Decomposition-Based-Sorting and Angle-Based-Selection for Evolutionary Multiobjective and Many-Objective Optimization

Xinye Cai, *Member, IEEE*, Zhixiang Yang, Zhun Fan, *Senior Member, IEEE*,
and Qingfu Zhang, *Senior Member, IEEE*

***Abstract*—Multiobjective evolutionary algorithm based on decomposition (MOEA/D) decomposes a multiobjective optimization problem (MOP) into a number of scalar optimization subproblems and then solves them in parallel. In many MOEA/D variants, each subproblem is associated with one and only one solution. An underlying assumption is that each subproblem has a different Pareto-optimal solution, which may not be held, for irregular Pareto fronts (PFs), e.g., disconnected and degenerate ones. In this paper, we propose a new variant of MOEA/D with sorting-and-selection (MOEA/D-SAS). Different from other selection schemes, the balance between convergence and diversity is achieved by two distinctive components, decomposition-based-sorting (DBS) and angle-based-selection (ABS). DBS only sorts *L* closest solutions to each subproblem to control the convergence and reduce the computational cost. The parameter *L* has been made adaptive based on the evolutionary process. ABS takes use of angle information between solutions in the objective space to maintain a more fine-grained diversity. In MOEA/D-SAS, different solutions can be associated with the same subproblems; and some subproblems are allowed to have no associated solution, more flexible to MOPs or many-objective optimization problems (MaOPs) with different shapes of PFs. Comprehensive experimental studies have shown that MOEA/D-SAS outperforms other approaches; and is especially effective on MOPs or MaOPs with irregular PFs. Moreover, the computational efficiency of DBS and the effects of ABS in MOEA/D-SAS are also investigated and discussed in detail.**

***Index Terms*—Angle-based-selection (ABS), decomposition-based-sorting (DBS), diversity, evolutionary multiobjective optimization, many-objective optimization.**

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs) involve the optimization of more than one objective function. Since these objectives usually conflict with each other, no single optimal solution exists to optimize all the objectives simultaneously. Instead, Pareto-optimal solutions, which are their best tradeoff candidates, can help decision makers to understand the tradeoff relationship among different objectives and choose their preferred solutions. In the field of multiobjective optimization, the set of all the Pareto-optimal solutions is usually called the Pareto set (PS) and the image of (PS) on the objective vector space is called the Pareto front (PF) [30]. Over the past decades, multiobjective evolutionary algorithms (MOEAs) have been recognized as a major methodology for approximating the PF [5], [9], [10], [12], [18], [36]–[38].

In MOEAs, selection is of great importance for the performance of MOEAs. Usually, it is desirable to balance between convergence and diversity for obtaining good approximation to the set of Pareto-optimal solutions [4], [17]. Convergence can be measured as the distance of solutions toward the PF, which should be as small as possible. Diversity can be measured as the spread of solutions along the PF, which should be as uniform as possible.

Based on the above two requirements for selection, the current MOEAs can be categorized into the domination-based (see [15], [42], [44]), the indicator-based (see [2], [3], [24], [43]), and the decomposition-based MOEAs (see [20], [21], [35], [40]). A representative of decomposition-based MOEAs is MOEA based on decomposition (MOEA/D) [40], which can be regarded as a generalization of cMOGA [31]. MOEA/D decomposes an MOP into a number of single objective optimization subproblems and then solves them in parallel. The objective function in each subproblem can be a linear or nonlinear weighted aggregation function of all the objective functions in the MOP in question.

Manuscript received January 5, 2016; revised May 25, 2016; accepted June 15, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61300159, Grant 61473241, Grant 61332002, Grant 61370185 and Grant 61175073, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20130808, in part by the China Post-Doctoral Science Foundation under Grant 2015M571751, in part by the Science and Technology Planning Project of Guangdong Province of China under Grant 2013B011304002, in part by the Educational Commission of Guangdong Province of China under Grant 2015KGJHZ014, in part by the Fundamental Research Funds for the Central Universities of China under Grant NZ2013306, and in part by the Guangdong High-Level University Project "Green Technologies" for Marine Industries. This paper was recommended by Associate Editor G. G. Yen.

X. Cai and Z. Yang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: xinye@nuaa.edu.cn; xiang052@163.com).

Z. Fan is with the Guangdong Provincial Key Laboratory of Digital Signal and Image Processing and the Department of Electronic Engineering, School of Engineering, Shantou University, Shantou 515063, China (e-mail: zfan@stu.edu.cn).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qingfu.zhang@cityu.edu.hk).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org as a PDF file provided by the authors. The total size of the file is 414 KMB in size.

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.2016.2586191

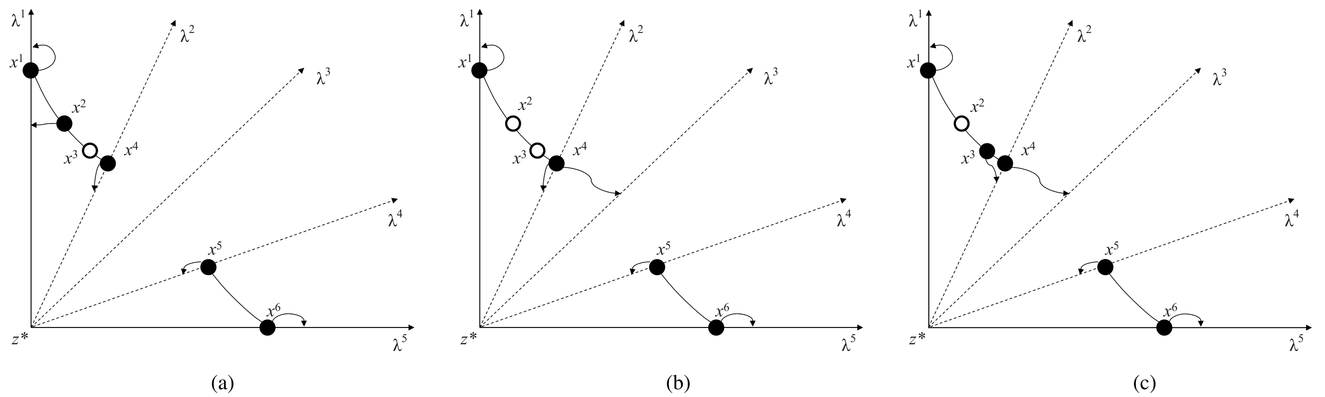


Fig. 1. Illustrative examples of different selection results. (a) Ideal selection result. Selection result obtained by (b) MOEA/D and (c) MOEA/D-STM.

In MOEA/D, each solution is associated with a subproblem, and two subproblems are called neighbors if their weight vectors are close to each other. MOEA/D explores correlation relationships among neighboring subproblems to speed up its search. The diversity is implicitly achieved by specifying a wide spread of the directions in the objective space. Several variants of MOEA/D have been proposed and studied (see [1], [6], [7], [19], [23], [26], [28], [29], [39]). For example, an online geometrical metric was proposed to enhance the diversity of MOEA/D in [19]. In [28], a global stable matching (STM) model is integrated into MOEA/D for suitable matches between subproblems and solutions. In MOEA/D-STM, each subproblem prefers the solution with better aggregation function value, which indicates a better convergence along its search direction. Therefore, the preferences of subproblems encourage convergence. Meanwhile, each solution agent ranks all subproblems according to its distance to the weight vector of these subproblems. Therefore, the preferences of the solutions can promote the diversity. The STM between subproblems and solutions achieves an equilibrium between their mutual-preferences and thus, a balance between convergence and diversity can be achieved.

The motivations of this paper are based on the following considerations.

1) In many MOEA/D variants, e.g., MOEA/D-STM, each subproblem is allowed to associate with one and only one solution. An underlying assumption is that each subproblem leads to a diversely located Pareto optimal solution in PF. It could hold if the weight vectors of the subproblems are appropriately assigned priori. However, for the real-world MOPs, both shape and spread of the PFs are unknown and this assumption is not likely to be held, especially for disconnected and degenerate PFs [22]. Fig. 1 shows the ideal selection results for such a PF as well as the ones obtained by MOEA/D and MOEA/D-STM. The population diversity for both MOEA/D and MOEA/D-STM has not been well-maintained due to the above assumption. Under this circumstance, it is not reasonable to force one subproblem to associate with one solution. The framework of MOEA/D can be more flexible to accommodate MOPs with different shapes of PFs.
2) As a state-of-the-art variant of MOEA/D, MOEA/D-STM can usually, achieve good balance between the convergence and diversity. However, the computational cost of its selection scheme is still high [$O(NM \log M)$, where N is the population size and M is 2*N], due to the use of global STM model.[1] The selection of local solutions for each subproblem can be used to reduce the computational complexity.
3) Although some advanced diversity maintenance schemes, e.g., niche-counts [14], have been adopted for MOEA/D to further increase its diversity [27]. Nevertheless, such scheme is, in some sense, very coarse-grained: it does not distinguish subproblems with the same niche-counts and it is possible that solutions associated with different subproblems may be close to each other while solutions associated with the same subproblems may be far from each other. To further increase the diversity, a more fine-grained diversity maintenance scheme is desired.

Based on the above considerations, this paper proposes a new variant of MOEA/D with sorting-and-selection (MOEA/D-SAS) for MOPs. Different from other selection schemes, the balance between convergence and diversity is achieved by two distinctive components, decomposition-based-sorting (DBS) and angle-based-selection (ABS). DBS only sorts L closest solutions to each subproblem to control the convergence and reduce the computational cost. The parameter L has been made adaptive based on the evolutionary process. ABS takes use of angle information between solutions in the objective space to maintain a more fine-grained diversity. In addition, different solutions can be associated with the same subproblems; and some subproblems are allowed to have no associated solution, which is more flexible to MOPs with different shapes of PFs.

The rest of this paper is organized as follows. Section II introduces some preliminaries on multiobjective optimization and decomposition methods. Section III describes the proposed sorting-and-selection (SAS) scheme, which contains two important components, DBS and ABS. In Section IV,

[1] Most high computational cost originates from sorting for the preference orderings of the whole population.

SAS is integrated into MOEA/D. Section V introduces the benchmark test functions and the performance indicators used in this paper. Experimental studies and discussion are presented in Section VI, where we compare our proposed algorithm with four classical MOEAs: 1) NSGA-II; 2) MOSOPS-II; 3) MOEA/D; and 4) MOEA/D-DE; and three state-of-the-art MOEA: 1) MOEA/D-STM; 2) NSGA-III; and 3) MOEA/D-AWA on MOPs or many-objective optimization problem (MaOPs). The effects of DBS and ABS are also investigated and discussed in Section VI. Section VII concludes this paper.

II. Preliminaries

This section first gives some basic definitions of multiobjective optimization. Then, some basic knowledge about the decomposition methods used in this paper is also introduced.

A. Basic Definitions

An MOP can be defined as follows:

$$\begin{aligned} &\text{minimize} \;\; F(x) = (f_1(x), \ldots, f_m(x))^T \\ &\text{subject to} \;\; x \in \Omega \end{aligned} \tag{1}$$

where Ω is the decision space, $F : \Omega \to R^m$ consists of m real-valued objective functions. The attainable objective set is $\{F(x) | x \in \Omega\}$.

Let $u, v \in R^m$, u is said to dominate v, denoted by $u \prec v$, if and only if $u_i \leq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$.[2] A solution $x^* \in \Omega$ is Pareto-optimal to (1) if there exists no solution $x \in \Omega$ such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is then called a Pareto-optimal (objective) vector. In other words, any improvement in one objective of a Pareto-optimal solution is bound to deteriorate at least another objective.

B. Decomposition Methods

In principle, many methods can be used to decompose an MOP into a number of scalar optimization subproblems [30]. Among them, the most popular ones are weighted sum (WS), Tchebycheff (TCH), and penalty boundary intersection (PBI) approaches [40]. The mathematical definition of these decomposition methods are as follows.

1) *WS Approach:* This approach considers a convex combination of all the objectives. One single objective subproblem s^k is defined as

$$\begin{aligned} &\text{minimize} \;\; g^{ws}\Big(x|\lambda^k\Big) = \sum_{i=1}^{m} \lambda_i^k f_i(x) \\ &\text{subject to} \;\; x \in \Omega \end{aligned} \tag{2}$$

where $\lambda^k = (\lambda_1^k, \ldots, \lambda_m^k)^T$ is the direction vector of subproblem s^k, and $\lambda_i^k \geq 0$, $i \in 1, \ldots, m$ and $\sum_{i=1}^m \lambda_i^k = 1$. The optimal solution to (2) is a Pareto-optimal solution to (1). A set of different Pareto-optimal solutions can be obtained simply by using different direction vectors, to approximate the PF when it is convex.

2) *TCH Approach:* In this approach, one single objective subproblem s^k is defined as

$$\begin{aligned} &\text{minimize} \;\; g^{te}\Big(x|\lambda^k, z^*\Big) = \max_{1 \leq i \leq m} \Big\{ |f_i(x) - z_i^*| / \lambda_i^k \Big\} \\ &\text{subject to} \;\; x \in \Omega \end{aligned} \tag{3}$$

$z^* = (z_1^*, \ldots, z_m^*)^T$ is the ideal objective vector, where $z_i^* < \min\{f_i(x) | x \in \Omega\}$, $i \in 1, \ldots, m$. For convenience, $\lambda_i^k = 0$ is replaced by $\lambda_i^k = 10^{-6}$, because $\lambda_i^k = 0$ is not allowed as a denominator in (3).

3) *PBI Approach:* This approach is a variant of normal-boundary intersection approach [11]. A subproblem s^k is defined as

$$\begin{aligned} &\text{minimize} \;\; g^{pbi}\Big(x|\lambda^k, z^*\Big) = d_1 + \beta d_2 \\ &\qquad d_1 = \big(F(x) - z^*\big)^T \lambda^k / ||\lambda^k|| \\ &\qquad d_2 = ||F(x) - z^* - \Big(d_1 / ||\lambda^k||\Big) \lambda^k|| \\ &\text{subject to} \;\; x \in \Omega \end{aligned} \tag{4}$$

where $||.||$ denotes the L_2-norm, and β is the penalty parameter.

III. Selection Operators

This section elaborates the selection operator based on the DBS and the ABS (SAS) scheme.

Given a set of N subproblems S and a set of M solutions, Z, the goal of SAS is to select N solutions from Z to form P.

For each subproblem j that has a direction vector λ^j, $P^j(L)$ denotes the set of the first L closest solutions to λ^j in Z. The "closeness" is defined by the acute angle between the solution x and the direction vector λ^j, based on

$$\text{angle}\big(x, \lambda^j\big) = \arccos\left(\frac{(F(x) - z^*)^T \lambda^j}{\|F(x) - z^*\| \|\lambda^j\|} \right). \tag{5}$$

The input parameter L in the current call of SAS has been adaptive, based on the value of output α in the previous call of SAS (see step 3 of Algorithm 4). L is defined as the number of closest solutions to each subproblem and α is the number of selected solution sets, which is explained in Section III-A.

A. Framework of SAS

The pseudo-code of SAS is presented in Algorithm 1.

1) Sorting: In step 1, P is first initialized to be an empty set. The DBS is conducted iteratively. DBS, presented in Algorithm 2, is detailed in Section III-B. In the ith iteration of DBS, L solution sets (fronts) $Q^{(i-1)*L+1}, \ldots, Q^{(i-1)*L+j}, \ldots, Q^{i*L}$ can be obtained by sorting population Z (or part of the population Z, depending on the value of L). Then, these sorted solution sets are added to P and eliminated from Z (lines 5–8). This process is repeated until the total number of sorted solutions $|P|$ exceeds the population size N.

After step 1, the population Z (or part of the population Z) is divided into $L*(i-1)$ solution sets (fronts): $Q^1, \ldots, Q^{L*(i-1)}$, where $i - 1$ is the total number of iterations of DBS and L is the number of closest solutions in the population Z to each subproblem. Note that possible overlapped solutions may exist

[2] In the case of maximization, the inequality signs should be reversed.

Algorithm 1: SAS(Z, z^*, L, N)

```
Input:
    1) $Z$: the solution set;
    2) $z^*$: the ideal objective vector;
    3) $L$: the number of closest solutions to each
       subproblem;
    4) $N$: the size of $P$.
Output:
    1) the population $P$;
    2) the number of selected fronts $\alpha$.
Step 1 Sorting:
1  $P = \emptyset$;
2  $i = 1$;
3  do
4      $[Q^{(i-1)*L+1}, \ldots, Q^{i*L}]$ = DBS (Z, $z^*, L$);
5      for $k = (i-1)*L+1$ to $i*L$ do
6          $P = P \bigcup Q^k$;
7          $Z = Z \setminus Q^k$;
8      end
9      $i = i + 1$;
10 while $|P| < N$;
Step 2 Selection:
11 $P = \emptyset$;
12 $k = 1$;
13 while $|P \bigcup Q^k| \leq N$ do
14     $P = P \bigcup Q^k$;
15     $k = k + 1$;
16 end
17 if $|P \bigcup Q^k| > N$ then
18     $A = Q^k \setminus P$;
19     $P$ = ABS$(A, P, z^*, N)$;
20     $\alpha = k$;
21 else
22     $\alpha = k - 1$;
23 end
24 return $P$ and $\alpha$;
```

Algorithm 2: DBS(Z, z^*, L)

```
Input:
    1) $Z$: the solution set;
    2) $z^*$: the ideal objective vector;
    3) $L$: the number of closest solutions to each
       subproblem.
Output:
    1) $L$ solution sets: $Q^1, \ldots, Q^L$;
1  $Q^1 = Q^2 = \ldots = Q^L = \emptyset$.
2  for $j = 1$ to $N$ do
3      $P^j(L) = Associate(Z, \lambda^j)$;    /* Associate
       subproblem $j$ with the $L$ closest
       solutions $P^j$ in $Z$, based on (5). */
4      for $k = 1$ to $L$ do
5          $g(P^j(k)|\lambda^j, z^*)$;    /* Computing the
           aggregation function values. */
6      end
7      $P^j = Sort(P^j)$;  /* Sort $P^j$ in an ascending
       order, based on $g(P^j|\lambda^j, z^*)$ values. */
8  end
9  for $k = 1$ to $L$ do
10     for $j = 1$ to $N$ do
11         $Q^k = Q^k \bigcup \{P^j(k)\}$;
12     end
13 end
14 return $Q^1, \ldots, Q^L$ ;
```

in different fronts Q^k. The value of i $(2 \leq i \leq (N+1))$ is determined by both value of L $(1 \leq L \leq |Z|)$ and evolutionary status of the algorithm. However, two extreme cases in terms of the value of L can be analyzed as follows. When $L = 1$, only the closest solution to each subproblem is chosen, which indicates only the solution closest to the direction vector of each subproblem gets involved in sorting. In this case, the diversity is emphasized and DBS is conducted for multiple times $(i > 2)$. When $L = |Z|$, the whole population Z is sorted for every subproblem and Z is divided into at most $|Z|$ fronts. In this case, convergence is emphasized and DBS is conducted for only one time $(i = 2)$. Therefore, more convergence is likely to be emphasized with the increase of the value of L.

2) Selection: In step 2, N solutions are selected out of the $L*(i-1)$ solution sets (fronts) obtained from step 1, as follows. P is initialized to be an empty set. For the kth front, if the size of the combined set $(P \cup Q^k)$ is smaller than N, then Q^k is added to P and the remaining members of P are chosen from Q^{k+1}. This procedure is continued until no more sets can be accommodated (lines 13–16). If the size of the combined set $(P \cup Q^k)$ is larger than N and say that the set Q^α is the last set beyond which no other set can be accommodated. Then the previously selected solutions in P are eliminated from Q^α $(Q^\alpha \backslash P)$ and stored in an intermediate set A. The ABS is activated to select solutions from A to fill P. More details of ABS are presented in Algorithm 3 in Section III-C. The number of actually selected sets $(Q^1, \ldots, Q^\alpha)$ is saved as α.

3) Termination: The solution set P and α are returned as the outputs.

B. Decomposition-Based-Sorting

The detailed procedures of DBS is presented in Algorithm 2 as follows. At the beginning, each subproblem j chooses its closest L solutions P^j in Z, based on (5) (line 3). The chosen solutions are sorted into L solution sets (fronts), $Q^1, \ldots, Q^k, \ldots, Q^L$ (lines 9–13), where Q^k contains the solutions with the kth best $g(*|\lambda^j, z^*)$ values, in P^j for every subproblem j, $(1 \leq j \leq N)$ (lines 4–7). Note that it is possible that $|Q^k| \leq N$ since two different subproblems may have the same kth best solution in Q^k. An illustrative example of DBS can be found in Section I of supplementary material.

C. Angle-Based-Selection

To further improve the diversity in the population, our selection scheme needs to consider the diversity relationship between the selecting solutions and the solutions in the previously

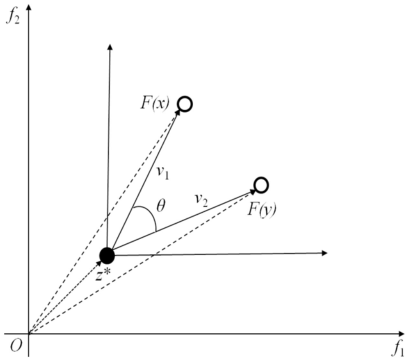


Fig. 2. Illustration of the angle between solution x and y.

```
Algorithm 3: ABS($A, P, z^*, N$)
   Input: two populations $A$ and $P$, the ideal objective
          vector $z^*$, size of $P$: $N$
   Output: the population $P$
1  foreach $x^i \in A$ do
2      $\theta_i \leftarrow \min_{x^j \in P}\{angle(x^i, x^j)\}$;
3  end
4  for $i \leftarrow 1$ to $|A|$ do
5      $k \leftarrow \mathrm{argmax}_k\{\theta_k\}$;
6      if $|P| < N$ then
7          $A \leftarrow A \setminus \{x^k\}$;
8          $\theta \leftarrow \theta \setminus \{\theta_k\}$;
9          $P \leftarrow P \bigcup \{x^k\}$;
10         foreach $x^j \in A$ do
11             $\theta_j \leftarrow \min\{\theta_j, angle(x^k, x^j)\}$;
12         end
13     else
14         break;
15     end
16 end
17 return $P$;
```

selected fronts. In this paper, we use acute angles between objective vectors of solutions to quantify diversity, as follows:

$$\mathrm{angle}(x, y) = \arccos\left(\frac{v_1^T v_2}{\|v_1\|\|v_2\|}\right) \tag{6}$$

where

$$v_1 = F(x) - z^*$$
$$v_2 = F(y) - z^*.$$

$F(x)$ and $F(y)$ are, respectively, the objective vectors of solution $x \in P$ and $y \in A$, and z^* is the ideal objective vector. Fig. 2 shows the calculation of diversity, as presented in (6). The idea of ABS is that, a member is added to P if and only if its angle with elements in P is the largest.

The pseudo-code of ABS is presented in Algorithm 3. For each solution $x^i \in A$ in the selecting front, its minimum angle, θ_i, to each solution in P, is calculated (lines 1–3), based on (6). To maximize the diversity, line 5 obtains the solution (x^k) with the largest angle (θ_k) to P. If the size of P is less than N, it will be deleted from A and added to P (lines 7–9); and the corresponding minimum angle between x^j and P is updated (lines 10–12); otherwise, the loop is terminated. Finally, P is returned as the output (line 17). An illustrative example of ABS can be found in Section II of supplementary material.

D. Computational Cost of the SAS

In DBS (Algorithm 2), the computational cost of association operators (N cycles of line 3) for all the subproblems can be easily reduced by the following two steps. The first step approximates the objective vectors of all the solutions in Z to their closest direction vectors, which requires $O(mM)$ computations, where m is the number of objectives and $M = 2N$ is the size of population Z. The second step needs $O(L)$ computations to obtain L neighboring solutions for each subproblem based on the first step. Therefore, the total computational cost of association operators for all the subproblems can be reduced to $O(LN)$. The complexity to calculate $g(*)$ for all the subproblems is $O(mLN)$ (N cycles of lines 4–6). $O(NL\log L)$ comparisons are used to sort $g(*)$ values (N cycles of line 7). So the complexity of Algorithm 2 is the larger one of $O(mLN)$ and $O(NL\log L)$.

In ABS (Algorithm 3), the computational cost can also be reduced if each solution in A only calculates its angle to each solution x^i in $S \subset P$, where S is $T \ll |P|$ neighboring solutions of $x^i \in P$. Suppose that the size of solution set A is N_a. The calculation of minimum angles between each solution in A and S (lines 1–3) needs $O(mN_aT)$ computations. The cycles from lines 4–16 are executed at most N_a times. Line 11 is executed $N_a * (N_a - 1)/2$ times. Therefore, in the worst case, the complexity of Algorithm 3 is $O(N_a^2)$ ($N_a \ll N$).

In SAS (Algorithm 1), at most N solutions are added to P in step 2, so the computational cost of for-loop in step 1 is $O(N\log N)$. The computational cost of DBS in step 1 is $O(mLN)$ or $O(NL\log L)$. At most N solutions are added to P in step 2, so the computational cost of while-loop in step 2 is also $O(N\log N)$. ABS in step 2 needs $O(N_aN_a)$ ($N_a \ll N$). So the computational cost of the SAS is the larger one of $O(mLN)$, $O(NL\log L)$ ($L \ll N$) and $O(N\log N)$. On the contrary, the computational cost of the STM model is $O(NM\log M)$. Apparently, either one of $O(mLN)$, $O(NL\log L)$, and $O(N\log N)$ is much smaller than $O(NM\log M)$ and the computational complexity of SAS has been greatly reduced compared with STM.

IV. Integration of SAS With MOEA/D

In this section, SAS is integrated into MOEA/D. The pseudo-code of our algorithm, called MOEA/D-SAS, is demonstrated in Algorithm 3.

At each generation, MOEA/D-SAS maintains the following.

1) A population of N solutions, $P = \{x^1, \ldots, x^N\}$.
2) A set of N subproblems, $S = \{s^1, \ldots, s^N\}$.
3) Objective function values, $\mathrm{FV}^1, \ldots, \mathrm{FV}^N$, where FV^i is the F-value of x^i.

The algorithm works as follows.

Step 1: *Initialization:* Initialize P.

```
Algorithm 4: MOEA/D-SAS
Input:
  1) MOP(1);
  2) a stopping criterion;
  3) $N$: the number of subproblems; the population size of
     $P$ and $Y$;
  4) $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N$: a set of $N$ weight vectors;
  5) $T$: the size of the neighborhood for each subproblem.
Output: population $P$.
Step 1 Initialization:
  a) Compute the Euclidean distances between any two
     weight vectors and obtain $T$ closest weight vectors to
     each weight vector. For each $i = 1, \ldots, N$, set
     $B(i) = \{i_1, \ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the $T$ closest
     weight vectors to $\lambda^i$.
  b) Generate an initial population $P = \{x^1, \ldots, x^N\}$
     randomly.
  c) Initialize $z^* = (z_1^*, \ldots, z_m^*)^T$ by setting
     $z_i^* = \min\{f_i(x^1), \ldots, f_i(x^N)\}$.
  d) Initialize $\alpha = 2 * N$.
Step 2 New Solution Generation:
  For each $i = 1, \ldots, |P|$, do:
  a) Selection of the Mating Solutions:
     1) Associate each solution $x^i$ with its closest
        subproblem $k$ based on (5).
     2) If $rand(0, 1) < \delta$, then set $D$ to the set of
        solutions associated with all the subproblems
        in $B(k)$, else, set $D = P$.
  b) Reproduction: Set $x^{r_1} = x^i$ and randomly select two
     indices $r_2$ and $r_3$ from $D$, and then generate a new
     solution $y^i$ from $x^{r_1}$, $x^{r_2}$ and $x^{r_3}$ by DE.
  c) Evaluation $y^i$ : $FV^i = F(y^i)$.
  d) Update of $z^*$ : For each $j = 1, \ldots, m$, if $z_j^* > f_j(y^i)$,
     then set $z_j^* = f_j(y^i)$.
Step 3 Sorting-and-Selection: Set $L = \min\{\alpha + T, 2N\}$,
$[P, \alpha]$ = SAS($P \bigcup Y, z^*, L, N$).
Step 4 Stopping Criteria: If stopping criteria is satisfied,
then stop and output $P$. Otherwise, go to Step 2.
```

Step 2: *New Solution Generation:* Generate a set of new solutions Y.

Step 3: *Sorting and Selection:* Use Y to update P.

Step 4: *Stopping Condition:* If a preset stopping condition is met, output P. Otherwise, go to step 2.

The pseudocode of MOEA/D-SAS is given in Algorithm 4. The details of steps 1–3 are given as follows.

A. Initialization

MOEA/D-SAS decomposes an MOP into N single objective optimization subproblems by using a decomposition approach (WS, TCH, or PBI) with N weight vectors

$$\lambda^k = \left(\lambda_1^k, \ldots, \lambda_m^k\right)^T \quad k = 1, \ldots, N \tag{7}$$

where $\lambda^k \in R_+^m$ and $\sum_{i=1}^m \lambda_i^k = 1$. The subproblem s^k is defined by (2), (3), or (4), in Section II-B.

For each $k = 1, \ldots, N$, let $B(k)$ be the set containing the indices of the T closest weight vectors to λ^k in terms of the Euclidean distance. If $i \in B(k)$, subproblem i is called a neighbor of subproblem k.

Solution x^i in P can be generated randomly or by using a single objective heuristic on the subproblem i. The ideal objective vector is initialized as the minimum values of all the solutions in P along each objective. α is initialized as $2N$.

B. New Solution Generation

An offspring population Y (size of N) is generated in step 2. For each solution x^i in P, the process for generating a new solution y^i is as follows.

In step 2a, the mating pool D for solution x^i is set to the set of solutions associated with all the subproblems in $B(k)$ with probability δ or the population P with probability $1 - \delta$. In step 2b, an offspring solution is reproduced, using parent solutions from the mating pool D. Any genetic operator or mathematical programming technique can serve this purpose, although differential evolution (DE) [32] and polynomial mutation [13] are used in this paper. We set one parent solution $x^{r_1} = x^i$. The other two parent solutions, x^{r_2} and x^{r_3}, are randomly selected from mating pool D, for generating an offspring solution as follows:

$$\bar{y}_j = \begin{cases} x_j^{r_1} + F \times \left(x_j^{r_2} - x_j^{r_3}\right), & \text{if rand} \le CR \text{ or } j = j_{\text{rand}} \\ x_j^{r_1}, & \text{otherwise} \end{cases} \tag{8}$$

where $j = 1, \ldots, n$, rand $\in [0, 1]$, $j_{\text{rand}} \in [1, n]$ is a random integer uniformly generated from 1 to n; and CR and F are two control parameters.

The polynomial mutation operator is applied on $\bar{y}$ to generate $y = (y_1, \ldots, y_n)^T$

$$y_j = \begin{cases} \bar{y}_j + \sigma_j \times (b_j - a_j), & \text{with probability } p_m \\ \bar{y}_j, & \text{with probability } 1 - p_m \end{cases} \tag{9}$$

with

$$\sigma_j = \begin{cases} (2 \times \text{rand})^{\frac{1}{\eta+1}} - 1, & \text{if rand} < 0.5 \\ 1 - (2 - 2 \times \text{rand})^{\frac{1}{\eta+1}}, & \text{otherwise} \end{cases} \tag{10}$$

where rand is a random number uniformly generated from $[0, 1]$; the distribution index η and the mutation rate p_m are two control parameters; and a_j and b_j are the lower and upper bounds of the jth decision variable.

In step 2c, the new solution y^i is evaluated. The ideal objective vector z^* is updated in step 2d. The procedure (steps 2a–2d) is repeated N times, so a population $Y = \{y^1, \ldots, y^N\}$ can be obtained.

C. Sorting-and-Selection

SAS is called to updated P, that is to select N solutions out of combined population $P \cup Y$. The neighborhood size L for sorting in SAS, is adaptively controlled by the number of selected fronts in the last call of SAS. The maximum value of L is set to $2 * N$ while the minimum value is set to $\alpha + T$.

TABLE I
PARAMETER SETTINGS IN MOEA/D-SAS, MOEA/D-STM, MOEA/D-DE, MOEA/D, MSOPS-II, AND NSGA-II FOR 2-OBJECTIVE TEST INSTANCES

–	Population size	# of neighbors	Crossover rates	Mutation rates	# of function evaluations
	UF(1-7)	UF(1-7)	UF(1-7)	UF(1-7)	UF(1-7)
MOEA/D-SAS	300	10 (20 for UF3)	1.0	$1/n$	300,000
MOEA/D-STM	300	20	1.0	$1/n$	300,000
MOEA/D-DE	300	20	1.0	$1/n$	300,000
MOEA/D	300	20	1.0	$1/n$	300,000
MSOPS-II	300	–	0.7	–	300,000
NSGA-II	300	–	0.9	$1/n$	300,000

TABLE II
PARAMETER SETTINGS IN MOEA/D-SAS, MOEA/D-STM, MOEA/D-DE, MOEA/D, MSOPS-II, AND NSGA-II FOR 3-OBJECTIVE TEST INSTANCES

–	Population size		# of neighbors		Crossover rates		Mutation rates		# of function evaluations	
	UF(8-10)	DTLZ(1-7)	UF(8-10)	DTLZ(1-7)	UF(8-10)	DTLZ(1-7)	UF(8-10)	DTLZ(1-7)	UF(8-10)	DTLZ(1-7)
MOEA/D-SAS	600	300	20	20	1.0	1.0	$1/n$	$1/n$	300,000	300,000
MOEA/D-STM	600	300	20	20	1.0	1.0	$1/n$	$1/n$	300,000	300,000
MOEA/D-DE	600	300	20	20	1.0	1.0	$1/n$	$1/n$	300,000	300,000
MOEA/D	600	300	20	20	1.0	1.0	$1/n$	$1/n$	300,000	300,000
MSOPS-II	600	300	–	–	0.7	0.7	–	–	300,000	300,000
NSGA-II	600	300	–	–	0.9	0.9	$1/n$	$1/n$	300,000	300,000

D. More Discussions on MOEA/D-SAS

In a very recent work [34], a decomposition-based MOEA (named WASF-GA) is proposed. In WASF-GA, the population is also divided into different fronts based on decomposition function values for subproblems. However, the selection of MOEA/D-SAS is fundamentally different from WASF-GA in the following two aspects.

1) MOEA/D-SAS can also deal with MOPs with irregular PFs, e.g., the disconnected and degenerate ones. Therefore, for the same case in Fig. 1, WASF-GA can only achieve the selection results in Fig. 1(b), while MOEA/D-SAS can achieve the ideal results in Fig. 1(a), due to the following two distinctive characteristics of MOEA/D-SAS.
   a) Different solutions are allowed to associate with the same subproblems and some subproblems may have no associated solutions.
   b) ABS adopts the angle information to select solutions with the best diversity.
2) Different from WASF-GA, which conducts sorting on all the solutions for each subproblem, DBS only sorts L closest solutions to each subproblem to control the convergence and reduce the computational cost.

V. EXPERIMENTAL SETTING

A. Test Problems

Two well-known test suites are considered in our experimental studies. One is the UF test suite which contains ten unconstrained MOP test instances (UF1–UF10) from the CEC2009 MOEA competition [41]. Seven of them (UF1–UF7) are 2-objective test functions, and the rest (UF8–UF10) are 3-objective functions. For all UF test functions, the number of decision variables is set to 30. Another test suite is DTLZ [16]. All DTLZ instances can be scaled to any number of objectives and decision variables. In this paper, the number of objectives is set to 3 and the number of decision variables is set to 10.

B. Parameter Settings

All the algorithms were implemented in MATLAB. The parameters of NSGA-II, MSOPS-II, MOEA/D, MOEA/D-DE, and MOEA/D-STM were set according to [15], [20], [25], [28], and [40]. The parameters of MSOPS-II, MOEA/D, MOEA/D-STM, and MOEA/D-SAS were set in such a way that they shared the same key parameter values with MOEA/D-DE. Their parameter settings for 2- and 3-objective benchmark functions are listed in Tables I and II, respectively.

The setting of N weight vectors $(\lambda^1, \ldots, \lambda^N)$ is controlled by a positive integer parameter H, which specifies the granularity or resolution of weight vectors, as in [40]. Each individual weight takes a value from

$$\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}.$$

The number of weight vectors is determined by both parameter H and the number of objectives m: $N = C_{H+m-1}^{m-1}$.

C. Performance Metrics

Inverted generational distance (IGD) [8], [45] is used as the performance metric in our studies. IGD measures the average distance from a set of reference points P^* in the PF to the approximation set P. It can be formulated as follows:

$$\mathrm{IGD}(P, P^*) = \frac{1}{|P^*|} \sum_{v \in P^*} \mathrm{dist}(v, P) \tag{11}$$

where $\mathrm{dist}(v, P)$ is the Euclidean distance between the solution v and its nearest point in P, and $|P^*|$ is the cardinality of P^*. If $|P^*|$ is large enough to represent the PF very well, $\mathrm{IGD}(P, P^*)$ could measure both diversity and convergence of P in a sense.

TABLE III
MEAN AND STANDARD DEVIATION VALUES OF IGD, OBTAINED BY MOEA/D-SAS, MOEA/D, MOEA/D-DE, MSOPS-II, AND NSGA-II ON UF AND DTLZ INSTANCES

instance	IGD				
	MOEA/D-SAS	MOEA/D	MOEA/D-DE	MSOPS-II	NSGA-II
UF1	**1.644E-03 (5.52E-05)**	9.807E-02 (5.92E-02) †	2.439E-03 (4.94E-04) †	7.425E-02 (5.44E-03) †	8.390E-02 (1.17E-02) †
UF2	**7.162E-03 (2.57E-03)**	7.712E-02 (3.40E-02) †	1.118E-02 (3.21E-03) †	5.716E-02 (2.10E-02) †	3.272E-02 (2.32E-03) †
UF3	**9.987E-03 (8.25E-03)**	2.912E-01 (5.23E-02) †	2.539E-02 (2.12E-02) †	3.141E-01 (1.75E-02) †	7.031E-02 (1.14E-02) †
UF4	5.699E-02 (2.52E-03)	9.186E-02 (6.82E-03) †	6.767E-02 (2.80E-03) †	**5.671E-02 (4.09E-03)**	7.606E-02 (1.35E-02) †
UF5	**2.358E-01 (3.14E-02)**	1.178E+00 (1.62E-01) †	2.901E-01 (4.56E-02) †	3.437E-01 (9.80E-02) †	6.793E-01 (9.88E-02) †
UF6	**9.963E-02 (5.76E-02)**	4.958E-01 (1.58E-01) †	1.868E-01 (1.34E-01) †	2.985E-01 (2.29E-01) †	3.217E-01 (7.60E-02) †
UF7	**2.480E-03 (6.28E-04)**	2.546E-01 (2.50E-01) †	4.067E-03 (9.31E-04) †	4.181E-02 (7.21E-03) †	3.504E-01 (8.65E-03) †
UF8	**3.356E-02 (4.03E-03)**	7.324E-02 (2.35E-02) †	6.213E-02 (7.45E-03) †	1.912E-01 (5.38E-03) †	2.671E-01 (5.44E-02) †
UF9	**2.925E-02 (3.88E-03)**	1.228E-01 (4.90E-02) †	6.111E-02 (3.85E-02) †	2.354E-01 (3.42E-02) †	1.840E-01 (6.91E-02) †
UF10	5.236E-01 (6.87E-02)	4.827E-01 (7.29E-02)	4.971E-01 (4.44E-02)	**2.490E-01 (1.20E-01)** ‡	6.632E-01 (6.81E-02) †
DTLZ1	**1.081E-02 (1.21E-04)**	1.600E-02 (5.88E-05) †	1.611E-02 (1.32E-04) †	1.573E+00 (6.27E-01) †	4.158E+00 (2.56E+00) †
DTLZ2	**2.897E-02 (3.68E-05)**	3.727E-02 (8.30E-05) †	3.725E-02 (8.69E-05) †	3.757E-02 (2.16E-03) †	5.972E-02 (7.90E-03) †
DTLZ3	**3.363E-02 (3.33E-04)**	3.868E-02 (2.52E-04) †	3.885E-02 (1.86E-04) †	4.866E+00 (2.64E+00) †	1.458E+01 (8.01E+00) †
DTLZ4	3.404E-02 (2.41E-04)	1.292E-01 (1.83E-01) †	5.971E-02 (7.34E-02) †	**3.321E-02 (4.57E-04)** ‡	7.318E-02 (2.84E-02) †
DTLZ5	3.598E-03 (1.43E-04)	4.394E-03 (1.11E-05) †	4.402E-03 (9.04E-06) †	1.578E-02 (3.82E-03) †	**1.836E-03 (1.34E-04)** ‡
DTLZ6	**3.685E-03 (1.95E-04)**	4.420E-03 (3.18E-06) †	4.583E-03 (1.56E-04) †	9.139E-01 (7.54E-02) †	5.388E-03 (1.53E-03) †
DTLZ7	**6.837E-02 (5.13E-02)**	3.586E-01 (2.82E-01) †	1.151E-01 (2.70E-03) †	1.240E-01 (1.29E-01) †	7.828E-01 (1.20E-01) †

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-SAS and each of the other competing algorithms. † and ‡ denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-SAS, respectively. The best mean is highlighted in boldface.

VI. EXPERIMENTAL STUDIES AND DISCUSSION

To study the performance of MOEA/D-SAS and understand its behavior, this section conducts the following experimental works.

1) Comparisons of MOEA/D-SAS with NSGA-II [15], MOEA/D [40], MOEA/D-DE [25], and MSOPS-II [20].
2) Comparisons of MOEA/D-SAS with MOEA/D-STM [28].
3) Investigation of the computational efficiency of DBS.
4) Investigation of the effects of ABS in MOEA/D-SAS.

In our experiments, each algorithm was run 30 times independently for each test instance. To make fair comparisons, TCH approach has been used as decomposition approach in MOEA/D, MOEA/D-DE, and MOEA/D-SAS on 2- or 3-objective optimization problems.

A. Comparisons With Classical MOEAs

In this section, we compare MOEA/D-SAS with four classical domination or decomposition-based MOEAs—NSGA-II, MSOPS-II, MOEA/D, and MOEA/D-DE.

The performances of MOEA/D-SAS, MOEA/D, MOEA/D-DE, NSGA-II, and MSOPS-II, in terms of IGD, is presented in Table III. MOEA/D-SAS has the significantly best performance among all the compared algorithms, on all the test functions, except for UF4, UF10, DTLZ4, and DTLZ5. MSOPS-II has the best performance on UF4, UF10, and DTLZ4; NSGA-II has the best performance on DTLZ5.

To compare the performance of algorithms during the optimization process, the evolution of the average IGD values, versus the number of function evaluations, for UF and DTLZ test instances are plotted in Fig. 3. It can be seen clearly from these figures that MOEA/D-SAS almost always has the best performances on both convergence speed and quality of the final nondominated sets. This observation is consistent with our motivations in Section I.

TABLE IV
MEAN AND STANDARD DEVIATION VALUES OF IGD, OBTAINED BY MOEA/D-SAS AND MOEA/D-STM ON UF AND DTLZ INSTANCES

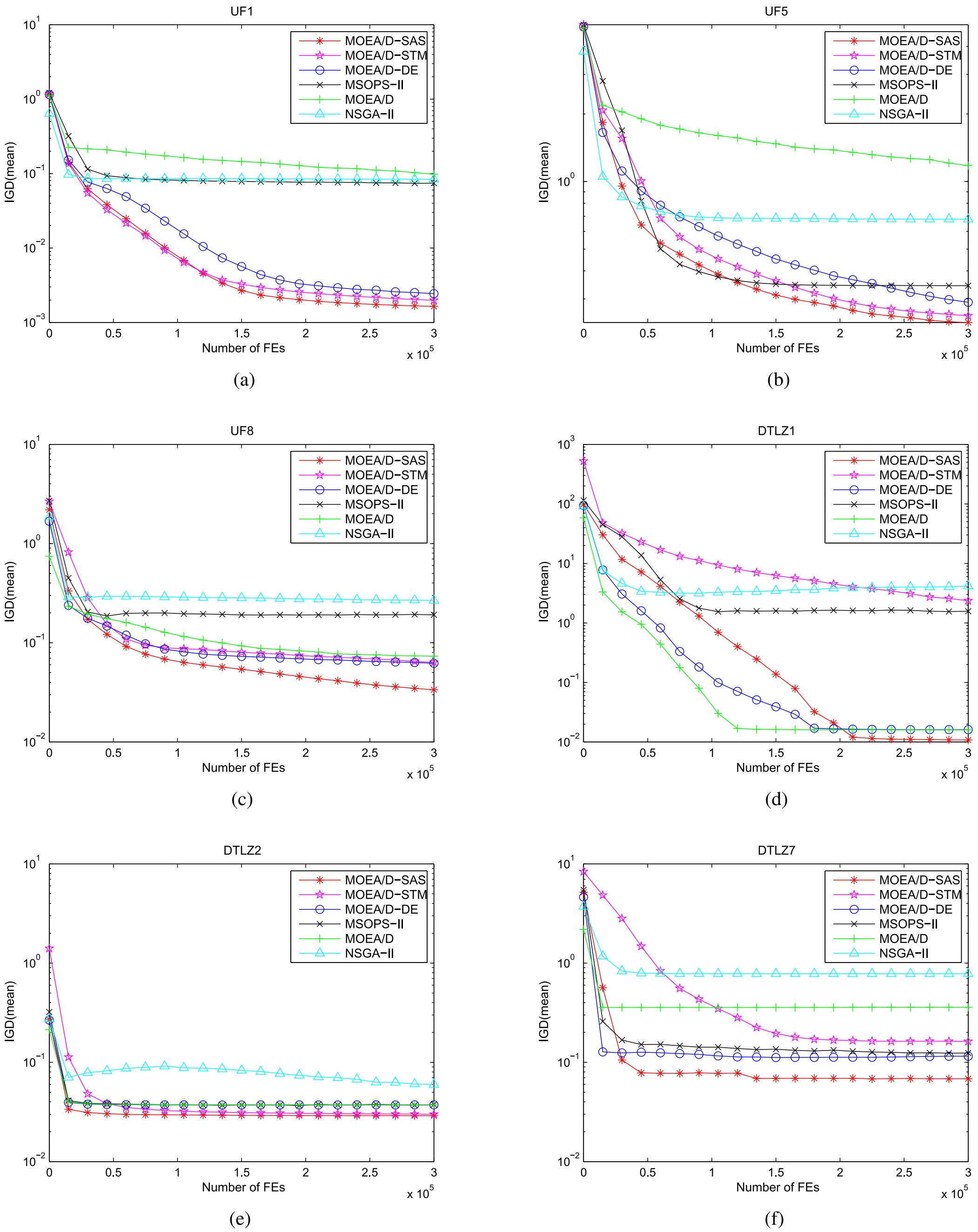
instance	IGD		
	MOEA/D-SAS	MOEA/D-STM	p-value
UF1	**1.644E-03 (5.52E-05)**	1.980E-03 (6.32E-05)	3.02E-11
UF2	7.162E-03 (2.57E-03)	7.074E-03 (1.84E-03)	0.7845
UF3	9.987E-03 (8.25E-03)	**3.721E-03 (2.48E-03)**	5.09E-06
UF4	5.699E-02 (2.52E-03)	5.614E-02 (3.75E-03)	0.2340
UF5	**2.358E-01 (3.14E-02)**	2.522E-01 (2.45E-02)	2.15E-02
UF6	9.963E-02 (5.76E-02)	8.202E-02 (3.75E-02)	0.6520
UF7	**2.480E-03 (6.28E-04)**	2.658E-03 (6.35E-04)	4.86E-03
UF8	**3.356E-02 (4.03E-03)**	6.331E-02 (1.04E-02)	4.50E-11
UF9	2.925E-02 (3.88E-03)	2.748E-02 (2.26E-03)	0.0555
UF10	**5.236E-01 (6.87E-02)**	1.213E+00 (2.09E-01)	3.02E-11
DTLZ1	**1.081E-02 (1.21E-04)**	2.367E+00 (7.91E+00)	3.82E-10
DTLZ2	**2.897E-02 (3.68E-05)**	3.000E-02 (2.11E-04)	3.02E-11
DTLZ3	**3.363E-02 (3.33E-04)**	2.499E+00 (7.85E+00)	6.07E-11
DTLZ4	**3.404E-02 (2.41E-04)**	8.748E-02 (1.10E-01)	3.02E-11
DTLZ5	**3.598E-03 (1.43E-04)**	7.083E-03 (2.63E-05)	3.02E-11
DTLZ6	**3.685E-03 (1.95E-04)**	7.091E-03 (8.72E-07)	3.02E-11
DTLZ7	**6.837E-02 (5.13E-02)**	1.627E-01 (4.52E-02)	5.07E-10

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-SAS and MOEA/D-STM. Boldface denotes that the performance of the corresponding algorithm is significantly better than that of the other.

B. Comparisons With MOEA/D-STM

MOEA/D-STM [28] is a state-of-the-art MOEA/D variant, which adopts an STM model to balance the convergence and diversity in the selection process of MOEA/D. In this section, we compare MOEA/D-SAS with it.

Table IV shows the performance of MOEA/D-SAS and MOEA/D-STM in terms of IGD. We can observe that the performances of MOEA/D-SAS are significantly better than that of MOEA/D-STM on 12 out of 17 test functions, although its performance is significantly worse than that of MOEA/D-STM on UF3. The two compared algorithms have very similar performances on UF2, UF4, UF6, and UF9.



(a)

(b)

(c)

(d)

(e)

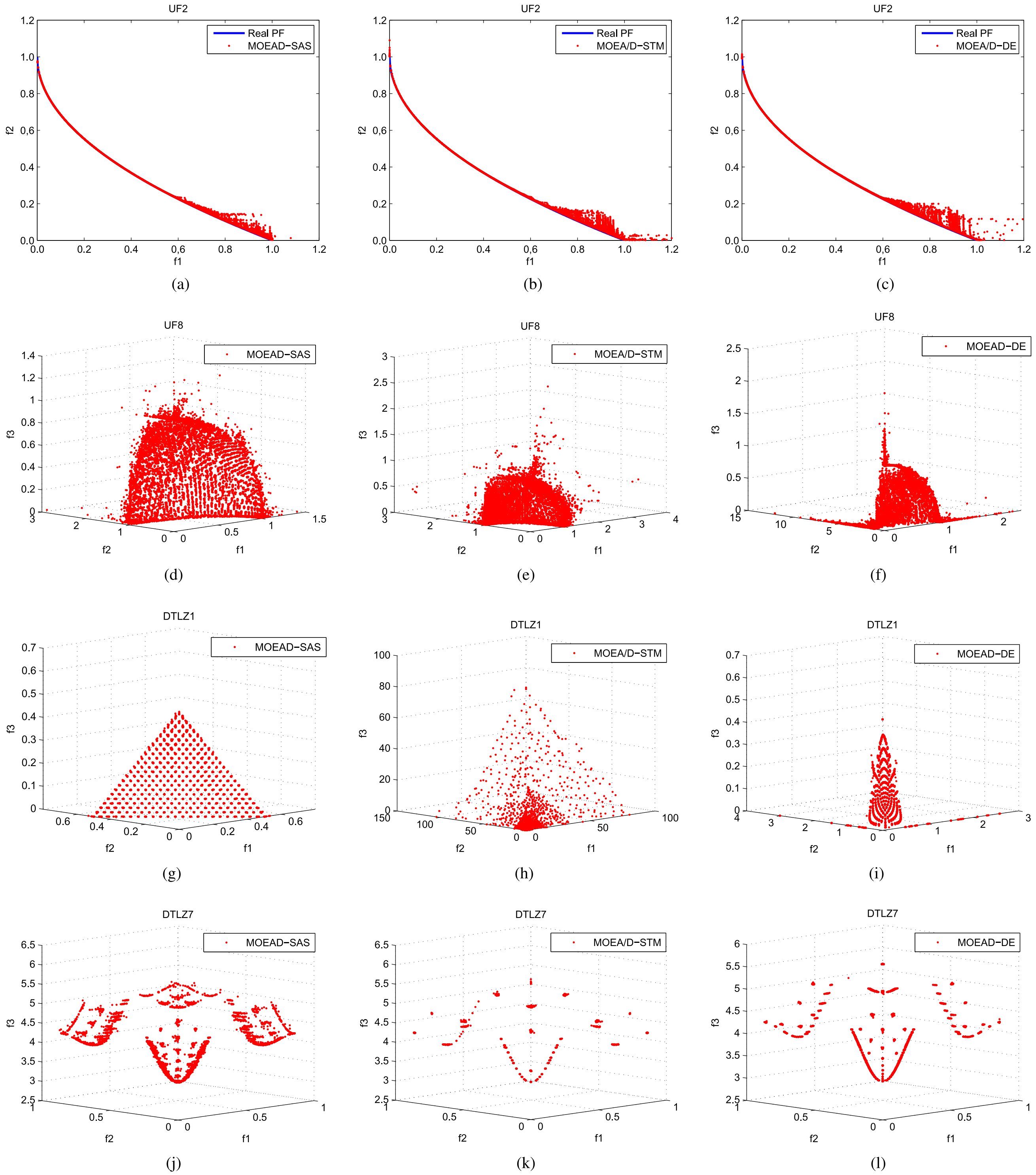
(f)

Fig. 3. Convergence graphs in terms of IGD (mean) obtained by MOEA/D-SAS, MOEA/D, MOEA/D-DE, MSOPS-II, NSGA-II, and MOEA/D-STM on three UF and three DTLZ instances. Convergence plot of six algorithms on (a) UF1, (b) UF5, (c) UF8, (d) DTLZ1, (e) DTLZ2, and (f) DTLZ7.

Fig. 4 plots all the populations over 30 independent runs obtained by MOEA/D-SAS, MOEA/D-STM, and MOEA/D-DE on UF2, UF8, DTLZ1, and DTLZ7. It is very clear that MOEA/D-SAS performs best among the three algorithms. It is worth to note that, for the benchmark problem with a disconnected and degenerate PF, such as DTLZ7, MOEA/D-STM tends to obtain the boundary solutions in PF, as illustrated in Fig. 1 and explained in Section I, while MOEA/D-SAS is able to obtain more diverse Pareto approximate solutions.

C. Computational Efficiency of Decomposition-Based-Sorting

DBS conducts sorting only among the neighboring solutions for each subproblem, which effectively reduces its computational cost. The number of selected fronts (α) at each generation adaptively determines the number of closest solutions to the direction vectors of subproblems L, for the next generation, and thus, plays an important role on the computational efficiency of DBS. Fig. 5 plots the evolution of α at each generations on different benchmark problems. It can be observed that the values of α decrease very quickly during the evolutionary process and level off at a very small value ($\alpha \ll N$), in all the benchmark problems. These observations further support the motivations in Section I and analysis in Section III, that DBS is able to use local neighborhood information to reduce its computational cost.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

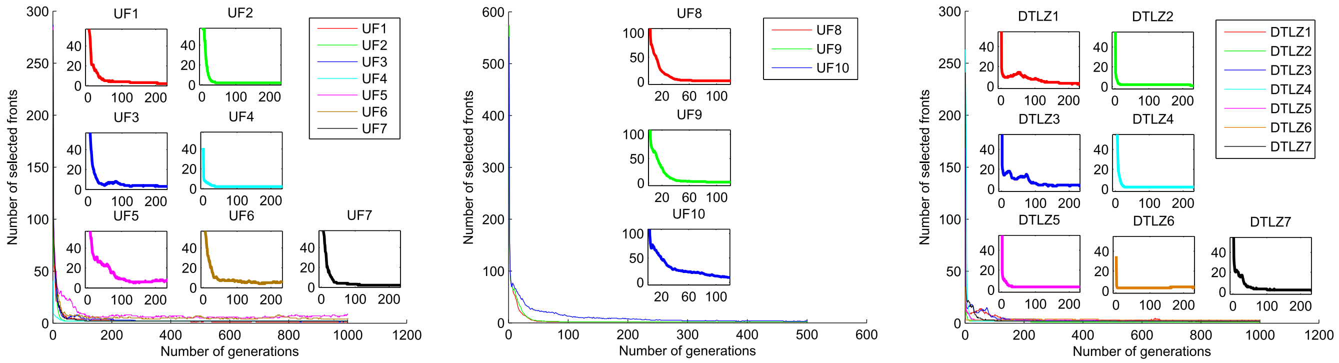
(j) (k) (l)

Fig. 4. Plots of all the final populations over 30 independent runs obtained by MOEA/D-SAS, MOEA/D-STM, and MOEA/D-DE on two UF instances and two DTLZ instances. The solution set obtained by (a) MOEA/D-SAS on UF2, (b) MOEA/D-STM on UF2, (c) MOEA/D-DE on UF2, (d) MOEA/D-SAS on UF8, (e) MOEA/D-STM on UF8, (f) MOEA/D-DE on UF8, (g) MOEA/D-SAS on DTLZ1, (h) MOEA/D-STM on DTLZ1, (i) MOEA/D-DE on DTLZ1, (j) MOEA/D-SAS on DTLZ7, (k) MOEA/D-STM on DTLZ7, and (l) MOEA/D-DE on DTLZ7.

D. Effects of Angle-Based-Selection

The ABS is proposed as a fine-grained diversity maintenance scheme in SAS. In this section, the effects of it are investigated and analyzed. We compare MOEA/D-SAS with a variant of itself [named MOEA/D-SAS(a)], in which the ABS is eliminated. The comparisons between these two algorithms can be considered as a way to test the effects of ABS. In addition, we also replace ABS with niche-counts [14], for MOEA/D-SAS. This variant of MOEA/D-SAS [named MOEA/D-SAS(n)] is also compared with the original MOEA/D-SAS.

The experimental results of comparing MOEA/D-SAS with MOEA/D-SAS(a) and MOEA/D-SAS(n) are presented in Table V. It can be observed that the performances



(a) (b) (c)

Fig. 5. Number of selected fronts versus the number of generations in the evolutionary process. (a) UF1-7. (b) UF8-10. (c) DTLZ107.

TABLE V
MEAN AND STANDARD DEVIATION VALUES OF IGD, OBTAINED BY MOEA/D-SAS, MOEA/D-SAS(a), AND MOEA/D-SAS(n) ON UF AND DTLZ INSTANCES

instance	*IGD*		
	MOEA/D-SAS	MOEA/D-SAS(a)	MOEA/D-SAS(n)
UF1	**1.644E-03 (5.52E-05)**	1.664E-03 (4.68E-05) †	1.655E-03 (4.64E-05)
UF2	7.162E-03 (2.57E-03)	6.888E-03 (1.77E-03)	**6.458E-03 (2.10E-03)**
UF3	9.987E-03 (8.25E-03)	7.360E-03 (4.17E-03)	**7.042E-03 (2.89E-03)**
UF4	**5.699E-02 (2.52E-03)**	5.736E-02 (3.86E-03)	5.773E-02 (3.41E-03)
UF5	**2.358E-01 (3.14E-02)**	2.541E-01 (3.57E-02) †	2.497E-01 (3.16E-02) †
UF6	**9.963E-02 (5.76E-02)**	1.072E-01 (5.27E-02)	1.178E-01 (4.80E-02) †
UF7	**2.480E-03 (6.28E-04)**	2.724E-03 (5.59E-04) †	2.814E-03 (9.01E-04)
UF8	**3.356E-02 (4.03E-03)**	4.069E-02 (7.37E-03) †	3.813E-02 (5.02E-03) †
UF9	2.925E-02 (3.88E-03)	2.885E-02 (2.84E-03)	**2.790E-02 (2.05E-03)**
UF10	5.236E-01 (6.87E-02)	**5.021E-01 (7.26E-02)**	5.178E-01 (6.27E-02)
DTLZ1	1.081E-02 (1.21E-04)	1.084E-02 (2.16E-04)	**1.077E-02 (1.94E-04)** ‡
DTLZ2	**2.897E-02 (3.68E-05)**	2.898E-02 (5.48E-05)	2.897E-02 (3.14E-05)
DTLZ3	3.363E-02 (3.33E-04)	3.368E-02 (3.76E-04)	**3.355E-02 (2.65E-04)**
DTLZ4	**3.404E-02 (2.41E-04)**	3.419E-02 (2.23E-04) †	3.411E-02 (1.75E-04)
DTLZ5	**3.598E-03 (1.43E-04)**	6.301E-03 (5.96E-06) †	6.300E-03 (4.98E-06) †
DTLZ6	**3.685E-03 (1.95E-04)**	6.294E-03 (1.04E-04) †	6.276E-03 (1.75E-04) †
DTLZ7	**6.837E-02 (5.13E-02)**	7.466E-02 (8.81E-03) †	9.645E-02 (1.30E-01) †

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-SAS and each of the other competing algorithms. † and ‡ denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-SAS, respectively. Boldface denotes the corresponding algorithm is better than the two other algorithms.

TABLE VI
PARAMETER SETTINGS IN MOEA/D-SAS AND NSGA-III FOR MANY-OBJECTIVE BENCHMARK PROBLEMS

–	Population size				# of neighbors	Crossover rates	Mutation rates	# of function evaluations
	M=5	M=8	M=10	M=15				
MOEA/D-SAS	212	156	276	136	10	1.0	1/n	300,000
NSGA-III	210	156	275	135	–	1.0	1/n	300,000

of MOEA/D-SAS are significantly better than that of MOEA/D-SAS(a) on 8 out of 17 benchmark problems. The performances between these two algorithms have no significant differences on the other nine benchmark problems. The results validate that ABS is very effective to improve the diversity of the population in most cases.

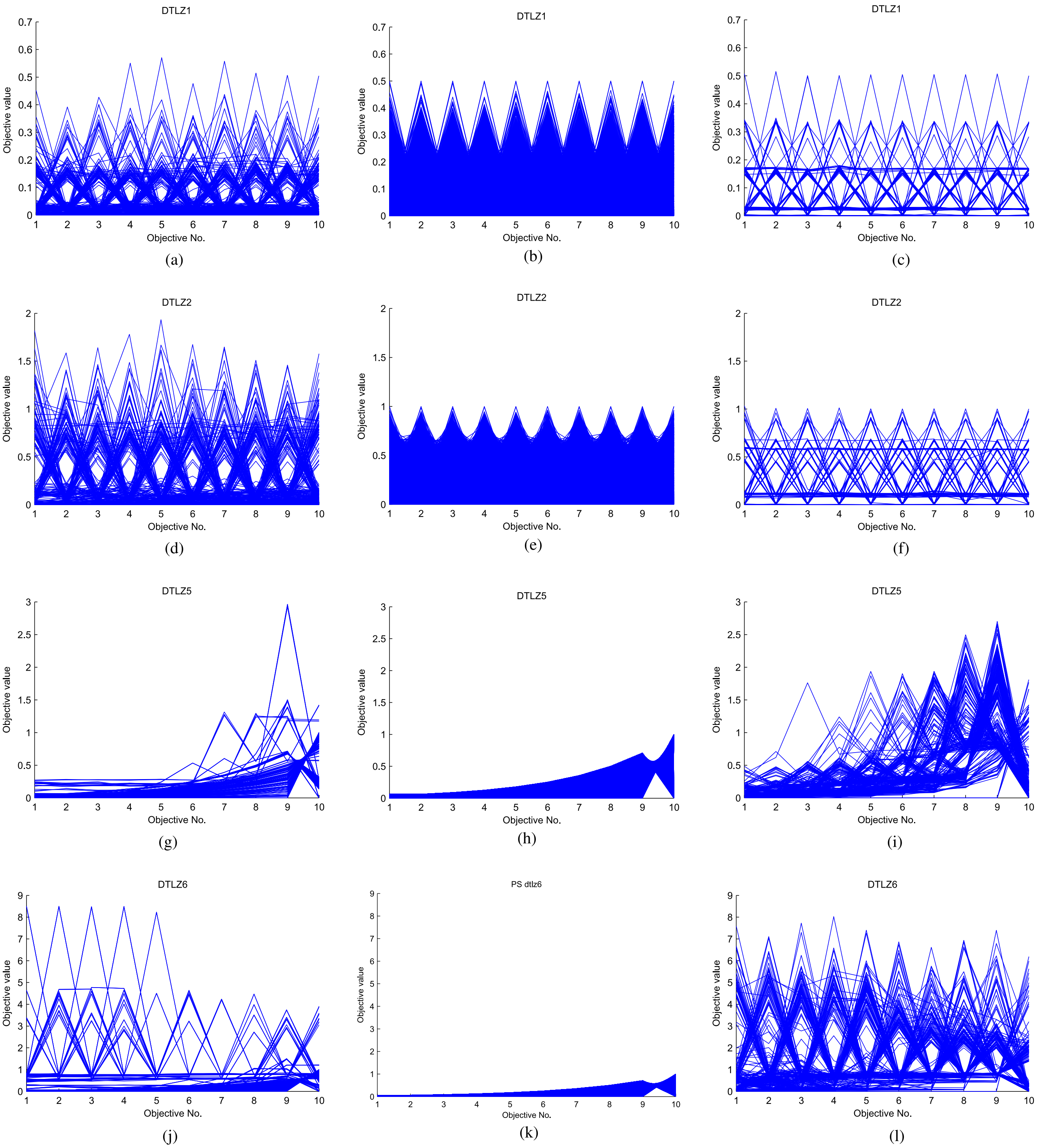
In addition, the performances of MOEA/D-SAS are significantly better on six benchmark problems and worse on one problem than that of MOEA/D-SAS(n). Both algorithms have very similar performances on the rest of benchmark problems. The above results are consistent with our motivations in Section I that ABS is more fine-grained than niche-counts scheme.

E. Performance of MOEA/D-SAS on Many-Objective Optimization Problems

1) MOEA/D-SAS Versus NSGA-III: NSGA-III [14], which is a state-of-the-art variant of NSGA-II, has shown very good performance on MOPs. In this section, MOEA/D-SAS and NSGA-III are compared on the 5-, 8-, 10-, and 15-objective DTLZ test problems. PBI is used as the decomposition approach for MOEA/D-SAS, where the penalty parameter β is set to 3 for DTLZ3 and 10 for all other test problems. More details with regard to the parameter settings are listed in Table VI.

The performance of MOEA/D-SAS and NSGA-III, in terms of IGD values, are presented in Table VII. It can be observed

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 6. Parallel coordinate plots for the nondominated solution set in the best run obtained by MOEA/D-SAS (left column) and NSGA-III (right column) on 10-objective DTLZ problems. The middle column shows the parallel coordinate plots of the reference solution sets sampled from the true PFs. (a), (d), (g), and (j) MOEA/D-SAS. (b), (e), (h), and (k) True PFs. (c), (f), (i), and (l) NSGA-III.

that MOEA/D-SAS is able to outperform NSGA-III on most test problems and MOEA/D-SAS has the increasingly better performance than NSGA-III when the number of objectives increases. It is also worth to note that DTLZ5 and DTLZ6 are the degenerated test problems, whose PFs are irregular. Nevertheless, MOEA/D-SAS constantly achieves better performance than NSGA-III on these two test problems.

To show the convergence and diversity for these two compared algorithms, the parallel coordinate plots of the solution sets obtained from the best run for MOEA/D-SAS and NSGA-III are shown in Fig. 6. It is clear to see that MOEA/D-SAS achieves much better diversity than NSGA-III on DTLZ1 and DTLZ2. For DTLZ5 and DTLZ6 whose PFs are degenerate, the shapes of parallel coordinate plots obtained by

TABLE VII
MEAN AND STANDARD DEVIATION VALUES OF IGD, OBTAINED BY MOEA/D-SAS AND NSGA-III ON DTLZ INSTANCES

M	instance	MOEA/D-SAS	NSGA-III	p-value
5	DTLZ1	6.667E-02 (1.447E-03)	**5.193E-02 (1.753E-04)**	3.02E-11
	DTLZ2	2.337E-01 (9.387E-03)	**1.332E-01 (2.645E-04)**	3.02E-11
	DTLZ3	1.851E+00 (2.254E+00)	**1.341E-01 (3.200E-03)**	3.02E-11
	DTLZ4	2.108E-01 (1.228E-02)	**1.331E-01 (1.266E-04)**	3.02E-11
	DTLZ5	**2.043E-02 (1.241E-03)**	9.550E-02 (2.769E-02)	3.02E-11
	DTLZ6	**2.029E-02 (1.010E-03)**	1.364E+00 (9.325E-02)	3.02E-11
	DTLZ7	5.926E-01 (4.230E-02)	**4.116E-01 (1.255E-01)**	6.52E-09
8	DTLZ1	**9.579E-02 (7.327E-04)**	2.001E-01 (5.059E-02)	1.11E-06
	DTLZ2	4.999E-01 (1.240E-02)	5.535E-01 (2.323E-01)	0.1858
	DTLZ3	5.735E-01 (6.574E-01)	6.850E-01 (2.437E-01)	0.2282
	DTLZ4	5.010E-01 (6.964E-03)	**3.654E-01 (1.242E-03)**	3.02E-11
	DTLZ5	**2.524E-02 (8.936E-04)**	3.256E-01 (1.126E-01)	3.02E-11
	DTLZ6	**2.391E-01 (5.993E-01)**	4.014E+00 (4.411E-01)	3.02E-11
	DTLZ7	**1.364E+00 (1.642E-01)**	2.286E+00 (3.537E-01)	3.02E-11
10	DTLZ1	**9.380E-02 (7.515E-04)**	2.318E-01 (4.330E-02)	8.48E-09
	DTLZ2	**5.215E-01 (5.599E-03)**	7.467E-01 (2.207E-01)	7.96E-03
	DTLZ3	1.267E+00 (1.907E+00)	**8.784E-01 (9.647E-02)**	4.94E-05
	DTLZ4	5.675E-01 (5.178E-03)	**4.159E-01 (2.819E-03)**	3.02E-11
	DTLZ5	**2.064E-02 (2.453E-04)**	4.465E-01 (1.092E-01)	3.02E-11
	DTLZ6	**1.789E-01 (4.577E-01)**	4.961E+00 (4.647E-01)	3.02E-11
	DTLZ7	**1.765E+00 (9.144E-02)**	4.572E+00 (7.510E-01)	3.02E-11
15	DTLZ1	**7.082E-02 (5.327E-03)**	3.121E-01 (8.653E-03)	3.02E-11
	DTLZ2	**4.510E-01 (3.219E-02)**	1.014E+00 (2.716E-02)	3.02E-11
	DTLZ3	**3.593E-01 (2.267E-01)**	1.137E+00 (3.675E-01)	5.57E-10
	DTLZ4	**1.478E-01 (1.124E-02)**	1.031E+00 (2.401E-02)	6.77E-05

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-SAS and NSGA-III. Boldface denotes that the performance of the corresponding algorithm is significantly better than that of the other.

TABLE VIII
MEAN AND STANDARD DEVIATION VALUES OF IGD, OBTAINED BY MOEA/D-SAS AND MOEA/D-AWA ON DTLZ TEST PROBLEMS WITH DISCONNECTED AND DEGENERATE PFS

M	instance	MOEA/D-SAS	MOEA/D-AWA
5	DTLZ5	**2.043E-02 (1.241E-03)**	1.134E-01 (1.560E-02)
	DTLZ6	**2.029E-02 (1.010E-03)**	2.173E-01 (3.213E-02)
	DTLZ7	5.926E-01 (4.230E-02)	**2.353E-01 (1.035E-02)**
8	DTLZ5	**2.524E-02 (8.936E-04)**	1.627E-01 (4.401-02)
	DTLZ6	**2.391E-01 (5.993E-01)**	5.790E-01 (2.192E-01)
	DTLZ7	1.364E+00 (1.642E-01)	**9.570E-01 (1.370E-01)**
10	DTLZ5	**2.064E-02 (2.453E-04)**	2.604E-01 (6.477E-02)
	DTLZ6	**1.789E-01 (4.577E-01)**	7.450E-01 (3.223E-03)
	DTLZ7	1.765E+00 (9.144E-02)	**1.210E+00 (1.175E-01)**

Wilcoxon's rank sum test at a 0.05 significance level is performed between MOEA/D-SAS and MOEA/D-AWA. Boldface denotes that the performance of the corresponding algorithm is significantly better than that of the other.

MOEA/D-SAS is much more similar to that of true PFs than the ones obtained by NSGA-III.

2) MOEA/D-SAS Versus MOEA/D-AWA: In the MOEA/D with adaptive weight adjustment (MOEA/D-AWA) [33], an adaptive weight vector adjustment strategy is introduced. The weight vectors of subproblems are adjusted periodically to be redistributed adaptively for obtaining better uniformity of solutions. Different from MOEA/D-AWA, MOEA/D-SAS uses the fixed set of weight vectors. However, different solutions can be associated with the same subproblems; and some subproblems are allowed to have no associated solution. To compare the effects of MOEA/D-AWA and MOEA/D-SAS on irregular MOPs, experiments are conducted between these two algorithms on 5-, 8-, and 10-objective DTLZ5–7 test problems that have disconnected or degenerate PFs, as shown in Table VIII. It can be seen that MOEA/D-SAS performs significantly better than MOEA/D-AWA on all DTLZ5-6 test problems though it performs worse than MOEA/D-AWA on DTLZ7.

VII. CONCLUSION

This paper proposed an SAS as the selection operator for MOEA/D to address MOPs. In SAS, the balance between convergence and diversity is achieved by two components, DBS and ABS. Different from other selection schemes, e.g., global STM model, DBS only conducts sorting within the local neighboring solutions, which drastically reduce the computational cost of SAS. Meanwhile, ABS utilizes the angle information in the objective space to maintain a fine-grained diversity. Different from many other MOEA/D variants, SAS allows one subproblem to associate with any number of solutions, or even no solutions, which makes it more flexible for MOPs with different shapes of PFs. SAS is integrated into MOEA/D and the algorithm, called MOEA/D-SAS, is compared with four classical (NSGA-II, MSOPS-II, MOEA/D, and MOEA/D-DE) and three state-of-the-art MOEAs (MOEA/D-STM, NSGA-III, and MOEA/D-AWA) on continuous MOPs or MaOPs. The experimental results show that MOEA/D-SAS outperforms other compared algorithms. In addition, the computational efficiency of DBS and the effects of ABS are also discussed in this paper in detail.

REFERENCES

[1] M. Asafuddoula, T. Ray, and R. A. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

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

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

[4] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[5] Q. Cai, M. Gong, S. Ruan, Q. Miao, and H. Du, "Network structural balance based on evolutionary multiobjective optimization: A two-step approach," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 903–916, Dec. 2015.

[6] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.

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

[8] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evol. Mach.*, vol. 6, no. 2, pp. 163–190, 2005.

[9] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

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

[11] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.

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

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

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

[15] 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.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," *Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005, pp. 105–145.

[17] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "On the effect of the steady-state selection scheme in multi-objective genetic algorithms," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, vol. 5467. Nantes, France, Apr. 2009, pp. 183–197.

[18] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[19] S. B. Gee, K. C. Tan, V. A. Shim, and N. R. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 542–559, Aug. 2015.

[20] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, Sep. 2007, pp. 3944–3951.

[21] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 4. Canberra, ACT, Australia, Dec. 2003, pp. 2678–2684.

[22] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. Evol. Comput.*, to be published.

[23] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. 5th Int. Conf. Evol. Multi Criterion Optim. (EMO)*, vol. 5467. Nantes, France, Apr. 2009, pp. 438–452.

[24] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Trans. Evol. Comput.*, Mar. 2016, doi: 10.1109/TEVC.2016.2549267. .

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

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

[27] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Inter-relationship based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

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

[29] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

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

[31] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optim.*, Zürich, Switzerland, 2001, pp. 82–95.

[32] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization* (Natural Computing Series). Heidelberg, Germany: Springer, 2005.

[33] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[34] A. B. Ruiz, R. Saborido, and M. Luque, "A preference-based evolutionary algorithm for multiobjective optimization: The weighting achievement scalarizing function genetic algorithm," *J. Glob. Optim.*, vol. 62, no. 1, pp. 101–129, 2015.

[35] J. D. Schaffer and J. J. Grefenstette, "Multi-objective learning via genetic algorithms," in *Proc. 9th Int. Joint Conf. Artif. Intell. (IJCAI)*, Los Angeles, CA, USA, 1985, pp. 593–595.

[36] X.-N. Shen and X. Yao, "Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems," *Inf. Sci.*, vol. 298, pp. 198–224, Mar. 2015.

[37] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications* (Advanced Information and Knowledge Processing). London, U.K.: Springer, 2005.

[38] J. Wang *et al.*, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.

[39] R. Wang, Q. Zhang, and T. Zhang, "Decomposition based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, to be published.

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

[41] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Elect. Eng., Univ. Essex, Colchester, U.K. and Nanyang Technol. Univ., Singapore, and Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, 2008.

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

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

[44] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems—EUROGEN 2001*, K. C. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds. Athens, Greece: CIMNE, 2002, pp. 95–100.

[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

Authors' photographs and biographies not available at the time of publication.