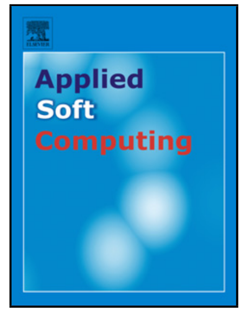
Accepted Manuscript



Title: An Improved Decomposition-Based Multiobjective Evolutionary Algorithm with a better Balance of Convergence and Diversity

Author: Senliang Ying Li Li Zheng Wang Weikun Li Wanliang Wang

PII: S1568-4946(17)30165-5
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.03.041
Reference: ASOC 4126

To appear in: *Applied Soft Computing*

Received date: 3-1-2017
Revised date: 1-3-2017
Accepted date: 27-3-2017

Please cite this article as: Senliang Ying, Li Li, Zheng Wang, Weikun Li, Wanliang Wang, An Improved Decomposition-Based Multiobjective Evolutionary Algorithm with a better Balance of Convergence and Diversity, *<![CDATA[Applied Soft Computing Journal]]>* (2017), http://dx.doi.org/10.1016/j.asoc.2017.03.041

An Improved Decomposition-Based Multiobjective Evolutionary Algorithm with a better Balance of Convergence and Diversity

Senliang Ying, Li Li, Zheng Wang, Weikun Li, Wanliang Wang*

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310023, P.R.China

Abstract

In Decomposition-Based multiobjective evolutionary algorithms (MOEAs), a good balance between convergence and diversity is very important to the performance of an algorithm. However, only the aggregation functions enough to achieve a good balance, especially in high-dimensional objective space. So we considered using the value of related acute angle between a solution and a direction vector as an other consider index. This idea is implemented to enhance the famous decomposition-based algorithm, i.e., MOEA/D. The enhanced algorithm is compared to its predecessor and other state-of-the-art algorithms on a several well-known test suites. Our experimental results show that the proposed algorithm performs better than its predecessor in keeping a better balance between the convergence and diversity, and also as effective as other state-of-the-art algorithms.

Keywords:
Decomposition, diversity, convergence, related angle value, evolutionary multi-objective optimization

*Corresponding author.
Email address: `zjutwwl@zjut.edu.cn` (Wanliang Wang)

Preprint submitted to Applied Soft Computing *March 1, 2017*

1. Introduction

A multi-objective optimization problem(MOP) can be defined as follows:

$$\begin{cases} min\boldsymbol{F}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x}))^T \\ subject \quad to \quad \mathbf{x} \in \boldsymbol{\Omega} \end{cases} \tag{1}$$

where $\mathbf{x} \in \boldsymbol{\Omega}$ is the feasible search region and $\boldsymbol{x} = (x_1, x_2, ...x_n)^T$ is the decision variable vector. $\boldsymbol{F} : \boldsymbol{\Omega} \rightarrow \boldsymbol{R^m}$, m is the number of the objective function and $\boldsymbol{R^m}$ is the objective space.

Due to the conflicting nature of the objectives, a single solution is not available to minimize all the objectives in most practical optimization problems. Instead, a set of trade-off optimal solutions known as Pareto-optimal (PO) solutions [1] can optimize all the conflicting objectives at the same time. The Pareto-optimal solutions are known as the comprised members of Pareto set (PS), and the Pareto front(PF) is defined as $PF = \{\boldsymbol{F}(\mathbf{x})|\mathbf{x} \in \boldsymbol{PS}\}$ [2].

Over the past two decades, Multi-objective evolutionary algorithms (MOEAs) have witnessed a boom of development [3], and also play a role in real world [4, 5]. MOEAs can be recognized as successors when we deal with MOPs having less than three objectives. But massive difficulties have been experienced when they are adopted to deal with the MaOPs [6, 7], whose objectives are more than three. Particularly for the popular Pareto dominance-based MOEAs, such like strength Pareto evolutionary algorithm SPEA-II [8] and nondominated sorting genetic algorithm NSGA-II [9], experimental and analytical studies can be found in [10, 11].

One major possible reason for the failure can be attributed to the loss of selection pressure. The selection pressure toward the Pareto front(PF) would be reduced as the proportion of nondominated solutions in a population increases exponentially with the number of the objectives increased. These situation happens in NSGA-II and SPEA-II mentioned before.

Another important reason can be regarded as the difficulties in maintaining a good balance between the population diversity and the convergence in a high-dimensional objective space. A set of evenly distributed representative solutions is aimed to be found by MOEAs to approximate the PF, and it is not hard work when the number of objectives is two or three. However, as the dimension of the objective space increased, the diversity of the population becomes increasingly hard to be maintained, as the distribution of each solution is very sparse in the high-dimensional objective space, and the diversity management strategies such as the crowding distance in NSGA-II is difficult to be used in this situation [12, 13, 14].

In the past several years, a number of approaches have been proposed to overcome the shortages of most traditional MOEAs when we solve MaOPs. They can be divided into four types roughly.

The first approach can be grouped into convergence enhancement. In order to increase the selection pressure toward the PF, some modified dominance definitions have been proposed, include ϵ-dominance [15, 16], preference order ranking [17], and fuzzy dominance [18]. In [19], a novel ranking strategy called Global Margin Ranking (GMR) is proposed by Li, which deploys the position information of individuals in objective space to gain the margin of dominance throughout the population. In [20], a grid-based EA(GrEA) has been proposed, in which a grid dominance-based metric for solving MaOPs has been defined and the dominance criterion has been modified to accelerate the convergence to PF. In [21], a novel multi-objective immune algorithm with double modules is presented in order to locate the entire Pareto-optimal front. In [22], a Baldwinian learning strategy is designed for improving the nondominated neighbor immune algorithm and a multi-objective immune algorithm with Baldwinian learning (MIAB) is proposed. In [23], a new decomposition based evolutionary algorithm with uniform designs is proposed to get a set of solutions with good convergence and diversity. Some people also combine the Pareto dominance-based criterion with additional convergence-related metrics. For instance, a binary ϵ-indicator-based preference which is combined with dominance, has been proposed in [24], which speeds up the convergence of NSGA-II for solving MaOPs. In [25], a knee point-driven EA (KnEA) is proposed which makes the knee point-based the secondary selection to enhance convergence pressure.

The second category is known as the performance indicator-based approaches. As hypervolume (HV) may be the most widely used performance indicator for the nice theoretical nature it has [26], a dynamic neighborhood MOEA based on HV indicator [27] and the fast HV-based EA (HypE) [28] have been proposed as well as few others [29], [30]. However, the shortage of this kind of algorithms is that the computational cost of HV grows exponentially as the number of objectives increase. Although, there are other indicators like $R2$ [31] and Δ_p [32], which have been implemented into some recent MOEAs [33, 34] but none of them can get a good performance when we solving MaOPs.

The third type of MOEAs are based on the use of preference, known as the Preference-inspired co-evolutionary algorithms (PICEAs). PICEA-g [35] is one realization of PICEAs proposed by Wang, in which goal vectors are taken as preferences and are co-evolved with the candidate solutions during the search . PICEA-w [36] has been proposed recently, in which weights are co-evolved with candidate solutions during the search process, and guiding candidate solutions towards the Pareto optimal front effectively. This kind of algorithms have gotten a good performance in experiments and are less sensitive to the problem geometry.

The fourth type is known as the decomposition-based MOEAs, in which a MOP is generally divided into a number of subproblems, and optimized by optimizing all the objectives simultaneously. These kind of MOEAs aggregate the objectives of a MOP into an aggregation function by using a unique weight vector, and a set of weight vector will generate multiple aggregation functions to present each single-objective optimization problem. MOEA based on decomposition(MOEA/D) [37, 38] has been the very popular MOEA since it was proposed. MOEA/D performs well on approaching the PF even if in high-dimensional objective space, but maintaining the diversity in many-objective optimization is still hard for it, and as a result, MOEA/D usually fail to achieve a good coverage of PF [20, 39]. Recent years, MOEA/D has spawned a large amount of research work [40, 41, 42]. But the problem of diversity still exists, and even the decomposition-based algorithms proposed recently, such like MOEA/D-M2M [43], NSGA-III [39], did not address this problem well, and the convergence to the PF in many-objective optimization is still unsatisfactory. In [44], a perpendicular distance from a solution to the weight vector has been exploited, which has enhanced the convergence and diversity of solutions.

This paper focuses on the decomposition-based MOEAs that boost the convergence by exploiting the aggregation functions in many-objective optimization. Compared with existing decomposition-based MOEAs, the new major contributions of this paper can be summarized as follows.

1) In order to get a better balance between convergence and diversity in decomposition-based many-objective optimizations, the acute angle between a solution and a direction vector is exploited, which is totally different from [44].
2) The idea has been implemented in MOEA/D and get an enhanced algorithm, i.e., MOEA/D variant (MOEA/D-AU) with an angle-based updating strategy.

3) Different existing ideas are combined together in our algorithm. MOEA/D-AU inherits the merit of tow-layer weight vectors (the same as [39]), notion of neighborhood (the same of [38]), online normalization and the preferred order of replacement (the same as [44]).
4) The differences between our proposed algorithm and other existing decomposition-based algorithms have been discussed and the possible reasons why the compared algorithms performs good or not in the experiments are discussed in our analysis.

The rest of this paper is organized as follows. In Section 2, the background knowledge and framework of MOEA/D is introduced. Section 3 gives the basic idea and describes in detail how to improve the performance of MOEA/D by implemented our idea. In Section 4 , the designs of experiments are given in detail. Section 5 analyses the influence of a key parameter G in the proposed algorithm. In Section 6 , the proposed algorithm is compared to its predecessors and some state-of-the-art methods. Finally, Section 7 concludes this paper.

2. Preliminaries and background

In this section, some basic knowledge about the decomposition method used in MOEA/D is given at first. Then, some related approaches and framework of MOEA/D are introduced.

2.1. Decomposition Methods

In the original MOEA/D framework [37], a set of uniformly spread weight vectors $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2, \ldots,}$ $\boldsymbol{\lambda_N}$ are generated, where $\boldsymbol{\lambda_i} = (\lambda_i^1, \lambda_i^2, .., \lambda_i^m)^T$, subject to the conditions $\lambda_i^j \geq 0$ and $\sum\limits_{j=1}^{m} \lambda_i^j = 1$ for all $j = 1, 2, \ldots, m$ as well as $i = 1, 2, \ldots, N$. To generate these weight vectors, there are three widely used decomposition approaches as fellow:

1) Weighted Sum (WS) Approach: In this approach, the ith subproblem is defined in the form

$$minimize \quad g^{ws}(\mathbf{x}|\boldsymbol{\lambda_i}) = \sum_{j=1}^{m} \lambda_i^j f_j\left(\mathbf{x}\right) \tag{2}$$

This approach works very well for convex PFs. The strength is the search ability of the WS method can keep well and will not be affected by the increasing number of the objectives. It is also the best in terms of minimizing the projector distance of solutions on direction vectors. The weakness of this approach is that this method may not be able to find all Pareto optimal solutions in the case of non-convex PFs. [2, 45] .

2) Tchebycheff (TCH) Approach: In this approach, the th subproblem is defined in the form

$$minimize \quad g^{te}(\mathbf{x}|\boldsymbol{\lambda_i}, \boldsymbol{Z}^*) = \max_{1\leq j\leq m}\{\lambda_i^j|f_j(\mathbf{x}) - z_j^*|\} \tag{3}$$

where $\boldsymbol{Z}^* = (z_1^*, \ldots, z_m^*)^T$ is the ideal reference point with

$$z_j^* < \min\{f_j(\mathbf{x})\,|\mathbf{x} \in \boldsymbol{\Omega}\} \quad for \quad j = 1, 2, \ldots, m. \tag{4}$$

This approach is one of the most common used approaches. The strength of this approach is that the solutions selected by Tchebycheff method are closer to the ideal point compared to WS method and have a better convergence than the Pareto dominance relation based selection. It also performs well in terms of capturing the small changes of solutions in objective space and can nd solutions in both convex and non-convex regions. The weakness of this method is that the search ability is affected by the increasing number of objectives while the WS method is not [45].

3) Penalty-based Boundary Intersection(PBI) Approach:In this approach, the ith subproblem is defined in the form

$$minimize \quad g^{pbi}(\mathbf{x}|\boldsymbol{\lambda_i}, \boldsymbol{Z}^*) = d_1 + \theta d_2 \tag{5}$$

where, $\begin{cases} d_1 = \frac{\left\|(\boldsymbol{F}(\mathbf{x})-\boldsymbol{z}^*)^T\boldsymbol{\lambda_i}\right\|}{\boldsymbol{\lambda_i}} \\ d_2 = \left\|\boldsymbol{F}(\mathbf{x}) - (\boldsymbol{z}^* - d_1\frac{\boldsymbol{\lambda_i}}{\|\boldsymbol{\lambda_i}\|})\right\| \end{cases}$

In Eq. (5), $\boldsymbol{Z}^*$ is set of the reference points as defined in the Tchebycheff approach and θ is a penalty parameter. This approach works well to approximate a PF if the weight vectors are set in an appropriate manner [11]. Usually, the smaller the θ value, the faster the algorithm converges. However, for PBI with a small θ value, the optimal solution could be far away from its corresponding direction vectors and may not work effectively on non-convex PFs, but on the other side, a small penalty value is more prone to convergence performance [46]. For PBI with a large θ value, it become robust on various PF shapes.

2.2. MOEA/D Framework

By using the three decomposition methods discussed above (WS, TCH, PBI), an MOP is decomposed into a set of N scalar objective optimization subproblems in MOEA/D. In the original MOEA/D framework [37], the Tchebycheff approach is adopted. By using the TCH approach, the solution to the target MOP is equivalent to optimizing N scalar optimization subproblems, and the j-th objective function of the ith subproblem can be expressed as

$$g^{te}(\mathbf{x}|\boldsymbol{\lambda_i}, \boldsymbol{Z^*}) = \max_{1\leq j\leq m} \{\lambda_i^j |f_j(x) - z_j^*|\} \tag{6}$$

In MOEA/D, a population consisting of N solutions is randomly generated and each solution is allocated to a particular subproblem as well. x^i is the solution of subproblem i. A neighborhood of weight vector $\boldsymbol{\lambda_i}$ is defined as the T closest weight vectors which in $\boldsymbol{\lambda_1, \lambda_2 \ldots, \lambda_N}$. $\boldsymbol{B^i(T)}$ is used in this paper to denote the set of the T indexes of neighboring subproblems of subproblem i. In the reproduction operation, two solutions have been randomly chosen from the neighborhood of the subproblem i in order to generate an offspring by genetic operators. After that, the neighborhood of the subproblem i is updated by the offspring generated before. At last, the solutions in the population are the best solutions for each subproblem. In recent years, many different MOEA/D variants have been proposed [44, 47, 41]. The framework of proposed MOEA/D [38] is given in Algorithm 1.

2.3. Related Algorithms

In this section, several representative MOEAs relating to this paper are introduced.

1) Li et al. [47] enhanced the MOEA/D with a stable matching (STM) model to coordinate the selection process, and got a variant of MOEA/D called MOEA/D-STM. This algorithm considers two parts, the aggregation function values and the direction vectors which are close, to get a good balance between convergence and diversity, a matching algorithm is used to apportion each solution to each subproblems in MOEA/D-STM.
2) Deb and Himanshu [39] proposed NSGA-III, which is a reference point-based many-objective algorithm based on NSGA-II. In the last accepted front during the evolutionary process, the solutions are selected based on a niche-preservation operator but the crowding distance, the less crowded reference line a solution connected, the higher priority the solution to be chosen.

Algorithm 1: Framework of MOEA/D

InPut: N: population size, T: neighborhood size.
n_r: maximal number of updated subproblems.
$\boldsymbol{B^i}\left(\boldsymbol{T}\right)$: index set of the neighbors of subproblem i.
δ: a control parameter.
Termination condition.

OutPut: all solutions in the population

begin

1 `/* Initialization */`
population $\boldsymbol{P} \leftarrow \{x^i\}$, weight vectors $\boldsymbol{\Lambda} \leftarrow \{\boldsymbol{\lambda_i}\}$,
the ideal point $\boldsymbol{Z^*} \leftarrow \{z_i^*\}$, neighbor index set $\boldsymbol{B^i}\left(\boldsymbol{T}\right)$,
set $F^i = F\left(x^i\right)$ for $i = 1, \ldots, N$.
`/* Update */`

2 **while** *the termination condition is not met* **do**

3   **for** *each subproblem* $i = 1, \ldots, N$ **do**

4   `/* determine the mating or update pool */`
Set $\boldsymbol{E} \leftarrow \boldsymbol{B^i}\left(\boldsymbol{T}\right)$, if $rand() \leq \delta$, otherwise
$\boldsymbol{E} \leftarrow \{1, 2, \ldots, N\}$

5   `/* recombination */`
Randomly select two index k, l from E
Generate a new solution
$\boldsymbol{y} \leftarrow recombination\ operators\left(x^k, x^l\right)$

6   `/* population replacement */`
Set $c = 0$.
while $c < n_r$ *and* $\boldsymbol{P} \neq \phi$ **do**
  Randomly pick an index j from P
  $\boldsymbol{P} := \boldsymbol{P} \backslash \{j\}$
  If $\boldsymbol{y}$ has a better aggregation function value than x^j for subproblem j
  then
  c = c + 1
  end
    end
  end

7 **return** population $\boldsymbol{P} = \left\{x^1, \ldots, x^N\right\}$.

end

3) Wang et al. [48] proposed a decomposition-based multi-objective evolutionary algorithm with constrained subproblems termed MOEA/D-CD or MOEA/D-ACD. Some constraints on the subproblems are imposed, which reduce the improvement regions for commonly used decomposition approaches. This strategy gets a good balance between the population diversity and convergence, significantly improving the algorithm performance.
4) Yuan et al. [44] proposed an enhanced version of MOEA/D termed MOEA/D-DU. A perpendicular distance is introduced in [44] to identify which solution is closet to the weight vector, and select k closet new solutions to update the old solutions. This algorithm also gets a good balance between the diversity and convergence and gives us the inspiration of this paper.

3. Basic idea and implementation

3.1. Basic Idea

The Tchebycheff function may be the most popular type of aggregation functions used in MOEA/D or some other variants. As the origin Tchebycheff function may not work well when the target MOP has a complex Pareto front, the modified Tchebycheff function [48] is employed in this paper, and the function for jth subproblem can be defined as fellows

$$F_j(\mathbf{x}) = \sum_{k=1}^{m} \left\{ \frac{1}{\lambda_{j,k}} \left| f_k(\mathbf{x}) - z_k^* \right| \right\} \tag{7}$$

where z_k^* is the ideal reference point, $\boldsymbol{\lambda_j} = (\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,m})^T$, $j = 1, 2, \ldots, N$, $\lambda_{j,k} \geq 0$ for all $k \in \{1, 2, \ldots, m\}$ and $\sum_{k=1}^{m} \lambda_{j,k} = 1$. If $\lambda_{j,k} = 0$, $\lambda_{j,k}$ is set to 10^{-6}.

Compared to the original one, the modified version of the Tchebycheff function used in some decomposition-based MOEAs has two advantages over the original one. One is the uniformly distributed weight vectors leading to the uniformly distributed search directions in the objective space. The other is each of the weight vectors corresponding to a unique solution located on PF.

However, the modified version still faces problems in practice. Assuming that N weight vectors : $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, ..., $\boldsymbol{\lambda_N}$ are used, and the PF is divided into N

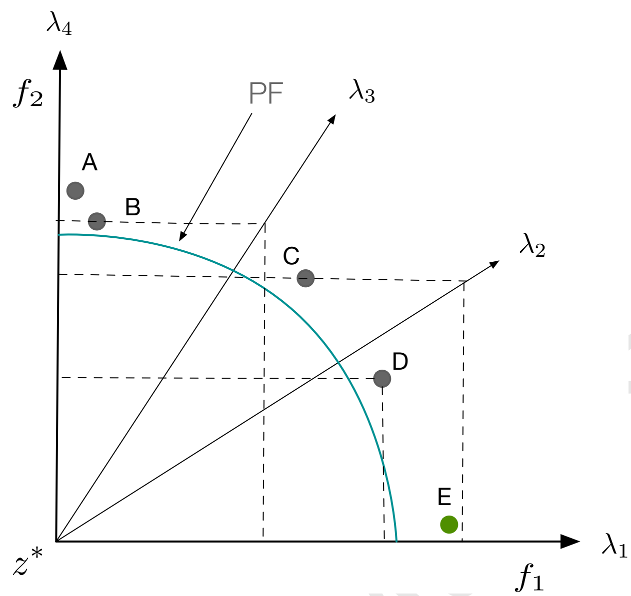


Figure 1: Illustration of the distribution of solution in the 2-D objective space

parts by a decomposition based algorithm. In Fig. 1, there are four different and even spread weight vectorsvectors: $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, $\boldsymbol{\lambda_3}$, $\boldsymbol{\lambda_4}$. Each Tchebycheff method with a different weight vector tries to find a Pareto optimal solution. A-D are four solutions found so far according to these four different vectors. A for λ_4, B for λ_3, C for λ_2 and D for λ_1. It is obvious that the solutions do not spread as well as the weight vectors. Solution B, C and D deviate far from their corresponding directions $\boldsymbol{\lambda_3}$, $\boldsymbol{\lambda_2}$ and $\boldsymbol{\lambda_1}$, although they achieve good aggregation function values on their corresponding directions. Assuming E is also an solution gotten by Tchebycheff method with weight vector $\boldsymbol{\lambda_1}$. Comparing solution E and solution D, it can be found that although E is a little worse than D in terms of the Tchebycheff function value, but E is in fact more preferred by the weight vector $\boldsymbol{\lambda_1}$. Unfortunately, if the selection of solutions only takes the aggregation function values as criterion, solution E would be replaced by solution D in the updating procedure without a doubt. It should be pointed out that, in the early stage of evolution, the solutions are usually far away from the PF, and the search may be restricted to a small part of the PF if the misleading selection occurs a lot, which influences the diversity of the solutions a lot. The situation can be more aggravate in the high-dimensional objective space due to the few and scattered distribution of solutions.

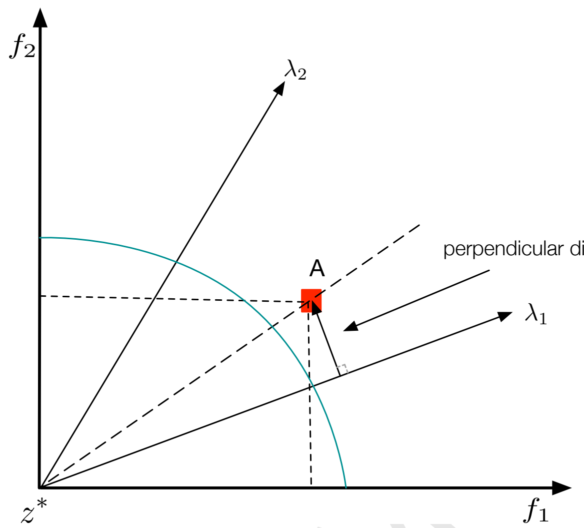


Figure 2: Illustrations of the perpendicular distance between solutions and corresponding weight vector

To address the problems mentioned above, we begin to consider not only the aggregation function value of a solution, but also something else which could be taken as an other measurement.

A perpendicular distance from a solution to the weight vector is used in [44], and this idea is illustrated in Fig. 2. There is no doubt that this idea is a good approach to deal with the problem mentioned above, and can keep the solutions close to the direction vectors. But what will happen if the acute angle between a solution and the direction vector used instead of the perpendicular distance. The difference is illustrated in Fig. 3. In Fig. 3, there are two solutions A and B, B is little closer to the direction vector α^i than A in terms of the perpendicular distance, but if the acute angle between a solution and the corresponding direction vector is used to identify which solution is closer to the direction, A will be the answer. So, in general, if the perpendicular distance is replaced by acute angle, more different solutions will have the chance to be selected compared to the selection in [44]. Intuitively, this kind of selection method may be helpful to increase the possibility to generate solutions having more difference and avoid to be caught in local optimum when the Pareto optimal front is complex. That is why we considered the acute angle between a solution and a direction to be used as

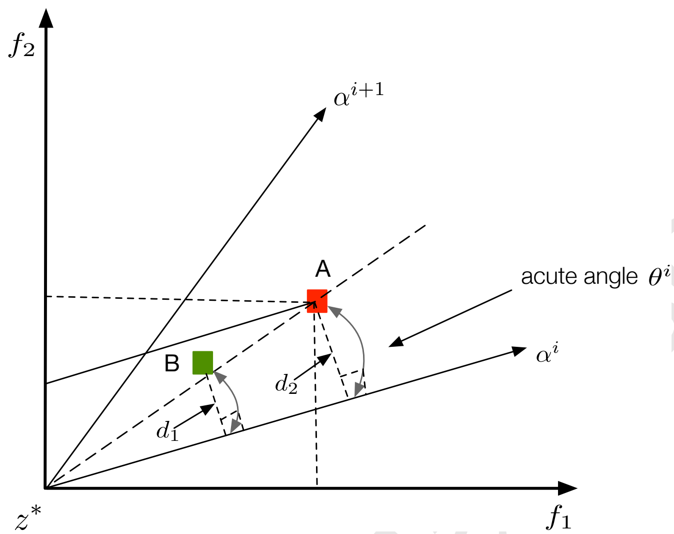


Figure 3: Illustrations of the difference between perpendicular distance and acute angle

the measurement to decide which solution is closer to the direction vector in this paper and it has been illustrated in Fig. 4.

This idea is expected to keep the solutions which are closed to the direction vectors to be selected, and so to maintain a good distribution of the solutions in the evolutionary process. As a result, the balance between convergence and diversity may become better in many-objective optimization.

The idea has been illustrated in Fig. 4. In Fig. 4, solution A is the current solution and $\boldsymbol{\alpha^i}$ is the corresponding direction vector, θ^i is the acute angle between solution A and its corresponding direction vector $\boldsymbol{\alpha^i}$, $\boldsymbol{\alpha^i} = (1/\lambda_i^1, \ldots, 1/\lambda_i^m)^T$, when the weighted Tchebycheff approach is used. And when the WS approach and the PBI are used, $\boldsymbol{\alpha^i} = \boldsymbol{\lambda_i}$. The acute angle between vector $\boldsymbol{\alpha^i}$ and vector $\boldsymbol{F}(\mathbf{x}) - \boldsymbol{Z^*}$. $\theta^i(\mathbf{x})$ is denoted as $\left\langle \boldsymbol{\alpha^i}, \boldsymbol{F}(\mathbf{x}) - \boldsymbol{Z^*} \right\rangle$. $\theta^i(\mathbf{x})$ can be obtained by

$$\theta^i(\mathbf{x}) = \left\langle \boldsymbol{\alpha^i}, \boldsymbol{F}(\mathbf{x}) - \boldsymbol{Z^*} \right\rangle = \arccos\left(\frac{\boldsymbol{\alpha^i} \times (\boldsymbol{F}(\mathbf{x}) - \boldsymbol{Z^*})}{\left\| \boldsymbol{\alpha^i} \right\| \times \left\| \boldsymbol{F}(\mathbf{x}) - \boldsymbol{Z^*} \right\|} \right) \tag{8}$$

According to the value of acute angle $\theta^i(\mathbf{x})$, the smaller value of $\theta^i(\mathbf{x})$, the closer the solution x to the direction vector $\boldsymbol{\alpha^i}$. Thus, the solution which is close to the direction vector could be selected.

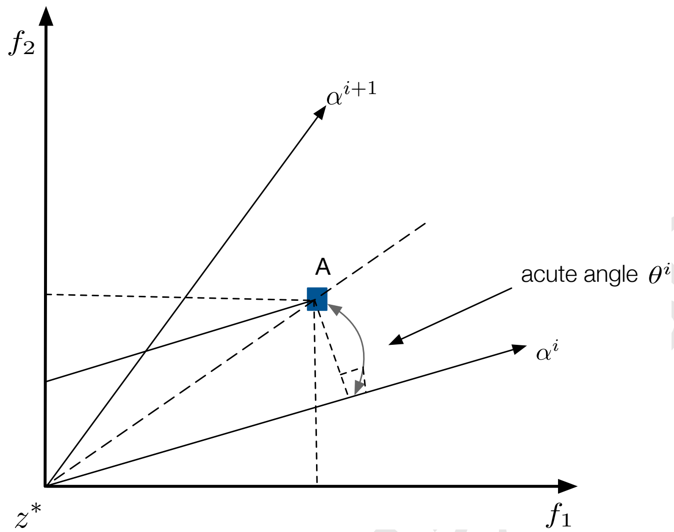


Figure 4: Illustrations of the acute angle between solutions and corresponding weight vector

3.2. Implementation

The idea has been implemented in the updating strategy of population replacement, which is significantly different from the original MOEA/D in step 6 of Algorithm 1. In order to control how many old solutions corresponding to the direction vectors are needed to be updated, a parameter G is introduced in the updating procedure and the influence of G is illustrated in Section 5. The framework of MOEA/D-AU is illustrated in Algorithm 2 briefly. The detail of this proposed strategy is presented in Algorithm 3. In Algorithm 3, a new solution y is produced at first, then the acute angle to each direction vector $\boldsymbol{\alpha^j}$, i.e.$\theta^i\left(x\right)$, $j = 1, \ldots, N$, are computed respectively. In step 4 of Algorithm 3, G minimum angles are selected from all the N angles got before, where G is much smaller than N. Then, the G angles are arranged in the nondecreasing order, i.e.$\langle a^{j_1}, \boldsymbol{F}\left(\boldsymbol{y}\right) - \boldsymbol{Z}^* \rangle \leq \langle a^{j_2}, F\left(\boldsymbol{y}\right) - \boldsymbol{Z}^* \rangle \ldots \leq \langle a^{j_G}, F\left(\boldsymbol{y}\right) - \boldsymbol{Z}^* \rangle$. After that, solution $\boldsymbol{y}$ is compared to the solutions x_{j1},x_{j2} ,.., x_{jG} one by one, $\boldsymbol{g} = (1, \ldots, G)$,and x_{jg} would be replaced if the aggregation function value of $\boldsymbol{y}$ is better than the function value of x_{jg}. From the process mentioned above, it can be seen that one solution in the current population can be replaced by the new solution $\boldsymbol{y}$ at most.

Algorithm 2: Framework of MOEA/D-AU

InPut: N: population size, T: neighborhood size .
n_r: maximal number of updated subproblems.
$\boldsymbol{B^i}\,(\boldsymbol{T})$: index set of the neighbors of subproblem i.
δ: a control parameter.
Termination condition.

OutPut: an approximation to the PF (PS).

begin
- Initialization: $\boldsymbol{P} \leftarrow \{x^i\}$// weight vectors $\boldsymbol{\Lambda} \leftarrow \{\boldsymbol{\lambda_i}\}$
  the ideal point $\boldsymbol{Z^*} \leftarrow \{z_i^*\}$
  neighbor index set $\boldsymbol{B^i}\,(\boldsymbol{T})$,
  set $F^i = F\left(x^i\right)$ for $i = 1, \ldots, N$.
- **while** *the termination condition is not met* **do**
  - **for** *each subproblem* $i = 1, \ldots, N$ **do**
    - $\boldsymbol{E} \leftarrow$ *determine the mating or updatepool*(δ);
    - Randomly select an index k from $\boldsymbol{E}$,
      $\boldsymbol{y} \leftarrow$ *recombination operators* $\left(x^i, x^k\right)$;
    - $\boldsymbol{Z^*} \leftarrow$ *updating* $\mathrm{Z}^*(\boldsymbol{y})$;
    - $\boldsymbol{P} \leftarrow$ *population replacement*$(\boldsymbol{y})$;

    end

  end
- **return** population $\boldsymbol{P}$.

end

Algorithm 3: Population Replacement of MOEA/D-AU

begin
  for $j \leftarrow 1$ *to* N **do**
    Compute the acute angle between $\boldsymbol{\alpha^i}$ and $(\boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^*)$, i.e., $\theta^j(\boldsymbol{y}) = \left\langle \boldsymbol{\alpha^i}, \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^* \right\rangle$
  end
  Select G minimum angle from N angles $\left\langle \boldsymbol{\alpha^j}, \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^* \right\rangle$ $j = 1, \ldots, N$, and get $\langle a^{j_1}, \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^* \rangle \leq \langle a^{j_2}, \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^* \rangle \ldots \leq \langle a^{j_G}, \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{Z}^* \rangle$
  for $g \leftarrow 1$ *to* G **do**
    if $F^{jg}(\boldsymbol{y})$ is better than $F^{jg}(x_{jg})$ **then**
      $x_{jg} \leftarrow \boldsymbol{y}$
    return
  end
end

3.3. Normalization

For an algorithm, when the objective values on a PF are not the same scaled, it may be a trouble for the algorithm to work well, but normalization can deal with this problem very well.

In this paper, we use the normalization procedure proposed in [44], which is similar to the normalization in NSGA-III, the objective value of ith subproblem can be replaced by the function below

$$\widetilde{f}_i(\mathbf{x}) = (f_i(\mathbf{x}) - z_i^*)/({z_i}^{nad} - z_i^*) \tag{9}$$

where z_i^* is the minimum value of ith objective and ${z_i}^{nad}$ is the intercept of the constructed hyperplane with ith objective axis in the last one generation, $i \in \{1, 2, \ldots, n\}$.

The different between the normalization we used and the normalization in NSGA-III is the achievement scalarizing function to identify extreme points. Suppose $\boldsymbol{S_t}$ is the population needs to be normalized, and the solution $\mathbf{x} \in \boldsymbol{S_t}$ could be identified as the extreme point if it can minimizes the achievement scalarizing function below:

$$ASF(\mathbf{x}, \boldsymbol{w_j}) = max_{i=1}^{m} \left\{ \frac{1}{w_{j,i}} \left| \frac{f_i(\mathbf{x}) - {z_i}^*}{{z_i}^{nad} - {z_i}^*} \right| \right\} \tag{10}$$

where $\boldsymbol{w_j} = (w_{j,1}, w_{j,2}, \ldots, w_{j,m})^T$, $j \in \{1, 2, \ldots, m\}$, $w_{j,i} = 0, when\ i \neq j$ and $w_{j,i} = 1$ if $w_{j,i} = 0$, it will be replaced by a small number10^{-6}.

3.4. Computational Complexity

For MOEA/D-AU in Algorithm 3, the updating procedure takes the major part of the computational costs. $O(mN)$ computations are required in step 1-3 to compute each angle. To select G minimum angels and sort all of them need $O(N \log G)$ computations. $O(mG)$ computations are needed at most to compare the solutions. Thus, $O(mN^2)$ is the total complexity of MOEA/D-AU to produce N test solutions in one generation, as m is larger than $logG$ for MaOPs in most cases .

4. Experimental design

In this section, the design of experiments to investigate the performance of our proposed algorithm MOEA/D-AU is illustrated. First, the benchmark test problems are given as well as the performance metrics. Then, the MOEAs which compared to MOEA/D-AU in the experiments are listed. At last, the experimental settings are given in detail.

4.1. Benchmark Test Problems

In order to compare the performance of each algorithm, several test problems form [49], [50] are used. DTLZ1-DTLZ4, DTLZ7, WFG1-WFG9. All of them are widely used benchmark test problems. In this paper we consider the number of objectives $m \in \{2, 5, 8, 10\}$. The main features of each test problems are illustrated in Table 1.

The problems are divided into two groups. The first group of test problems consists of the normalized test problems DTLZ1-DTLZ4, DTLZ7, and WFG1-WFG9. It should be noticed that the original DTLZ7 and WFG1-WFG9 test problems are scaled differently. Therefore, the objective functions of DTLZ7 and WFG1-WFG9 problems are modified as

$$f_i \leftarrow (f_i - z_i^*) \,/\, \left(z_i^{nad} - z_i^*\right), i = 1,\ 2,\ \ldots,\ m. \tag{11}$$

Then, the ideal point is 0 while the 1 is the nadir point.

The second group of test problems are scaled test problems, consisting of DTLZ1 and DTLZ2 problems [36] and the original version of WFG4-WFG9 problems. This group of problems are used to test the performance

Table 1: Illustration of the features of the test problems

Problem	Features
DTLZ1	Liner, Multi-modal
DTLZ2	Concave
DTLZ3	Multi-moda,Concave,
DTLZ4	Concave, Biased
DTLZ7	Mixed, Multi-modal,Disconnected
WFG1	Mixed, Biased
WFG2	Convex, Disconnected, Multi-modal, Non-separable
WFG3	Linear, Degenerate, Non-separable
WFG4	Concave, Multi-modal
WFG5	Concave, Deceptive
WFG6	Concave, Non-separable
WFG7	Concave, Biased
WFG8	Concave, Biased, Non-separable
WFG9	Concave, Biased, Multi-modal, Deceptive, Non-separable

of an algorithm with a normalization procedure to handle differently scaled objective values. The scaled DTZL1 and DTLZ2 problems are illustrated as follows: 10^i is supposed to be the scaling factor, and then the objective functions of the original DTLZ1 problem will be modified as

$$f_i \leftarrow 10^{i-1} f_i, \quad i = 1, 2, \ldots, m. \tag{12}$$

G is set to 5 for all the test instances. The decision variables in all DTLZ problems are set to $n = m + k - 1$. For all WFG test problems, 24 is the number of decision variables, $m - 1$ is the position-related parameter.

4.2. Performance Metrics

Two famous performance metrics are used in this paper to evaluate the performance of the algorithms compared. One is the inverted generational distance (IGD), the other is HV [50].

IGD is the performance metric which can evaluate not only the convergence but also the diversity of a solution set. The smaller values mean the better quality. HV also can measure both convergence and diversity of a solution set in a sense, and the larger values of HV means the better quality

Table 2: Scaling factors for scaled problems

No. of Objectives	Scaling Factor	
m	DTLZ1	DTLZ2
2	10^i	10^i
5	10^i	10^i
8	3^i	3^i
10	3^i	2^i

of the performance. However, in high-dimensional objective space, a huge calculation is required to calculate IGD as the number of points on the PF which is needed in the calculation is enormous. For this reason, HV has been chosen as the primary performance metric in this paper.

For calculating HV, it is very important to choose an appropriate reference point. In our experiments, the reference point is set to be $1.1z^{nad}$, and z^{nad} is easy to be obtained by analyzing for each test problem. Moreover, based on the practice in [50] and [51], the solutions will be discarded for the HV calculation if they do not dominate any of the reference points. And the proposed WFG algorithm [52] is used in this paper to calculate the HV.

4.3. MOEAs for Comparison

Five MOEAs are selected to evaluate the performance of MOEA/D-AU in this paper.

The first one is MOEA/D [38]. As MOEA/D-AU is the enhanced version of MOEA/D, to compare with the original algorithm. The advantage of the updating strategy proposed in this paper can be clearly illustrated by the performance of the experiments.

The other one is MOEA/D-DU [44]. As MOEA/D-DU is somewhat similar to MOEA/D-DU and can be regarded as anther predecessor of MOEA/D-AU except MOEA/D. The different performance of the selection strategy can be illustrated clearly by the experiments.

The other three compared MOEA/D variants are MOEA/D-STM [47], MOEA/D-GR [53] and MOEA/D-LWS [54], because they are somewhat similar to MOEA/D-AU and all of them are proposed in recent years. For fair comparison, the differential evolution (DE) operator in MOEA/D-STM is replaced by the recombination schemes in MOEA/D-AU.

Table 3: Settings of population size

No. of Objectives (m)	Divisions (H)	Population Size (N)
2	99	100
5	6	210
8	3,2	156
10	3,2	275

Table 4: Settings of crossover and mutation

Parameter	Value
Crossover probability	1.0
Mutation probability	1/n
Distribution index for crossover	20
Distribution index for mutation	20

All the algorithms compared in the experiments use the modified Tchebycheff function defined in Eq. (7) except MOEA/D-LWS using its own localized weighted sum method. And the JMetal framework [55] is used to implement these algorithms.

4.4. Experimental Settings

The experimental settings are stated as follows.

1) Population Size: In this paper, the population size is set according to a parameter $H(N = C^{(m-1)}_{(H+m-1)})$ as MOEA/D, and equal to the number of weight vectors in all the compared algorithms. For problems with more than five objectives, the two-layers weight vectors generation method [39] is adopted to generate weight vectors which can avoid producing only the boundary weight vectors. The settings of population size are listed in Table 3.
2) Settings of Crossover and Mutation: For all the algorithms compared, SBX crossover method and polynomial mutation method are employed to generate new solutions. The settings are listed in Table 4.
3) Neighbourhood Size T and Generate Probability δ : T is set to 20 in all the algorithms compared in experiments, i.e., MOEA/D, MOEA/D-STM,

MOEA/D-GR, and δ is equal to 0.9.

4) Parameter G in MOEA/D-AU is set to 5 and the influence of the setting of G is investigated in Section 5.
5) Runs Times and Termination Criterion: Each tested instance is run 30 times independently by the algorithms compared. The algorithm will be terminated after 2000xm function evaluations every run time.
6) Significance Test: In order to investigate the difference between two algorithms for statistical significance, the Wilconxon signed-rank test is used on the metric values gotten by two competing algorithms at a significance level of 5%.

In addition, the algorithms compared have their own specific parameters. The maximal number of solutions which can be replaced by each offspring solution is set to 1 in MOEA/D and MOEA/D-GR. While 5 is set to the number of replacement neighborhood T_r in MOEA/D-GR.

5. Investigation the influence of parameter *G*

In this section, the influence of different settings of parameter G is analyzed.

In our proposed algorithm MOEA/D-AU, G is a very important control parameter that influences the balance between the convergence and diversity. In order to investigate the influence of parameter G, different settings of $G \in [1, 20]$ are tested on all the normalized problem instances with a step size of 1. Other parameters except G are the same as in Section 4.4 .

In Fig. 5, the influence of parameter G on different test problems are showed by the value of HV. The instances tested are Normalized DTLZ1, DTLZ4, DTLZ7, and WFG1, WFG3, WFG9 with 2,5,8 and 10 objectives. From Fig. 5, the influence of different settings of parameter G can be observed as follows:

1) Different settings of parameter G exert different impact on the performance of the test instances.
2) The most suitable value of parameter G depends on the two parts: the problem tested and the number of objectives of the problems.
3) For some test problems, every value of G between 1 to 20 can get the similar performance, e.g., DTLZ1, DTLZ4. It can be seen from Fig. 5 (a), Fig. 5 (b).

4) Different setting of parameter G may cause a little variation of performance for some problems, e.g., WFG3 and WFG9.
5) For DTLZ7, a little change on the value of parameter G may lead to a huge change of the performance. For this kind of problems, the setting of parameter G should be more careful.

Above all, the influences of different settings of parameter G on the tested problems are affected. In most cases, it may be a good choice to adjust the value of parameter G between $[1, 20]$ since the algorithm can usually get a good performance with such G values. But when the convergences of the solutions are not good, larger G value may be helpful as well as a smaller G value may push the diversity of solutions.

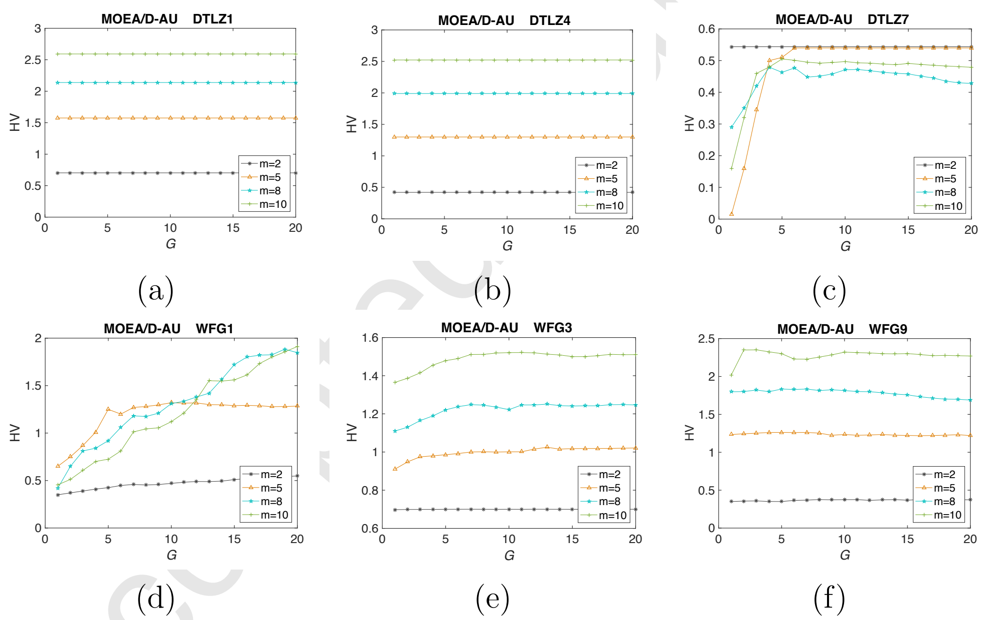


Figure 5: Influence of G on the performance of MOEA/D-AU for the normalized (a)DTLZ1, (b)DTLZ4, (c)DTLZ7, (d)WFG1, (e)WFG3, (f)WFG9 test problems with different number of objectives m. Illustrated by the average value of HV.

6. Comprsions with the original and state-of-the-art algorithms

In this section, MOEA/D-AU is compared with its predecessor and some state-of-the-art algorithms. Section 6.1 investigated the ability of MOEA/D-

AU to balance the convergence and diversity in many-objective optimizations compared to its predecessor. Section 6.2 compared MOEA/D-AU with other state-of-the-art algorithms. Section 6.3 compared MOEA/D-AU with NSGA-III on scaled test problems. In Section 6.4 some issuers relevant to the experiments are discussed.

6.1. Comparison with the predecessors

In this section, the performance of balancing the convergence and diversity of our proposed algorithm are investigated by IGD and HV metrics. Normalized DTLZ1-DTLZ4 problems are tested in this part.

In order to investigate the improvement of our proposed algorithm compared to its corresponding predecessor MOEA/D and MOEA/D-DU in the many-objective optimizations, DTLZ1-DTLZ4 problems are tested by both algorithms. The average IGD and HV results are provided in Table 5. From Table 5, it can be seen that, compared to MOEA/D, the better HV and IGD results are achieved by MOEA/D-AU in most concerned instances. Compared to MOEA/D-DU, the results of MOEA/D-AU are better on DTLZ2 and DTLZ4, and as effective as MOEA/D-DU on DTLZ1 and DLTZ3. From the results, it can be analyzed that MOEA/D-AU is generally better than its predecessor MOEA/D, and also performs better than MOEA/D-DU on more than half of the tested instances. Above all, MOEA/D-AU can achieve a better convergence and diversity than MOEA/D, and also superior to MOEA/D-DU on some kinds of test problems.

To illustrate the distribution of solutions intuitively when in high-dimensional objective space, the final nondominated solutions of three compared algorithms (MOEA/D-AU, MOEA/D, MOEA/D-DU) are plotted in Fig. 6 by blue solid lines, and the 10-objective DTLZ2 and DTLZ4 problem are used. It can be seen from Fig. 6 that, MOEA/D-AU can produce a well distributed solution for all the objectives in the range of [0,1], while MOEA/D does not perform as well as MOEA/D-AU, and a little difference between the distribution of solutions on different number of objectives exists compared to MOEA/D-DU, but both of them (MOEA/D-AU, MOEA/D-DU) perform much better than MOEA/D.

Above all, it can be concluded that MOEA/D-AU can balance the diversity and convergence better than MOEA/D and also performs better than MOEA/D-DU on some instances.

Table 5: Comparison between MOEA/D-AU, MOEA/D and MOEA/D-DU in terms of average IGD and HV values on normalized test problems, the better average result is highlighted in boldface

Problem	m	IGD			HV		
		MOEA/D -AU	MOEA/D	MOEA/D -DU	MOEA/D -AU	MOEA/D	MOEA/D -DU
DTLZ1	2	**1.4394E-4**	1.4402E-4	1.4507E-4	0.703576	0.703885	**0.704351**
	5	3.5658E-4	3.6342E-5	**3.5578E-4**	1.575239	1.561590	**1.577323**
	8	**6.0148E-4**	7.1838E-4	6.2237E-4	**2.137589**	2.095139	2.128645
	10	**1.7650E-3**	2.7625E-2	2.8587E-3	**2.592379**	2.587712	2.591574
DTLZ2	2	**2.0847E-4**	2.0893E-4	2.0882E-4	**0.420157**	0.420127	0.420135
	5	2.3981E-4	2.5687E-4	**2.2119E-4**	**1.305522**	1.279779	1.304998
	8	**1.6448E-2**	1.7761E-2	3.0238E-2	**1.972904**	1.880641	1.949237
	10	**3.8572E-2**	4.2453E-2	6.8944E-2	**2.502553**	2.469218	2.497638
DTLZ3	2	**2.0475E-4**	2.0488E-4	2.0686E-4	**0.403398**	0.351673	0.394324
	5	2.3109E-3	**1.5870E-3**	2.8928E-3	1.221362	1.211054	**1.224016**
	8	3.7904E-2	**3.3577E-2**	4.0828E-2	1.915801	1.777359	**1.926168**
	10	**6.0233E-2**	6.2112E-2	9.5051E-2	2.421219	2.377561	**2.448847**
DTLZ4	2	2.0879E-4	2.1040E-4	**2.0866E-4**	**0.420122**	0.378781	0.419287
	5	3.9768E-4	4.8488E-4	**3.7068E-4**	1.306910	1.300439	**1.306962**
	8	1.5253E-2	**1.1993E-2**	2.0954E-2	**1.992520**	1.987186	1.988963
	10	**1.9637E-2**	2.2120E-2	2.5576E-2	**2.522784**	2.518759	2.515798

6.2. Comparison with the art-of-the-state Algorithms

In this section, the proposed algorithm is compared with several state-of-the-art algorithms on the normalized test problems. The experimental results of each compared algorithms on normalized DTLZ test instances are showed in Table 6, in terms of HV metrics. The experimental results on normalized WFG test instances are demonstrated in Table 7 . The significance test is conducted between MOEA/D-AU and other compared algorithms after that, which makes the conclusions statistical. In the two Tables mentioned above, the results outperformed by MOEA/D-AU significantly are marked by + symbol. A summary of significance test of HV results is provided too. $B(W)$ is used to illustrate which of the compared algorithms between Algorithm A and Algorithm B is better or worse, $B(W)$ illustrates the results of Algorithm A are clearly better(worse) than those of Algorithm B, E illustrates that the performances of two algorithms are almost equal.

From the results showed above, some observations for the proposed MOEA/D-AU can be obtained as follows.

1) A remarkable performance obtained by MOEA/D-AU is than MOEA/D

Table 6: Performances of each compared algorithms on normalized DTLZ problems with respect to the average HV values. The best value of HV for each instance is highlighted in boldface

problem	m	MOEA/D -AU	MOEA/D	MOEA/D -STM	MOEA/D -GR	MOEA/D -DU	MOEA/D -LWS
DTLZ1	2	0.703576	0.703885	0.703974	**0.704425**	0.704362	0.704285
	5	1.577439	1.561590	1.553199	1.477211	1.577231	**1.582678**
	8	**2.137589**$^{+}$	2.095139	2.072321	2.094542	2.128046	2.111530
	10	**2.592379**$^{+}$	2.587712	2.584715	2.587284	2.591873	2.589721
DTLZ2	2	**0.420157**	0.420127	0.420122	0.420124	0.420017	0.420146
	5	**1.305522**	1.279779	1.256105	1.305501	1.305173	1.305441
	8	**1.972904**$^{+}$	1.880641	1.802215	1.799789	1.949348	1.943843
	10	2.502553	2.469218	2.437517	2.428532	2.498727	**2.510820**
DTLZ3	2	0.403398	0.351673	0.404438	0.394488	0.395233	**0.405484**
	5	1.221326	1.211054	1.220321	1.023840	**1.223997**	1.219475
	8	1.915801	1.777359	1.731220	1.820581	**1.926278**	1.926178
	10	2.421219	2.377561	2.377549	2.304109	2.449936	**2.458930**
DTLZ4	2	0.420122	0.378781	0.409782	**0.420127**	0.419347	0.420021
	5	1.306910	1.300439	1.304318	1.307969	1.306871	**1.308204**
	8	**1.992520**	1.987186	1.980211	1.985271	1.989979	1.991407
	10	2.522784	2.518759	2.517571	2.519717	2.516887	**2.528468**
DTLZ7	2	0.543922	0.**543948**	0.543947	0.495959	0.490513	0.543879
	5	0.519442	**0.549150**	0.548519	0.535317	0.548002	0.518637
	8	**0.469972**$^{+}$	0.390228	0.409171	0.469485	0.410125	0.467182
	10	**0.498776**$^{+}$	0.459658	0.460769	0.316692	0.457183	0.484380

$^{+}$ means significantly better performs by MOEA/D-AU

Table 7: Performances of each compared algorithms on normalized WFG problems with respect to the average HV values. The best value of HV for each instance is highlighted in boldface

prblem	m	MOEA/D -AU	MOEA/D	MOEA/D -STM	MOEA/D -GR	MOEA/D -DU	MOEA/D -LWS
WFG1	2	0.565281	0.593810	**0.641742**	0.568670	0.561129	0.603897
	5	1.035103	1.306572	**1.349239**	0.465752	1.001012	1.341093
	8	1.478539	1.766690	**1.935879**	1.482639	1.439981	1.801757
	10	1.653310	2.442013	2.439171	1.292317	1.665329	**2.452282**
WFG2	2	0.751872	0.761350	**0.762649**	0.742648	0.740179	0.762524
	5	**1.601375**	1.571652	1.589302	1.464047	1.599979	1.600812
	8	2.136676	2.119647	2.121199	2.107629	2.138917	**2.150236**
	10	2.585979	2.578579	2.583829	2.583831	2.590625	**2.604648**
WFG3	2	0.700212	0.700158	**0.700703**	0.700283	0.700545	0.700510
	5	1.000252	0.867279	0.866015	0.906812	**1.049987**	0.954034
	8	1.246797	1.222792	1.227954	1.136275	**1.280019**	1.248375
	10	1.467273	1.458072	1.460349	1.489879	**1.532269**	1.511733
WFG4	2	0.418159	0.416904	0.417199	0.417182	0.416982	**0.420179**
	5	**1.283313**	0.920678	0.962503	0.792781	1.282939	1.095514
	8	1.922142	1.679864	1.659223	1.715842	1.933527	**2.078933**
	10	**2.480921**	2.232118	2.247538	2.369641	2.467571	2.477581
WFG5	2	**0.379065**	0.373093	0.378516	0.378372	0.377825	0.379030
	5	**1.221267**$^{+}$	1.173120	1.175927	0.798742	1.213023	1.212237
	8	**1.846841**$^{+}$	1.523772	1.499357	1.610529	1.808435	1.828526
	10	**2.317688**$^{+}$	2.044103	2.111799	2.110549	2.287861	2.299802
WFG6	2	0.387901	0.388268	0.388157	0.386991	0.385872	**0.407504**
	5	**1.210158**$^{+}$	0.932132	0.989177	0.745469	1.200253	1.034665
	8	1.787884	1.524039	1.514135	1.632451	1.791281	**1.911250**
	10	2.271243	1.963312	2.019167	2.106369	2.263856	**2.284506**
WFG7	2	0.419257	0.419269	0.419267	0.419268	0.418991	**0.421195**
	5	**1.284139**	1.134059	1.134062	0.843452	1.277899	1.282762
	8	1.907811	1.640167	1.640170	1.741278	1.925682	**1.933756**
	10	**2.461546**$^{+}$	2.329329	2.319593	2.315839	2.457684	2.448291
WFG8	2	**0.381609**	0.380672	0.380572	0.380462	0.379897	0.381021
	5	1.160235	0.620308	0.606447	0.687012	**1.173569**	0.712725
	8	1.707206	1.426564	1.416871	1.545538	1.752459	**1.926583**
	10	2.135564	1.770678	1.781840	1.925299	2.226917	**2.238662**
WFG9	2	0.394901	0.400665	0.387579	0.388452	0.394179	**0.413887**
	5	**1.253801**$^{+}$	0.658929	0.674281	0.677243	1.236991	0.716321
	8	**1.836068**	1.628031	1.623611	1.639321	1.818987	1.832640
	10	2.313448	2.062378	2.074167	2.134859	2.340615	**2.457252**

$^{+}$ means significantly better performs by MOEA/D-AU

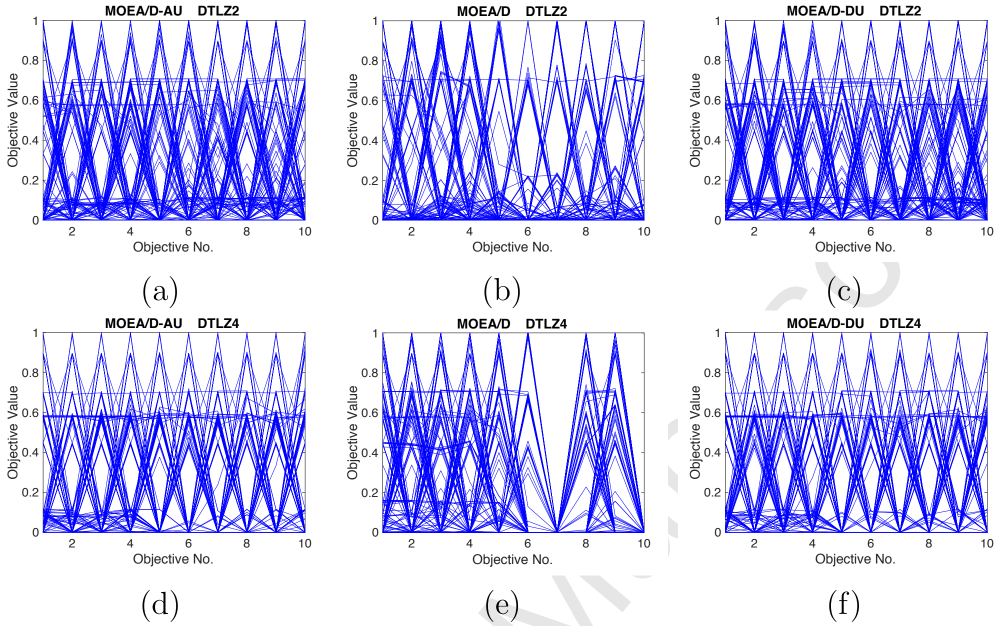
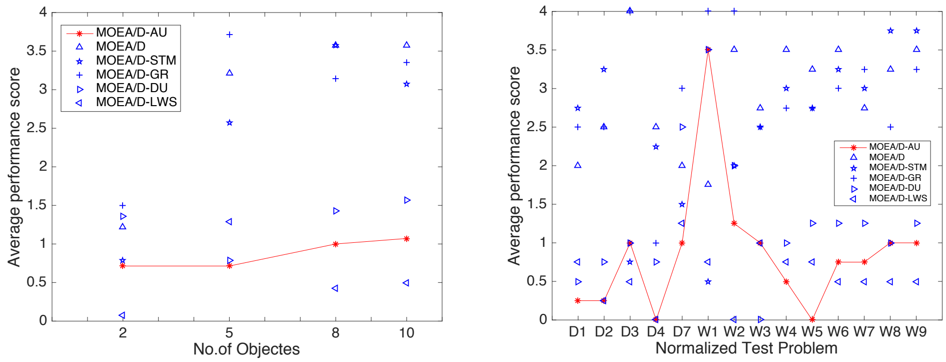


Figure 6: Final nondominated solution set of (a)DTLZ2 by MOED/-AU and (d)DTLZ4 by MOEA/D-AU, (b) DTLZ1 by MOEA/D and (e) DTLZ4 by MOEA/D, (c)DTLZ2 by MOEA/D-DU and (f)DTLZ4 by MOEA/D-DU on the normalized 10-objective instance, shown by parallel coordinates.

in 46 out of 56 test instances, which means MOEA/D-AU is distinctly superior to MOEA/D, and most of the instances outperformed by MOEA/D are the instances on 2-objective. The experimental results strongly point out that the updating strategy in our proposed algorithm is more effective for MaOPs compared to the one in MOEA/D.

2) Some superiorities have been showed by MOEA/D-AU over MOEA/D-STM and MOEA/D-GR. The performance of MOEA/D-AU is clearly better on most test instances, and also remains competitive on the lost instances. So it can be told that MOEA/D-AU is as comparative as these two state-of-the-art algorithms, even better at most cases.

3) Better performance than MOEA/D-DU has been obtained by MOEA/D-AU on DTLZ1, DTLZ2 and WFG4, WFG5, and perform as effective as MOEA/D-AU on DTLZ4, DTLZ7, WFG6, WFG7, WFG9, while the performance is a little worse on other test instances. It can be told that for some test problems, the updating strategy in MOEA/D-AU is more suitable than that in MOEA/D-DU, i.e., DTLZ1, DTLZ2, WFG4, WFG5.



(a) (b)

Figure 7: Average performance score got by MOEA/D-AU on all (a) dimensions for different normalized test problems, and (b) normalized test problems, Dx for DTLZ and Wx for WFG. The values of proposed MOEA/D-AU is connected by a red solid line.

4) Better performance has been obtained by MOEA/D-AU on most cases of DTLZ2, DTLZ7, WFG5, WFG7 compared to MOEA/D-LWS, and also performs as good as MOEA/D-LWS on some dimension of DTLZ1, DTLZ4, WFG4, WFG7, WFG9, but the performance of MOEA/D-AU is a little worse than MOEA/D-LWS on DTLZ3, WFG2, WFG6 and some cases of high dimension, i.e., dimension of 10.

Furthermore, the performance score [28] is introduced to measure the entire performance of all the algorithms compared, so that some options of the behaviors of these algorithms can be observed easily. For instance, l algorithms are compared in the experiments, i.e.,Alg_1, Alg_2,..., Alg_l, $\delta_{i,j}$ will be turned to 1 if ,Alg_j performs significantly better than Alg_i, according to the value of HV metric. Otherwise, it make s $\delta_{i,j}$ be 0. The performance score of algorithm $\mathrm{P}(\mathrm{Alg}_i)$ is defined as fellow

$$P(Alg_i) = \sum_{j=1, j \neq i}^{l} \delta_{i,j} \tag{13}$$

This value represents the number of algorithms which have significantly better performance than Alg_i on the entail tested instances, and smaller value means better performances of an algorithm.

From Fig. 7(a), the proposed algorithm MOEA/D-AU have small values on tested MaOPs. For 2-objective problems, MOEA/D-AU is slightly better

Table 8: Summary of the significance test of HV

		MOEA/D -AU	MOEA/D	MOEA/D -STM	MOEA/D -GR	MOEA/D -DU	MOEA/D -LWS
MOEA/D -AU vs	B	-	39	36	40	24	16
	W	-	7	7	4	15	22
	E	-	10	13	12	17	18

than MOEA/D and little worse than MOEA/D-LWS, almost all the compared algorithms also perform well on 2-objective test instances. MOEA/D-AU performs a little better than MOEA/D-DU on instances of 8- and 10-objectives, and has equal shares of instances on 5-objectives. The performance of MOEA/D-AU is a little worse than MOEA/D-LWS on 8- and 10-objectives, but is better on 5-objectives. It is obvious that not only MOEA/D but also MOEA/D-STM and MOEA/D-GR have a relative big score on the situation of more than three objectives, which means they do not perform well on high-dimensional problems.

From Fig. 7(b), MOEA/D-AU get the smallest score on DTLZ1, DTLZ2, DTLZ4, DTLZ7, WFG2, WFG4 and WFG5, which means it performs best on these instances, but the performance on WFG1 is unsatisfactory and a little worse than MOEA/D-DU on WFG3, and also performs a little worse than MOEA/D-LWS on some test problems. Although MOEA/D-AU not performs the best on some cases but it still has competitive performance compared to these art-of-the-state algorithms.

The average performance score of compared algorithms on all 56 instances is showed in Fig. 8. The proposed algorithm MOEA/D-AU ranks second, just a little worse than MOEA/D-LWS. It should be noticed that the proposed algorithm performs better than other algorithms on some instances tested not means it is better than these algorithms which performs not as well as MOEA/D-AU, no algorithm can always perform the best on all possible instances, there must be some algorithms which are more suitable than MOEA/D-AU to deal with some kinds of problems.

6.3. Comparison on Scaled Test Problems

In this section, the normalization procedure introduced in Section 3.3 is employed into the proposed MOEA/D-AU. In order to investigate the performance of MOEA/D-AU on scaled test problems, NSGA-III is chosen to be compared on the group of scaled test problems.

Table 9: Performances of MOEA/D-AU and NSGA-III on scaled test problems with respect to average HV values. The best average HV value for each instance is highlighted boldface

problem	m	MOEA/D-AU	NSGA-III
DTLZ1	2	**0.703570**	0.701756
	5	**1.577358**	1.572782
	8	**2.138478**	2.129725
	10	**2.592751**	2.585657
DTLZ2	2	**0.420142**	0.419948
	5	**1.305242**	1.297455
	8	1.970127	**1.970348**
	10	2.492199	**2.505279**
WFG4	2	0.417159	**0.417908**
	5	**1.281324**	1.265781
	8	**1.912489**	1.879854
	10	**2.481879**	2.442646
WFG5	2	0.378365	**0.378761**
	5	**1.222047**	1.213638
	8	**1.846758**	1.820268
	10	2.318691	**2.327392**
WFG6	2	**0.388012**	0.386412
	5	**1.211089**	1.201123
	8	1.715486	**1.803971**
	10	2.465761	**2.300159**
WFG7	2	**0.419354**	0.418767
	5	1.162337	**1.278557**
	8	1.715486	**1.920519**
	10	2.465760	**2.474062**
WFG8	2	**0.382645**	0.379502
	5	1.162143	**1.163528**
	8	**1.699150**	1.679487
	10	2.135761	**2.214109**
WFG9	2	0.395812	**0.401292**
	5	**1.251201**	1.179842
	8	**1.840058**	1.728558
	10	**2.304254**	2.274671

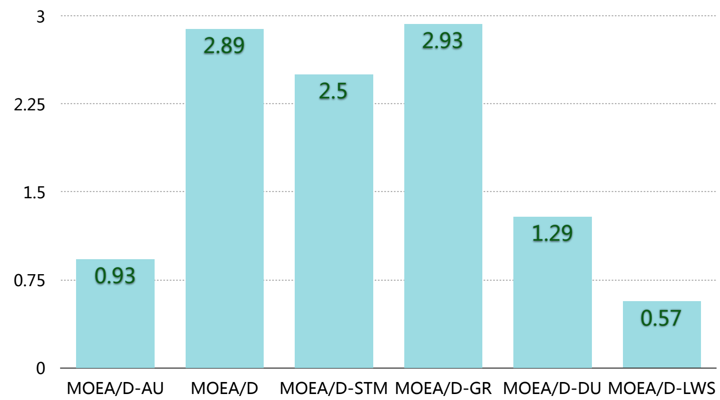


Figure 8: The score of average performance obtained by each compared algorithms

The average HV results of MOEA/D-AU and NSGA-III are showed in Table 9. It can be seen that MOEA/D-AU have equal shares with NSGA-III according to the problems tested or even better in some case. MOEA/D-AU performs better on DTLZ1, WFG4 and WFG9. While we deal with problems with 10-objective, NSGA-III wins in 5 out of 8 instances.

The trace of evolutionary for proposed MOEA/D-AU and NSGA-III on 8-objective WFG9 instance is showed in Fig. 9, by the value of HV metric. In Fig. 9, the values of HV obtained by both algorithms increase stably with the number of generation grows, and MOEA/D-AU converge to the PF a little faster than NSGA-III The final HV obtained by MOEA/D-AU is also higher than NSGA-III.

6.4. Discussion

In this section, some relevant issues are discussed according to the experimental results.

The first issue to be discussed is the setting of population size. With the increasing number of the dimensional, the points which are needed to represent an entail PF are increasing in a geometrical progression. Thus, a large population size will be inevitable if an entire PF needs to be approxi-

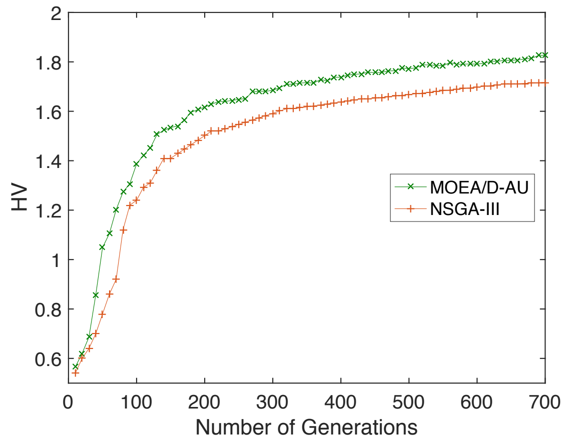


Figure 9: The score of average performance obtained by each compared algorithms

mated. But just as mentioned in [39], to comprehend and select a preferred solution is tough work in this scenario, and a large population size may make the computation of an algorithm become inefficient or even impractical. For instance, 100 is the population size used in 2-objective cases, so if the same dense distribution needs to be kept in 8-objective cases, a population size of 10^8 will be inevitable. However, to evolve such a number of population is unaffordable and unpractical. Therefore, as illustrated in [56], the nondominated solutions are distributed sparsely over the entire PF, and searched by globally search method which is one of the feasible ways in the moment. In this paper, the setting of population size in our experiment is showed in Table 3 , which is similar as most existing studies on MaOPs (see [20], [39]).

The second issue to be discussed is what leads to the performance of MOEA/D-AU is not significantly better than other compared algorithms on 2-objective problems, although the performances obtained by MOEA/D-AU are much better on other dimensions. Not like the performance on many-objective ones, when the number of objectives is 2, just a normal population size is used, e.g., 100, a very dense distribution of solutions can be got, which means it can capture almost every regions of PF in this situation for all the tested algorithms even including MOEA/D. But when we deal with the problems having high dimensional objective space, more degrees of freedom appear, it is much more likely to produce misleading solutions which means the solutions are far from the weight vector if only the aggregation function is considered. So the angle-based updating strategy and ranking restriction scheme proposed in MOEA/D-AU will play a role in dealing with the misleading solutions and maintain a good distribution of solutions in the evolutionary process.

The third issue is why the other three similar algorithms, i.e., MOEA/D-STM, MOEA/D-GR and MOEA/D-DU do not performs as well as MOEA/D-AU on the test instances they lost. For MOEA/D-STM, a matching algorithm is used to coordinate the preferences of solutions and subproblems, but as a result of using sparsely distributed solutions, missing track of solutions in some regions would be much more likely to appears when in high-dimensional objective space, so the preference of solutions may be needed a bias in this case. The experimental verification can be found in the study of the influence of parameter G in MOEA/D-AU, where a small value near 5 is suitable. For MOEA/D-GR, when in high-dimensional objective space, the weight vectors are most likely far from each other, so to allocated a subproblem for a new solution only by the value of aggregation function of subproblems may not be enough, just like the problems motioned at the beginning of Section 3. For MOEA/D-DU, it performs as equally as MOEA/D-AU on many test instances and even better on some test problems, like WFG3 and WFG8, but there are also worse performances than MOEA/D-AU on DTLZ2, WFG4, WFG5. The reason can be attributed to the different measurement between two algorithms in deciding which solution is closer to the direction vector, and different measurements suit different problems.

The fourth issue is why the proposed algorithm MOEA/D-AU performs a little worse on some of the test instances, especially the instances having high dimension. This may because of the different decomposition approaches used in MOEA/D-AU and MOEA/D-LWS. In MOEA/D-AU the Tchebycheff method is used whereas the localized weight sum method is used in MOEA/D-LWS. For localized weight sum method, the convergence ability is better than the one of Tchebycheff. Apart from that, the search ability of localized weight sum method will not be affected by the increasing number of objectives whereas the Tchebycheff method will.

The last issue which should be discussed is the setting of parameter G in MOEA/D-AU. As discussed in Section 5, the most suitable value of G depends on the problem instance to be solved. Different problem instance needs different parameter G to achieve the best performance. Indeed, G plays a key role in controlling the balance between diversity and convergence. The convergence is encouraged when a large value of G is used because the area of selections becomes big and the value of aggregation function plays a more important role in the selection procedure. When a small value of G is used, $\theta^i(x)$ is more emphasized and the diversity is promoted. So there must be an optimal value of G which can balance the diversity and convergence well for a specific instance.

7. Conclusions and Future work

In this paper, algorithm MOEA/D-AU is proposed to deal with the many-objective optimizations which enhance MOEA/D, and performs well on the test problems. The basic idea proposed in MOEA/D-AU is to maintain the distribution of solutions by using the acute angle between a solution and a direction vector in the objective space during the evolutionary process. In detail, when a new solution is generated in MOEA/D-AU, the updating strategy determines G closest direction vectors to this new solution in terms of the acute angle, and then the old solutions corresponding to the G direction vectors will be updated by this new solution according to the aggregation function values.

In the experimental studies, for the performance of MOEA/D-AU is mainly controlled by parameter G, so the investigation of the influence of the setting of parameter G has been done at first and some suggestions are given to set G suitably. It also can be seen that MOEA/D-AU shows a better ability in balancing the diversity and convergence compared to its predecessor, i.e., MOEA/D. In order to demonstrate the competitiveness, MOEA/D-AU is compared to five state-of-the-art MOEAs on a total of 56 test problems including DTLZ and WFG test suites, and the number of objectives is up to 10. The results indicate that MOEA/D-AU has an obvious better performance than MOEA/D-STM and MOEA/D-GR on most test instances, and performs as well or even better as MOEA/D-DU on two-thirds of the test instances, also performs as equal or even better as MOEA/D-LWS on more than half of the test instances. At the end of experiments, MOEA/D-AU is compared with NSGA-III on scaled test problems, and the results show that MOEA/D-AU with the normalization procedure performs as effectively as NSGA-III in dealing with different scales of objectives.

In the future, how to adjust the parameter G dynamically in the evaluate process will be studied, and some other search strategies may also be helpful if used in MOEA/D-AU.

Acknowledgements

This work was supported by the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period under Grant 2012BAD10B01 and National Natural Science Foundation of China under Grant 61572438, 61379123.

References

[1] Kalyanmoy Deb, . Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc; 2001.

[2] Miettinen, K.. Nonlinear Multiobjective Optimization; vol. 12 of *International Series in Operations Research & Management Science*. Boston, MA: Springer US; 1999. ISBN 978-1-4613-7544-9. URL: `http://link.springer.com/10.1007/978-1-4615-5563-6`. doi:`10.1007/978-1-4615-5563-6`.

[3] Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 2011;1(1):32–49. URL: `http://linkinghub.elsevier.com/retrieve/pii/S2210650211000058`. doi:`10.1016/j.swevo.2011.03.001`.

[4] Zheng, Y.J., Ling, H.F., Xue, J.Y., Chen, S.Y.. Population Classification in Fire Evacuation: A Multiobjective Particle Swarm Optimization Approach. IEEE Transactions on Evolutionary Computation 2014;18(1):70–81. URL: `http://ieeexplore.ieee.org/document/6595531/`. doi:`10.1109/TEVC.2013.2281396`.

[5] Zheng, Y.J., Chen, S.Y., Ling, H.F.. Evolutionary optimization for disaster relief operations: A survey. Applied Soft Computing 2015;27:553–566. URL: `http://linkinghub.elsevier.com/retrieve/pii/S1568494614004943`. doi:`10.1016/j.asoc.2014.09.041`.

[6] Ishibuchi, H., Tsukamoto, N., Nojima, Y.. Evolutionary many-objective optimization. In: 2008 3rd International Workshop on Genetic and Evolving Systems. IEEE. ISBN 978-1-4244-1612-7; 2008, p. 47–52. URL: `http://ieeexplore.ieee.org/document/4484566/`. doi:`10.1109/GEFS.2008.4484566`.

[7] Wagner, T., Beume, N., Naujoks, B.. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In: Evolutionary Multi-Criterion Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 742–756. URL: http://link.springer.com/10.1007/978-3-540-70928-2{_}56. doi:10.1007/978-3-540-70928-2{_}56.

[8] Zitzler, E., Laumanns, M., Thiele, L.. SPEA2: Improving the strength Pareto evolutionary algorithm. Swarm and Evolutionary Computation 2002;:98–100doi:10.1.1.28.7571.

[9] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002;6(2):182–197. URL: http://ieeexplore.ieee.org/document/996017/. doi:10.1109/4235.996017.

[10] Ikeda, K., Kita, H., Kobayashi, S.. Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546); vol. 2. IEEE. ISBN 0-7803-6657-3; 2001, p. 957–962. URL: http://ieeexplore.ieee.org/document/934293/. doi:10.1109/CEC.2001.934293.

[11] Khare, V., Yao, X., Deb, K.. Performance Scaling of Multi-objective Evolutionary Algorithms. 2003, p. 376–390. URL: http://link.springer.com/10.1007/3-540-36970-8{_}27. doi:10.1007/3-540-36970-8{_}27.

[12] Purshouse, R.C., Fleming, P.J.. On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Computation 2007;11(6):770–784. URL: http://ieeexplore.ieee.org/document/4384508/. doi:10.1109/TEVC.2007.910138.

[13] Adra, S.F., Fleming, P.J.. Diversity Management in Evolutionary Many-Objective Optimization. IEEE Transactions on Evolutionary Computation 2011;15(2):183–195. URL: http://ieeexplore.ieee.org/document/6782674/http://ieeexplore.ieee.org/document/5675670/. doi:10.1109/TEVC.2010.2058117.

[14] Miqing Li, , Shengxiang Yang, , Xiaohui Liu, . Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization. IEEE Transactions on Cybernetics 2014;44(12):2568–2584. URL: http://ieeexplore.ieee.org/document/6782674/. doi:10.1109/TCYB.2014.2310651.

[15] Laumanns, M., Thiele, L., Deb, K., Zitzler, E.. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation 2002;10(3):263–282. URL: http://www.mitpressjournals.org/doi/10.1162/106365602760234108. doi:10.1162/106365602760234108.

[16] Hadka, D., Reed, P.. Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework. Evolutionary Computation 2013;21(2):231–259. URL: http://www.mitpressjournals.org/doi/10.1162/EVCO{_}a{_}00075. doi:10.1162/EVCO{_}a{_}00075.

[17] di Pierro, F., Khu, S.T., Savi, D.A.. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. IEEE Transactions on Evolutionary Computation 2007;11(1):17–45. URL: http://ieeexplore.ieee.org/document/4079613/. doi:10.1109/TEVC.2006.876362.

[18] Wang, G., Jiang, H.. Fuzzy-Dominance and Its Application in Evolutionary Many Objective Optimization. In: 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007). IEEE. ISBN 0-7695-3073-7; 2007, p. 195–198. URL: http://ieeexplore.ieee.org/document/4425478/. doi:10.1109/CISW.2007.4425478.

[19] Li, L., Wang, W., Xu, X.. Multi-objective particle swarm optimization based on global margin ranking. Information Sciences 2017;375:30–47. URL: http://linkinghub.elsevier.com/retrieve/pii/S0020025516306156. doi:10.1016/j.ins.2016.08.043.

[20] Yang, S., Li, M., Liu, X., Zheng, J.. A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation 2013;17(5):721–736. URL: http://ieeexplore.ieee.org/document/6400243/. doi:10.1109/TEVC.2012.2227145.

[21] Liang, Z., Song, R., Lin, Q., Du, Z., Chen, J., Ming, Z., et al. A double-module immune algorithm for multi-objective optimization problems. Applied Soft Computing 2015;35:161–174. URL: http://linkinghub.elsevier.com/retrieve/pii/S1568494615003713. doi:10.1016/j.asoc.2015.06.022.

[22] Qi, Y., Liu, F., Liu, M., Gong, M., Jiao, L.. Multi-objective immune algorithm with Baldwinian learning. Applied Soft Computing 2012;12(8):2654–2674. URL: http://linkinghub.elsevier.com/retrieve/pii/S1568494612001792. doi:10.1016/j.asoc.2012.04.005.

[23] Dai, C., Wang, Y.. A new decomposition based evolutionary algorithm with uniform designs for many-objective optimization. Applied Soft Computing 2015;30:238–248. URL: http://linkinghub.elsevier.com/retrieve/pii/S1568494615000836. doi:10.1016/j.asoc.2015.01.062.

[24] Lopez, A., Coello, C.A.C., Oyama, A., Fujii, K.. An Alternative Preference Relation to Deal with Many-Objective Optimization Problems. 2013, p. 291–306. URL: http://link.springer.com/10.1007/978-3-642-37140-0{_}24. doi:10.1007/978-3-642-37140-0{_}24.

[25] Zhang, X., Tian, Y., Jin, Y.. A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation 2015;19(6):761–776. URL: http://ieeexplore.ieee.org/document/6975108/. doi:10.1109/TEVC.2014.2378512.

[26] Fleischer, M.. The Measure of Pareto Optima Applications to Multi-objective Metaheuristics. 2003, p. 519–533. URL: http://link.springer.com/10.1007/3-540-36970-8{_}37. doi:10.1007/3-540-36970-8{_}37.

[27] Li, K., Kwong, S., Cao, J., Li, M., Zheng, J., Shen, R.. Achieving balance between proximity and diversity in multi-objective evolutionary algorithm. Information Sciences 2012;182(1):220–242. URL: http://linkinghub.elsevier.com/retrieve/pii/S0020025511004464. doi:10.1016/j.ins.2011.08.027.

[28] Bader, J., Zitzler, E.. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation 2011;19(1):45–76. URL: http://www.mitpressjournals.org/doi/10.1162/EVCO{_}a{_}00009. doi:10.1162/EVCO{_}a{_}00009.

[29] Beume, N., Naujoks, B., Emmerich, M.. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 2007;181(3):1653–1669. URL: http://linkinghub.elsevier.com/retrieve/pii/S0377221706005443. doi:10.1016/j.ejor.2006.08.008.

[30] Igel, C., Hansen, N., Roth, S.. Covariance Matrix Adaptation for Multi-objective Optimization. Evolutionary Computation 2007;15(1):1–28. URL: http://www.mitpressjournals.org/doi/10.1162/evco.2007.15.1.1. doi:10.1162/evco.2007.15.1.1.

[31] Brockhoff, D., Wagner, T., Trautmann, H.. On the properties of the R2 indicator. In: Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12. New York, New York, USA: ACM Press. ISBN 9781450311779; 2012, p. 465. URL: http://dl.acm.org/citation.cfm?doid=2330163.2330230. doi:10.1145/2330163.2330230.

[32] Schutze, O., Esquivel, X., Lara, A., Coello, C.A.C.. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 2012;16(4):504–522. URL: http://ieeexplore.ieee.org/document/6151115/. doi:10.1109/TEVC.2011.2161872.

[33] Trautmann, H., Wagner, T., Brockhoff, D.. R2-EMOA: Focused Multiobjective Search Using R2-Indicator-Based Selection. 2013, p. 70–74. URL: http://link.springer.com/10.1007/978-3-642-44973-4{_}8. doi:10.1007/978-3-642-44973-4{_}8.

[34] Gomez, R.H., Coello, C.A.C.. MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator. In: 2013 IEEE Congress on Evolutionary Computation. IEEE. ISBN 978-1-4799-0454-9; 2013, p. 2488–2495. URL: http://ieeexplore.ieee.org/document/6557868/. doi:10.1109/CEC.2013.6557868.

[35] Wang, R., Purshouse, R.C., Fleming, P.J.. Preference-Inspired Coevolutionary Algorithms for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation 2013;17(4):474–494. URL: http://ieeexplore.ieee.org/document/6215034/. doi:10.1109/TEVC.2012.2204264.

[36] Wang, R., Purshouse, R.C., Fleming, P.J.. Preference-inspired coevolutionary algorithm using weights for many-objective optimization. In: Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion - GECCO '13 Companion. New York, New York, USA: ACM Press. ISBN 9781450319645; 2013, p. 101. URL: http://dl.acm.org/citation.cfm?doid=2464576.2464622. doi:10.1145/2464576.2464622.

[37] Qingfu Zhang, , Hui Li, . MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 2007;11(6):712–731. URL: http://ieeexplore.ieee.org/document/4358754/. doi:10.1109/TEVC.2007.892759.

[38] Hui Li, Q.Z.. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 2009;13(2):284–302. doi:10.1109/TEVC.2008.925798.

[39] Deb, K., Jain, H.. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Transactions on Evolutionary Computation 2014;18(4):577–601. URL: http://ieeexplore.ieee.org/document/6600851/. doi:10.1109/TEVC.2013.2281535.

[40] Zhao, S.Z., Suganthan, P.N., Zhang, Q.. Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes. IEEE Transactions on Evolutionary Computation 2012;16(3):442–446. URL: http://ieeexplore.ieee.org/document/6151117/. doi:10.1109/TEVC.2011.2166159.

[41] Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., Wu, J.. MOEA/D with Adaptive Weight Adjustment. Evolutionary Computation 2014;22(2):231–264. URL: http://www.mitpressjournals.org/doi/10.1162/EVCO{_}a{_}00109. doi:10.1162/EVCO{_}a{_}00109.

[42] Jiang, S., Yang, S.. An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts. IEEE Transactions on Cybernetics 2016;46(2):421–437. URL: http://ieeexplore.ieee.org/document/7060668/. doi:10.1109/TCYB.2015.2403131.

[43] Liu, H.L., Gu, F., Zhang, Q.. Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems. IEEE Transactions on Evolutionary Computation 2014;18(3):450–455. URL: http://ieeexplore.ieee.org/document/6595549/. doi:10.1109/TEVC.2013.2281533.

[44] Yuan, Y., Xu, H., Wang, B., Zhang, B., Yao, X.. Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers. IEEE Transactions on Evolutionary Computation 2016;20(2):180–198. URL: http://ieeexplore.ieee.org/document/7120115/. doi:10.1109/TEVC.2015.2443001.

[45] Wang, R., Zhang, Q., Zhang, T.. Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods. IEEE Transactions on Evolutionary Computation 2016;20(6):821–837. URL: http://ieeexplore.ieee.org/document/7390047/. doi:10.1109/TEVC.2016.2521175.

[46] Wang, R., Ishibuchi, H., Yan Zhang, , Xiaokun Zheng, , Zhang, T.. On the effect of localized PBI method in MOEA/D for multi-objective optimization. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. ISBN 978-1-5090-4240-1; 2016, p. 1–8. URL: http://ieeexplore.ieee.org/document/7850222/. doi:10.1109/SSCI.2016.7850222.

[47] Ke Li, , Qingfu Zhang, , Sam Kwong, , Miqing Li, , Ran Wang, . Stable Matching-Based Selection in Evolutionary Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 2014;18(6):909–923. URL: http://ieeexplore.ieee.org/document/6678563/. doi:10.1109/TEVC.2013.2293776.

[48] Luping Wang,Qingfu Zhang, A.Z.. Constrained Subproblems in a Decomposition- Based Multiobjective Evolutionary Algorithm. IEEE Transactions on Evolutionary Computation 2016;20(3):475 – 480. doi:10.1109/TEVC.2015.2457616.

[49] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.. Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600); vol. 1. IEEE. ISBN 0-7803-7282-4; ????, p. 825–830. URL: http://ieeexplore.ieee.org/document/1007032/. doi:10.1109/CEC.2002.1007032.

[50] Huband, S., Hingston, P., Barone, L., While, L.. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 2006;10(5):477–506. URL: http://ieeexplore.ieee.org/document/1705400/. doi:10.1109/TEVC.2005.861417.

[51] J. Knowles, L. Thiele, E.Z.. A tutorial on the performance assessment of stochastic multiobjective optimizers. EMO 2006;:327–332URL: http://link.springer.com/10.1007/3-540-61723-X{_}1022http://ieeexplore.ieee.org/document/5766730/.

[52] While, L., Bradstreet, L., Barone, L.. A Fast Way of Calculating Exact Hypervolumes. IEEE Transactions on Evolutionary Computation 2012;16(1):86–95. URL: http://ieeexplore.ieee.org/document/5766730/. doi:10.1109/TEVC.2010.2077298.

[53] Wang, Z., Zhang, Q., Gong, M., Zhou, A.. A replacement strategy for balancing convergence and diversity in MOEA/D. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE. ISBN 978-1-4799-1488-3; 2014, p. 2132–2139. URL: http://ieeexplore.ieee.org/document/6900319/. doi:10.1109/CEC.2014.6900319.

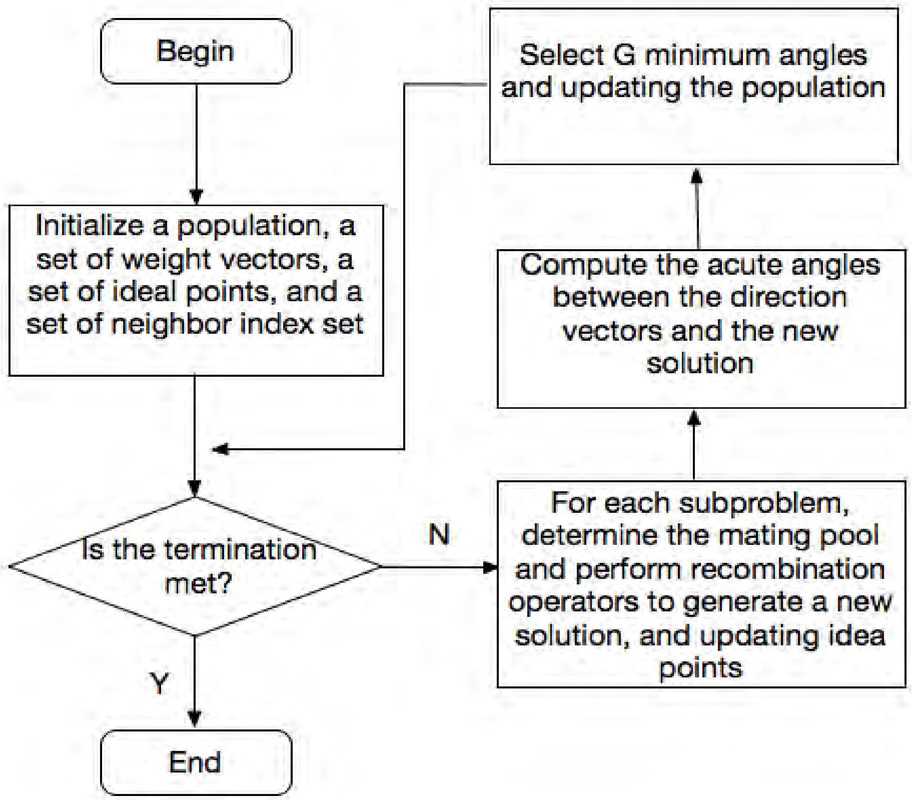
[54] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., Zhang, T.. Localized weighted sum method for many-objective optimization. IEEE Transactions on Evolutionary Computation 2016;:1–1URL: http://ieeexplore.ieee.org/document/7572016/. doi:10.1109/TEVC.2016.2611642.

[55] Durillo, J.J., Nebro, A.J.. jMetal: A Java framework for multi-objective optimization. Advances in Engineering Software 2011;42(10):760–771. URL: http://linkinghub.elsevier.com/retrieve/pii/S0965997811001219. doi:10.1016/j.advengsoft.2011.05.014.

[56] Ishibuchi, H., Akedo, N., Nojima, Y.. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. IEEE Transactions on Evolutionary Computation 2015;19(2):264–283. URL: http://ieeexplore.ieee.org/document/6782742/. doi:10.1109/TEVC.2014.2315442.

Highlights:

- A measure exploits the acute angle between a solution and a direction vector is proposed.
- A famous algorithm MOEA/D is improved through the updating strategy.
- An enhanced version of MOEA/D called MOEA/D-AU is proposed.



This above figure illustrates the flowchart of the proposed decomposition-based multiobjective evolutionary algorithm with an angle-based updating strategy (MOEA/D-AU). Its basic procedure is similar to most generational multi-objective evolutionary algorithms. First, an initial population is formed by randomly generating individuals. Then, for each subproblem, genetic operators are performed to obtain an offspring population. Next, the acute angles to each direction vector are computed respectively, G minimum angles are selected, the new solution is compared to G solutions corresponding to the G minimum angles. Finally, superior solutions are selected according to the value of aggregation function to update the parent population. It can be seen that there are two key operators in MOEA/D-AU: computation the acute angle between a solution and a direction vector, rank the angles and select G minimum angles. In this study, an acute angle between a solution and a direction vector is used to keep the solution close to the direction vectors, and we select G minimum angles to update the population, as a result, a good balance between convergence and direction is achieved.