases between MS-DE and rank-DE. For the functions at 50D, according to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in four cases between MS-DE and ProDE, and in four cases between MS-DE and rank-DE. According to the Wilcoxon's test at $\alpha = 0.1$, significant differences between MS-DE and ProDE are observed in four cases, while in five cases between MS-DE and rank-DE. The results of Table III indicate that MS-DE is significantly better than ProDE and rank-DE in most of the cases based on the multiple-problem statistical analysis.

As shown in [18] and [19], rank-DE and ProDE are mainly able to enhance the exploitation ability for unimodal functions, but they may be over-exploitative and lead to premature convergence to the local optima in the multimodal problems especially for exploitative mutation operators. MS-DE versions can obtain better results for most of cases in different kinds of functions since they can achieve the balance between exploration and exploitation.

The superior performance of MS-DE attributes to the good balance between exploration and exploitation. The balance is achieved by considering both fitness and diversity information for parent selection. The balance between exploration and exploitation is a core of all kinds of EAs. The idea of MS-DE can be seen as a general and fundamental scheme to select promising or healthy individuals, which can be extended to further improve other EAs.

F. Computation Cost of MS-DE

As discussed in Section IV-D, MS-DE has additional computation burden compared with the original DE. Here, we compare the average runtime between MS-DE and the corresponding DE algorithms on the 28 functions in CEC 2013 at 30D and 50D. The results are shown in the supplemental file. The runtime value for each function means the average overhead of all DE algorithms considered in this paper on the corresponding function. The ratio value is defined as the value that the cost of MS-DE is divided by that of the original DE.

For the functions at 30D, the ratio values are higher than 2.5 for all unimodal and basic multimodal functions except for $F9$ and $F17$. The evaluation for these functions is not costly; thus, the runtime of MS-DE mainly comes from updating the pairwise distance and nondominated sorting. On the hybrid composition functions $F21 - F28$, most of the ratio values, however, are lower than 2. The evaluations of $F15 - F25$ are costly, thus the overhead of calculating pairwise distance and nondominated sorting becomes trivial.

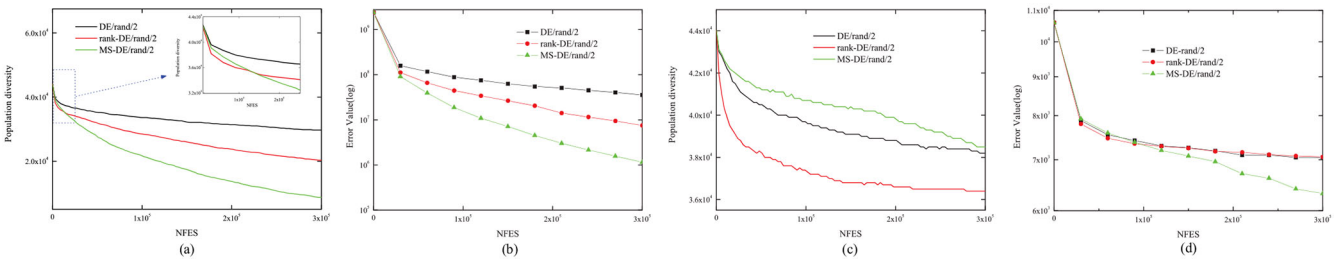


Fig. 2. Evolution graphs of diversity and fitness during the search progress in DE/rand/2, rank-DE/rand/2, and MS-DE/rand/2 on unimodal function F2, and multimodal function F22. (a) Diversity on F2. (b) Fitness on F2. (c) Diversity on F22. (d) Fitness on F22.

For the functions at 50D, most of the ratio values are lower than 2. Compared with the functions at 30D, higher dimensional functions are more difficult to be solved, which leads to a decrease in the frequency of updating the pairwise distance. Therefore, the ratio values are not as high as those of the functions at 30D.

Overall, the average runtime ratio values are 2.592 and 1.991 for the functions in CEC 2013 at 30D and 50D, respectively, which are acceptable for the practical applications.

MS-DE has almost the same computational complexity with ProDE since both approaches have to update the pairwise distance.

In order to further reduce computational effort, on the one hand, the diversity measure of each individual can be computed approximately. For example, it can be updated every 20 generations instead of every generation. On the other hand, other cheaper diversity measure can be designed in a cleverer fashion without the need of calculating pairwise Euclidean distance in future work.

G. Discussion on Search Behavior of MS-DE

In this section, the search mechanism of MS-DE is briefly explained, then the experimental study is given to visually show the search behavior of MS-DE. Finally, how to control the search behavior and further improve the performance of MS-DE are discussed. These discussions on search behavior of MS-DE versions can show, to some extent, how and why the proposed selection helps DE algorithms to find better results.

In original DE, the distribution information is learned by sampling random difference vectors from the population. It is argued in [20] that DE may sample the local topology of the function better, and thus improves its efficiency by learning distribution information from elite difference vectors in the population. This idea is originally inspired by CMA-ES [48]. In order to focus on elite difference vectors, a fitness rank-based parent selection scheme is used to restrict the set of parents to elite individuals in the population [19], [20]. The difference vectors that are likely to be selected with and without fitness-biased selective pressure are shown [20], and thus the effect of fitness-biased selection is analyzed and discussed. It is argued that the fitness-biased difference vectors describe the function topology better and with fewer members. Similarly, the base vector in mutation operator is more likely to be in a fitter region of the space. Thus, fitness-biased selection helps to focus the search, which contributes to the performance improvement over original DE. However, one inherent danger with high fitness-biased selective pressure is the rapid loss of diversity within the population. It is clearly shown in [20] that the rank-based variants lead to a lower population dispersion which benefits differential mutation by focusing the search. That is, the increase of exploitative power comes at a cost to potential exploration [19]. This raises some important questions regarding stagnation.

In fact, the worth of an individual is not considered to rely only on its fitness value. When mating with other individuals, the ability of an individual to produce offspring dispersed enough in the solution space is also very important. Keep this in mind and to mitigate the risk of rank-DE using only fitness to select parent, we introduce an additional diversity measure and thus fitness-and-diversity biased selection is proposed. In MS-DE, the elite difference vectors to be focused on are redefined in terms of both fitness and diversity. It is expected that the increase of exploitative power does not necessarily come at a cost to potential exploration. That is, MS-DE can still focus the search using the fitness information while avoiding rapid loss of diversity or trying to maintain diversity of the population by using diversity information. Since MS-DE versions consider both fitness and diversity information to select promising or healthy individuals, a good balance between exploration, and exploitation can be achieved.

Two representative mutation strategies, DE/rand/2, and DE/best/1, are selected to visually show and compare the evolution behaviors among original DE algorithms, rank-DE, and MS-DE versions. DE/rand/2 is the most explorative mutation operator since a random base vector and two random difference vectors are used. DE/best/1 is the most exploitative mutation operator since the best vector is selected as base vector and only one difference vector is used. These mutation strategies have different exploration/exploitation features. Fig. 2 shows the evolution of diversity and fitness in DE/rand/2, rank-DE/rand/2, and MS-DE/rand/2 on unimodal function F_2, and multimodal function F_{22}, respectively. Fig. 3 shows the evolution of diversity and fitness in DE/best/1, rank-DE/best/1, and MS-DE/best/1 on unimodal function F_2, and multimodal function F_{22} at $D = 30$, respectively. The convergence graphs are used to depict the evolution of fitness. The evolution graphs of diversity illustrate the evolution of the mean population diversity over all runs, where the population diversity PD is the sum of all individuals' diversity in

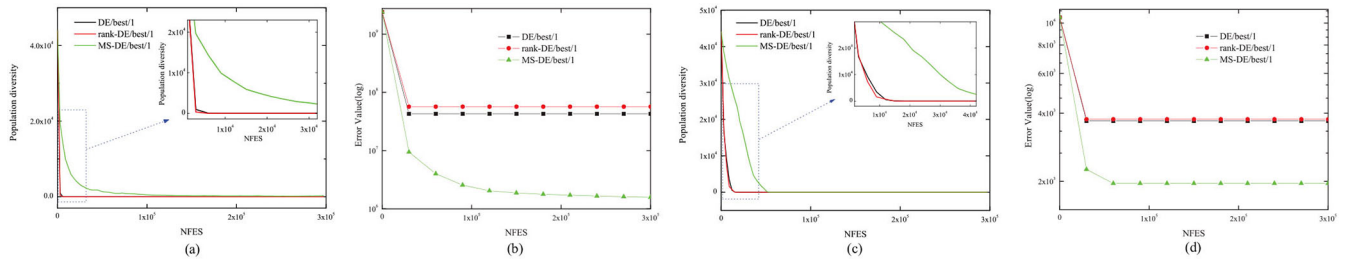


Fig. 3. Evolution graphs of diversity and fitness during the search progress in DE/best/1, rank-DE/best/1, and MS-DE/best/1 on unimodal function F2, and multimodal function F22. (a) Diversity on F2. (b) Fitness on F2. (c) Diversity on F22. (d) Fitness on F22.

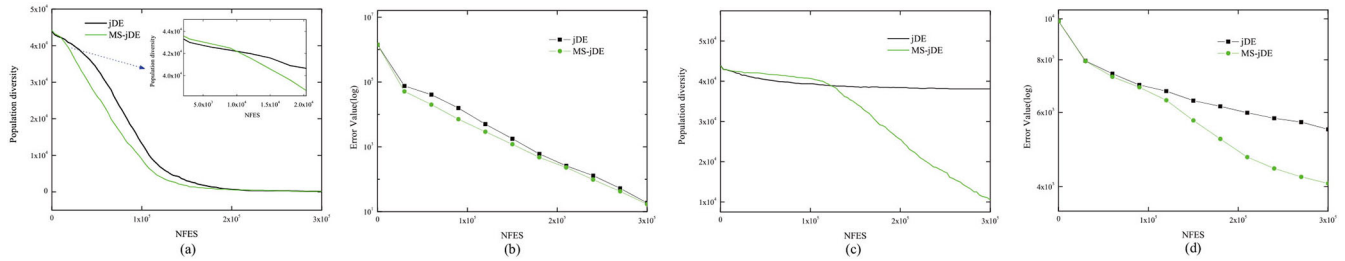


Fig. 4. Evolution graphs of diversity and fitness during the search progress in jDE and MS-jDE on unimodal function F4, and multimodal function F23. (a) Diversity on F4. (b) Fitness on F4. (c) Diversity on F23. (d) Fitness on F23.

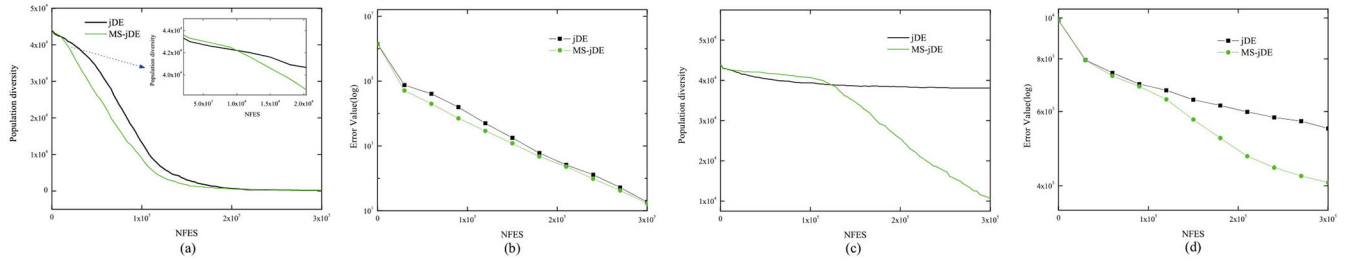


Fig. 5. Evolution graphs of diversity and fitness during the search progress in OXDE and MS-OXDE on unimodal function F4, and multimodal function F6. (a) Diversity on F4. (b) Fitness on F4. (c) Diversity on F6. (d) Fitness on F6.

the population, i.e., $PD = \sum_{i=1}^{NP} \Psi_i$. The diversity evolution graphs combined with the corresponding fitness convergence graphs on both unimodal and multimodal functions can show the behavior of the algorithms comprehensively.

For explorative mutation DE/rand/2, MS-DE version maintains higher diversity than rank-DE at early stage on unimodal function F_2 in Fig. 2(a) and (b), and then rapidly converges to better final result with decreasing the diversity. On multimodal function F_{22} in Fig. 2(c) and (d), MS-DE version maintains higher diversity than rank-DE all the time, and obtains better results finally. Compared with original DE, MS-DE version leads to a lower diversity on unimodal function F_2 for exploitation, and leads to a higher diversity on multimodal function F_{22} at early stage for exploration. In contrast, rank-DE version leads to a much lower diversity on all functions compared with original DE. This may lead to worse result. For exploitative mutation DE/best/1, MS-DE version maintains higher diversity than original DE and rank-DE version, and obtains the best results on both unimodal and multimodal functions. The best individual in DE/best/1 is redefined in terms of fitness and diversity, and entire population is guided by the different best vectors randomly selected from the first front instead of a single globally best vector. This, combined with the elite difference vector redefined, greatly improves the diversity of population, which promotes the exploration ability of the exploitative mutation to escape from local minima. In contrast, rank-DE leads to a lower population diversity on both unimodal and multimodal functions. Thus, rank-DE obtains worse results than DE/best/1 for unimodal function F_2 and multimodal functioin F_{22} due to over-exploitation. These results support our explanation that MS-DE can focus the search using fitness information, avoiding rapid loss of diversity or trying to maintain diversity using diversity information.

For advanced DE variants, two representatives, jDE and OXDE, are selected because the results for the others are similar. Fig. 4 shows the evolution of diversity and fitness in jDE and MS-jDE on unimodal function F_4, and multimodal function F_{23} at $D = 30$, respectively. Fig. 5 shows the evolution of diversity and fitness in OXDE and MS-OXDE on unimodal function F_4, and multimodal functions F_6 at $D = 30$, respectively. Compared with the corresponding advanced DE variants, MS-DE versions lead to lower or higher population diversity on different functions or at different search stages on the same function, trying to achieve better balance of exploration and exploitation for better results. For example, MS-jDE maintains higher diversity at early stage for

exploration and then leads to lower diversity for exploitation or convergence on unimodal function F_4 and multimodal function F_{23}. MS-OXDE leads to lower diversity on unimolal function F_4 for exploitation, and maintains higher diversity at the later stage on the multimodal function F_6 for exploration.

Note that we should not expect that the diversity of population in MS-DE versions can necessarily be maintained to a much higher level. The reason is that, whatever mating selection is used, all generated trial vectors will be filtered by the same fitness based greedy survival selection to form the population of next generation. The diverse individuals with bad fitness have no chance of survival. Both DE and the corresponding MS-DE version try to find the single global optimum. Thus, the diversity evolution in both DE algorithms and the corresponding MS-DE versions shows similar decreasing trend for convergence. Since the diversity used in MS-DE versions is a helper objective, the proposed selection method can be seen as a diversity maintenance method [21].

Different problems, or different stages during the search progress on the same problem, need different ratios between exploration and exploitation [21]. The proposed parent selection scheme, as a simple diversity maintenance method, can not deal with all kinds of problems well. Hence, MS-DE can not obtain the best results on all unimodal and multimodal functions as shown in previous simulation results. To further improve the performance of MS-DE on more difficult functions, two research lines can be followed in future. Firstly, diversity control, diversity learning and some direct approaches can be adopted to promote the diversity of MS-DE to control exploration and exploitation [21]. For example, diversity can be injected or created by using a restart mechanism, or additional mutation. But this will lead to slower convergence because high diversity does not necessarily mean that a diverse population is obtained by a good balance between exploration and exploitation. Hence, local search should be introduced to speed convergence. Secondly, the diversity and fitness information defined in this paper can be further and fully exploited to improve the performance of MS-DE. For example, they can be used to estimate and thus monitor the real-time evolutionary states of individuals and population in each generation, as in adaptive DEs [42], [49] and adaptive PSOs [50], [51]. Then, the search behavior of each individual, determined by the parameters and operators, in MS-DE can be adaptively controlled according to the real-time evolutionary states estimated. Thus, different individuals can play difference roles (exploitation or exploration) during the search progress, and even the same individual can play different roles during different search progress.

H. Scalability Study on the CEC 2012 Optimization Problems

In order to study the scalability, MS-DE is tested on the set of 20 benchmark functions at $1000D$ from the CEC 2012 special session on large scale global optimization [27]. These functions, denoted as $LF1$-$LF20$, are classified into five types of functions: separable ($LF1$-$LF3$), single-group m-nonseparable ($LF4$-$LF8$), $D/2m$-group m-nonseparable ($LF9$-$LF13$), D/m-group m-nonseparable ($LF14$-$LF18$) and Nonseparable functions ($LF19$-$LF20$). Group size m is set to 50, and more details of the 20 functions can be found in [27]. In this experiment, 25 runs are performed independently for each problem, and *MNFEs* is set to 3×10^6 [27]. Evaluation and experiment of algorithms on these large scale benchmark functions are extremely time-consuming, hence several representative algorithms are chosen in this experiment due to time limit. Here, four representative original DE algorithms, including explorative mutation DE/rand/2, and exploitative mutation DE/best/1, DE/current-to-best/1, and DE/rand-to-best/1, and two representative advanced DE variants including jDE and SaDE are used to provide a deeper analysis about the scalability. These mutation strategies have different exploration/exploitation features. The MS-DE versions of jDE and SaDE are the overall best algorithms for the CEC 2013 functions. Hence, the selected algorithms can be seen as representatives of DE variants for testing. The results are shown in the supplemental file.

Table IV provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding DE algorithm on most of functions. MS-DE obtains higher $R+$ values than $R-$ values in all cases. Further, the p values of all cases are less than 0.05, which indicates that MS-DE is significantly better than its corresponding DE algorithm. Hence, we can conclude that MS-DE brings benefit to the four original DE algorithms and two advanced DE variants for solving the large scale optimization problem at $1000D$.

Here, these functions are used just to study the scalability of the MS-DE, and the main aim of this paper is not to propose MS-DE to solve large scale global optimization. In order to further improve the performance of MS-DE for large-scale global optimization, other advanced techniques, see parallel mechanism [36] and cooperative coevolution [37], can be introduced into or imposed on MS-DE. Recent work [37] shows that it is possible to make a given high performance evolutionary optimizer scale well to high dimensional problems by using it as a subcomponent optimizer in a cooperative coevolution framework with an automatic decomposition strategy such as differential grouping [37].

I. Application to Real-World Problems

In this section, we evaluate the potential of MS-DE for real-world application problems selected from the CEC 2011 competition on testing evolutionary algorithm on real-world numerical optimization problems [28]. There are total 22 (by taking different instance of problems as a separate one) real-world numerical optimization problems including both unconstrained and constrained optimization problems. This paper considers the selected 12 unconstrained problems: parameter estimation of frequency-modulated sound waves (T1), Lennard-Jones potential (T2), optimal control of a nonlinear stirred tank reactor (T4), Tersoff potential function minimization problem-Si(B) (T5.1), Tersoff potential function minimization problem-Si(C) (T5.2), spread spectrum radar polly phase code design (T6), transmission network expansion

planning (T7), static economic load dispatch-instance 2 (T11.2), static economic load dispatch-instance 3 (T11.3), static economic load dispatch-instance 4 (T11.4), Messenger: spacecraft trajectory optimization (T13), and Cassini 2: spacecraft trajectory optimization (T14). Details about these 12 problems can be referred to [28].

In this section, six original DE algorithms and five advanced DE variants mentioned above, and their MS-DE versions are all applied to the selected real-world problems for comparison. The *MNFEs* for these problems is set to 150 000, and 25 independent runs are implemented for each problem [28]. The results are shown in the supplemental file.

Table V provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding DE algorithm on most of the real-world application problems. Moreover, MS-DE obtains higher $R+$ values than $R-$ values in all cases. According to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in six cases between DE algorithms and their MS-DE version. The results of Table V indicate that MS-DE is significantly better than original DE algorithms and advanced DE variants in most of the cases.

The final rankings of all MS-DE versions for all real-world problems are shown in the supplemental file. Overall, MS-SaDE gets the first rank, followed by MS-OXDE and MS-jDE.

V. Conclusion

This paper presents a DE framework with multiobjective sorting-based mutation operator. In the proposed mutation operator, individuals in the current population are first sorted according to their fitness and diversity information by nondominated sorting. Then parents in the mutation operators are proportionally selected according to their rankings based on fitness and diversity. Since fitness and diversity information is simultaneously considered to select the parents, a good balance between exploration and exploitation can be achieved. The proposed operator is applied to the original DE algorithms, as well as several advanced DE variants. Experimental results on benchmark functions and real-world application problems show that the proposed operator is an effective approach to enhance the performance of most of the DE algorithms studied. The scalability of MS-DE is also discussed. Through the simulation results on benchmark functions and real-world problems, MS-jDE, MS-SaDE, and MS-OXDE, among all MS-DE versions studied, perform well overall in terms of both simplicity and efficiency.

As a new DE framework, there are still many interesting issues worth further studying in MS-DE. Firstly, to reduce computational effort of MS-DE framework, the diversity measure can be designed in a cleverer fashion without the need of calculating pairwise Euclidean distance. Other kinds of cheaper diversity measures can also be introduced. Secondly, the fitness and diversity information defined in this paper can be further used to estimate and monitor the real-time evolutionary states of individuals and population in each generation. Thus, the search behavior of each individual in MS-DE can be adaptively controlled according to the real-time evolutionary states estimated. Thirdly, MS-DE can be extended to solve constraint optimization problems. Finally, the multiobjective sorting-based selection scheme can be seen as a general mating selection scheme simultaneously considering the fitness and diversity. Since mating selection operator is important to most of EAs, the proposed approach can be used to further improve other EAs [38], [39].

References

[1] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[2] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Secaucus, NJ, USA: Springer-Verlag, 2005.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[4] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.

[5] A. Iorio, and X. Li, "Improving the performance and scalability of differential evolution on problems exhibiting parameter interactions," *Soft Comput.*, vol. 15, no. 9, pp. 1769–1792, Sep. 2011.

[6] J.-B. Xiao and J. Xiao, "Classification-based self-adaptive differential evolution with fast and reliable convergence performance," *Soft Comput.*, vol. 15, no. 8, pp. 1581–1599, Jan. 2011.

[7] B. Dorronsoro and P. Bouvry, "Improving classical and decentralized differential evolution with new mutation operator and population topologies," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 67–98, Feb. 2011.

[8] H. Wang, S. Rahnamayan, S. Hui, and M. G. Omran, "Gaussian barebones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.

[9] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.

[10] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[11] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[12] P. Kaelo and M. Ali, "A numerical study of some modified differential evolution algorithms," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 1176–1184, 2006.

[13] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[14] J. Zhang and A. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[15] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[16] R.-J. He and Z.-Y. Yang, "Differential evolution with adaptive mutation and parameter control using Levy probability distribution," *J. Comput. Sci. Tech.*, vol. 27, no. 5, pp. 1035–1055, 2012.

[17] C. García-Martínez, F. Rodríguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: An analysis on large-scale optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2109–2126, Nov. 2011.

[18] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.

[19] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.

[20] A. M. Sutton, M. Lunacek, and L. D. Whitley, "Differential evolution and non-separability: Using selective pressure to focus search," in *Proc. 9th Annu. Conf. GECCO*, England. U.K., Jul. 2007, pp. 1428–1435.

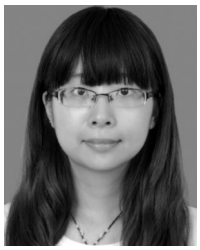
[21] M. Crepinsek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM CSUR*, vol. 45, no. 3, Article 35, 2013.
[22] G. Corriveau, R. Guilbault, A. Tahan, and R. Sabourin, "Review of phenotypic diversity formulations for diagnostic tool," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 9–26, 2013.
[23] G. Corriveau, R. Guilbault, A. Tahan, and R. Sabourin, "Review and study of genotypic diversity measures for real-coded representations," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 695–710, Oct. 2012.
[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[25] Y.-S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.
[26] J. J. Liang, B. Y. Qu, P. N. Suganthan, and G. H.-D. Alfredo, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Tech. Univ., Singapore, Tech. Rep. 201212, Jan. 2013.
[27] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large scale global optimization," Nature Inspired Comput. Appl. Lab., USTC, China, Tech. Rep., 2009.
[28] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," Jadavpur Univ., India, and Nanyang Tech. Univ., Singapore, Tech. Rep., Dec. 2010.
[29] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[30] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb. 2008.
[31] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.
[32] S. Garcäa, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 16, pp. 617–644, 2009.
[33] J. Derrac, S. Garcäa, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[34] J. Alcalá-Fdez, L. Sánchez, and S. Garcáa, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Oct. 2008.
[35] Y. Wang, Z. Cai, and Q. Zhang. "Enhancing the search ability of differential evolution through orthogonal crossover," *Inf. Sci.*, vol. 185, no. 1, pp. 153–177, 2012.
[36] M. Weber, F. Neri, and V. Tirronen, "Shuffle or update parallel differential evolution for large scale optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2089–2107, Oct. 2011.
[37] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, 2014, to be published.
[38] H. Ma, S. Dan, M. Fei, and Z. Chen, "On the equivalences and differences of evolutionary algorithms," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2397–2407, Nov. 2013.
[39] S. Kern *et al.*, "Learning probability distributions in continuous evolutionary algorithms—A comparative review," *Natural Comput.*, vol. 3, no. 1, pp. 77–112, 2004.
[40] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 107–124, Feb. 2012.
[41] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 397–413, Apr. 2011.
[42] W.-J. Yu *et al.*, "Differential evolution with two-level parameter adaptation," *IEEE Trans. Cybern.*, 2014, to be published.
[43] B. Lacroix, D. Molina, and F. Herrera, "Dynamically updated region based memetic algorithm for the 2013 CEC special session and competition on real parameter single objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1945–1951.
[44] J. L. Rueda and I. Erlich, "Hybrid mean-variance mapping optimization for solving the IEEE-CEC competition problems," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1664–1671.
[45] F. Caraffini, G. Iacca, F. Neri, L. Picinali, and E. Mininno, "A CMAES super-fit scheme for the re-sampled inheritance search," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1123–1130.
[46] M. El-Abd, "Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2215–2220.
[47] S. M. M. Elsayed, R. A. Sarker, and D. L. Essam, "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 356–360.
[48] S. D. Müller, N. Hansen, and P. Koumoutsakos. "Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations," in *Proc. PPSN*, Granada, Spain, 2002, pp. 422–431.
[49] X. Zhou, Z. Wu, H. Wang, and S. Rahnamayan, "Enhancing differential evolution with role assignment scheme," *Soft Comput.*, 2014, to be published.
[50] Z.-H. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.
[51] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 627–646, Jun. 2012.

Jiahai Wang (M'07) received the Ph.D. degree from the University of Toyama, Toyama, Japan, in 2005.

He is currently an Associate Professor at the Department of Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include computational intelligence and its applications.

Jianjun Liao received the B.S. degree from Hunan University of Science and Engineering, Yongzhou, China, in 2011. He is currently pursuing the M.S. degree with Sun Yat-sen University, Guangzhou, China.

His current research interests include differential evolution and evolutionary computation techniques.



Ying Zhou received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2009, where she is currently pursuing the Ph.D. degree.

Her current research interests include local search algorithms, multiobjective optimization, and other evolutionary computation techniques.

Yiqiao Cai received the B.S. degree from Hunan University, Changsha, China, and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2007 and 2012, respectively.

He is currently a Lecturer with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. His current research interests include differential evolution, multiobjective optimization, and other evolutionary computation techniques.

Differential Evolution Enhanced With Multiobjective Sorting-Based Mutation Operators

Jiahai Wang, *Member, IEEE*, Jianjun Liao, Ying Zhou, and Yiqiao Cai

***Abstract*—Differential evolution (DE) is a simple and powerful population-based evolutionary algorithm. The salient feature of DE lies in its mutation mechanism. Generally, the parents in the mutation operator of DE are randomly selected from the population. Hence, all vectors are equally likely to be selected as parents without selective pressure at all. Additionally, the diversity information is always ignored. In order to fully exploit the fitness and diversity information of the population, this paper presents a DE framework with multiobjective sorting-based mutation operator. In the proposed mutation operator, individuals in the current population are firstly sorted according to their fitness and diversity contribution by nondominated sorting. Then parents in the mutation operators are proportionally selected according to their rankings based on fitness and diversity, thus, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. Since fitness and diversity information is simultaneously considered for parent selection, a good balance between exploration and exploitation can be achieved. The proposed operator is applied to original DE algorithms, as well as several advanced DE variants. Experimental results on 48 benchmark functions and 12 real-world application problems show that the proposed operator is an effective approach to enhance the performance of most DE algorithms studied.**

***Index Terms*—Differential evolution, diversity, exploration and exploitation, mutation operator, nondominated sorting.**

I. Introduction

DIFFERENTIAL evolution (DE) [1], [2] is one of the most successful and popular evolutionary algorithms (EAs) for global optimization. The advantages of DE are its easiness of use, simple structure, efficacy, and robustness. Recently, DE is successfully applied to various scientific and engineering fields. Details can be found in recent two reviews of DE [3], [4].

The salient feature of DE lies in its mutation mechanism that distinguishes it from other EAs. In the mutation operator of DE, a mutant vector can be treated as a leading individual to explore the search space and generated by adding a difference vector to a base vector. We have observed, however, on the one hand, both vectors (i.e., base and difference vectors) in most of DE are selected randomly, which does not fully utilize the fitness information. Hence, all vectors are equally likely to be selected as parents without selective pressure at all [3]. On the other hand, the diversity information is always ignored. This will make the population lose diversity. The algorithm is trapped in local optimum more frequently, especially when the optimization problem is multimodal [3], [4]. Hence, the fitness and diversity information of the population is not fully exploited in the designing of DE.

In view of the limitations of previous works, this paper proposes an alternative to the uniform random selection of parents during mutation in original DE. During the selection of parents, the fitness and diversity information is introduced, with the aim of favoring those individuals with both high fitness values and high diversity contributions. In this way, two underlying objectives, fitness and diversity, are both considered simultaneously. The proposed scheme is named as multiobjective sorting-based mutation operator. More specifically, individuals in the current population are first sorted according to their fitness and diversity contribution by nondominated sorting. Then some of the parents in the mutation operators are proportionally selected according to their rankings, thus, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. DE enhanced with multiobjective sorting-based mutation operators is named as MS-DE framework. In order to evaluate the effectiveness of MS-DE, the proposed framework is applied to original DE algorithms, as well as several advanced DE variants. Extensive experiments have been carried out on two sets of benchmark functions and some real-world application problems. The results show that MS-DE is able to enhance the performance of DE for most of the considered problems.

The main contribution of this paper is to select promising or healthy individuals with better fitness and diversity simultaneously as parents in DE mutation. This leads to the major advantages of the proposed algorithm as follows.

1) Since fitness and diversity information is considered by multiobjective sorting to select the parents, a good balance between exploration and exploitation can be achieved. The balance between exploration and exploitation is a core of all kinds of EAs. The proposed

Manuscript received September 1, 2013; revised January 10, 2014, March 23, 2014, and March 29, 2014; accepted April 6, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 60805026, Grant 61070076, and Grant 61305085, and in part by the Zhujiang New Star of Science and Technology in Guangzhou City under Grant 2011J2200093. This paper was recommended by Associate Editor Y. Jin.

J. Wang, J. Liao, Y. Zhou are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wjiahai@hotmail.com; liaojianj@mail2.sysu.edu.cn; dyingdy@gmail.com).

Y. Cai is with the College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China (e-mail: yiqiao00@163.com).

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

Digital Object Identifier 10.1109/TCYB.2014.2316552

approach provides a simple yet efficient way to deal with the balance.
2) The proposed approach is very simple because it keeps the simple structure of the original DE algorithm, and no additional parameter is introduced.
3) The proposed approach can be easily applied to other advanced DE variants. Thus, it cooperates with different kinds of modifications in advanced DE variants to further improve the performance. This provides a promising approach for solving benchmark and real-world problems.
4) The multiobjective sorting-based selection scheme can be seen as a general and fundamental mating selection scheme to select promising or healthy individuals. Since mating selection operator plays an important role in most of EAs, the proposed approach can be used to further improve other EAs.

The rest of this paper is organized as follows. In Section II, DE and related work are reviewed. MS-DE framework is presented in Section III. In Section IV, experimental results are reported. Finally, conclusions are drawn in Section V.

II. Related Work

A. DE

The following global optimization problem is considered:

$$\text{Minimize} \quad f(X) \tag{1}$$

where $X = (x_1, x_2, \ldots, x_D) \in R^D$ is a vector in the D-dimensional decision (variable) space (solution space), and the feasible solution space is $x_i \in [L_i, U_i]$, where L_i and U_i represent the lower and upper bound of the ith parameter, respectively. DE evolves a population of NP candidate individuals (solutions) with D-dimensional parameter vectors [1]. Each individual is denoted as $X_{i,G} = [x_{i,1,G}, x_{i,2,G}, \ldots, x_{i,D,G}]$, where $i = 1, 2, \ldots, NP$, NP is the population size and G is the current generation. It has three main operators: mutation, crossover, and selection.

1) Mutation: This operator generates a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ (named as target vector). The originally proposed and most frequently used mutation strategies in the literature are [1] and [2].

1) DE/rand/1

$$V_{i,G} = X_{i_{1,G}} + F \cdot (X_{i_{2,G}} - X_{i_{3,G}}). \tag{2}$$

2) DE/rand/2

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{i_2,G} - X_{i_3,G}) + F \cdot (X_{i_4,G} - X_{i_5,G}). \tag{3}$$

3) DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{i_1,G} - X_{i_2,G}). \tag{4}$$

4) DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{i_1,G} - X_{i_2,G}) + F \cdot (X_{i_3,G} - X_{i_4,G}). \tag{5}$$

5) DE/current-to-best/1

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{i_1,G} - X_{i_2,G}). \tag{6}$$

6) DE/rand-to-best/1

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{best,G} - X_{i_1,G}) + F \cdot (X_{i_2,G} - X_{i_3,G}). \tag{7}$$

The indices i_1, i_2, i_3, i_4 and i_5 are mutually different random indices chosen from the set $\{1, 2, \ldots, NP\} \setminus \{i\}$. $X_{best,G}$ is the best vector in terms of fitness value at the current generation G. In the mutation defined by (2), the first-term at the right hand of the equation is called base vector, and remaining term is called difference vector. F, called mutation scaling factor, is a positive control parameter for scaling the difference vectors.

In general, we can distinguish between mutation operators that promote exploration (called explorative strategies) and operators that promote exploitation (called exploitative strategies). Operators that incorporate the best individual (see DE/best/1, DE/best/2, DE/current-to-best/1, DE/rand-to-best/1) favor exploitation, since the mutant individuals are attracted around the current best individual. DE/rand-to-best/1 has a stronger exploration capability by introducing more perturbation with the random individual. In contrast, operators that incorporate randomly selected individuals (see DE/rand/1, DE/rand/2) enhance the exploration ability, since a high degree of random variability is introduced.

2) Crossover: When the mutant vector is generated, crossover operator is applied to each pair of target vector $X_{i,j,G}$ and mutant vector $V_{i,j,G}$ to generate a trial vector $U_{i,G} = [u_{i,1,G}, u_{i,2,G}, \ldots, u_{i,D,G}]$. There are two kind of crossover operators, binomial and exponential. The most widely used is the binomial crossover as follows:

$$u_{i,j,G} = \begin{cases} v_{i,j,g}, & \text{if } rand_j(0,1) \le CR \quad \text{or} \quad j = j_{rand} \\ x_{i,j,g}, & \text{otherwise} \end{cases} \tag{8}$$

where $\text{rand}_j(0,1)$ is a uniform random number in (0,1) for jth dimension, $CR \in (0, 1)$ is a predefined crossover control parameter, and $j_{rand} \in (1, D)$ is an integer randomly chosen from 1 to D.

3) Selection: This operator uses a one-to-one greedy scheme to select the better one between target vector $X_{i,G}$ and trial vector $U_{i,G}$ to survive in the next generation as follows:

$$X_{G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \tag{9}$$

B. Improved Mutation Operators

DE has drawn the attention of many researchers [3], [4] due to the attractive characteristics. However, DE is not completely free from problems of stagnation and premature convergence due to the limited amount of exploratory moves [4]. To overcome these problems, a number of DE variants are proposed. According to [4], these modern DE variants can be classified into two categories: DE integrated with an extra component and DE with modified structures. Modifications on DE in these DE variants mainly focus on control parameters (i.e., NP, F and CR), operators (mutation, crossover, and selection), population (multipopulation or parallel population) and hybridization of DE with other operators or algorithms. Compared to many empirical works, few research has so far been devoted to theoretically analyze the search mechanism and convergence properties of DE. Recently, [40]

takes a first significant step toward the convergence analysis of DE.

In this section, we focus on the related work that how the mutation operators can be modified in DE to improve its performance. These modifications can be classified roughly into three categories: proposal of new mutation operator, ensemble of different mutation operators, and selection of vectors in the existing mutation operator.

Several new mutation operators are proposed, for example, rotation-invariant mutation operator [5], DE/rand-to-best/*p*best [6], Gaussian PBX-α [7], and Gaussian mutation [8]. Direction information of the best and worst near neighbors is incorporated into the mutation strategies [9].

Two representatives of algorithms with ensemble of different mutation operators are self-adaptive DE (SaDE) [10] and composite DE (CoDE) [11]. In SaDE, both mutation strategies and their associated control parameter values are gradually self-adapted by learning from their previous experiences in generating promising solutions. It is a learning-based ensemble method. In CoDE, three mutation strategies and three control parameter settings are randomly combined to generate new solutions. It is a random combination-based ensemble method. [41] also proposes a mutation ensemble method. It adopts different existing parameter control methods of DE to adaptively choose a suitable mutation strategy.

Selection of suitable vectors in the existing mutation operator is also an effective approach to improve DE. Tournament best base vector selection for DE is proposed in [12]. Several selection methods for the best individual in DE/current-to-best/1, DE/rand-to-best/1, and DE/best/1 are proposed in [13]–[16] and [42], respectively. These selection methods all aim to appropriately make use of the bias of the best individuals, but they introduce additional control parameters or sophisticated control mechanisms. Role differentiation and malleable mating for DE are proposed in [17]. Recently, two general frameworks for the selection of suitable vectors in the existing mutation operators, called proximity-based mutation in DE (ProDE) [18] and rank-based mutation in DE (rank-DE) [19], are proposed. ProDE advocates a stochastic selection in which the probability of selecting an individual to become a parent is inversely proportional to its distance from the individual undergoing mutation. In rank-DE, some of the parents in the mutation operators are proportionally selected according to their fitness rankings in the current population. The higher ranking a parent obtains, the more frequently it will be selected. Similar rank-based parent selection scheme is also proposed in earlier work [20]. Unlike the selection methods in [12]–[16], and [42] designed for a specific mutation strategy, ProDE and rank-DE are general frameworks and can be applied to all mutation strategies.

III. Multiobjective Sorting-Based Mutation Operators

A. Motivations

In order to be successful, an EA algorithm, for example, DE, needs to establish a good balance between exploration and exploitation [21]. There are some additional characteristics in DE. On the one hand, not all DE search operators have the same impact on the balance of exploration and exploitation [18]. DE search operators can be roughly classified into explorative and exploitative mutation operators as mentioned above. Hence, the choice of the most efficient mutation operator can be cumbersome and problem dependent. On the other hand, most of DE mutation strategies have clustering tendency [18], which easily leads to premature convergence.

Recent works, including rank-DE and ProDE, aim to promote the exploitation ability of DE. Rank-DE increases the selective pressure on fitter solutions in the population, and hence promotes efficient exploitation. But it may be over-exploitative and lead to premature convergence to the local optima in the multimodal problems especially for exploitative mutation operators. Earlier work [20] has pointed out that one inherent danger of rank-based parent selection with high selective pressure is the rapid loss of diversity within the population, and the increase of exploitative power comes at a cost to potential exploration. By favoring search in the vicinity of the mutated individual, ProDE promotes efficient exploitation, while diminishing the exploration abilities of the mutation operator. Therefore, the proximity scheme does not lead to great benefits for exploitative mutation strategies, and sometimes deteriorates their performance. This highlights the importance of a good balance between exploration and exploitation in DE search.

Exploration is associated with the distribution of individuals on a landscape, and can be estimated by a genotypic diversity measure. In contrast, exploitation is related to individual response (fitness), which can be described with phenotypic (fitness) convergence measure [21]–[23]. Since the exploration and exploitation are measured indirectly using diversity and fitness, respectively, balance between exploration and exploitation can be guided or achieved by diversity and fitness. High diversity does not necessarily mean that a diverse population is obtained by a good balance between exploration and exploitation [21]. We are interested in diversity that helps to find fit individuals, i.e., useful diversity or healthy diversity. In rank-DE and ProDE, the diversity information is always ignored. This will make the population lose diversity and thus be trapped in local optimum more frequently. Therefore, in this paper, the fitness and diversity are considered to select parents for mutation operator in DE, with the aim of favoring those promising or healthy individuals with both high fitness values and high diversity contributions. To achieve this, two key problems arise: how to measure the contribution of diversity of an individual to the whole population and how to consider fitness and diversity simultaneously to guide the evolution.

B. Multiobjective Sorting-Based Mutation Operators

1) Fitness and Diversity Measures: The aim of using two measures (objectives) is to select promising or healthy individuals with better fitness and diversity. The fitness is associated with exploitation and the diversity helps exploration. Since greedy convergence and maintenance of diversity (i.e., exploitation and exploration) are often conflicting objectives of search process, we may consider the diversity versus selective pressure (fitness) problem as a multiobjective problem.

Prior to discuss about the multiobjective sorting procedure, one more measure that would have a direct relation with the population diversity for each individual is defined. There are a lot of diversity measures for whole population (i.e., on population level) as reviewed in [23], but there is few diversity contribution measures for a single individual to whole population (i.e., on individual level). Here, an Euclidean distance-based diversity metric for each individual is defined:

$$\Psi_i = \sum_{j=1}^{NP} ||X_i - X_j|| \tag{10}$$

where $i \in (1, NP)$, NP is the population size, $||X_i - X_j|| = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2}$, and D denotes the dimensionality of the search space. Ψ_i equals to the sum of the distances of ith individual from other members in the population. High diversity value means the ith individual is far from other ones in the population. This measures the contribution of diversity of an individual to the whole population.

In fact, the worth of an individual is not considered to rely only on its fitness value. When mating with other individuals, the ability of an individual to produce offspring dispersed enough in the solution space is also very important, because it helps to explore the solution space efficiently and exhaustively. An individual that is distant from the others in the population has more chances, when mating, to produce offspring in the regions of the search space not covered by the current population. Thus, two measures or objectives associated with each individual in the population can be defined as

$$\text{Minimize} \quad f_{fitness}(X_i) = f(X_i) \tag{11}$$

$$\text{Minimize} \quad f_{diversity}(X_i) = -\Psi_i \tag{12}$$

where $f(X_i)$ is the fitness function value of ith individual and Ψ_i is the contribution of diversity of ith individual to the population. The diversity (12) is not the final objective of the single objective optimization problem, and thus can be seen as a helper objective.

2) Nondominated Sorting: To understand the nondominated sorting procedure, some definitions about multiobjective optimization are first briefly reviewed. The concept of dominance can be formally stated in the following way. In general, a multiobjective optimization problem can be state as follows:

$$\text{Minimize} \quad F(X) = \{f_1(X), f_2(X), \ldots, f_M(X)\} \tag{13}$$

where $X = (x_1, x_2, \ldots, x_D) \in R^D$ is a vector in the D-dimensional decision space as in single optimization problem (1), and $F = (f_1, f_2, \ldots, f_M) \in \Omega^M$ is the objective space with M minimization objectives. A solution X dominates Y iff X is at least as good as Y in all objectives, and better in at least one. Two solutions are incomparable or nondominated iff they neither dominate the other.

In this paper, the nondominated sorting procedure [24] is adopted to sort the individuals of a given population according to the level of nondomination. The nondominated sorting procedure mainly consists of two steps [24]. In the first step, for each individual, domination count c_p, the number of individuals which dominate the solution p, and a set of individuals S_p that solution p dominates, are calculated. All individuals in the first nondominated front will have their domination count as zero. In the second step, for each solution p with $c_p = 0$, each member q in its set S_p is visited and its domination count is reduced by one. In doing so, if the domination count becomes zero for any member q, it is added into a separate list Q. These members belong to the second nondominated front. The above procedure is continued with each member of Q and the third front is identified. This process continues until all fronts are identified. Details of the nondominated sorting procedure can be found in nondominated sorting genetic algorithm II (NSGA-II) [24].

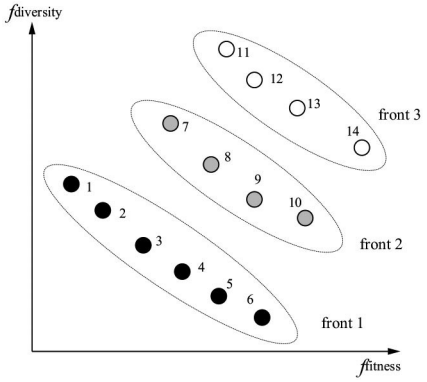


Fig. 1. Typical output of the nondominated sorting procedure.

The nondominated sorting procedure is mainly used to partition the population of individuals into different fronts based on the multiple objectives or measures considered. Fig. 1 illustrates the typical output of the nondominated sorting procedure. The individuals in the lowest layer belong to front 1. The second lowest layer of individuals form front 2, and remaining individuals belong to front 3.

The time complexity of the nondominated sorting is $O(M \cdot NP^2)$, where M is the number of objectives and NP is the population size. Since only two objectives are considered in this paper, the time complexity can be simplified as $O(NP^2)$.

3) Selection Procedure With Nondominated Sorting: Using nondominated sorting procedure, the population is sorted into several fronts, thus we can differentiate the individuals among different fronts. In order to further differentiate the individuals within the same front, an objective (fitness or diversity) is randomly selected to sort the individuals. This sorting mechanism within the same front is simple, and can automatically switch back and forth between fitness-based selection and diversity-based selection. Finally, the population can be fully sorted in ascending order based on the fitness and diversity. Fig. 1 shows the ascending order of 14 individuals in a typical population, where fitness is selected to sort the individuals within the same front. The number or index of an individual in the sorted population shown in Fig. 1 indicates its order.

Thus, the ranking of ith individual (vector) in the sorted population is assigned as follows:

$$R_i = NP + 1 - i, i = 1, 2, \ldots, NP \tag{14}$$

where NP is the population size, i is the index of ith individual in the sorted population. According to (14), better individual

Algorithm 1 MS-DE/rand/1: Multiobjective Sorting Based Mutation for DE/rand/1 /***Mutation Operator***/

1: Compute the fitness and diversity of each individual according to (11) and (12).
2: Sorting the population according to the fitness and diversity of each individual using the nondominated sorting procedure.
3: Calculate the probability vector based on (15).
4: Utilize a roulette wheel to select parents based on the probability vector.
5: For each target vector, generate the corresponding mutant vector using DE/rand/1 mutation (2).

in terms of bi-objective, defined by (11) and (12), in the sorted population will obtain better ranking.

Then, the selection probability of the ith individual for mutation operation is proportional to

$$P_i = \frac{R_i}{NP}, i = 1, 2, \ldots, NP. \tag{15}$$

Hence, the promising individuals with better fitness and diversity have more opportunity to be selected as parents. Note that the simplest linear model is adopted to compute the selection probability vector based on the ranking, which is inspired by the will to make things as simple as possible, neither introducing arbitrary parameters nor using sophisticated heuristics. In the future, other sophisticated models, for example, those defined in [19] and [20], can be used.

Finally, we incorporate a stochastic selection procedure, in the form of a simple roulette wheel selection with replacement, to select parents for mutation operators. Note that all selection probabilities of individuals, in the roulette wheel selection process, are firstly normalized since all selection probabilities obtained by (15) do not sum to 1.

C. MS-DE Framework

The proposed multiobjective sorting-based mutation framework affects only the mutation step of the original DE, hence it can be directly applied to any DE mutation strategy. The application of this framework for DE/rand/1, named MS-DE/rand/1, is described in Algorithm 1.

In the application of this framework for operators that incorporate the best individual (see DE/best/1, DE/best/2, DE/current-to-best/1, DE/rand-to-best/1), an individual randomly selected from the first front is set as the best individual (Note that remaining parent individuals are still selected by roulette wheel selection process). Thus, the mutant individuals are attracted around the best individual in terms of both fitness and diversity values. Here, what is the best individual is redefined. This new definition of the best individual can be used in other EAs, for example, particle swarm optimization (PSO), to select global best individual.

Ong and Keane [25] stated that the simplicity of an algorithm is also very important. Simplicity means ease of implementation and a minimum number of control parameters of the algorithm [25]. The proposed MS-DE is very simple. MS-DE modifies the original DE only in the selection of parents in mutation parents, thus it keeps the simple structure of the original DE without introducing additional parameter. Further, it can be easily applied to other advanced DE variants.

Although MS-DE adopts the nondominated sorting procedure from multiobjective optimization, it is different from multiobjective optimization algorithm. Multiobjective optimization algorithms, for example, NSGA-II, aim to simultaneously optimize multiple objectives. The nondominated sorting is originally used in environmental selection in NSGA-II. Environmental selection in multiobjective optimization algorithms aims to obtain a well-approximated and well-distribution solution set, which is implemented by picking out the best solutions from the previous population and the newly created population. In NSGA-II, the nondominated sorting and crowding distance are used for approximation and distribution, respectively. MS-DE aims to optimize only a single objective to find a global optimum instead of a non-dominated solution set, and the nondominated sorting is introduced for mating selection instead of environmental selection.

The fitness and diversity are associated with exploitation and exploration, respectively. Thus, the classic trade-off or balance between exploration and exploitation is explicitly modeled using multiobjective optimization. Mating selection aims to make a good preparation for exchanging the information of individuals, which plays an important role in EAs. In MS-DE, the nondominated sorting for mating selection selects promising individuals in mutation operator for sharing the information among them, which is expected to improve the performance of DE. Further, the proposed nondominated sorting for mating selection also can be used in other EAs.

1) Complexity Analysis: Compared with the original DE, MS-DE has additional computation burden on calculation of diversity measure and nondominated sorting procedure. During one generation, the computation complexity of diversity measure mainly depends on the distance measure, which takes $\frac{NP\cdot(NP-2)}{2}$ times of calculating the pairwise distances between individuals (owing to the symmetric property of the distance measure). Therefore, the complexity of diversity measure is $O\left(D\cdot\frac{NP\cdot(NP-2)}{2}\right)$. As mentioned above, the complexity of nondominated sorting is $O(NP^2)$. Since the complexity of the original DE algorithm is $O(G_{max}\cdot NP\cdot D)$, where G_{max} is the maximal number of generation, the overall complexity of MS-DE is $O\left(G_{max}\cdot NP\cdot D + G_{max}\cdot\left(D\cdot\frac{NP\cdot(NP-2)}{2}+NP^2\right)\right)$, that is, $O(G_{max}\cdot NP\cdot NP\cdot D)$.

It should be noted that the recalculation of distance measure in MS-DE is performed only when at least one individual of every pair is changed at current generation. As a result, the complexity of MS-DE is far lower than $O(G_{max}\cdot NP\cdot NP\cdot D)$. In addition, under the circumstances of that the function evaluation is costly (see the CEC 2013 benchmark problems [26]), the computational burden of MS-DE is mostly determined by the function evaluation. According to the study in [18], the additional computational cost by computing the pairwise distance between individuals is shown to be negligible for the functions with costly evaluation. Therefore, the additional

overhead caused by diversity calculation and nondominated sorting is relatively small when the function evaluation is costly. The comparison of MS-DE with the original DE in terms of computational cost will be given in Section IV-F.

2) Comparison With ProDE and Rank-DE: MS-DE, ProDE and rank-DE are all general frameworks for the selection of suitable parent vectors in the existing mutation operators of DE. However, the ways to select the parent vectors in these frameworks are totally different. ProDE advocates a stochastic selection of some individuals according to their distances from the individual undergoing mutation. It is a neighborhood (distance) biased parent selection and favors search in the vicinity of the mutated individual. rank-DE selects parents according to their fitness ranking in the current population. It is a fitness biased parent selection and increases the selective pressure on fitter solutions. As a result, both ProDE and rank-DE promote efficient exploitation but diminish exploration abilities of the mutation operators. The proposed MS-DE selects parents according to the fitness and diversity information of individuals in the current population, and thus it is a fitness-and-diversity biased parent selection. In MS-DE, the exploitation is represented by fitness value, and the exploration is represented by a defined diversity measure. It is expected that balance of exploration and exploitation can be achieved by simultaneously considering fitness and diversity. The experimental comparisons of three frameworks will be given in Section IV-E.

IV. Experimental Results

In this section, comprehensive experiments are carried out to evaluate the effectiveness of MS-DE. Firstly, MS-DE is tested on 28 benchmark functions from CEC 2013 [26], and the comparisons of MS-DE with corresponding DE algorithms, advanced DE variants and related frameworks, respectively, are given in detail. Then, MS-DE is tested on 20 benchmark functions at 1000D from CEC 2012 special session on large scale global optimization [27] to study the scalability of MS-DE. Finally, MS-DE is applied to solve 12 real-world application problems from CEC 2011 [28]. The simulations are carried out on an Intel Core Dual-Core E5400 PC with 2.70 GHz CPU and 2GB RAM.

A. Experiment Setup

MS-DE is firstly tested on CEC 2013 test suite with 28 benchmark functions. These functions are divided into three groups: unimodal functions ($F1$-$F5$), basic multimodal functions ($F6$-$F20$), and composition functions ($F21$-$F28$). Elaborate descriptions of the 28 functions can be found in [26].

To evaluate the performances of original DE and MS-DE in each of the six original DE mutation strategies, we select parameters as follows.

1) Population size: $NP = 100$ [9], [29], and [30].
2) Scaling factor and crossover factor: $F = 0.5$, $CR = 0.9$ [9], [30].
3) Dimension of each function: $D = 30$ and $D = 50$ [26].
4) Maximum number of function evaluations (MaxFEs): $MaxFEs = 10^4 D$ [26].
5) Independent Number of runs (NumR): $NumR = 51$ [26].

Three evaluation criteria are adopted to measure the final performance of each algorithm.

1) *Solution error measure:* As an evaluation indicator, the function error value for solution X is defined as $f(X) - f(X^*)$, where X^* is the global optimum of the corresponding function. The best error value of each run is recorded when the maximal number of generation is reached. The average and standard deviation of the best error values are also calculated for comparison.

2) *Convergence:* The convergence graphs for some typical functions are plotted to illustrate the mean error performance of the best solution over the total run in the respective experiments.

3) *Statistics by Wilcoxon and Friedman tests [31]–[34]:* To show the significant differences between two algorithms on single problem, Wilcoxon signed-rank test [31]–[34] at 5% significance level is conducted. The result of the test is represented as "$w/t/l$," which means that one algorithm is significantly better than, equal to and worse than the corresponding competitor on w, t, and l functions, respectively. The significantly better values in terms of mean solution error between MS-DE and its competitor are highlighted in boldface in tables. To identify differences between pair of algorithms on all problems, the multiproblem Wilcoxon signed-rank test is carried out. The Friedman test, conducted by the KEEL software [34], is used to obtain the rankings of multiple algorithms (more than two algorithms) on all problems.

Due to the space limit, all numerical values (solution errors) of the simulations are presented in the supplemental file (available at http://ieeexplore.ieee.org). Statistics summarizing those numerical values are shown in Tables I–V.

B. Comparison With Original DE Algorithms

MS-DE is compared with the six original DE algorithms presented earlier (i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/current-to-best/1, and DE/rand-to-best/1) on the 28 benchmark functions at 30D and 50D. Table I provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms corresponding DE algorithm on most of functions. Then, multiproblem Wilcoxon signed-rank test is carried out to identify differences between each pair of algorithms on all problems [33]. The results are also summarized in Table I. It is clear that MS-DE obtains higher $R+$ values than $R-$ values in all cases. Further, the p values of all cases are less than 0.05, which indicates that MS-DE is significantly better than its corresponding DE algorithm.

The overall results of Table I clearly show that MS-DE can improve the performance of the original DE for both explorative and exploitative mutation strategies. The improvements on the mutation operators with the best vector, i.e., DE/best/1, DE/best/2, DE/current-to-best/1, and DE/rand-to-best/1, on different kinds of functions are more obvious. The reason is that the best individual is redefined. Additionally, the entire population is guided by the different best vectors randomly selected from the first front instead of a single globally best vector, which greatly improves the diversity of population.

TABLE I
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH ORIGINAL DE ALGORITHMS FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Alogrithm at 30D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	17/4/7	288.5	89.5	1.60E-02	Yes	Yes
MS-DE/rand/2 *vs* DE/rand/2	20/7/1	348	58	8.92E-04	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	22/2/4	306	72	4.76E-03	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	17/8/3	296	92	1.92E-02	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	21/5/2	322	56	1.34E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	20/4/4	308	70	4.09E-03	Yes	Yes
Alogrithm at 50D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	12/9/7	280	98	2.79E-02	Yes	Yes
MS-DE/rand/2 *vs* DE/rand/2	22/5/1	383.5	22.5	3.80E-05	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	23/2/3	327	51	8.76E-04	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	19/4/5	292.5	113.5	4.04E-02	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	22/3/3	321	57	1.46E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	20/5/3	334	44	4.73E-04	Yes	Yes

TABLE II
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH ADVANCE DE VARIANTS FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Alogrithm at 30D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-ODE *vs* ODE	14/7/7	262	116	7.74E-02	No	Yes
MS-jDE *vs* jDE	10/14/4	273	105	4.23E-02	Yes	Yes
MS-SaDE *vs* SaDE	18/5/5	338.5	39.5	3.14E-04	Yes	Yes
MS-CoDE *vs* CoDE	18/5/5	391	15	1.80E-05	Yes	Yes
MS-OXDE *vs* OXDE	17/4/7	304.5	101.5	2.02E-02	Yes	Yes
Alogrithm at 50D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-ODE *vs* ODE	11/11/6	264	114	6.89E-02	No	Yes
MS-jDE *vs* jDE	12/10/6	229.5	148.5	3.25E-01	No	No
MS-SaDE *vs* SaDE	21/2/5	346	32	1.54E-04	Yes	Yes
MS-CoDE *vs* CoDE	20/3/5	308.5	69.5	3.83E-03	Yes	Yes
MS-OXDE *vs* OXDE	14/8/6	280	126	7.76E-02	No	Yes

TABLE III
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH PRODE AND RANK-DE, RESPECTIVELY, FOR CEC 2013 FUNCTIONS AT 30D AND 50D

Mutation	**Alogrithm at 30D**	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
DE/rand/1	ProDE	11/10/7	244	134	1.81E-01	No	No
	rank-DE	12/9/7	239	167	4.06E-01	No	No
DE/rand/2	ProDE	16/9/3	269	137	1.28E-01	No	No
	rank-DE	20/5/3	305.5	100.5	1.87E-02	Yes	Yes
DE/best/1	ProDE	22/3/3	329	49	7.37E-04	Yes	Yes
	rank-DE	14/8/6	285.5	120.5	5.88E-02	No	Yes
DE/best/2	ProDE	12/8/8	199.5	178.5	9.09E-01	No	No
	rank-DE	12/10/6	220	158	4.48E-01	No	No
DE/rand-to-best/1	Pro-DE	21/3/4	309	69	3.79E-03	Yes	Yes
	rank-DE	21/4/3	337.5	68.5	2.05E-03	Yes	Yes
DE/current-to-best/1	ProDE	19/4/5	292	86	1.29E-02	Yes	Yes
	rank-DE	20/2/6	304	102	2.08E-02	Yes	Yes
Mutation	**Alogrithm at 50D**	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
DE/rand/1	ProDE	8/15/5	222.5	155.5	4.14E-01	No	No
	rank-DE	10/11/7	288.5	117.5	5.02E-02	No	Yes
DE/rand/2	ProDE	21/6/1	348	58	9.22E-04	Yes	Yes
	rank-DE	20/8/0	324	54	1.13E-03	Yes	Yes
DE/best/1	ProDE	23/1/4	330	48	6.75E-04	Yes	Yes
	rank-DE	22/3/3	329	49	7.37E-04	Yes	Yes
DE/best/2	ProDE	14/6/8	226.5	179.5	5.85E-01	No	No
	rank-DE	15/3/10	247	131	1.60E-01	No	No
DE/rand-to-best/1	ProDE	22/3/3	309	69	3.79E-03	Yes	Yes
	rank-DE	22/3/3	337.5	68.5	2.05E-03	Yes	Yes
DE/current-to-best/1	ProDE	20/4/4	333	45	5.17E-04	Yes	Yes
	rank-DE	19/5/4	326	52	9.22E-04	Yes	Yes

With respect to the features of the benchmark functions, a close inspection of the numerical values of the simulations presented in the supplemental file indicates that MS-DE versions can find better results for most of cases in different kinds of functions. In the case of unimodal functions ($F_1 - F_5$), MS-DE versions greatly improve the performances of original DE/best/1, DE/current-to-best/1, DE/rand/2, and DE/rand-to-best/1. MS-DE versions obtain even one order of magnitude

TABLE IV
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH SELECTED DE ALGORITHMS AND ADVANCED DE VARIANTS, RESPECTIVELY, ON CEC 2012 FUNCTIONS AT 1000D

Alogrithm at 1000D	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/2 *vs* DE/rand/2	17/2/1	194	16	8.34E-04	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	18/1/1	210	0	8.20E-05	Yes	Yes
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	16/4/0	159	31	9.09E-03	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	18/1/1	207	3	1.30E-04	Yes	Yes
MS-jDE *vs* jDE	10/3/7	165	45	2.39E-02	Yes	Yes
MS-SaDE *vs* SaDE	12/3/5	158	52	4.58E-03	Yes	Yes

TABLE V
STATISTICS OF PERFORMANCE COMPARISONS OF MS-DE WITH CORRESPONDING DE ALGORITHMS ON REAL-WORLD PROBLEMS

Alogrithm	$w/t/l$	$R+$	$R-$	$p-$value	$\alpha = 0.05$	$\alpha = 0.1$
MS-DE/rand/1 *vs* DE/rand/1	5/7/0	42.5	23.5	3.74E-01	No	No
MS-DE/rand/2 *vs* DE/rand/2	4/8/0	57.5	8.5	2.62E-02	Yes	Yes
MS-DE/best/1 *vs* DE/best/1	8/4/0	61.5	4.5	9.93E-03	Yes	Yes
MS-DE/best/2 *vs* DE/best/2	8/1/3	39	27	5.63E-01	No	No
MS-DE/rand-to-best/1 *vs* DE/rand-to-best/1	7/4/1	58.5	7.5	2.08E-02	Yes	Yes
MS-DE/current-to-best/1 *vs* DE/current-to-best/1	4/4/4	40.5	37.5	8.75E-01	No	No
MS-ODE *vs* ODE	4/8/0	66	12	2.94E-02	Yes	Yes
MS-jDE *vs* jDE	4/6/2	55.5	22.5	1.82E-01	No	No
MS-SaDE *vs* SaDE	8/3/1	68.5	9.5	1.75E-02	Yes	Yes
MS-CoDE *vs* CoDE	10/2/0	76.5	1.5	2.87E-03	Yes	Yes
MS-OXDE *vs* OXDE	6/5/1	59	19	1.08E-01	No	No

improvement for $F_1 - F_3$. In the cases of multimodal functions and hybrid composition functions ($F_6 - F_{28}$), MS-DE versions also obtain better performance on most of functions. MS-DE versions obtain one order of magnitude improvement on F_6, F_7, F_{10}, F_{11}, F_{19}, and F_{28}. Since MS-DE versions consider both fitness and diversity information, a good balance between exploration and exploitation can be achieved. Thus, MS-DE versions can obtain good results for most of cases in unimodal problems, multimodal problems, and composition problems. Detailed discussion about the search behavior of MS-DE will be given in Section IV-G.

C. Comparison With Advanced DE Variants

To further evaluate the effectiveness of proposed framework, MS-DE is then applied to five state-of-the-art advanced DE variants, including ODE [30], jDE [29], SaDE [10], CoDE [11], and OXDE [35]. These advanced DE variants introduce different kinds of modifications on initialization, control parameters, operators (including ensemble of mutation, and orthogonal crossover), and their hybrid (see ensemble of mutation and parameter adaptation in SaDE) into DE, respectively. They have different characteristics and all obtain very promising results. The comparisons of MS-DE with the corresponding advanced DE variants are carried out on 28 benchmark functions with 30D and 50D. All parameter settings of the advanced DE variants are kept the same as the original papers. The results are shown in the supplemental file.

Table II provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding advanced DE variants on most of functions. From the multiproblem Wilcoxon signed-rank test, it is clear that MS-DE obtains higher $R+$ values than $R-$ values in all cases. It means that MS-DE is better than its corresponding advanced DE variant for all functions. Additionally, for the functions at 30D, the p values are less than 0.05 in four cases (except in ODE case), while the p values are less than 0.1 in all cases. For the functions at 50D, the p values of two cases (MS-SaDE *vs* SaDE and MS-CoDE *vs* CoDE) are less than 0.05, while the p values in all cases, except for MS-jDE *vs* jDE, are less than 0.1. These results indicate that MS-DE is significantly better than most of its corresponding DE variants according to the multiple-problem statistical analysis.

With respect to the features of the benchmark functions, a close inspection of the results in the supplemental file also indicates that MS-DE versions can find better results for most of unimodal, multimodal, and composition functions. For solving unimodal functions, more exploitation is beneficial, while more exploration is beneficial for solving multimodal functions. MS-DE versions consider both fitness and diversity information for parent selection, where the fitness is associated with exploitation and the diversity helps exploration. Thus, MS-DE versions can simultaneously deal with most of unimodal, multimodal, and composition functions.

In summary, MS-DE can improve the performance of the advanced DE variants. It means that the proposed scheme in mutation operator can cooperate with other different kinds of modifications to further improve the performance of DE. This provides a promising approach for solving benchmark and real-world problems.

D. More Comparison of MS-DE Versions

In this section, we firstly compare different MS-DE versions. Then the best overall performing MS-DE versions are further compared with other state-of-the-art algorithms.

The Friedman test is used to obtain the rankings of different MS-DE versions on all problems. The final rankings of all MS-DE versions for all functions are shown in the supplemental file. Overall, MS-SaDE gets the first rank, followed

by MS-OXDE and MS-jDE for all functions at $D = 30$, while MS-jDE gets the first rank, followed by MS-SaDE and MS-OXDE for all function at $D = 50$.

Note that the main contribution of this paper is to propose a multiobjective sorting-based parent selection scheme, and not to propose a "Best" algorithm or competitor to defeat other state-of-the-art algorithms. However, to provide additional comparison for reference, the best overall performing algorithms, MS-SaDE for $D = 30$ and MS-jDE for $D = 50$ are selected to further compare with five selected state-of-the-art algorithms from CEC 2013 competition [26]. The selected algorithms include: two memetic algorithms, named DRMA-LSCh-CMA [43] and MVMO-SH [44], a hybrid CMA-ES, named CMA-ES-RIS [45], a hybrid PSO and artificial bee colony (ABC) algorithm, named ABC-SPSO [46], and a advanced genetic algorithm, named GA-TPC [47]. These algorithms are very powerful algorithms with different features, and thus can be seen as representatives of the state-of-the-art algorithms for comparison.

The results for the functions at $D = 30$ and $D = 50$ are shown in the supplemental file. The Friedman test is used to obtain the ranking of different algorithms on all problems [33], [34]. Overall, MS-SaDE gets the third rank for the functions at $D = 30$. It outperforms CMA-ES-RIS, ABC-SPSO and GA-TPC, but is outperformed by two memetic algorithms, DRMA-LSCh-CMA and MVMO-SH. MS-jDE also gets the third rank for the functions at $D = 50$. It is also outperformed by two memetic algorithms and outperforms remaining algorithms. Note that DRMA-LSCh-CMA and MVMO-SH are advanced memetic algorithms combing advanced evolutionary algorithm and sophisticated local search. It is not surprising that they obtain superior results. MS-DE versions are pure EAs, and their performance can be greatly enhanced if the sophisticated local search is introduced.

E. Comparison With ProDE and Rank-DE

In this section, the proposed algorithm is compared with rank-DE and ProDE. Comparison of MS-DE with rank-DE also helps to show the contribution or effect of diversity information introduced in MS-DE.

The experimental comparisons of MS-DE with ProDE and rank-DE are carried out on 28 benchmark functions with 30D and 50D. All six DE mutation strategies are used in three frameworks, and all statistics are summarized in Table III. From Table III, it is clear that MS-DE obtains the significantly better results than ProDE and rank-DE in most of the cases. In some cases, the differences are more significant. For example, with DE/rand/2, MS-DE significantly outperforms rank-DE on 20 functions without losing any function; and with DE/best/1, MS-DE significantly outperforms rank-DE on 22 functions, and is outperformed by it on only three functions.

In order to further show the significant differences between MS-DE and two related frameworks, Friedman test is carried out and the results are shown in the supplemental file. With respect to the average rankings of different algorithms by Friedman test, MS-DE consistently obtains the best rankings in all six cases for the functions at 30D and 50D.

Moreover, the multiproblem Wilcoxon signed-rank test is also conducted between MS-DE and the two related frameworks on all problems for each case. The results in Table III show that MS-DE obtains higher $R+$ values than $R-$ values in all cases. For the functions at 30D, according to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in three cases between MS-DE and ProDE, and in three cases between MS-DE and rank-DE. According to the Wilcoxon test at $\alpha = 0.1$, significant differences between MS-DE and ProDE are observed in three cases, while in four cases between MS-DE and rank-DE. For the functions at 50D, according to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in four cases between MS-DE and ProDE, and in four cases between MS-DE and rank-DE. According to the Wilcoxon's test at $\alpha = 0.1$, significant differences between MS-DE and ProDE are observed in four cases, while in five cases between MS-DE and rank-DE. The results of Table III indicate that MS-DE is significantly better than ProDE and rank-DE in most of the cases based on the multiple-problem statistical analysis.

As shown in [18] and [19], rank-DE and ProDE are mainly able to enhance the exploitation ability for unimodal functions, but they may be over-exploitative and lead to premature convergence to the local optima in the multimodal problems especially for exploitative mutation operators. MS-DE versions can obtain better results for most of cases in different kinds of functions since they can achieve the balance between exploration and exploitation.

The superior performance of MS-DE attributes to the good balance between exploration and exploitation. The balance is achieved by considering both fitness and diversity information for parent selection. The balance between exploration and exploitation is a core of all kinds of EAs. The idea of MS-DE can be seen as a general and fundamental scheme to select promising or healthy individuals, which can be extended to further improve other EAs.

F. Computation Cost of MS-DE

As discussed in Section IV-D, MS-DE has additional computation burden compared with the original DE. Here, we compare the average runtime between MS-DE and the corresponding DE algorithms on the 28 functions in CEC 2013 at 30D and 50D. The results are shown in the supplemental file. The runtime value for each function means the average overhead of all DE algorithms considered in this paper on the corresponding function. The ratio value is defined as the value that the cost of MS-DE is divided by that of the original DE.

For the functions at 30D, the ratio values are higher than 2.5 for all unimodal and basic multimodal functions except for $F9$ and $F17$. The evaluation for these functions is not costly; thus, the runtime of MS-DE mainly comes from updating the pairwise distance and nondominated sorting. On the hybrid composition functions $F21 - F28$, most of the ratio values, however, are lower than 2. The evaluations of $F15 - F25$ are costly, thus the overhead of calculating pairwise distance and nondominated sorting becomes trivial.

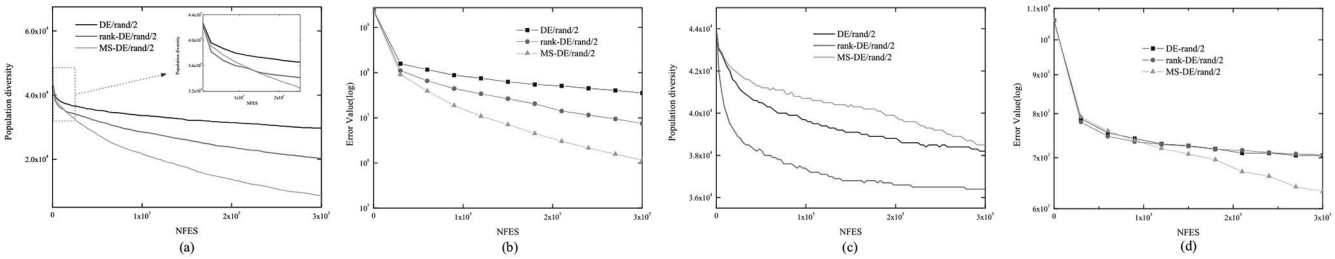


Fig. 2. Evolution graphs of diversity and fitness during the search progress in DE/rand/2, rank-DE/rand/2, and MS-DE/rand/2 on unimodal function F2, and multimodal function F22. (a) Diversity on F2. (b) Fitness on F2. (c) Diversity on F22. (d) Fitness on F22.

For the functions at 50*D*, most of the ratio values are lower than 2. Compared with the functions at 30*D*, higher dimensional functions are more difficult to be solved, which leads to a decrease in the frequency of updating the pairwise distance. Therefore, the ratio values are not as high as those of the functions at 30*D*.

Overall, the average runtime ratio values are 2.592 and 1.991 for the functions in CEC 2013 at 30*D* and 50*D*, respectively, which are acceptable for the practical applications.

MS-DE has almost the same computational complexity with ProDE since both approaches have to update the pairwise distance.

In order to further reduce computational effort, on the one hand, the diversity measure of each individual can be computed approximately. For example, it can be updated every 20 generations instead of every generation. On the other hand, other cheaper diversity measure can be designed in a cleverer fashion without the need of calculating pairwise Euclidean distance in future work.

G. Discussion on Search Behavior of MS-DE

In this section, the search mechanism of MS-DE is briefly explained, then the experimental study is given to visually show the search behavior of MS-DE. Finally, how to control the search behavior and further improve the performance of MS-DE are discussed. These discussions on search behavior of MS-DE versions can show, to some extent, how and why the proposed selection helps DE algorithms to find better results.

In original DE, the distribution information is learned by sampling random difference vectors from the population. It is argued in [20] that DE may sample the local topology of the function better, and thus improves its efficiency by learning distribution information from elite difference vectors in the population. This idea is originally inspired by CMA-ES [48]. In order to focus on elite difference vectors, a fitness rank-based parent selection scheme is used to restrict the set of parents to elite individuals in the population [19], [20]. The difference vectors that are likely to be selected with and without fitness-biased selective pressure are shown [20], and thus the effect of fitness-biased selection is analyzed and discussed. It is argued that the fitness-biased difference vectors describe the function topology better and with fewer members. Similarly, the base vector in mutation operator is more likely to be in a fitter region of the space. Thus, fitness-biased selection helps to focus the search, which contributes to the performance improvement over original DE. However, one inherent danger with high fitness-biased selective pressure is the rapid loss of diversity within the population. It is clearly shown in [20] that the rank-based variants lead to a lower population dispersion which benefits differential mutation by focusing the search. That is, the increase of exploitative power comes at a cost to potential exploration [19]. This raises some important questions regarding stagnation.

In fact, the worth of an individual is not considered to rely only on its fitness value. When mating with other individuals, the ability of an individual to produce offspring dispersed enough in the solution space is also very important. Keep this in mind and to mitigate the risk of rank-DE using only fitness to select parent, we introduce an additional diversity measure and thus fitness-and-diversity biased selection is proposed. In MS-DE, the elite difference vectors to be focused on are redefined in terms of both fitness and diversity. It is expected that the increase of exploitative power does not necessarily come at a cost to potential exploration. That is, MS-DE can still focus the search using the fitness information while avoiding rapid loss of diversity or trying to maintain diversity of the population by using diversity information. Since MS-DE versions consider both fitness and diversity information to select promising or healthy individuals, a good balance between exploration, and exploitation can be achieved.

Two representative mutation strategies, DE/rand/2, and DE/best/1, are selected to visually show and compare the evolution behaviors among original DE algorithms, rank-DE, and MS-DE versions. DE/rand/2 is the most explorative mutation operator since a random base vector and two random difference vectors are used. DE/best/1 is the most exploitative mutation operator since the best vector is selected as base vector and only one difference vector is used. These mutation strategies have different exploration/exploitation features. Fig. 2 shows the evolution of diversity and fitness in DE/rand/2, rank-DE/rand/2, and MS-DE/rand/2 on unimodal function F_2, and multimodal function F_{22}, respectively. Fig. 3 shows the evolution of diversity and fitness in DE/best/1, rank-DE/best/1, and MS-DE/best/1 on unimodal function F_2, and multimodal function F_{22} at $D = 30$, respectively. The convergence graphs are used to depict the evolution of fitness. The evolution graphs of diversity illustrate the evolution of the mean population diversity over all runs, where the population diversity *PD* is the sum of all individuals' diversity in

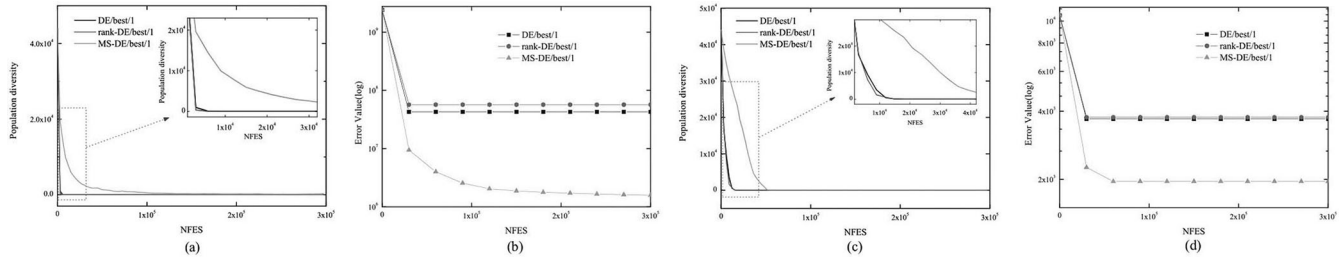


Fig. 3. Evolution graphs of diversity and fitness during the search progress in DE/best/1, rank-DE/best/1, and MS-DE/best/1 on unimodal function F2, and multimodal function F22. (a) Diversity on F2. (b) Fitness on F2. (c) Diversity on F22. (d) Fitness on F22.

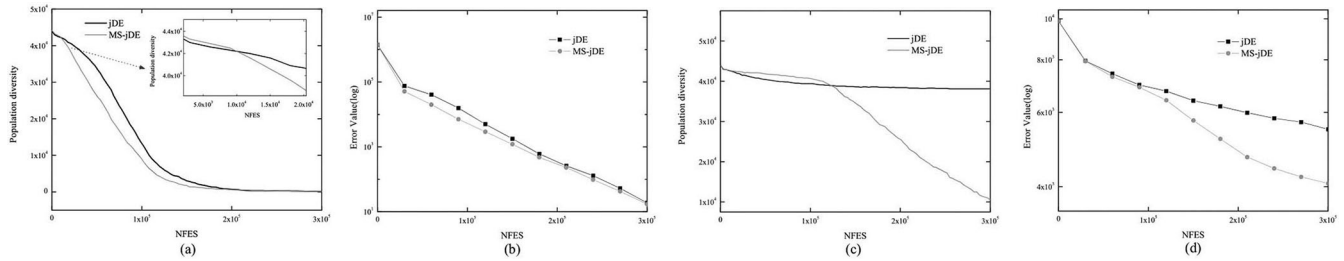


Fig. 4. Evolution graphs of diversity and fitness during the search progress in jDE and MS-jDE on unimodal function F4, and multimodal function F23. (a) Diversity on F4. (b) Fitness on F4. (c) Diversity on F23. (d) Fitness on F23.

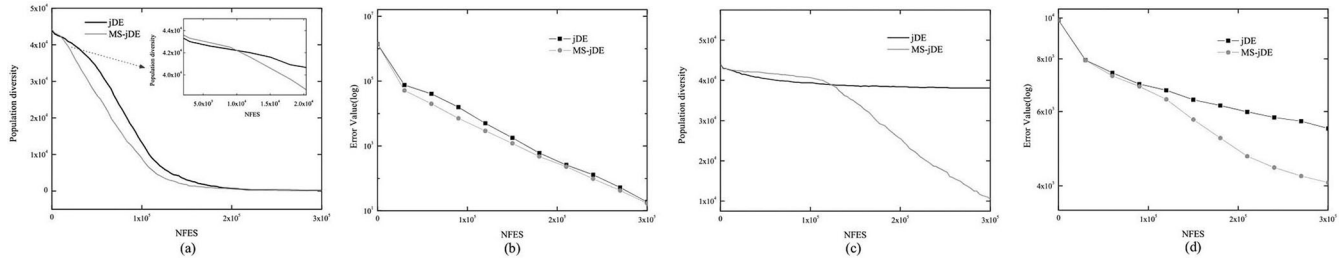


Fig. 5. Evolution graphs of diversity and fitness during the search progress in OXDE and MS-OXDE on unimodal function F4, and multimodal function F6. (a) Diversity on F4. (b) Fitness on F4. (c) Diversity on F6. (d) Fitness on F6.

the population, i.e., $PD = \sum_{i=1}^{NP} \Psi_i$. The diversity evolution graphs combined with the corresponding fitness convergence graphs on both unimodal and multimodal functions can show the behavior of the algorithms comprehensively.

For explorative mutation DE/rand/2, MS-DE version maintains higher diversity than rank-DE at early stage on unimodal function F_2 in Fig. 2(a) and (b), and then rapidly converges to better final result with decreasing the diversity. On multimodal function F_{22} in Fig. 2(c) and (d), MS-DE version maintains higher diversity than rank-DE all the time, and obtains better results finally. Compared with original DE, MS-DE version leads to a lower diversity on unimodal function F_2 for exploitation, and leads to a higher diversity on multimodal function F_{22} at early stage for exploration. In contrast, rank-DE version leads to a much lower diversity on all functions compared with original DE. This may lead to worse result. For exploitative mutation DE/best/1, MS-DE version maintains higher diversity than original DE and rank-DE version, and obtains the best results on both unimodal and multimodal functions. The best individual in DE/best/1 is redefined in terms of fitness and diversity, and entire population is guided by the different best vectors randomly selected from the first front instead of a single globally best vector. This, combined with the elite difference vector redefined, greatly improves the diversity of population, which promotes the exploration ability of the exploitative mutation to escape from local minima. In contrast, rank-DE leads to a lower population diversity on both unimodal and multimodal functions. Thus, rank-DE obtains worse results than DE/best/1 for unimodal function F_2 and multimodal functioin F_{22} due to over-exploitation. These results support our explanation that MS-DE can focus the search using fitness information, avoiding rapid loss of diversity or trying to maintain diversity using diversity information.

For advanced DE variants, two representatives, jDE and OXDE, are selected because the results for the others are similar. Fig. 4 shows the evolution of diversity and fitness in jDE and MS-jDE on unimodal function F_4, and multimodal function F_{23} at $D = 30$, respectively. Fig. 5 shows the evolution of diversity and fitness in OXDE and MS-OXDE on unimodal function F_4, and multimodal functions F_6 at $D = 30$, respectively. Compared with the corresponding advanced DE variants, MS-DE versions lead to lower or higher population diversity on different functions or at different search stages on the same function, trying to achieve better balance of exploration and exploitation for better results. For example, MS-jDE maintains higher diversity at early stage for

exploration and then leads to lower diversity for exploitation or convergence on unimodal function F_4 and multimodal function F_{23}. MS-OXDE leads to lower diversity on unimolal function F_4 for exploitation, and maintains higher diversity at the later stage on the multimodal function F_6 for exploration.

Note that we should not expect that the diversity of population in MS-DE versions can necessarily be maintained to a much higher level. The reason is that, whatever mating selection is used, all generated trial vectors will be filtered by the same fitness based greedy survival selection to form the population of next generation. The diverse individuals with bad fitness have no chance of survival. Both DE and the corresponding MS-DE version try to find the single global optimum. Thus, the diversity evolution in both DE algorithms and the corresponding MS-DE versions shows similar decreasing trend for convergence. Since the diversity used in MS-DE versions is a helper objective, the proposed selection method can be seen as a diversity maintenance method [21].

Different problems, or different stages during the search progress on the same problem, need different ratios between exploration and exploitation [21]. The proposed parent selection scheme, as a simple diversity maintenance method, can not deal with all kinds of problems well. Hence, MS-DE can not obtain the best results on all unimodal and multimodal functions as shown in previous simulation results. To further improve the performance of MS-DE on more difficult functions, two research lines can be followed in future. Firstly, diversity control, diversity learning and some direct approaches can be adopted to promote the diversity of MS-DE to control exploration and exploitation [21]. For example, diversity can be injected or created by using a restart mechanism, or additional mutation. But this will lead to slower convergence because high diversity does not necessarily mean that a diverse population is obtained by a good balance between exploration and exploitation. Hence, local search should be introduced to speed convergence. Secondly, the diversity and fitness information defined in this paper can be further and fully exploited to improve the performance of MS-DE. For example, they can be used to estimate and thus monitor the real-time evolutionary states of individuals and population in each generation, as in adaptive DEs [42], [49] and adaptive PSOs [50], [51]. Then, the search behavior of each individual, determined by the parameters and operators, in MS-DE can be adaptively controlled according to the real-time evolutionary states estimated. Thus, different individuals can play difference roles (exploitation or exploration) during the search progress, and even the same individual can play different roles during different search progress.

H. Scalability Study on the CEC 2012 Optimization Problems

In order to study the scalability, MS-DE is tested on the set of 20 benchmark functions at $1000D$ from the CEC 2012 special session on large scale global optimization [27]. These functions, denoted as $LF1$-$LF20$, are classified into five types of functions: separable ($LF1$-$LF3$), single-group m-nonseparable ($LF4$-$LF8$), $D/2m$-group m-nonseparable ($LF9$-$LF13$), D/m-group m-nonseparable ($LF14$-$LF18$) and Nonseparable functions ($LF19$-$LF20$). Group size m is set to 50, and more details of the 20 functions can be found in [27]. In this experiment, 25 runs are performed independently for each problem, and *MNFEs* is set to 3×10^6 [27]. Evaluation and experiment of algorithms on these large scale benchmark functions are extremely time-consuming, hence several representative algorithms are chosen in this experiment due to time limit. Here, four representative original DE algorithms, including explorative mutation DE/rand/2, and exploitative mutation DE/best/1, DE/current-to-best/1, and DE/rand-to-best/1, and two representative advanced DE variants including jDE and SaDE are used to provide a deeper analysis about the scalability. These mutation strategies have different exploration/exploitation features. The MS-DE versions of jDE and SaDE are the overall best algorithms for the CEC 2013 functions. Hence, the selected algorithms can be seen as representatives of DE variants for testing. The results are shown in the supplemental file.

Table IV provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding DE algorithm on most of functions. MS-DE obtains higher $R+$ values than $R-$ values in all cases. Further, the p values of all cases are less than 0.05, which indicates that MS-DE is significantly better than its corresponding DE algorithm. Hence, we can conclude that MS-DE brings benefit to the four original DE algorithms and two advanced DE variants for solving the large scale optimization problem at $1000D$.

Here, these functions are used just to study the scalability of the MS-DE, and the main aim of this paper is not to propose MS-DE to solve large scale global optimization. In order to further improve the performance of MS-DE for large-scale global optimization, other advanced techniques, see parallel mechanism [36] and cooperative coevolution [37], can be introduced into or imposed on MS-DE. Recent work [37] shows that it is possible to make a given high performance evolutionary optimizer scale well to high dimensional problems by using it as a subcomponent optimizer in a cooperative coevolution framework with an automatic decomposition strategy such as differential grouping [37].

I. Application to Real-World Problems

In this section, we evaluate the potential of MS-DE for real-world application problems selected from the CEC 2011 competition on testing evolutionary algorithm on real-world numerical optimization problems [28]. There are total 22 (by taking different instance of problems as a separate one) real-world numerical optimization problems including both unconstrained and constrained optimization problems. This paper considers the selected 12 unconstrained problems: parameter estimation of frequency-modulated sound waves (T1), Lennard-Jones potential (T2), optimal control of a nonlinear stirred tank reactor (T4), Tersoff potential function minimization problem-Si(B) (T5.1), Tersoff potential function minimization problem-Si(C) (T5.2), spread spectrum radar polly phase code design (T6), transmission network expansion

planning (T7), static economic load dispatch-instance 2 (T11.2), static economic load dispatch-instance 3 (T11.3), static economic load dispatch-instance 4 (T11.4), Messenger: spacecraft trajectory optimization (T13), and Cassini 2: spacecraft trajectory optimization (T14). Details about these 12 problems can be referred to [28].

In this section, six original DE algorithms and five advanced DE variants mentioned above, and their MS-DE versions are all applied to the selected real-world problems for comparison. The *MNFEs* for these problems is set to 150 000, and 25 independent runs are implemented for each problem [28]. The results are shown in the supplemental file.

Table V provides the statistics summarizing the performance comparisons. The results in column $w/t/l$ show that MS-DE significantly outperforms the corresponding DE algorithm on most of the real-world application problems. Moreover, MS-DE obtains higher $R+$ values than $R-$ values in all cases. According to the Wilcoxon test at $\alpha = 0.05$, there are significant differences in six cases between DE algorithms and their MS-DE version. The results of Table V indicate that MS-DE is significantly better than original DE algorithms and advanced DE variants in most of the cases.

The final rankings of all MS-DE versions for all real-world problems are shown in the supplemental file. Overall, MS-SaDE gets the first rank, followed by MS-OXDE and MS-jDE.

V. Conclusion

This paper presents a DE framework with multiobjective sorting-based mutation operator. In the proposed mutation operator, individuals in the current population are first sorted according to their fitness and diversity information by nondominated sorting. Then parents in the mutation operators are proportionally selected according to their rankings based on fitness and diversity. Since fitness and diversity information is simultaneously considered to select the parents, a good balance between exploration and exploitation can be achieved. The proposed operator is applied to the original DE algorithms, as well as several advanced DE variants. Experimental results on benchmark functions and real-world application problems show that the proposed operator is an effective approach to enhance the performance of most of the DE algorithms studied. The scalability of MS-DE is also discussed. Through the simulation results on benchmark functions and real-world problems, MS-jDE, MS-SaDE, and MS-OXDE, among all MS-DE versions studied, perform well overall in terms of both simplicity and efficiency.

As a new DE framework, there are still many interesting issues worth further studying in MS-DE. Firstly, to reduce computational effort of MS-DE framework, the diversity measure can be designed in a cleverer fashion without the need of calculating pairwise Euclidean distance. Other kinds of cheaper diversity measures can also be introduced. Secondly, the fitness and diversity information defined in this paper can be further used to estimate and monitor the real-time evolutionary states of individuals and population in each generation. Thus, the search behavior of each individual in MS-DE can be adaptively controlled according to the real-time evolutionary states estimated. Thirdly, MS-DE can be extended to solve constraint optimization problems. Finally, the multiobjective sorting-based selection scheme can be seen as a general mating selection scheme simultaneously considering the fitness and diversity. Since mating selection operator is important to most of EAs, the proposed approach can be used to further improve other EAs [38], [39].

References

[1] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[2] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Secaucus, NJ, USA: Springer-Verlag, 2005.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[4] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.

[5] A. Iorio, and X. Li, "Improving the performance and scalability of differential evolution on problems exhibiting parameter interactions," *Soft Comput.*, vol. 15, no. 9, pp. 1769–1792, Sep. 2011.

[6] J.-B. Xiao and J. Xiao, "Classification-based self-adaptive differential evolution with fast and reliable convergence performance," *Soft Comput.*, vol. 15, no. 8, pp. 1581–1599, Jan. 2011.

[7] B. Dorronsoro and P. Bouvry, "Improving classical and decentralized differential evolution with new mutation operator and population topologies," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 67–98, Feb. 2011.

[8] H. Wang, S. Rahnamayan, S. Hui, and M. G. Omran, "Gaussian barebones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.

[9] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.

[10] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[11] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[12] P. Kaelo and M. Ali, "A numerical study of some modified differential evolution algorithms," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 1176–1184, 2006.

[13] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[14] J. Zhang and A. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[15] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[16] R.-J. He and Z.-Y. Yang, "Differential evolution with adaptive mutation and parameter control using Levy probability distribution," *J. Comput. Sci. Tech.*, vol. 27, no. 5, pp. 1035–1055, 2012.

[17] C. García-Martínez, F. Rodríguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: An analysis on large-scale optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2109–2126, Nov. 2011.

[18] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.

[19] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.

[20] A. M. Sutton, M. Lunacek, and L. D. Whitley, "Differential evolution and non-separability: Using selective pressure to focus search," in *Proc. 9th Annu. Conf. GECCO*, England. U.K., Jul. 2007, pp. 1428–1435.

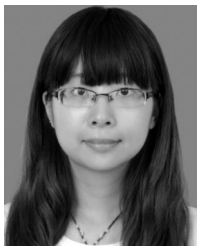
[21] M. Crepinsek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM CSUR*, vol. 45, no. 3, Article 35, 2013.
[22] G. Corriveau, R. Guilbault, A. Tahan, and R. Sabourin, "Review of phenotypic diversity formulations for diagnostic tool," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 9–26, 2013.
[23] G. Corriveau, R. Guilbault, A. Tahan, and R. Sabourin, "Review and study of genotypic diversity measures for real-coded representations," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 695–710, Oct. 2012.
[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[25] Y.-S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.
[26] J. J. Liang, B. Y. Qu, P. N. Suganthan, and G. H.-D. Alfredo, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Tech. Univ., Singapore, Tech. Rep. 201212, Jan. 2013.
[27] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large scale global optimization," Nature Inspired Comput. Appl. Lab., USTC, China, Tech. Rep., 2009.
[28] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," Jadavpur Univ., India, and Nanyang Tech. Univ., Singapore, Tech. Rep., Dec. 2010.
[29] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[30] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb. 2008.
[31] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.
[32] S. Garcäa, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 16, pp. 617–644, 2009.
[33] J. Derrac, S. Garcäa, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[34] J. Alcalá-Fdez, L. Sánchez, and S. Garcáa, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Oct. 2008.
[35] Y. Wang, Z. Cai, and Q. Zhang. "Enhancing the search ability of differential evolution through orthogonal crossover," *Inf. Sci.*, vol. 185, no. 1, pp. 153–177, 2012.
[36] M. Weber, F. Neri, and V. Tirronen, "Shuffle or update parallel differential evolution for large scale optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2089–2107, Oct. 2011.
[37] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, 2014, to be published.
[38] H. Ma, S. Dan, M. Fei, and Z. Chen, "On the equivalences and differences of evolutionary algorithms," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2397–2407, Nov. 2013.
[39] S. Kern *et al.*, "Learning probability distributions in continuous evolutionary algorithms—A comparative review," *Natural Comput.*, vol. 3, no. 1, pp. 77–112, 2004.
[40] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 107–124, Feb. 2012.
[41] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 397–413, Apr. 2011.
[42] W.-J. Yu *et al.*, "Differential evolution with two-level parameter adaptation," *IEEE Trans. Cybern.*, 2014, to be published.
[43] B. Lacroix, D. Molina, and F. Herrera, "Dynamically updated region based memetic algorithm for the 2013 CEC special session and competition on real parameter single objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1945–1951.
[44] J. L. Rueda and I. Erlich, "Hybrid mean-variance mapping optimization for solving the IEEE-CEC competition problems," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1664–1671.
[45] F. Caraffini, G. Iacca, F. Neri, L. Picinali, and E. Mininno, "A CMAES super-fit scheme for the re-sampled inheritance search," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1123–1130.
[46] M. El-Abd, "Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2215–2220.
[47] S. M. M. Elsayed, R. A. Sarker, and D. L. Essam, "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 356–360.
[48] S. D. Müller, N. Hansen, and P. Koumoutsakos. "Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations," in *Proc. PPSN*, Granada, Spain, 2002, pp. 422–431.
[49] X. Zhou, Z. Wu, H. Wang, and S. Rahnamayan, "Enhancing differential evolution with role assignment scheme," *Soft Comput.*, 2014, to be published.
[50] Z.-H. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.
[51] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 627–646, Jun. 2012.

Jiahai Wang (M'07) received the Ph.D. degree from the University of Toyama, Toyama, Japan, in 2005.

He is currently an Associate Professor at the Department of Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include computational intelligence and its applications.

Jianjun Liao received the B.S. degree from Hunan University of Science and Engineering, Yongzhou, China, in 2011. He is currently pursuing the M.S. degree with Sun Yat-sen University, Guangzhou, China.

His current research interests include differential evolution and evolutionary computation techniques.



Ying Zhou received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2009, where she is currently pursuing the Ph.D. degree.

Her current research interests include local search algorithms, multiobjective optimization, and other evolutionary computation techniques.

Yiqiao Cai received the B.S. degree from Hunan University, Changsha, China, and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2007 and 2012, respectively.

He is currently a Lecturer with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. His current research interests include differential evolution, multiobjective optimization, and other evolutionary computation techniques.