2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

An External Archive Guided Multiobjective Evolutionary Approach Based on Decomposition for Continuous Optimization

Yexing Li
School of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Jiangsu, China
Email:lyx19910329@126.com

Xinye Cai
School of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Jiangsu, China
Email:xinye@nuaa.edu.cn

Zhun Fan
School of Engineering
Shantou University
Guangdong, China
Email:mailto:zfan@stu.edu.cn

Qingfu Zhang
Computer Science and Electronic Engineering
University of Essex
Colchester, UK
Email:qzhang@essex.ac.uk

***Abstract*—In this paper, we propose a decomposition based multiobjective evolutionary algorithm that extracts information from an external archive to guide the evolutionary search for continuous optimization problem. The proposed algorithm used a mechanism to identify the promising regions(subproblems) through learning information from the external archive to guide evolutionary search process. In order to demonstrate the performance of the algorithm, we conduct experiments to compare it with other decomposition based approaches. The results validate that our proposed algorithm is very competitive.**

I. INTRODUCTION

Multiobjective optimization problems(MOPs) involve multiple objectives to be satisfied simultaneously. The objectives to be optimized are usually conflicting with each other, thus MOPs do not have a single optimal solution but rather a set of Pareto optimal solutions, which represent the trade-off among different objectives. Along with Pareto dominance based [1]–[3] and performance indicator based algorithms [4], the multiobjective evolutionary algorithms based on decomposition (MOEA/D) [5] have been widely used and investigated in evolutionary computation community. In MOEA/D, a MOP is decomposed into a number of single objective optimization subproblems and then solve them in parallel. The objective function in each subproblem can be an linear or nonlinear weighted aggregation function of all the objective functions in the MOP in question. Two subproblems are called neighbors if their weight vectors are close to each other. Recent studies show that MOEA/D is very competitive compared with other types of multiobjective evolutionary approaches [6].

Over the recent years, many variants of MOEA/D have been proposed to further enhance the performance of MOEA/D [6]–[10]. For example, Li et. al [10] proposed a variant of MOEA/D which uses adaptive weights in the aggregation function to obtain more evenly distributed non-dominated solution. In addition, genetic algorithm is replaced with simulated annealing approach to prevent the search process stuck into local optima. The hybridization of MOEA/D with Pareto dominance based approach has been proposed in [7], [9]. Besides the original decomposition based population, these approaches all adopt a secondary archive that select diversified non-dominated solutions through Pareto dominance and diversity maintenance mechanisms. However, none of the above work use the information in the secondary archive to further improve MOEA/D.

From a more general perspective, multiobjective evolutionary algorithms(MOEAs) is a class of population-based iterative algorithms, which generate abundant data about the search space, problem feature and population information during the optimization process. These information can be learned to further improve the performance of MOEAs. For example, Chia et. al. [11] proposed a Bayesian rule miner, and incorporated it into Pareto dominance based MOEA to identify the promising region in the decision space where the Pareto set is most likely located. Their approach proves to be very effective in handling MOP with noises while maintaining competitive in other types of benchmark problems. Zhang et.al. [6] applied the similar idea into the MOEA/D framework and proposed a variant of MOEA/D based on dynamic resource allocation(MOEA/D-DRA). In their work, the utility value for each subproblem is calculated based on each subproblem's historical convergence information in each generation, then the dynamic resource allocation mechanism is applied to provide different computational effort for different subproblems based on their utility values. MOEA/D-DRA works well on continuous MOPs and won the CEC2009 competition due to its best performance among other 12 algorithms. However, MOEA/D-DRA only considers convergence progress for each subproblem but does not consider the diversity information for all individual subproblems during the different phases of evolutionary optimization. Inspired by above, we propose an adaptive mechanism to identify the promising regions(subproblems) through learning each subproblem's previous convergence and diversity performance in the secondary archive, to further improve MOEA/D. The main contributions of this paper are as follows.

- An external archive is adopted to store diversified non-dominated solutions for MOEA/D. An adaptive learning mechanism is proposed to identify the promising regions(subproblems) through mining their historical performance in the secondary archive. The adaptive learning mechanism is the first approach to learn both the convergence and diversity information during the optimization process in MOEA/D.

- We conduct experiments to compare the proposed approach with other three decomposition based MOEAs on ZDT test instances. Experimental results show the proposed aMOEA/DD outperforms other compared approaches on most test instances. In addition, the effects of the adaptive learning mechanism are also investigated and discussed in the paper.

The rest of this paper is organized as follows. Section II revisits basic concepts of multi-objective optimization(MOP). Section III introduces the MOEA/D and its several variants. The following Section IV mainly describes the proposed adaptive MOEA/D based on external archive(aMOEA/DD). Experimental settings and performance indicators for MOEAs are detailed in Section V. In Section VI, we conduct experiments and present the results to compare our proposed algorithm with three decomposition based MOEAs.

II. Multi-objective Optimization Problem

A. Multi-objective Optimization Problem revisited

A generic *multiobjective optimization problem* (MOP) can be stated as follows:

$$\begin{array}{ll} \text{maximize} & F(x) = (f_1(x), \ldots, f_m(x)) \\ \text{subject to} & x \in \Omega \end{array} \quad (1)$$

where Ω is the *decision space*, $F : \Omega \to R^m$ consists of m real-valued objective functions. The *attainable objective set* is $\{F(x)|x \in \Omega\}$. In the case when Ω is a finite set, (1) is called a discrete MOP.

Let $u, v \in R^m$, u is said to *dominate* v, denoted by $u \succ v$, if and only if $u_i \geq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j > v_j$ for at least one index $j \in \{1, \ldots, m\}$Given a set S in R^m, a point in it is called non-dominated in S if no other point in S can dominate it. A point $x^* \in \Omega$ is *Pareto-optimal* if $F(x^*)$ is non-dominated in the attainable objective set. $F(x^*)$ is then called a *Pareto-optimal (objective) vector*. In other words, any improvement in one objective of a Pareto optimal point must lead to deterioration to at least another objective. The set of all the Pareto-optimal points is called the *Pareto set* (*PS*) and the set of all the Pareto-optimal objective vectors is the *Pareto front* (*PF*) [12]. In many real life applications, the PF is of great interest to decision makers for understanding the tradeoff nature of different objectives and selecting their final solutions.

B. MOEA/D and its variants

This section is dedicated to introducing MOEA/D and its state-of-art variants.

1) MOEA/D: The decomposition based framework(MOEA/D) was proposed in [5], and currently, the decomposition based framework is proved to be an efficient method to solve multi-objective problems(see [5]). This framework decomposes an MOP into several single-objective subproblems, which are defined by a scalar function with different weight vector; and all the subproblems are optimized concurrently. Each subproblem has a single elite solution with respect to its own weight vector. To generate a new solution for each subproblem, parents are selected from its neighboring subproblems. If a better solution is generated by genetic operations, the current solution is replaced with the newly generated one. This solution replacement mechanism is applied to not only the current subproblem but also its neighboring subproblems. That is, a good solution has a chance to survive at multiple subproblems. The diversity of solutions is maintained implicitly by the use of a number of weight vectors guiding different search directions in MOEA/D. High search ability of decomposition based approach on various test functions and real-world problems has been demonstrated in the literature [13]–[19].

2) MOEA/DD: To further improve MOEA/D, an external archive has been adopted; and Pareto dominance and crowding distance method have been applied to save the diversified non-dominated solutions for this archive. For example, cai et.al [9] proposed a variant of MOEA/D with domination archive(MOEA/DD) to tackle multiobjective next release problem. Similarly, Mei et.al. [7] proposed hybrid MOEA/D with NSGAII [3] to tackle the multi-objective capacitated arc routing problem.

3) MOEA/D-DRA: In [6], Zhang et.al proposed a utility based selection mechanism to enhance the efficiency of MOEA/D, which is termed MOEA/D with dynamical resource allocation (MOEA/D-DRA). MOEA/D-DRA calculate each subproblem's previous performance in terms of local convergence(termed utility) and use it to dynamically allocate computational efforts to subproblems. MOEA/D-DRA is the winner of CEC2009 competition in multiobjective continues optimization problems among other 12 algorithms.

III. Motivation of Our work

In this paper, we present an improved MOEA/D, which is termed aMOEA/DD, to tackle multiobjective continuous optimization problems. Our proposed algorithm maintains both a decomposition population and an external archive applied with domination based methods(such as non-dominated sorting and crowding distance methods in NSGAII [3]). Analogous to the population of subproblems in the decomposition based MOEAs, the decomposition population consists the best solutions in different subproblems. The external archive is used to store the diversified non-dominated solutions found by all subproblems during the entire search process. After that, an adaptive learning mechanism is applied to learn the historical convergence and diversity information from the external archive, in order to identify the promising regions(subproblems). Consequently, the identified promising subproblems are inclined to be allocated more computational resources. This is based on our idea that the subproblems that generate more diverse non-dominated solutions survived in the final domination archive are more likely to be the promising regions, which should be given more intensive search.

IV. aMOEA/DD

A. Framework of aMOEA/DD

This paper proposes an improved MOEA/D based On an adaptive mechanism(aMOEA/DD). In aMOEA/DD, a MOP is originally decomposed into a number of subproblems by the weighted sum approach. To be more specific, the objective function of a subproblem can be stated as

$$g^{ws}(x|\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i f_i(x) \tag{2}$$

where $\boldsymbol{F}(x) = (f_1(x), \ldots, f_i(x), \ldots, f_n(x))$ is the objective vector to be minimized in instance and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n)$ is the weight vector that is used to decompose a subproblem of MOCOP. Suppose there are N uniformly distributed weight vectors $\boldsymbol{\lambda^1}, \ldots, \boldsymbol{\lambda^N}$, thus instance is decomposed into corresponding N subproblems. The above eq.(2) assigns the *j*th subproblem its own objective function $g^{ws}(x|\boldsymbol{\lambda}^j)$.

aMOEA/DD maintains a decomposition population $A1$ and an external archive $A2$ to save non-dominated solutions throughout the optimization process. The decomposition population and the archive are of size N. The pseudocode of aMOEA/DD is presented in Algorithm 1. At each generation, N subproblems are selected to pursue the search. The selection of subproblems are based on an adaptive mechanism which will be explained later. For the *i*th selected subproblem, two parents are selected from the T neighboring subproblems of it. Crossover and mutation operators are the same as in [5]. By repeating this procedure for N selected subproblems, an offspring population $Y = \{y_1, \ldots, y_N\}$ is generated. This newly generated offspring population Y is used to update both decomposition and an external archives. For the decomposition population $A1$, the optimal solution for each subproblem is updated based on eq.(2) above. For the external archive $A2$, solutions Y is merged with solutions X' in $A2$, and then the combined population is sorted by the fast non-dominated sorting method and the crowding distance procedure in NSGAII [3]. The best N solutions are kept to replace solutions X' in domination archive $A2$.

During the search process, some subproblems may contribute to multiple non-dominated solutions while others may contribute none to the Pareto approximation. The subproblems that contribute more non-dominated solutions can be considered as more promising regions(subproblems). In order to encourage intensive search for the promising subproblems, aMOEA/DD adopts an adaptive mechanism to select the promising subproblems that are more likely to generate non-dominated solutions. This is implemented based on learning subproblems' historical performance of generating non-dominated solutions in the external archive $A2$.

The selection mechanism based on adaptive learning can be described as follows. In Step2b of Algorithm 1, a certain fixed number of pervious generation, defined as the learning generations(LGs), are considered to calculate the probability of the promising regions for each subproblem based on the its previous performance of generating non-dominated solutions in the domination archive. Consequently, based on this probability, a subproblem will be chosen out of N subproblems to generate new solutions.At each generation $G > LGs-1$, the probability of choosing the ith$(i = 1, 2, \ldots, N)$ subproblem is updated by

$$prob_{i,G} = \frac{D_{i,G}}{\sum_{i=1}^{N} D_{i,G}} \tag{3}$$

Algorithm 1:aMOEA/DD

Input:

1) test instance;
2) a stopping criterion;
3) N: the number of decomposed subproblems(the size of decomposition archive $A1$ and the size of domination archive $A2$;
4) a uniform spread of N weight vectors: $\boldsymbol{\lambda^1}, \ldots, \boldsymbol{\lambda^N}$;
5) the size of the neighborhood of each subproblem, denoted as T;

Output: A set of non-dominated solutions X';

Step1: Initialization:

a) Decompose the original test instance into a set of subproblems $\{A_1, \ldots, A_N\}$ with $\boldsymbol{\lambda^1}, \ldots, \boldsymbol{\lambda^N}$.
b) Initialize a population $X = \{x_1, \ldots, x_N\}$ randomly in $A1$.
c) Set $X' = X$ in $A2$.
d) Compute the Euclidean distance 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 λ^i.

Step2: Search for new solutions

a) Set $j = 1$.
b) Adaptively select a subproblem A_i for search based on eq. 3 and 4.
d) Randomly select two indexes k and l from $B(i)$.
e) Apply one point crossover and bit-wise mutation operators to x_k and x_l to generate y_i for subproblem A_i.
f) Set $j \to j + 1$. If $j \leq N$, go back to Step 2b.

Step3: Update Solutions

/* update optimal solution $x(i)$ of ith subproblem in decomposition archive*/

a) Set $i = 1$.
b) For each $j \in y_j$, if $g^{ws}(y_j|\lambda) \leq g^{ws}(x_i|\lambda)$, then set $x_i = y_j$.
c) Set $i \to i + 1$. If $i \leq N$, go back to Step 3b.
/* update domination archive*/
d) Merge $Y = \{y_1, \ldots, y_N\}$ with X' in $A2$ to obtain $Z = X' \cup Y$; sort the merged population Z with fast non-dominated sorting and crowding distance approach of NSGA-II [3] and the best N solutions replace X' in $A2$.

Step4: Termination

1) If stopping criteria are satisfied, terminate the algorithm. Otherwise, go to Step 2.

---

where

$$D_{i,G} = \frac{\sum_{g=G-LGs}^{G-1} ds_{i,g}}{\sum_{g=G-LGs}^{G-1} total_{i,g}} + \epsilon, (i = 1, 2, \dots, N; G > LGs) \quad (4)$$

$D_{i,G}$ represents the proportion of non-dominated solutions generated by ith subproblem within the previous LGs. $ds_{i,g}$ is the number of non-dominated solutions generated by ith subproblem and $total_{i,g}$ is the total number of non-dominated solutions generated by all subproblems within the previous LGs. The non-dominated solutions mentioned above are the solutions that successfully enter the domination archive $A2$.

A small constant value $\epsilon = 0.001$ is used to avoid the possible zero selection probabilities. We also normalize $D_{i,G}$ in order to make the summation of $D_{i,G}$ for all subproblems into 1.

V. Experiments

A. Parameter Setting

In this experiment, we consider ZDT benchmark problems to test our proposed algorithm. All the compared algorithms are terminated after 50000 function evaluations. The population size is set to 200, and the neighborhood size defined between subproblems is set to 10. For the parameter of LGs in aMOEA/DD, we conduct a sensitivity test. As shown in Fig. 1, aMOEA/DD has the best performance when the value LPs is equal to 10, which is what we've adopted in this paper.

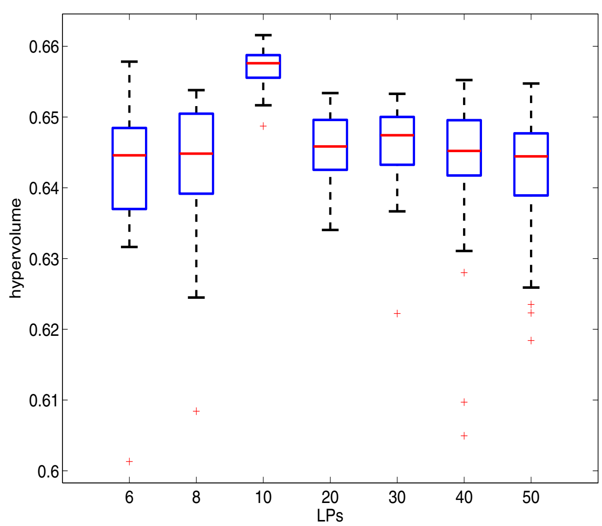


Fig. 1: Sensitivity Test for Parameter LGs

B. Performance Indicators

The performance of a MOEA is usually evaluated in two aspects. First, the obtained non-dominated set should be as close to the true Pareto front as possible. This aspect is usually called convergence. Second, the obtained non-dominated set should be distributed as diversely and uniformly as possible. This aspect is usually called diversity. There has been various metrics designed to reflect either one aspect or both of them to evaluate the performance of a MOEA [20]. In this paper, we adopt two most well-known performance indicators IGD-metric(IGD) [21]and H-metric(I_H) [22], both of which is able to reflect performance in the two respects at the same time.

VI. Experimental Results

A. Comparison with other decomposition based MOEA

We compares aMOEA/DD with MOEA/D [5], MOEA/DD [9] and MOEA/D-DRA [6] on ZDT benchmark functions. All algorithms run 30 times. The mean and standard deviation of IGD-metric after running each compared algorithm is shown in table I. In addition, t-test between the best results and the second best results for each benchmark function is conducted and presented in table I. If the result obtained by aMOEA/DD is not the best, then we conduct t-test between the best obtained result and the result obtained by aMOEA/DD. From table 1, we can see that both mean and standard deviation of IGD-metric obtained by aMOEA/DD for ZDT1, ZDT2 and ZDT4 has the smallest value among all the compared algorithms with statistical significance, which indicates that our proposed aMOEA/DD outperforms other algorithms on these functions. For function ZDT6, MOEA/D is slightly better than aMOEA/DD, although the result is not statistically significant. For ZDT3, MOEA/DD is significantly better than aMOEA/DD and aMOEA/DD outperforms other two algorithms.

The mean and standard deviation of H-metric value are presented in table II, as well as t-test results. We use (1;1) as the reference point. As we can see in table II, the results in terms of H-metric is consistent with that in terms of IGD. For ZDT1, ZDT2 and ZDT4, the mean of H-metric obtained by aMOEA/DD is larger than that of other algorithms, while the standard deviation is smallest for aMOEA/DD. In other word, aMOEA/DD is the most efficient and stable compared with other approaches on these benchmark functions.

Fig.2 present the convergence of each compared algorithm in terms of average IGD-metric value. All the results indicate that aMOEA/DD converges much faster than other algorithms in terms of IGD metric on ZDT1, ZDT2, ZDT3 and ZDT6.

B. aMOEA/DD vs. MOEA/D-DRA

aMOEA/DD outperforms MOEA/D-DRA with statistical significance on all the ZDT benchmark functions except for ZDT6 in terms of IGD; and ZDT3, ZDT6 in terms of H-metric. In addition, we adopt box-plot to compare aMOEA/DD and MOEA/D-DRA in terms of IGD, as show in fig.3. The above observations all confirm that aMOEA/DD is very competitive compared with MOEA/D-DRA.

VII. The Effects of Adaptive Learning Mechanism

In this paper, an adaptive learning mechanism is proposed in aMOEA/DD. The adaptive learning mechanism aims to identify the promising regions(subproblems) through learning the search information based on each subproblem's previous performance, and then use the learned information to guide the selection of subproblems. This section is dedicated to validating the effectiveness and efficiency of this adaptive learning mechanism.

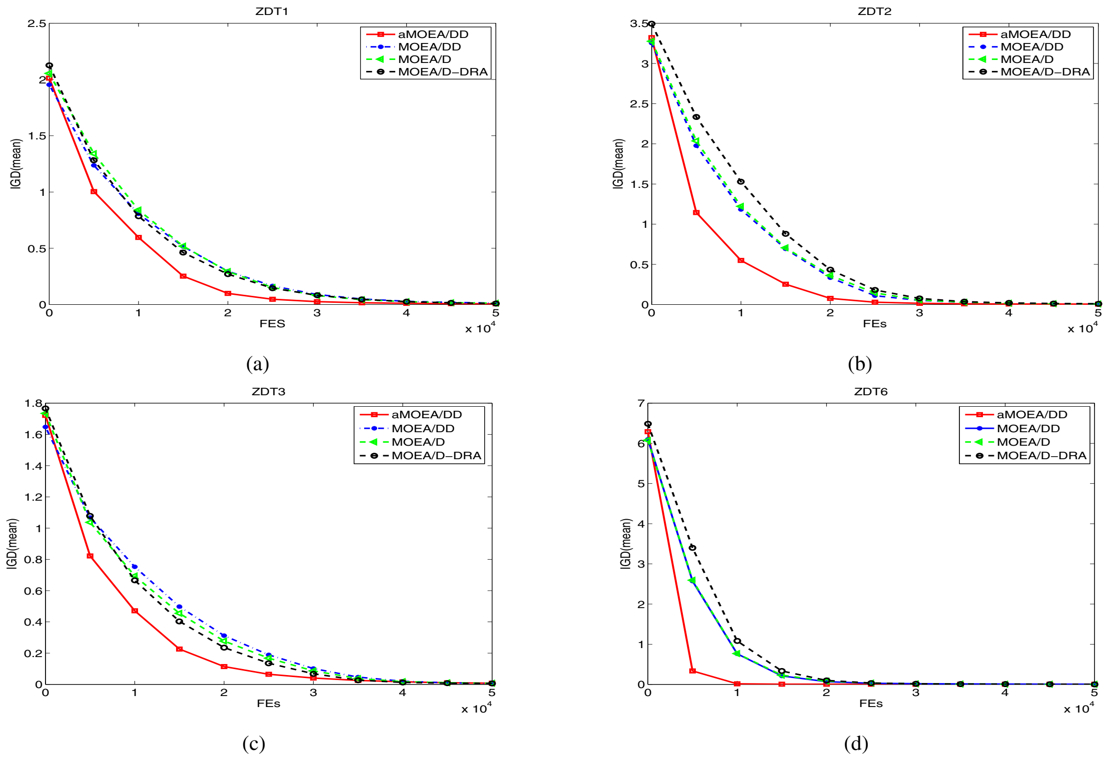


Fig. 2: Convergence graphs in term of mean of IGD obtained by three approaches on (a) ZDT1 (b) ZDT2 (c) ZDT3 (d) ZDT6 .

TABLE I: Mean and standard deviation of IGD-metric values obtained by four algorithms on ZDT instance

instance	*IGD*				t-test	
P=200,n=10	aMOEA/DD	MOEA/DD	MOEA/D	MOEA/D-DRA	h	p
zdt1	**0.0061**(**0.0019**)	0.0127(0.0032)	0.0117(0.0032)	0.0089(0.0023)	1	8.7025e-009
zdt2	**0.0047**(**0.0017**)	0.0077(0.0019)	0.0070(0.0018)	0.0086(0.0021)	1	1.0992e-005
zdt3	0.0034(**2.0412e-4**)	**0.0032**(2.4770e-4)	0.0050(5.8801e-5)	0.0052(2.3420e-4)	1	4.7508e-004
zdt4	**0.0161**(**0.0214**)	0.0680(0.0822)	0.1158(0.1182)	0.1047(0.0728)	1	0.0019
zdt6	0.0034(0.0028)	0.0047(0.0030)	**0.0029**(0.0013)	0.0030(**0.0009**)	0	0.3471

TABLE II: Mean and standard deviation of H-metric values obtained by three algorithms on ZDT instance.

instance	*hypervolume*				t-test	
P=200,n=10	aMOEA/DD	MOEA/DD	MOEA/D	MOEA/D-DRA	h	p
zdt1	**0.6571**(**0.0029**)	0.6470(0.0046)	0.6486(0.0045)	0.6499(0.0037)	1	5.4118e-011
zdt2	**0.3246**(**0.0031**)	0.3186(0.0033)	0.3201(0.0033)	0.3184(0.0031)	1	3.5699e-009
zdt3	0.1915(0.0013)	0.1907(**0.0011**)	**0.1932**(0.0013)	0.1924(0.0013)	1	1.6908e-007
zdt4	**0.5947**(**0.1117**)	0.5232(0.2294)	0.4711(0.1986)	0.4114(0.1906)	1	0.0124
zdt6	0.3176(0.0298)	0.3250(0.0040)	0.3251(0.0040)	**0.3271**(**0.0016**)	0	0.1985

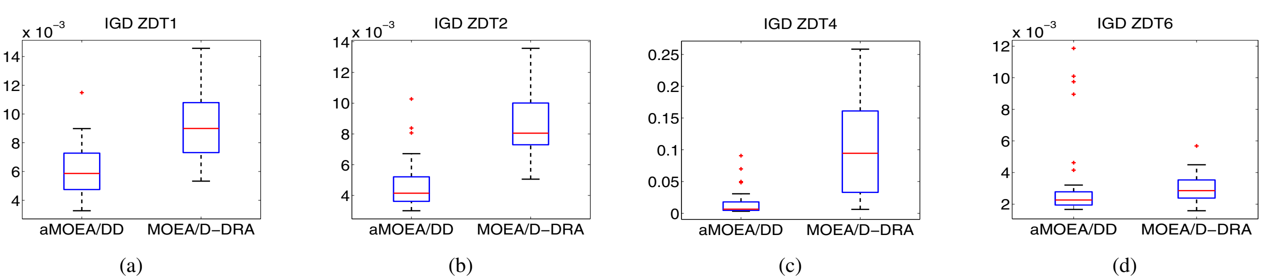
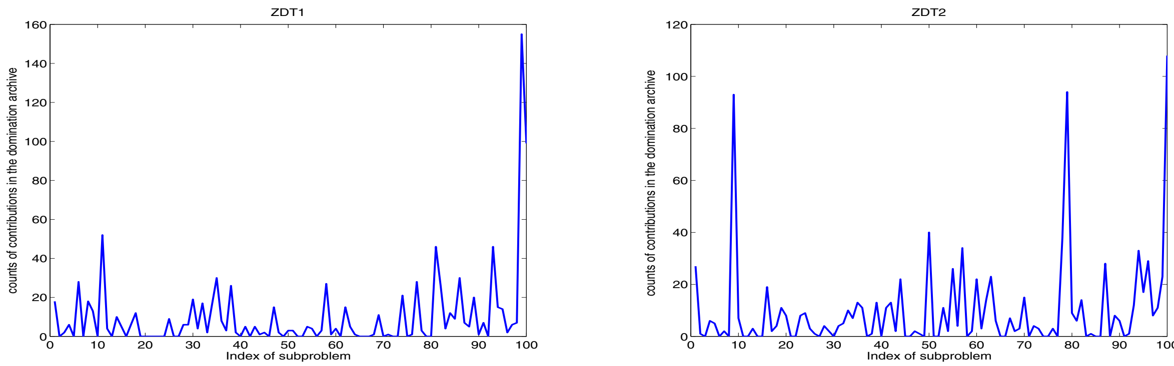


Fig. 3: Comparison of aMOEA/DD and MOEA/D-DRA in terms of IGD metric for a) ZDT1, b) ZDT2, c) ZDT4, d) ZDT6.



(a) (b)

Fig. 4: Contributions of subproblems to the domination archive over 10 generations at the final stage of the optimization process for ZDT instances.(a) ZDT1 (b) ZDT2 .

The motivation of the adaptive learning mechanism is based on our hypothesis that different subproblems give different contributions to non-dominated set stored in the external archive. Therefore, we conduct experiments to demonstrate the distributions of counts of non-dominated solutions various subproblems contribute to the external archive over 10 generations at the final stage of the optimization process for ZDT1 and ZDT2, as shown in Fig.4. Overall, different subproblems have very different contributions to the external archive, which supports our hypothesis that adaptive learning mechanism is based on. It is clear to see that the contributions of non-dominated solutions of ZDT1 in the external archive is more uniform than ZDT2.

VIII. Conclusion

This paper mainly analyzed decomposition based approaches and proposed a variant of MOEA/D based on an adaptive learning mechanism. The proposed approach, aMOEA/DD, is compared with MOEA/D and its variants on ZDT benchmark functions. Experimental results show the proposed aMOEA/DD give the best performance among all the compared approaches.

Further work includes investigation of the proposed approach for discrete optimization problem. It is also interesting to use other techniques to manage the external domination archive, besides non-dominated sorting and crowding distance in NSGA-II. Moreover, we also intend to extend aMOEA/DD to tackle more challenging multi-objective optimization problems, such as many objective problems and MOPs with complex constraints.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant 61300159 and 61175073, by the Natural Science Foundation of Jiangsu Province under grant BK20130808, by the Research Fund for the Doctoral Program of Higher Education of China under grant 20123218120041.

References

[1] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.

[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001.

[3] 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, April 2002.

[4] E. Zitzler and S. Künzli, "Indicator-based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Y. et al., Ed. Birmingham, UK: Springer-Verlag. Lecture Notes in Computer Science Vol. 3242, September 2004, pp. 832–842.

[5] 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, December 2007.

[6] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of moea/d on cec09 unconstrained mop test instances," School of CS and EE, University of Essex, Working Report CES-491, Feburary 2009.

[7] K. T. Yi Mei and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.

[8] H. Li and D. Landa-Silva, "Evolutionary Multi-Objective Simulated Annealing with Adaptive and Competitive Search Direction," in *2008 Congress on Evolutionary Computation (CEC'2008)*. Hong Kong: IEEE Service Center, June 2008, pp. 3310–3317.

[9] X. Cai and O. Wei, "A hybrid of decomposition and domination based evolutionary algorithm for multi-objective software next release problem," in *10th IEEE International Conference on Control and Automation*, 2013.

[10] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, pp. 561–595, 2011.

[11] V. A. S. J. Y. Chia, C. K. Goh and K. C. Tan, "A data mining approach to evolutionary optimization of noisy multi-objective problems," *International Journal of System Science*, vol. 43, no. 7, pp. 1217–1247, 2012.

[12] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1999.

[13] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Lin, "MOEA/D for Flowshop Scheduling Problems," in *2008 Congress on Evolutionary Computation (CEC'2008)*. Hong Kong: IEEE Service Center, June 2008, pp. 1433–1438.

[14] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of Scalarizing Funtions in MOEA/D: An Adaptive Scalarizing

Funtion-Based Multiobjective Evolutionary Algorithm," in *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Nantes, France: Springer. Lecture Notes in Computer Science Vol. 5467, April 2009, pp. 438–452.

[15] 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, April 2009.

[16] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," *Computer Networks*, vol. 54, no. 6, pp. 960–976, 2010.

[17] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem," in *Multi-objective memetic algorithms*, vol. 171. Berlin: Springer, 2009, pp. 309–324.

[18] W. Peng and Q. Zhang, "Network Topology Planning Using MOEA/D with Objective-Guided Operators," in *Parallel Problem Solving from Nature - PPSN XII*, vol. 7492, 2012, pp. 62–71.

[19] A. Kafafy, A. Bounekkar, and S. Bonnevay, "Hybrid metaheuristics based on moea/d for 0/1 multiobjective knapsack problems: A comparative study," in *IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[20] J. Knowles, L. Thiele, and E. Zitzler, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 214, Feb. 2006, revised version.

[21] C. A. Coello Coello and N. Cruz Cortés, "Solving Multiobjective Optimization Problems using an Artificial Immune System," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, June 2005.

[22] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," Computer Engineering and Networks Laboratory, ETH Zurich, Tech. Rep. 139, June 2002.