



## MULTI-OBJECTIVE OPTIMIZATION DECOMPOSITION ALGORITHM BASED ON DIFFERENTIATED SELECTION STRATEGY

Yan Xiaofan

College of Computer Science and Technology, Tianjin Polytechnic University, Tianjin, 300387, China

### ABSTRACT

Decomposition-based multi-objective evolutionary algorithm can obtain effective solution sets for solving multi-objective optimization problems. However, the random way of selecting individuals is not conducive to the preservation of good solutions and reduces the convergence speed of the algorithm. In order to solve this problem, a multi-objective optimization decomposition algorithm based on differentiated selection strategy (MOEA/D-DFS) is proposed. The differentiated selection strategy adopted by the algorithm is to select the non-dominant solution from population, using the method of individual selection in NSGA-II. Meanwhile, the strategy of first substitution and stopping is adopted when the offspring replace parent. It is conducive to enhancing the diversity of the population. The experimental results of ZDT test function shows that the algorithm is superior to other algorithm in convergence and diversity, and has certain advantages in solving performance.

**KEYWORDS :** decomposition-based; multi-objective; genetic operator; selection strategy.

### 1 INTRODUCTION

For multi-objective optimization problems (MOPs), it is necessary to optimize many conflicting problems concurrently so as to achieve the best overall effect. Owing to the contradiction of objectives, there is usually to single optimal solution but a set of solutions, called the Pareto set. In the real world, it is not uncommon to face multiple objectives simultaneously, such as water resource engineering [1], industrial scheduling problem [2], control system design [3] and so on [4]. Evolutionary algorithms (EAs) are considered to be very suitable for MOPs because of their population-based characteristics and the approximation of Pareto sets can be obtained in a single run. In the past decades, many multi-objective evolutionary algorithms (MOEAs) have been proposed. Multi-objective evolutionary optimization algorithms can be roughly divided into three categories.

Many multi-objective optimization algorithms based on Pareto dominance, e.g., strength Pareto evolutionary algorithm [5, 6] and nondominated sorting genetic algorithm [7, 8]. These algorithms use the dominant relation of solutions as the first selection condition to ensure the selection of convergent Pareto solution sets. Then, the second selection strategy is adopted to ensure the diversity of solution sets. Unlike dominance-based MOEAs, indicator-based evolutionary algorithms (IBEAs) [9] is also popular, e.g., hypervolume indicator [10], R2 indicator [11]. Indicator parameters are used to solve convergence and distributed performance problems of solution sets. Different from the first two kinds of multi-objective evolutionary algorithms, decomposition-based multi-objective evolutionary algorithms are proposed. A typical algorithm is MOEA/D algorithm [12], in addition, there are a great many of variants algorithms, e.g., [13, 14]. The MOEA/D algorithm converts the multi-objective optimization problem into a scalar subproblem and obtains the value of the decomposed single objective function. And this algorithm has been demonstrated efficient in solving MOPs. Generally speaking, these algorithms have two common but often contradictory goals: minimizing the distance of solutions to the optimal frontier and maximizing the distribution of solutions at the optimal frontier.

Although MOEA/D algorithm can get good Pareto frontier. However, when the number of non-dominated solutions is more than the number of selected solutions, more than the number of solutions will be deleted randomly. This is not conducive to preserving the best individuals, and will reduce the convergence of the solution. In order to solve the above problems, we adopt different deletion strategies to increase the convergence and diversity of the final approximate Pareto

frontier. At the same time, for the operation that MOEA/D offspring may replace more than one parent, this paper adopts the strategy of first substitution and stop [14].

The rest of this paper is organized as follows. Section 2 briefly reviews the related basic concepts and the MOEA/D algorithm. In section 3, the strategy of differentiated deletion and the improved algorithm of MOEA/D-DFS are introduced concisely. Section 4 shows the results and analysis of the proposed algorithm. Section 5 concludes this paper with a summary.

### 2 Preliminaries and Background

#### 2.1. Basic Concepts

Generality, an MOP maybe stated as minimization problem and defined as follows:

(1) where  $x$  denotes a solution vector in the feasible solution space  $\Omega$ , and  $f_i (i = 1, 2, \dots, M)$  is the  $i$ th objective to be minimized.

Then, basic concepts used in multi-objective optimization problem will be briefly introduce.

**Definition 1:** iff  $i \{1, 2, \dots, M\}$ ,  $f_i(u) \leq f_i(v)$ , and  $j \{1, 2, \dots, M\}$ ,  $f_j(u) < f_j(v)$ . In this case, vector  $u$  is said to dominate another vector  $v$ , denoted by  $u < v$ .

**Definition 2:** iff  $\forall x \in \Omega, \exists x^* \in \Omega$  is the Pareto optimal.

**Definition 3:** the Pareto optimal set, PS, is the non-dominated solution set.

**Definition 4:** the Pareto frontier, PF, is the objective function of the non-dominated solutions.

#### 2.2. Introduce of MOEA/D

In the paper of MOEA/D algorithm, three based on decomposition approaches are introduced. It includes Chebyshev decomposition approach, and has a corresponding weight vector for any Pareto optimal solution. Therefore, different Pareto optimal solutions can be obtained by altering the weight vector.

The process of MOEA/D algorithm is as follows:

In the initialization part, the population  $P$  with the scale of  $N$  is generated randomly, and the weight vector  $\lambda_1, \dots, \lambda_n$  are uniformly distributed. Meanwhile, the nearest  $T$  individuals of each individual can be obtained by Euclidean distance, and

the nearest T neighborhood individuals of each individual can be obtained. Then, the objective function value of each individual and the ideal point  $z$  of the population are obtained. The non-dominant solution in the population is called elite population  $EP$ .

In the update part, the paternal population generates the offspring population  $P$  by genetic operators. Two paternal individuals are randomly selected from their neighborhoods. Then, using simulated binary crossover (SBX) and polynomial mutation operation (PM) to generate offspring. The offspring individual are compared with the individuals in the neighborhood of the paternal generation. When the offspring individuals are superior to the paternal individuals, the offspring individuals replace the paternal. The ideal point is updated and the non-dominant solution set  $EP$  in the updated population is selected. Combining  $EP$  and  $EP$ , the final non-dominant solution set is selected.

The update part is iterated until the stop condition is satisfied, and then the final  $EP$  is obtained.

Since in the update section, one offspring may replace more than one parent, which will reduce the diversity of the population. In this paper, we adopt the replacement operation of first substitution and stop.

### 3 Proposed Algorithm

In this section, the proposed-- deposition-based multi-objective evolutionary algorithm with differential selection strategy (MOEA/D-DFS) will be introduced in detail.

#### Algorithm 1: MOEA/D-DFS

**Input:**  $N$ : Population Size;  $T$ : the number of the weight vectors in the neighborhood of each weight vector.

**Output:**  $EP$ : Elite Population.

1. Generate an initial population  $P$  randomly.
2. Generate a uniform spread of  $N$  weight vectors  $\lambda_1, \dots, \lambda_n$ .
3. Work out the  $T$  closet weight vectors to each weight vector. For each  $i = 1, \dots, N$ , set  $B(i) = \{b_1, \dots, b_T\}$ , where  $\lambda_{b_1}, \dots, \lambda_{b_T}$  are the  $T$  closest weight vectors to  $\lambda_i$ .
4. Initialize the ideal point  $z^*$ .
5. Initialize the elite population ( $EP$ )
6. While the termination criterion is not satisfied do
7. For  $i = 1, \dots, N$
8. Randomly select two indexes  $k, l$  from  $B(i)$ , and generate a new solution  $y$ .
9. Compare offspring  $y$  with the  $T$  closet points of its parent using Tchebycheff values.
10. Update the ideal point  $z^*$ .
11. Update of population  $P$ .
12. Get a new elite population  $EP$ .
13. Merge  $EP$  into  $EP$  and adopt differential selection strategy to obtain individuals.
14. End for
15. End while

The pseudo-code of MOEA/D-DFS is given in **Algorithm 1**. Steps 1-5 are the initialized part. The population  $P$  with size  $N$  is initialized in step 1. Step 2 is to generate a uniformly distributed weight vector  $\lambda = \{\lambda_i = (\lambda_{i1}, \dots, \lambda_{im}) \mid i (1, \dots, N)\}$ , where  $\lambda_i = 1$ . In step 3, the  $T$  closet weight vectors  $B(i) = \{b_1, b_2, \dots, b_T\}$  of each weight vector are calculated, using the Euclidean distance. In step 4, the ideal point  $z^* = \{z_1, \dots, z_m\}$  is initialized, and its updated during the search process. In step 5, the elite individuals in the population are preserved in  $EP$ , that is to say, the non-dominant solution in the population is selected.

Steps 6-15 are iterated until the termination criterion is

satisfied. In step 8, simulated binary crossover (SBX) [] and polynomial mutation are applied as the crossover operation to produce offspring. This cross mutation drives two parents generate an offspring. Comparing the generated offspring with the individuals in the  $T$  neighborhood of the parent generation, the replacement operation of the current individual will be stopped when the replaceable parent generation is first encountered in step 9. It is conducive to preserving the diversity of the population. In step 10, the ideal point  $z^*$  is updated. Then the population is updated in step 11. In step 12, The non-dominant solution set of the new population is obtained, which is denoted as  $EP$ .

In step 13,  $EP$  and  $EP$  are combined to adopt differential selection strategy to obtain individuals. The obtained solution is stored in  $EP$ . This strategy adopts the method of NSGA-II [7] to select the solution. The solutions in  $EP$  are sorted non-dominantly ( $F_1, F_2, \dots$ ) and selected layer by layer. The selection process is as follows:

Initial  $Q$  is an empty set, which merges individuals into  $Q$  layer by layer according to non-dominant sorting. When  $F_i$  merges into  $Q$  and the number of individuals in  $Q$  is larger than the set number, select individuals in  $F_i$  layer and put them into  $Q$ , so that the number of individuals in  $Q$  is equals to the set number. The crowding distance [7] is used when selecting individuals in  $F_i$ . In this way, the  $Q$  obtained is the elite population selected from the population, where  $EP = Q$ .

## RESULTS AND ANALYSIS

### 4.1 Unconstrained ZDT Problems

ZDT test function series is used to verify the effectiveness of the proposed MOEA/D-DFS algorithm. Next, ZDT test function series is introduced.

**Table 1. ZDT test function**

Test function	Functional Formula	parameter
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) * h(x)$ $g(x) = 1 + \frac{9}{(n-1)} * \sum_{i=2}^n x_i$ $h(x) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$x = (x_1, \dots, x_n)^T \in [0, 1]^n$ , $n = 30$
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x) * h(x)$ $h(x) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$x = (x_1, \dots, x_n)^T \in [0, 1]^n$ , $n = 30$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) * h(x)$ $h(x) = 1 - \sqrt{\frac{f_1(x)}{g(x)} - \frac{f_1(x) * \sin(10\pi x_1)}{g(x)}}$	$x = (x_1, \dots, x_n)^T \in [0, 1]^n$ , $n = 30$
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x) * h(x)$ $g(x) = 91 + \sum_{i=2}^n x_i^2 - 10 \cos(4\pi \sum_{i=2}^n x_i)$ $h(x) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$x = (x_1, \dots, x_n)^T \in [0, 1]^{n-1} \times [-5, 5]$ , $n = 10$
ZDT6	$f_1(x) = 1 - \exp(-4x_1) (\sin(6\pi x_1))^6$ $f_2(x) = g(x) * h(x)$ $g(x) = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{(n-1)}\right)^{0.25}$ $h(x) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$x = (x_1, \dots, x_n)^T \in [0, 1]^n$ , $n = 10$

### 4.2 Metric and Setting

The inverted generational distance (IGD) is a comprehensive performance evaluation index. It is used to evaluate the

convergence and distribution performance of algorithm. IGD is one of the most widely used indicators [15]. The true Pareto frontier is needed to calculate the IGD value of the obtained solution set. For ZDT test function, the IGD calculation requires normalizing the given set of solutions at ideal and extreme points. For any algorithm, the final nondominated solutions is named  $S$  and the true Pareto frontier is named  $Z$ . Then, the IGD metric is calculated as follows:

$$IGD(S, Z) = \frac{1}{|Z|} \sum_{j=1}^{|Z|} \min_{i=1}^{|S|} d(z_j, s_i)$$

Where  $d(z_i, s_j)$  is the Euclidean distance between the points  $z_i$  and  $s_j$ . The smaller the value of IGD, the better  $S$ .

In order to ensure the reliability of the experiment results, the parameters of the algorithm are set as follows. The number of population individuals  $P$  is set to 100, which means  $N = 100$ . The number of iterations  $It$  is set to 250. The probability of crossover  $CR$  is set to 0.5 and the probability of mutation  $F$  is set to 0.02. The maximum value of  $T$  closet neighborhood is set to 15, and minimum value of  $T$  is set to 2. Generally, the neighborhood size  $T$  is set to  $T = 0.1N$ .

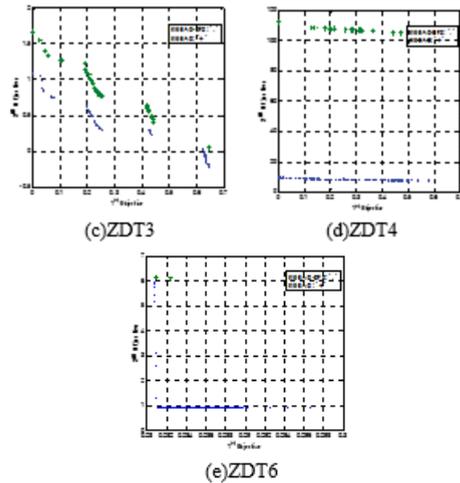
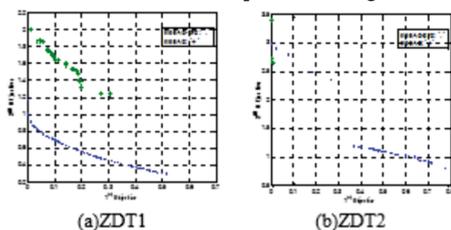
**4.3 Analysis**

Table 2 shows the experimental results of MOEA/D-DFS algorithm and MOEA/D algorithm on test functions ZDT1-4 and ZDT6. Static analysis is used to judge the performance of each algorithm by using IGD value. The smaller the IGD values, the better the performance of the algorithm. Table 2 shows that the performance of the MOEA/D-DFS algorithm is better than that of the MOEA/D algorithm on the whole.

**Table 2. The IGD (Min, Avg, Max) values of the MOEA/D-DFS and MOEA/D algorithms on test functions ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6.**

Test function	Item	MOEA/D-DFS	MOEA/D
ZDT1	Min	0.01001	0.05076
	Avg	0.040336	0.43379
	Max	0.040336	0.95388
ZDT2	Min	0.00227	0.00008
	Avg	0.29614	0.001564
	Max	0.74984	0.00640
ZDT3	Min	0.04068	0.16221
	Avg	0.25665	0.46492
	Max	0.38568	0.97953
ZDT4	Min	0.73323	15.09150
	Avg	10.72729	24.05082
	Max	24.1546	40.12033
ZDT6	Min	0.00952	0.00011
	Avg	1.39815	2.00117
	Max	4.12405	5.34535

Figure 1 shows the approximate Pareto frontier obtained by MOEA/D-DFS algorithm and MOEA/D algorithm on ZDT test function. from the comparison chart of Fig1, the improved algorithm MOEA/D-DFS gives better approximate Pareto frontier. Compared with MOEA/D algorithm, the convergence and distribution of MOEA/D-DFS are excellent. Although the ZDT2 test function does not obtain a good enough set of solutions, compared with the MOEA/D algorithm, a sufficient number of solutions are obtained. Obviously, on the test functions ZDT1, ZDT3, ZDT4 and ZDT6, the solution set obtained by MOEA/D-DFS has better convergence and distribution than that obtained by MOEA/D algorithm.



**Figure 1. the approximate Pareto frontier obtained by MOEA/D-DFS algorithm and MOEA/D algorithm on the ZDT test function.**

**5 CONCLUSION**

In order to achieve a good balance between convergence and diversity, this paper proposes an algorithm, MOEA/D-DFS, which adopts differently selecting strategy for population. This strategy adopts the non-dominated sorting method and gives priority to the selection of good individuals, which is conducive to enhancing the convergence of the final solution set. Meanwhile, it is beneficial to enhance the diversity of the population by using the first encounter and stop substitution operation. To prove the effectiveness of the proposed algorithm, the MOEA/D-DFS algorithm and MOEA/D algorithm are compared with ZDT series of test functions. Experiments show that the proposed algorithm can obtain good convergence and distribution solutions for different optimization problems.

**REFERENCES**

- [1] Trindade B C, Reed P M, Herman J D, et al. Reducing regional drought vulnerabilities and multi-city robustness conflicts using many-objective optimization under deep uncertainty[J]. *Advances in Water Resources*, 2017, 104(9):1117-1146.
- [2] Ringkamp M, Ober-BI?baum S, Dellnitz M, et al. Handling high-dimensional problems with multi-objective continuation methods via successive approximation of the tangent space[J]. *Engineering Optimization*, 2012, 44(9):1117-1146.
- [3] Fleming P J, Purshouse R C, Lygoe R J. *Many-Objective Optimization: An Engineering Design Perspective*[J]. 2005.
- [4] Goulart F, Campelo F. Preference-guided evolutionary algorithms for many-objective optimization[J]. *Information Sciences*, 2016, 329:236-255.
- [5] Kim M, Hiroyasu T, Miki M, et al. SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2.[J]. *Parallel Problem Solving from Nature-PPSN VIII*, 2004, 3242(4):742-751.
- [6] Sheng W, Liu Y, Meng X, et al. An Improved Strength Pareto Evolutionary Algorithm 2 with application to the optimization of distributed generations[J]. *Computers & Mathematics with Applications*, 2012, 64(5).
- [7] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2):0-197.
- [8] Deb K, Jain H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4):577-601.
- [9] Yao X, Burke E K, Lozano, José A, et al. [Lecture Notes in Computer Science] *Parallel Problem Solving from Nature - PPSN VIII Volume 3242 | Indicator-Based Selection in Multiobjective Search*[C]/2004:832-842.
- [10] Bader J, Zitzler E. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization[J]. *EVOLUTIONARY COMPUTATION*, 2011, 19(1):45-76.
- [11] Suzuki J, Phan D H. R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective