Tree-structured Decomposition and Adaptation in MOEA/D

Hanwei Zhang[1,2] and Aimin Zhou[1⋆]

[1] Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, Shanghai, China
[2] CNRS-IRISA, Rennes, France
hanwei.zhang@irisa.fr,amzhou@cs.ecnu.edu.cn

Abstract. The *multiobjective evolutionary algorithm based on decomposition (MOEA/D)* converts a *multiobjective optimization problem (MOP)* into a set of simple subproblems, and deals with them simultaneously to approximate the Pareto optimal set (PS) of the original MOP. Normally in MOEA/D, a set of weight vectors are predefined and kept unchanged during the search process. In the last few years, it has been demonstrated in some cases that a set of predefined subproblems may fail to achieve a good approximation to the Pareto optimal set. The major reason is that it is usually unable to define a proper set of subproblems, which take full consideration of the characteristics of the MOP beforehand. Therefore, it is imperative to develop a way to adaptively redefine the subproblems during the search process. This paper proposes a *tree-structured decomposition and adaptation (TDA)* strategy to achieve this goal. The basic idea is to use a tree structure to decompose the search domain into a set of subdomains that are related with some subproblems, and adaptively maintain these subdomains by analyzing the search behaviors of MOEA/D in these subdomains. The TDA strategy has been applied to a variety of test instances. Experimental results show the advantages of TDA on improving MOEA/D in dealing with MOPs with different characteristics.

1 Introduction

Decomposition based *multiobjective evolutionary algorithms (MOEAs)* [1–3] have recently been attracting much attention for dealing with *multiobjective optimization problems (MOPs)*. A main difference between the decomposition based MOEAs and the other two major MOEA paradigms, i.e., the Pareto domination based approaches [4, 5] and the indicator based approaches [6–8], lies in the environmental selection. To differentiate solutions in the environmental selection, the Pareto domination based approaches use the Pareto domination relationship and a density estimation strategy to define a complete ranking order of the solutions, and the indicator based approaches utilize a performance indicator to score a solution or a subpopulation. Since the decomposition based approaches

⋆ Corresponding author

convert an MOP into a set of subproblems, the environmental selection is implemented for each subproblem [9], i.e., if the subproblem is a scalar-objective problem, the subproblem objective value can be directly used to do selection; if the subproblem is a multiobjective problem, the above two selection approaches can be used. For both scalar-objective and multiobjective subproblems, they are tackled simultaneously.

The *multiobjective evolutionary algorithm based on decomposition (MOEA/D)* is a typical decomposition based MOEA [10]. The combination of the found optimal solutions of the subproblems will constitute an approximation to the Pareto optimal set of the original MOP. A variety of methods have been proposed to decompose an MOP into a set of subproblems in MOEA/D [10,11]. Let $\min_{x\in\Omega} F(x) = \{f_1(x), \cdots, f_m(x)\}$ be a general MOP, where x is a decision variable vector, Ω denotes the feasible region of the search space, and $f_i(x)$ is the ith objective. This paper considers the Tchebycheff approach [10] that defines a parameterized scalar-objective subproblem $g(x|w, z^*) = \max_{1\leq j\leq m} w^i|f_i(x) - z_j^*|$ with reference point $z^* = (z_1^*, \cdots, z_m^*)$ and weight vector $w = (w_1, \cdots, w_m)$, which is required that $w_i \geq 0$, for $i = 1, \cdots, m$, and $\sum_{i=1}^m w_i = 1$. It is clear that all the weight vectors are from an $(m-1)$-dimensional simplex. For simplicity, we use g^i to denote $g(x|w^i, z^*)$ in the sequel.

The approximation quality is determined by the weight vectors and the reference point. In different MOEA/D variants, the reference point is adaptively updated by the best solutions found so far and this strategy works well. However, when the Pareto Front (PF) shape of an MOP is complicated (e.g. disconnected, ill-scaled), the uniform sampling strategy may fail to find a good approximation of the PF. A natural way to deal with this problem is to adaptively adjust the weight vectors during the search process. Several works have been done along this direction [12–16]. This paper proposes a new way to adjust the weight vectors dynamically, called *tree-structured decomposition and adaptation (TDA)*, and name MOEA/D with TDA as MOEA/D-TDA. The basic idea is to maintain a tree structure to decompose the search domain into a set of subdomains that are related to some subproblems, and adaptively adjust the subdomains to find the Pareto optimal solutions of an MOP. The search domain, in both the objective and decision spaces, is recursively decomposed into a set of simplexes through a set of weight vectors in the weight space. The simplex is regarded as a basic unit of the search process. Each simplex is represented as a node in the tree structure. The sparseness of the simplexes is measured along the search process. According to the measurement, some new simplexes are added in the sparse areas and some old ones are removed from the dense areas. And a tree structure makes it efficient for these operations. Since it is hard to find a good approximation to both the PF and the PS, we utilize two populations: an internal working population that approximates the PS and an external archive that approximates the PF.

The rest of the paper is organized as follows. Section 2 presents the proposed method in detail. Sections 3 and 4 study the major components in the new approach. Section 5 gives the experimental studies. Finally, the paper is concluded in Section 6 with some suggestions for future work.

2 The Proposed Method

In this section, we introduce the strategy *tree-structured decomposition and adaptation (TDA)* in details. Firstly, the framework of the algorithm is given. After that, we explain how to partition the decision, the objective, and the weight domains into some subdomains by using a tree structure named decomposition tree. Last but not least, we depict the approach to adaptively change the weight vectors by adding or removing subdomains according to the search behaviors.

2.1 Algorithm Framework

MOEA/D-TDA maintains a set of scalar-objective subproblems, and the ith ($i = 1, \cdots, N$) subproblem is with (a) its weight vector w^i and objective function $g^i(x)$, (b) its current solution x^i and its objective vector $F^i = F(x^i)$, and (c) the index set of its neighboring subproblems, B^i, of which the weight vectors are closest to w^i.

With TDA, the domains are decomposed recursively using a Tree-structured $DT = \{D^k\}$, $k = 1, \cdots, K$, which is called the decomposition tree. Each node in DT represents a subdomain and is defined as $D^k =< p, O, W, E >$ where p is the index of its parent node, O contains the indices of its child nodes, W contains the weight vectors of the subproblems that are directly related to the domain, and E is the set of all the edges of the simplex that forms the subdomain. It should be noted that

- Each domain is with m subproblems, i.e., $|W| = m$, in which m is the number of objectives.
- $O = \emptyset$ if D^k is a leaf node or $|O| = 2^{m-1}$ otherwise.
- $\bigcup_{D \in DT} D.W$ contains the weight vectors of all the subproblems.
- Let $N = |\bigcup_{D \in DT} D.W|$ and K be the number of subproblems and the number of subdomains respectively. $D.W$ denotes the weight vectors for the domain D. Neither N nor K is fixed throughout the run, and we discuss this in the next section.

Algorithm 1: Main Framework of MOEA/D-TDA

```
1  Initialize a decomposition tree $DT$.
2  Initialize all the subproblems according to the weight vectors $\bigcup_{D \in DT} D.W$ with $DT$.
3  Initialize the reference point $z^* = (z_1^*, \cdots, z_m^*)$ as $z_j^* = \min_{i=1,\cdots,N} f_j(x^i)$ for $j = 1, , m$.
4  while not terminate do
5      Update the decomposition tree $DT$.
6      Update the neighborhood structure according to the weight vectors.
7      foreach subproblem $i \in \{1, \cdots, N\}$ do
8          Generate a new solution $y = Generate(i)$.
9          Update the reference point $z^*$ by resetting $z_j^* = f_j(y)$ if $z_j^* \geq f_j(y)$ for $j = 1, , m$.
10         Update the population by the new trial solution $y$.
```

MOEA/D-TDA also needs to maintain a reference point $z^* = (z_1^*, \cdots, z_m^*)$. The main framework of MOEA/D-TDA is shown in Algorithm 1. We would make the following comments on the framework.

- In *Line 2*, the weight vectors are generated in the decomposition tree initialization process. Each solution is initialized by a randomly sampled point from Ω and is assigned to subproblems according to the weight vectors.
- The reference point z^* is initialized in *Line 3* and updated in *Line 9*.
- In *Line 4*, a maximum number of generations is used as the termination condition.
- In *Line 6*, the neighborhood structure needs to be updated since the weight vectors may change in *Line 5*.
- In *Line 7*, each subproblem is selected for offspring generation and population update in each generation.

Basically, the above algorithm framework is following the original MOEA/D framework [17]. *Line 8* generates a new solution y. There are various ways to implement it. It should be noted that, in this paper, this procedure is the same generation procedure as in MOEA/D-DE [17]. *Line 10* tries to replace one solution in the current population by the trial solution y. In this paper, we use the approach defined in [18]. In the next section, we emphasize the decomposition tree initialization in *Line 1*, the decomposition tree update in *Line 5*.

2.2 Domain Decomposition

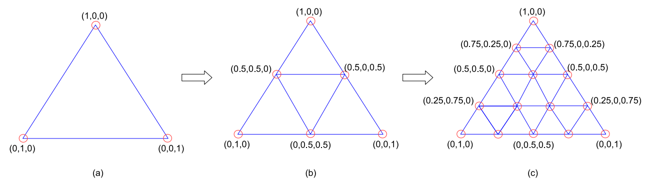

Fig. 1. An illustration of decomposition tree initialization in the case of tri-objective problems.

Let N_0 be the desired population size. The domain decomposition process starts by setting the decomposition tree as the weight domain, then recursively decomposes the subdomains until the number of weight vectors exceeds N_0. Fig. 1 illustrates the decomposition tree initialization process in the case of tri-objective problems. Each edge of the weight simplex is cut into two equal-length edges and some subdomains with equal size are generated. It can be deduced easily that when decomposing a subdomain, 2^{m-1} new subdomains and $2^{m-1}-1$ new weight vectors are generated.

Let $e_i = (e_{i,1}, e_{i,2}, \cdots, e_{i,m})$ denote the unit vector in the coordinate system where $e_{i,j} = 0$ if $j \neq i$, and $e_{i,i} = 1$. The decomposition tree initialization process is shown in Algorithm 2.

Algorithm 2: Decomposition Tree Initialization

```
1 Set $DT = \{D^1\}$ where $D^1.p = 0$, $D^1.O = \emptyset$, and $D^1.W = \{e_1, \cdots, e_m\}$.
2 while $|\bigcup_{D \in DT} D.W| < N_0$ do
3     Let $D \in DT$ be a randomly chosen leaf node that has the lowest depth.
4     Decompose domain $D$ into a set of subdomains, set the child nodes of $D$ be these
        subdomains, and add them to $DT$.
```

In Algorithm 2, we define the depth of the root node as 1, and the depth of a child node is the depth of its parent node plus 1. *Line 3* makes sure that it is always a leaf node, which is most closet to the root node, to be decomposed. It should also be noted that in *Line 3*, to keep the population size, not all the leaf nodes with the same depth will be decomposed.

2.3 Domain Adaptation

As discussed previously, MOEA/D can obtain a set of well-distributed solutions by setting proper weight vectors. To this end, we adaptively change the weight vectors by adding some nodes in sparse areas and removing some nodes in dense areas. This idea is implemented in TDA by adding some new subdomains and removing some old nodes respectively.

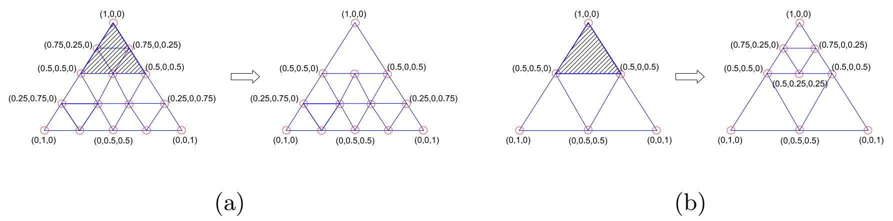

Fig. 2. An illustration of (a) deleting old domains and (b) inserting new domains in the case of tri-objective problems.

Fig. 2(a) illustrates, in the case of tri-objective problems, how to remove a subdomain. It should be noted that not all subdomains can be removed, and a removable subdomain is the one that contains only one level of child subdomains. Once a subdomain is removed, some of the corresponding weight vectors and subproblems are removed as well. Since a weight vector may be shared by several subdomains, only the unused weights can be removed. Fig. 2(b) illustrates, in the case of tri-objective problems, how to add some subdomains. Some new weight vectors and subproblems are added as well.

Let $d(D)$ be a function that measures the search behavior, which is the density in this paper, of subdomain D. We assume a lower $d(D)$ value denotes that subdomain D is dense while a higher $d(D)$ value denotes that subdomain D is sparse. The decomposition tree adaptation process is shown in Algorithm 3.

Algorithm 3: Decomposition Tree Adaptation

```
1 Let $D_1 \subset DT$ be the set of removable nodes, and sort them by an increasing order of their
    $d(\cdot)$ values.
2 Let $D_2 \subset S$ be the set of leaf nodes, and sort them by a decreasing order of their $d(\cdot)$ values.
3 Set $d_1 = first(D_1)$ and $d_2 = first(D_2)$.
4 while $|D_1| > 0$ and $|D_2| > 0$ and $d(d_1) < d(d_2)$ do
5     Delete node $d_1$ from DT, and set $D_1 = D_1 \backslash \{d_1\}$.
6     Decompose $d_2$, add new nodes to DT, and set $D_2 = D_2 \backslash \{d_2\}$.
7     Remove the parent node of $d_2$ from $D_1$ by setting $D_1 = D_1 \backslash \{parent(d_2)\}$.
8     Resort $D_1$ and $D_2$, set $d_1 = first(D_1)$ and $d_2 = first(D_2)$.
```

We would like to make some comments on the algorithm.

- The process stops in *Line 4* when there is no removable subdomain to delete, or no subdomain to decompose, or the density of the subdomain to delete is bigger than that of the one to decompose. The target is to make all subdomains have the same density values and thus to obtain a set of well-distributed final solutions.
- In each step, one subdomain is removed and one is decomposed. The target is to keep a stable population size although the number of added weight vectors may not be the same as the number of removed weight vectors.
- When a subdomain is deleted from DT in *Line 5*, the corresponding weight vectors and subproblems are deleted as well if the weight vectors are not used by other subdomains.
- When a subdomain is added to DT in *Line 6*, some new weight vectors and subproblems are also added. Each new subproblem is initialized with a randomly generated solution and with infinite objective values.
- In *Line 7*, the parent node of d_2 is removed from D_1 to prevent the newly added subdomains to be deleted again in the next steps.
- $d(\cdot)$ is a function to measure subdomain by measuring its density. Howe to define the function will be discussed later.

3 Subdomain Measurement

To implement MOEA/D-TDA, a key issue is on how to measure the subdomain. Density might be a good choice in this case. We define the density of a simplex as follows.

$$\begin{aligned} df(s) &= \sum_{w_i w_j \in s.E} ||F(x^i) - F(x^j)||_2 \\ dx(s) &= \sum_{w_i w_j \in s.E} ||x^i - x^j||_2 \\ dw(s) &= \sum_{w_i w_j \in s.E} ||w^i - w^j||_2 \end{aligned} \tag{1}$$

where $w_i w_j$ is an edge in the simplex s, and $|| \cdot ||_2$ denotes the L_2 norm, x^i and x^j are two solutions with w_i and w_j respectively. $df(\cdot)$, $dx(\cdot)$, and $dw(\cdot)$ measure the density of the subdomain in the objective space, in the decision space, and in the weight space respectively. It is clear that, if $dw(\cdot)$ is used, MOEA/D-TDA is actually the original MOEA/D because the initial weight vectors are well-distributed; otherwise if $df(\cdot)$ or $dx(\cdot)$ is used, MOEA/D-TDA will emphasize the search behavior in either the objective space or the decision space. If we attach more importance to the objective space, we name this version as TDA-F while TDA-X for the version underlines the decision space.

It should be noted that the above measurements are just examples and other subdomain measurements could be defined and used in MOEA/D-TDA. In following, we study the influence of the three subdomain measurements defined in (1). We choose two problems, i.e., LZ3 [17] and its variant SLZ3 [1], as examples in the study. The parameter settings are as follows: the population size $N = 300$, the number of decision variables $n = 30$, and the neighborhood size $T = 20$. The parameters in offspring reproduction are $\delta = 0.9$, $\mathrm{F} = 0.5$, and $\eta = 20$. The maximum FE number is 3×10^5 for all the algorithms. Each algorithm is executed in each problem with 50 independent runs. For quantitative comparison, the *Inverted Generational Distance (IGD)* metric [19] is used and the reference point set has 1000 points.

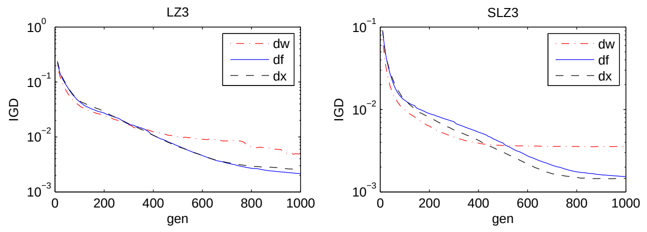

Fig. 3. The mean IGD metric values versus generations for MOEA/D-TDA with different density measurements on LZ3 and SLZ3.

The experimental results are shown in Fig. 3. From the figure, we can conclude that (a) it is hard to balance the population diversity in both the decision and the objective spaces if the diversity maintains strategy is used only in one space, and (b) in some cases, to keep the population diversity in the objective, it is necessary to keep the population diversity in the decision space.

4 External Population

As discussed in the above section, in order to balance population diversity in both the objective and decision spaces we need an external population (archive) to MOEA/D-TDA. A step to maintain the external population should be added to Algorithm 1 after *Line 10.* The two populations are with different usages: the

[1] The LZ test instances are scaled by replace the original $f_1(x)$ function by $0.1f_1(x)$.

internal population tries to approximate the PS in the decision space, and the *external population* tries to approximate the PF in the objective space.

In the new approach, the offspring generation operation is based on the internal population, and the density measurement dx is applied to tune the subproblems and thus to maintain the diversity of the internal population. The external population is initialized as the internal population. The newly generated solutions are used to update the external population. The solutions in the external population will not be used for offspring generation, but they will be output as the approximation result. It should be noted that any archive strategy can be integrated into MOEA/D-TDA. In the following experiment, we consider the following strategies: (a) NDS: the nondomination sorting scheme from NSGA-II [5], (b) HBS: the hypervolume based selection from SMS-EMOA [8], and (c) DBS: the population maintain strategy introduced in this paper with the density measurement $df(\cdot)$.

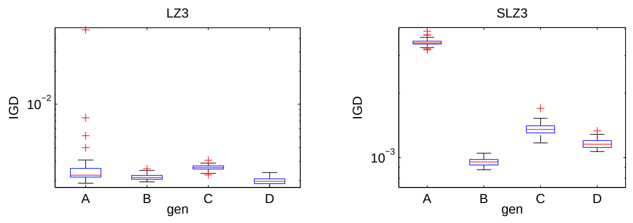

Fig. 4. Box-plots of IGD values of the final results obtained by the four algorithms over 50 independent runs.

To demonstrate the contribution of external population the corresponding maintain strategies, we empirically compare the following four algorithms on LZ3 and SLZ3: (a) A: MOEA/D-TDA with dw and without an external population, i.e., the original MOEA/D, (b) B: MOEA/D-TDA with dx and with an external population maintained by NDS, (c) C: MOEA/D-TDA with dx and with an external population maintained by HBS, and (d) D: MOEA/D-TDA with dx and with an external population maintained by DBS.

Fig. 4 shows the box-plots of the IGD metric values of the final results obtained by the four algorithms. From the figure, we can see that by using external population, the approximation quality can be significantly improved. Comparing the three external population maintain strategies, the experimental results suggest that MOEA/D-TDA with NDS performs the best. The reason might be that it is more suitable to approximate the PF especially when the PF is scaled.

Table 1. The mean and standard deviation of IGD values obtained by five algorithms over 50 runs on the LZ and SLZ suites.

	TDA-X	$1.407e-03_{1.681e-05}$[4]		TDA-X	$\mathbf{8.847e-04_{1.384e-05}}$[1]
	TDA-F	$1.403e-03_{1.288e-05}$[3]		TDA-F	$9.476e-04_{1.955e-05}$[2]
LZ1	DE	$\mathbf{1.280e-03_{3.029e-06}}$[1]	SLZ1	DE	$3.339e-03_{1.113e-05}$[4]
	M2M	$1.391e-03_{5.464e-05}$[2]		M2M	$3.411e-03_{1.655e-04}$[5]
	AWA	$1.809e-03_{7.184e-05}$[5]		AWA	$1.017e-03_{1.511e-05}$[3]
	TDA-X	$\mathbf{2.243e-03_{1.688e-04}}$[1]		TDA-X	$\mathbf{9.417e-04_{2.946e-05}}$[1]
	TDA-F	$2.539e-03_{1.767e-04}$[3]		TDA-F	$1.117e-03_{7.294e-05}$[2]
LZ2	DE	$2.429e-03_{2.395e-04}$[2]	SLZ2	DE	$3.709e-03_{2.853e-04}$[3]
	M2M	$3.157e-03_{7.434e-04}$[4]		M2M	$5.760e-03_{2.057e-03}$[4]
	AWA	$3.109e-02_{8.845e-03}$[5]		AWA	$1.219e-02_{5.529e-03}$[5]
	TDA-X	$\mathbf{2.128e-03_{1.131e-04}}$[1]		TDA-X	$\mathbf{9.521e-04_{2.799e-05}}$[1]
	TDA-F	$2.160e-03_{1.674e-04}$[2]		TDA-F	$1.420e-03_{9.339e-04}$[2]
LZ3	DE	$2.549e-03_{1.241e-03}$[4]	SLZ3	DE	$3.518e-03_{1.934e-04}$[3]
	M2M	$2.327e-03_{1.821e-04}$[3]		M2M	$3.774e-03_{2.733e-04}$[4]
	AWA	$7.092e-03_{2.240e-03}$[5]		AWA	$5.082e-03_{2.638e-03}$[5]
	TDA-X	$\mathbf{2.010e-03_{9.338e-05}}$[1]		TDA-X	$\mathbf{9.371e-04_{3.229e-05}}$[1]
	TDA-F	$2.192e-03_{2.031e-04}$[2]		TDA-F	$1.273e-03_{3.375e-04}$[2]
LZ4	DE	$3.016e-03_{1.512e-03}$[4]	SLZ4	DE	$3.544e-03_{1.481e-04}$[4]
	M2M	$2.966e-03_{6.310e-04}$[3]		M2M	$3.767e-03_{3.099e-04}$[5]
	AWA	$3.119e-03_{2.191e-04}$[5]		AWA	$1.457e-03_{2.386e-04}$[3]
	TDA-X	$7.867e-03_{2.919e-03}$[4]		TDA-X	$\mathbf{2.967e-03_{8.921e-04}}$[1]
	TDA-F	$6.872e-03_{1.391e-03}$[2]		TDA-F	$3.858e-03_{1.192e-03}$[2]
LZ5	DE	$7.091e-03_{1.537e-03}$[3]	SLZ5	DE	$4.185e-03_{8.924e-04}$[3]
	M2M	$\mathbf{4.240e-03_{4.501e-04}}$[1]		M2M	$4.728e-03_{6.535e-04}$[4]
	AWA	$1.163e-02_{2.879e-03}$[5]		AWA	$6.727e-03_{2.693e-03}$[5]
	TDA-X	$1.764e-01_{4.460e-02}$[4]		TDA-X	$1.787e-01_{9.593e-02}$[4]
	TDA-F	$2.587e-01_{4.757e-02}$[5]		TDA-F	$2.015e-01_{8.980e-02}$[5]
LZ6	DE	$\mathbf{3.015e-02_{5.592e-03}}$[1]	SLZ6	DE	$8.681e-02_{1.326e-02}$[3]
	M2M	$6.748e-02_{2.449e-02}$[3]		M2M	$7.559e-02_{9.218e-03}$[2]
	AWA	$5.286e-02_{4.625e-03}$[2]		AWA	$3.230e-02_{9.002e-03}$[1]
	TDA-X	$2.410e-01_{9.336e-02}$[5]		TDA-X	$5.807e-02_{5.907e-02}$[3]
	TDA-F	$2.342e-01_{8.378e-02}$[4]		TDA-F	$6.217e-02_{5.097e-02}$[4]
LZ7	DE	$8.053e-02_{8.047e-02}$[3]	SLZ7	DE	$3.842e-02_{2.723e-02}$[2]
	M2M	$5.082e-02_{6.167e-02}$[2]		M2M	$1.037e-01_{6.527e-02}$[5]
	AWA	$\mathbf{2.452e-03_{1.740e-04}}$[1]		AWA	$\mathbf{1.087e-03_{2.781e-05}}$[1]
	TDA-X	$1.774e-02_{1.737e-02}$[3]		TDA-X	$3.048e-03_{1.409e-03}$[2]
	TDA-F	$2.086e-02_{2.080e-02}$[4]		TDA-F	$\mathbf{2.696e-03_{1.727e-03}}$[1]
LZ8	DE	$\mathbf{3.653e-03_{5.015e-03}}$[1]	SLZ8	DE	$5.008e-03_{7.817e-04}$[3]
	M2M	$1.060e-02_{4.170e-03}$[2]		M2M	$5.383e-03_{7.574e-04}$[4]
	AWA	$6.847e-02_{2.812e-02}$[5]		AWA	$1.444e-02_{1.589e-02}$[5]
	TDA-X	$2.499e-03_{5.715e-04}$[2]		TDA-X	$\mathbf{1.080e-03_{9.872e-05}}$[1]
	TDA-F	$3.998e-03_{1.559e-03}$[3]		TDA-F	$1.484e-03_{1.236e-04}$[2]
LZ9	DE	$\mathbf{2.311e-03_{1.561e-04}}$[1]	SLZ9	DE	$6.135e-03_{1.927e-03}$[3]
	M2M	$4.874e-03_{1.894e-03}$[4]		M2M	$6.569e-03_{1.232e-03}$[4]
	AWA	$1.844e-01_{1.558e-02}$[5]		AWA	$3.887e-02_{2.324e-02}$[5]
	TDA-X	2.8		TDA-X	1.7
	TDA-F	3.1		TDA-F	2.4
Mean Rank	DE	2.2	Mean Rank	DE	3.1
	M2M	2.7		M2M	4.1
	AWA	4.2		AWA	4.7

5 Comparison Study

In this section, we study the performance of the proposed strategy with some state-of-the-art algorithms on some test suites. The following algorithms are compared: (a) *TDA*: MOEA/D-TDA with dx and with an external population maintained by NDS, (b) *DE*: MOEA/D-DE [20], which is a conceptual MOEA/D algorithm and is similar to MOEA/D-TDA with dw, (c)*AWA*: MOEA/D-AWA [13], which is a variation of MOEA/D by adapting weight vectors in evolution, and (d) *M2M*: MOEA/D-M2M [15], which decomposes an MOP into a set of MOPs and tackle these MOPs simultaneously.

The first five instances in the LZ test suite [17], their variants, in which f_1 is scaled to $10f_1$, and the GLT test suite [21] are used in the comparison study. The variants of LZ1-LZ5 are called SLZ1-SLZ5 respectively. The experimental

Table 2. The mean and standard deviation of IGD values obtained by five algorithms after different percentages of function evaluations over 50 runs on the GLT suite.

		20%	60%	100%
GLT1	TDA-X	$1.909e-02_{1.528e-02}$[4]	$3.881e-03_{4.800e-03}$[4]	$2.785e-03_{3.607e-03}$[4]
	TDA-F	$\mathbf{7.517e-04}_{\mathbf{4.872e-05}}$[1]	$\mathbf{6.961e-04}_{\mathbf{3.897e-05}}$[1]	$\mathbf{6.619e-04}_{\mathbf{3.406e-05}}$[1]
	DE	$1.259e-03_{3.918e-04}$[3]	$1.178e-03_{4.929e-07}$[3]	$1.177e-03_{1.519e-07}$[3]
	M2M	$1.182e-03_{8.326e-06}$[2]	$1.160e-03_{5.941e-06}$[2]	$1.146e-03_{6.267e-06}$[2]
	AWA	$4.146e-01_{1.495e-01}$[5]	$3.395e-02_{2.862e-02}$[5]	$1.612e-02_{2.325e-02}$[5]
GLT2	TDA-X	$1.730e-01_{1.445e-01}$[3]	$5.778e-02_{4.546e-02}$[2]	$4.960e-02_{3.610e-02}$[3]
	TDA-F	$\mathbf{5.411e-02}_{\mathbf{6.426e-02}}$[1]	$\mathbf{1.187e-02}_{\mathbf{3.297e-03}}$[1]	$\mathbf{1.024e-02}_{\mathbf{1.009e-03}}$[1]
	DE	$1.506e-01_{1.592e-02}$[2]	$1.524e-01_{5.333e-03}$[3]	$1.527e-01_{3.876e-03}$[4]
	M2M	$1.962e-01_{5.300e-02}$[4]	$1.654e-01_{6.687e-04}$[4]	$1.656e-01_{2.359e-04}$[5]
	AWA	$2.040e+00_{1.076e+00}$[5]	$2.386e-01_{2.329e-01}$[5]	$1.569e-02_{1.060e-03}$[2]
GLT3	TDA-X	$2.482e-02_{7.303e-03}$[4]	$1.417e-02_{9.307e-03}$[4]	$9.753e-03_{8.699e-03}$[4]
	TDA-F	$1.943e-02_{9.349e-03}$[3]	$6.295e-03_{6.039e-03}$[2]	$\mathbf{3.197e-03}_{\mathbf{3.577e-03}}$[1]
	DE	$1.607e-02_{1.004e-02}$[2]	$8.553e-03_{5.942e-03}$[3]	$8.115e-03_{5.284e-03}$[3]
	M2M	$\mathbf{6.466e-03}_{\mathbf{4.056e-04}}$[1]	$\mathbf{5.982e-03}_{\mathbf{2.015e-04}}$[1]	$5.881e-03_{9.950e-05}$[2]
	AWA	$2.151e-01_{4.782e-02}$[5]	$1.094e-01_{4.817e-02}$[5]	$6.073e-02_{2.317e-02}$[5]
GLT4	TDA-X	$3.734e-02_{8.125e-02}$[4]	$2.913e-02_{8.167e-02}$[4]	$2.225e-02_{6.917e-02}$[4]
	TDA-F	$\mathbf{2.487e-03}_{\mathbf{2.644e-03}}$[1]	$\mathbf{1.891e-03}_{\mathbf{4.997e-05}}$[1]	$\mathbf{1.874e-03}_{\mathbf{3.510e-05}}$[1]
	DE	$1.218e-02_{4.398e-02}$[3]	$5.185e-03_{1.110e-04}$[2]	$5.167e-03_{1.129e-04}$[3]
	M2M	$5.550e-03_{4.575e-04}$[2]	$5.217e-03_{2.316e-04}$[3]	$5.155e-03_{1.298e-05}$[2]
	AWA	$4.843e-01_{1.649e-01}$[5]	$6.700e-02_{5.839e-02}$[5]	$2.883e-02_{5.044e-02}$[5]
GLT5	TDA-X	$3.749e-02_{9.307e-03}$[2]	$2.302e-02_{3.629e-03}$[3]	$2.177e-02_{1.744e-03}$[3]
	TDA-F	$4.794e-02_{9.241e-03}$[3]	$2.279e-02_{4.658e-03}$[2]	$\mathbf{2.086e-02}_{\mathbf{9.415e-04}}$[1]
	DE	$\mathbf{2.589e-02}_{\mathbf{1.601e-03}}$[1]	$\mathbf{2.187e-02}_{\mathbf{1.618e-03}}$[1]	$2.098e-02_{1.171e-03}$[2]
	M2M	$5.220e-02_{6.313e-03}$[4]	$5.402e-02_{6.722e-03}$[4]	$5.471e-02_{6.731e-03}$[4]
	AWA	$2.404e-01_{2.345e-02}$[5]	$1.917e-01_{2.291e-03}$[5]	$1.860e-01_{1.156e-03}$[5]
GLT6	TDA-X	$1.632e-01_{8.498e-03}$[3]	$1.617e-01_{7.846e-03}$[3]	$1.616e-01_{7.832e-03}$[3]
	TDA-F	$1.631e-01_{1.011e-03}$[2]	$1.618e-01_{4.125e-04}$[4]	$1.616e-01_{3.808e-04}$[4]
	DE	$1.636e-01_{4.821e-02}$[4]	$1.496e-01_{5.359e-02}$[2]	$1.436e-01_{5.559e-02}$[2]
	M2M	$\mathbf{3.800e-02}_{\mathbf{5.870e-03}}$[1]	$\mathbf{3.813e-02}_{\mathbf{6.428e-03}}$[1]	$\mathbf{4.077e-02}_{\mathbf{6.389e-03}}$[1]
	AWA	$3.375e-01_{4.023e-02}$[5]	$2.354e-01_{1.011e-02}$[5]	$2.301e-01_{9.980e-03}$[5]
Mean Rank	TDA-X	3.3	3.3	3.5
	TDA-F	1.8	1.8	1.5
	DE	2.5	2.3	2.8
	M2M	2.3	2.5	2.7
	AWA	5.0	5.0	4.5

settings are as follows. For MOEA/D-TDA and MOEA/D-DE the experimental settings are the same as it is in Section 3. And for MOEA/D-AWA and MOEA/D-M2M, the experimental settings are the same as it is in the original paper.

Table 1 presents the mean and variance of IGD values obtained over 50 runs on LZ and SLZ test suites. On the LZ test suite, DE works the best and TDA-X achieves the best performance on LZ2, LZ3, and LZ4. In the SLZ test suite, TDA-X performs the best on all problems. The rank values obtained by the algorithms also indicate similar results. Comparing to LZ, the SLZ problems have more complex PF. This might be the reason that maintaining a well distributed population in the decision space is helpful for approximating the PF in the objective space especially when problems are with complicated PFs. Table 2 presents the mean and variance of IGD values obtained by the algorithms with different percentages of function evaluations. TDA-F achieves the best performance on all problems except on GLT6. Besides, TDA-F always gains the best rank value in every stage. The results indicate that TDA-F has better performance in the problems complicated in objective space than other state-of-art evolutionary algorithms.

6 Conclusions

This paper proposed a new adaptive strategy, called *domain decomposition and adaptation (TDA)*, to tune the weight vectors online in MOEA/D so as to find good approximations to both PF and PS. The empirical studies indicated that the search behavior measurement in the decision space is helpful and necessary to maintain a good approximation to the PS. Since it is hard to approximate both PS and PF well with a single population, an external archive is added to maintain a good approximation to the PF. Therefore, the proposed algorithm, called MOEA/D-TDA, has two populations: an internal population, which is with the subproblems that are adjusted by TDA, to approximate PS, and an external population to approximate PF. Comparing to the basic algorithm MOEA/D-DE, MOEA/D-TDA does not introduce additional control parameters.

The experimental study has demonstrated: (a) MOEA/D with a single population is hard to approximate both PS and PF, and a good approximation to the PS is necessary to find a good approximation to the PF; (b) TDA with a search behavior measurement in the decision space is helpful to find a good approximation to the PS; and (c) an external archive is helpful to find a good approximation to PF. A further systematic comparison study on several test suites has indicated the advantages of MOEA/D-TDA over some state-of-the-art MOEAs.

The success of TDA depends on two key issues: one is the domain decomposition strategy, and the other is the search behavior measurement. For the former, we use a simplex to represent the domain, and for the latter, we give an initial study based on L2 norm in the three domains. It is no doubt that there might be better ways to do so, and this is the target for future work. Besides, in the proposed approach, the fineness of the weight vector distribution has a fix pattern and the number of subdomains increases rapidly when the number of objective increasing. These are the issues to be improved in the future.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant No. 61731009, 61673180, and 61703382.

References

1. K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, 2001.
2. G. B. L. Cartos Coelle Coello, Daivid A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, 2002.
3. K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective evolutionary algorithms and applications.* Springer Science & Business Media, 2006.
4. E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” *Evolutionary Methods for Design Optimisation and Control*, pp. 95–100, 2001.

5. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
6. S. Rostami and F. Neri, "Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm," *Integrated Computer-Aided Engineering*, vol. 23, no. 4, pp. 313–329, 2016.
7. E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII.* Springer, 2004, pp. 832–842.
8. N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
9. H.-L. Liu, F. Gu, and Y. Cheung, "T-MOEA/D: MOEA/D with objective transform in multi-objective problems," in *2010 International Conference of Information Science and Management Engineering*, vol. 2. IEEE, 2010, pp. 282–285.
10. Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
11. A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–23, 2016.
12. H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.
13. Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
14. S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *2011 Seventh International Conference on Natural Computation (ICNC)*, vol. 3. IEEE, 2011, pp. 1260–1264.
15. H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.
16. H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach." *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
17. H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
18. A. Zhou and Q. Zhang, "Are all the subproblems equally important? resource allocation in decomposition based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016.
19. A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe, "A model-based evolutionary algorithm for bi-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC)*, vol. 3, 2005, pp. 2568–2575.
20. Y. Li, A. Zhou, and G. Zhang, "An MOEA/D with multiple differential evolution mutation operators," in *IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 397–404.
21. H. Zhang, A. Zhou, S. Song, Q. Zhang, X.-Z. Gao, and J. Zhang, "A self-organizing multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 792–806, 2016.