ICIC Express Letters
Volume **10**, Number **9**, September **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2059–2065**

RESOURCE ALLOCATION BASED ON THE ACCESS FREQUENCY OF EACH SUBPROBLEM IN DECOMPOSITION BASED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Weikang Ning, Baolong Guo, Yunyi Yan and Jinfu Wu

School of Aerospace Science and Technology
Xidian University
No. 2, South Taibai Road, Xi'an 710071, P. R. China
ningweikang@126.com

Received January 2016; accepted April 2016

Abstract. *Multi-objective evolutionary algorithm based on decomposition (MOEA/D) decomposes a multi-objective optimization problem (MOP) into a series of subproblems and solves them simultaneously in a single run. Recent studies have shown that by allocating different computational resources to each subproblem based on their difficulty, the performance of MOEA/D could be significantly enhanced. However, how to define the difficulty of a subproblem is rather hard. In this paper, a new dynamic resource allocation mechanism based on the access frequency of each subproblem is proposed. It assigns more computational resources for those subproblems that are less visited. The basic idea is that less accessed subproblems are not fully explored and thus need more efforts. Experimental results show that the strategy proposed is very competitive.*
Keywords: Multi-objective optimization, Decomposition, Dynamic resource allocation, Access frequency

1. **Introduction.** Without loss of generality, a multi-objective optimization problem (MOP) could be stated as:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ \text{s.t.} \quad & \mathbf{x} \in \Omega = [l_k, u_k]^n \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$ is an n dimensional decision vector lying in the decision space R^n. l_k, u_k are the lower and upper bounds of x_k $(k = 1, \ldots, n)$, respectively. $\mathbf{F} : \Omega \to R^m$ consists of m objective functions $(f_1, \ldots, f_m)$ to be minimized and R^m is called the objective space.

In solving a MOP, the improvement of one objective often leads to the deterioration of another. Thus multi-objective optimization algorithms aim to find the best trade-offs among the objectives. The best trade-offs among different objectives can be defined using the concept of Pareto optimality. In a minimization problem, a domination relationship between two solutions could be defined as follows: let $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x}$ is said to dominate $\mathbf{y}$, denoted by $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{y})$, if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for every $i \in \{1, \ldots, m\}$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, \ldots, m\}$. A solution $\mathbf{x}^*$ is called a Pareto optimal solution if there is no other solution $\mathbf{s} \in \Omega$ such that $\mathbf{s}$ dominates $\mathbf{x}^*$. The set of all the Pareto solutions in Ω is called the Pareto set (PS) and its image in the objective space is called the Pareto front (PF) [1].

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [2] has been proved efficient in solving MOPs. It decomposes a MOP into a series of subproblems and solves them simultaneously in a single run. However, the performance (search balance between exploitation and exploration) of MOEA/D is closely related to several

components such as T_m, T_r, n_r, and the decomposition method. Tuning these components adaptively based on the feedback of evolution thus becomes a prospective research direction. A lot of MOEA/D variants that tune these components adaptively have been proposed recently [3-6]. Among these techniques, dynamic resource allocation mechanism adaptively assigns different computational efforts for each subproblem based on their difficulty. In [7], each subproblem is assigned a utility π and the subproblems with higher utility will have higher probability to get more computational resources. Continuing this work, Zhou and Zhang [8] proposed a probability of improvement (PoI) vector that is maintained at each iteration and used to select the subproblems to invest.

The definition of subproblem difficulty, as noted in [8], is rather hard. In this paper, a new dynamic resource allocation mechanism based on the access frequency of each subproblem is proposed, which tries to put more efforts into the less explored regions. A probability vector is also adopted in the proposed algorithm to assign different subproblems into different computational resources. The probability vector adopted in the proposed algorithm is used in a similar way to Zhou and Zhang's algorithm [8], but the elements of our probability vector are calculated based on the access frequency of each subproblem. Though a hybrid utility function considering the density in the objective space is also suggested in [8], we actually propose a more simple measure from a different view to quantify the difficulty of each subproblem, which contributes to the main novelty of our method.

The rest of the paper is organized as follows. Section 2 details the proposed algorithm. Section 3 presents some experimental results. Section 4 concludes this paper.

2. **Proposed Algorithm.** The proposed algorithm adopts similar framework to MOEA /D [2], and thus a set of evenly spread weight vectors $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ is needed to decompose (1) into N subproblems. In this paper, the Tchebycheff approach that is most widely used in MOEA/D study is adopted (other decomposition methods such as boundary intersection approach [2] could also be adopted) as the decomposition method and the ith subproblem could be defined as:

$$\begin{aligned} \min \quad & g^{te}\left(\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*\right) = \max_{1\le j\le m} \left\{ w_j^i \left| f_j(\mathbf{x}) - z_j^* \right| \right\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{2}$$

where $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$ is the ideal reference point, i.e., $z_j^* = \min\{f_j(\mathbf{x})|\mathbf{x} \in \Omega\}$ for each $j = 1, \ldots, m$. $\mathbf{w}^i = \{w_1^i, \ldots, w_m^i\}$ $(i = 1, \ldots, N)$ is the ith weight vector in W and $\sum_{j=1}^m w_j^i = 1$. All solutions of these N subproblems constitute a good approximation to the PF of (1).

Similar to [8], the proposed algorithm also uses a probability vector $\mathbf{p} = \{p_1, \ldots, p_N\}$ to assign the computational resources for each subproblem. Initially, T (neighbor size) closest (based on the Euclidean distance between the corresponding weight vectors of each subproblem) neighbors of the ith $(i = 1, \ldots, N)$ subproblem are indexed by $B(i) \subset \{1, \ldots, N\}$ where $|B(i)| = T$. Each element of $\mathbf{p}$ is initialized as 1. Each element of *acc* (the number of solutions accessed by each subproblem at each iteration) is initialized as 0. At each iteration, the algorithm maintains:

- A population $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ of size N;
- The investment probability vector $\mathbf{p}$;
- *acc*: the number of solutions accessed by each subproblem at each iteration;
- The ideal reference point $\mathbf{z}^*$ estimated by all encountered solutions so far.

Algorithm 1 presents the general framework of the proposed algorithm. In line 7, the ith subproblem is invested with a probability of p_i, where $rand() \in [0, 1]$ is a random number generator. In line 9, two parents are selected from the mating pool P and a new solution $\mathbf{y}$ is generated using DE and mutation operator, as is done in [7]. In line 11,

the best suitable subproblem with index $bestId$ for the newly generated solution $\mathbf{y}$ is first selected using:

$$bestId = \arg_{i=1,\ldots,N} \min \left\{ g^{te}\left(\mathbf{y}|\mathbf{w}^i, \mathbf{z}^*\right)\right\} \tag{3}$$

as is done in [4]. Then $\mathbf{y}$ is used to update T neighbor solutions of subproblem $bestId$. That is, if $g^{te}\left(\mathbf{y}|\mathbf{w}^k, \mathbf{z}^*\right) < g^{te}\left(\mathbf{x}^k|\mathbf{w}^k, \mathbf{z}^*\right)$ $(k \in B(bestId))$, set $\mathbf{x}^k = \mathbf{y}$. Note that maximum n_r solution is replaced during this process.

In line 14, each element of the probability vector is updated at iteration t as follows:

$$c_i = \sum\nolimits_{h=t-\Delta T}^{t} acc[h][i] \tag{4}$$

$$u_i = 1 - \frac{c_i}{\max_{j=1,\ldots,N} c_j} \tag{5}$$

$$p_i = u_i \tag{6}$$

where $acc[h][i]$ denotes the number of solutions accessed by the ith $(i = 1, \ldots, N)$ subproblem at iteration h. Higher p_i means that the ith subproblem is less accessed and is given more efforts in line 7 of Algorithm 1.

We denote the proposed algorithm above as MOEA/D-AF (MOEA/D based on the access frequency of each subproblem). Besides, the proposed strategy is combined with the strategy adopted in [8], which allocates the resources based on the relative improvement of each subproblem. We denote the hybrid algorithm as MOEA/D-AF+RI (MOEA/D based on the access frequency and relative improvement of each subproblem). Each element of the investment probability vector $\mathbf{p}^{AF+RI}$ adopted in MOEA/D-AF+RI is calculated as follows:

$$u_i^{AF+RI} = u_i + u_i^{RI} \tag{7}$$

$$p_i^{AF+RI} = \frac{u_i^{AF+RI} + \epsilon}{\max_{j=1,\ldots,N} u_j^{AF+RI} + \epsilon} \tag{8}$$

where u_i^{RI} is the utility function of the ith subproblem based on Equation (5) in [8] and $\epsilon = 1.0 \times 10^{-50}$ is a small value.

Algorithm 1: General framework of the proposed algorithm

Input: a stopping criteria; W; T; δ; ΔT;

1 Initialize $B(i)$ for each subproblem;
2 Sample N solutions randomly from the search space to form the initial population;
3 Initialize $\mathbf{z}^*$;
4 $\mathbf{p} = (1, \ldots, 1)$; Each element of acc is initialized as 0; $t = 0$;
5 **while** *the stopping criteria are not satisfied* **do**
6 **foreach** $i \in \{1, \ldots, N\}$ **do**
7 **if** $rand() < p_i$ **then**
8 $P = \begin{cases} B(i), & \text{if } rand() \leq \delta \\ \{1, \ldots, N\}, & \text{otherwise} \end{cases}$
9 Randomly select two parent indexes from P and generate a new solution $\mathbf{y}$ using DE and mutation operator;
10 Update $\mathbf{z}^*$: if $f_j(\mathbf{y}) < z_j^*$, then set $z_j^* = f_j(\mathbf{y})$ $(j = 1, \ldots, m)$;
11 Select the most suitable subproblem with index $bestId$ for $\mathbf{y}$ and update the solutions that are the T closest neighbors of the $bestId$th subproblem, $acc[t][bestId] + +$;
12 t++;
13 **if** $t\%\Delta T == 0$ **then**
14 Update $\mathbf{p}$;

3. **Experimental Results.** The performance of MOEA/D-AF is tested on ten MOPs of CEC'09 test suite [9] and MOEA/D-AF is compared with MOEA/D-DRA (MOEA/D with dynamic resource allocation) [7] and MOEA/D-AF+RI [1]. Twenty independent runs are performed on each test instance for each algorithm. Maximum 300,000 function evaluations are used for each test instance. For all the three algorithms, the population size N is 600 for UF1-UF7 (two objectives) and 1000 for UF8-UF10 (three objectives); $T = 0.1N$ and $n_r = 0.01N$; $\delta = 0.9$. The DE operator with $CR = 1.0$, $F = 0.5$ and the mutation operator with $\eta = 20$, $p_m = 1/n$ are adopted. The parameters settings above are the same with those adopted in [7]. As a matter of fact, these parameters are set to some typical values which are adopted in most MOEA/D studies. For fair comparison, the utility of each subproblem is updated every 30 iterations for MOEA/D-DRA. For MOEA/D-AF and MOEA/D-AF+RI, $\Delta T = 30$.

Two indicators (hypervolume I_{hv} [11] and inverted generational distance I_{IGD} [12]) that quantify both the convergence and spread of the Pareto front approximation are adopted.

Let $\mathbf{r}^*$ be a reference point which denotes an upper bound over all the objectives. And it is defined as the biggest objective value of the real Pareto front. Let A be the Pareto front approximation, and I_{hv} could be calculated as:

$$I_{hv}(A) = \Lambda\left(\bigcup_{\mathbf{a}\in A}\{\mathbf{x}|\mathbf{a} \prec \mathbf{x} \prec \mathbf{r}^*\}\right) \tag{9}$$

where Λ is the Lebesgue measure. Higher I_{hv} indicates better convergence and spread performance of the Pareto front approximation.

Let A be the Pareto front approximation, R be the real Pareto front. I_{IGD} is calculated as follows:

$$I_{IGD}(R, A) = \frac{\sum_{\mathbf{r}\in R} d(\mathbf{r}, A)}{|R|} \tag{10}$$

where $d(\mathbf{r}, A)$ is the minimum Euclidean distance between $\mathbf{r}$ and the points in A. Lower I_{IGD} indicates better convergence and spread performance of the Pareto front approximation.

The box plots of I_{hv} and I_{IGD} based on twenty runs obtained by the three algorithms in solving UF1-UF10 are presented in Figure 1 and Figure 2, respectively. It could be observed from these two figures that MOEA/D-AF and MOEA/D-AF+RI have similar performances and they generally perform better than MOEA/D-DRA in UF1, UF3, UF4 and UF6. MOEA/D-DRA performs significantly better than MOEA/D-AF and MOEA/D-AF+RI only in UF8 and MOEA/D-DRA has better performance in UF2 with respect to the medias of I_{hv} and I_{IGD}. For UF5 and UF9, MOEA/D-AF+RI performs the best. For UF7, MOEA/D-AF performs the best with respect to I_{hv} and MOEA/D-DRA performs the best with respect to I_{IGD}. For UF10, MOEA/D-AF performs the best.

4. **Conclusions.** A new dynamic resource allocation strategy in MOEA/D that is based on the access frequency of each subproblem is proposed in this paper. More computational resources are allocated for those subproblems that are less visited. Besides, the strategy proposed is combined with a recently proposed dynamic resource allocation strategy that is based on the relative improvement of each subproblem. Experimental results suggest that the proposed resource allocation strategy is very competitive. When combined with the strategy that is based on the relative improvement of each subproblem, the strategy proposed in this paper could perform even better in some test instances. In the future, other measures that could be used to quantify the difficulty of a subproblem are also preferred and the adaptability of each measure needs further investigation.

[1]The jMetal framework [10] is used to implement MOEA/D-DRA. MOEA/D-AF and MOEA/D-AF+RI are implemented by the authors of this paper.

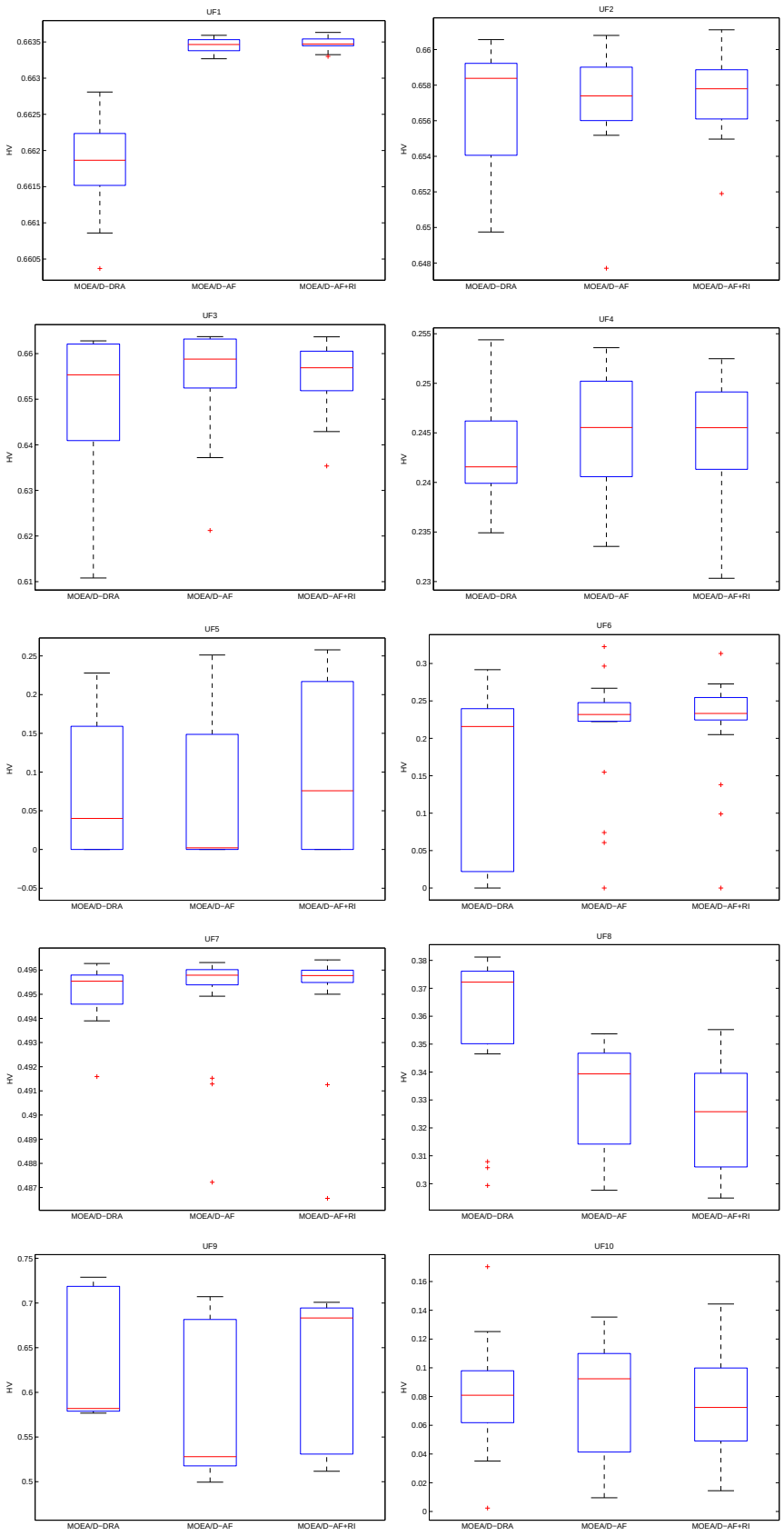

FIGURE 1. Box plots of I_{hv} based on twenty independent runs obtained by MOEA/D-DRA, MOEA/D-AF, and MOEA/D-AF+RI in solving UF1-UF10

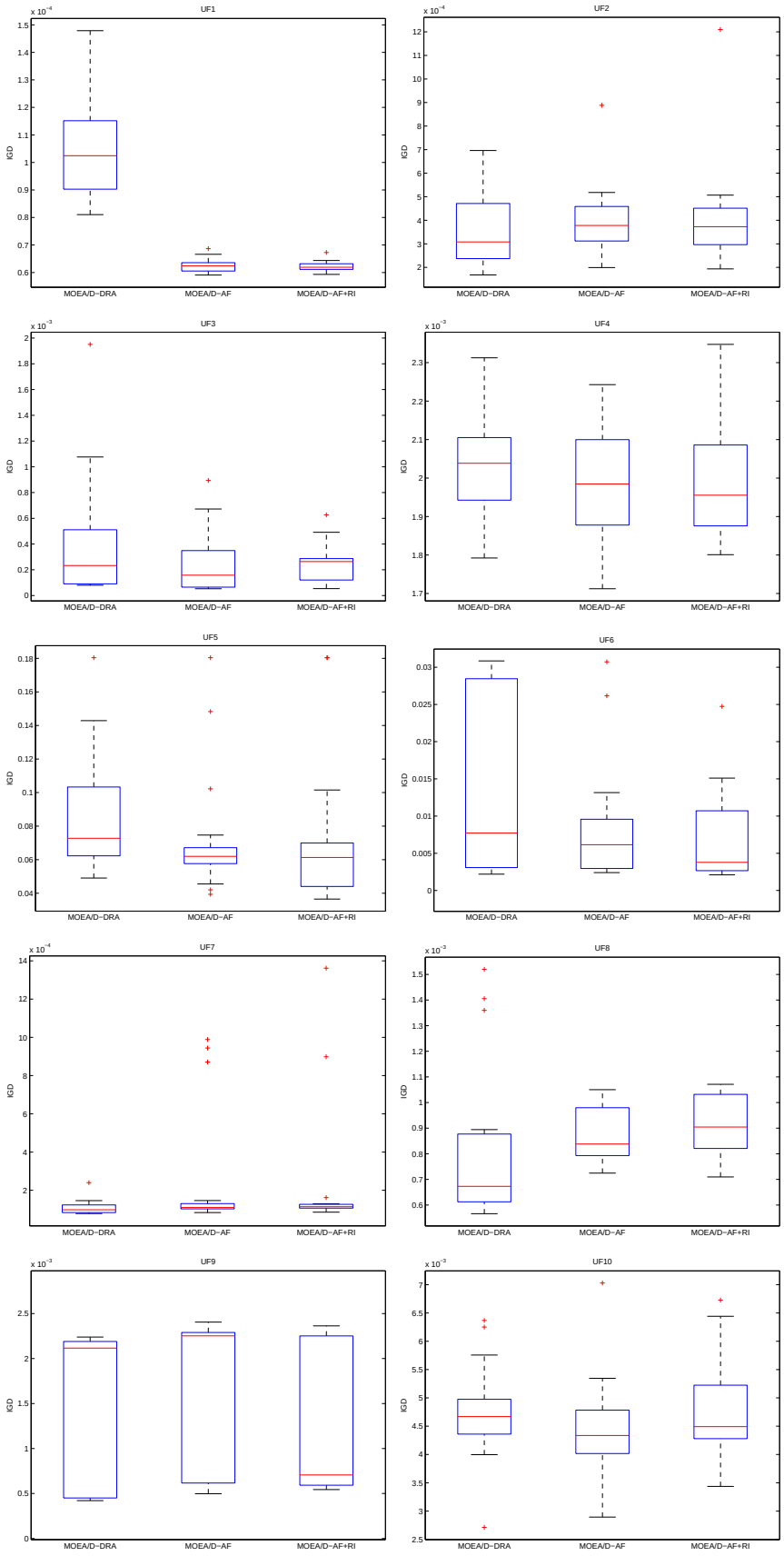

FIGURE 2. Box plots of I_{IGD} based on twenty independent runs obtained by MOEA/D-DRA, MOEA/D-AF, and MOEA/D-AF+RI in solving UF1-UF10

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 61571346, 61305041, 61105066 and 61305040. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*, 1999.

[2] Q. Zhang and H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition, *IEEE Trans. Evolutionary Computation*, vol.11, no.6, pp.712-731, 2007.

[3] S. Zhao, P. N. Suganthan and Q. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, *IEEE Trans. Evolutionary Computation*, vol.16, no.3, pp.442-446, 2012.

[4] Z. Wang, Q. Zhang, A. Zhou et al., Adaptive replacement strategies for MOEA/D, *IEEE Trans. Cybernetics*, vol.46, no.2, pp.474-486, 2015.

[5] H. Ishibuchi, Y. Sakane, N. Tsukamoto et al., Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm, *Evolutionary Multi-Criterion Optimization*, pp.438-452, 2009.

[6] W. K. Mashwani and A. Salhi, A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation, *Applied Soft Computing*, vol.12, no.9, pp.2765-2780, 2012.

[7] Q. Zhang, W. Liu and H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, *IEEE Congress on Evolutionary Computation*, pp.203-208, 2009.

[8] A. Zhou and Q. Zhang, Are all the subproblems equally important? Resource allocation in decomposition based multiobjective evolutionary algorithms, *IEEE Trans. Evolutionary Computation*, vol.20, no.1, pp.52-64, 2015.

[9] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu and S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, *Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report*, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, 2008.

[10] J. J. Durillo and A. J. Nebro, jMetal: A java framework for multi-objective optimization, *Advances in Engineering Software*, vol.42, no.10, pp.760-771, 2011.

[11] E. Zitzler and L. Thiele, Multiobjective optimization using evolutionary algorithms – A comparative case study, *Parallel Problem Solving from Nature – PPSN V*, pp.292-301, 1998.

[12] D. A. V. Veldhuizen and G. B. Lamont, *Multiobjective Evolutionary Algorithm Research: A History and Analysis*, 1998.