An Evolutionary Many Objective Optimisation Algorithm with Adaptive Region Decomposition

Hai-Lin Liu, Lei Chen, Qingfu Zhang, *Senior Member, IEEE* and Kalyanmoy Deb, *Fellow, IEEE*

COIN Report Number 2016014

Abstract—**When optimizing an multiobjective optimization problem, the evolution of population can be regarded as a approximation to the Pareto Front (PF). Motivated by this idea, we propose an adaptive region decomposition framework: MOEA/D-AM2M for the degenerated Many objective optimization problem (MaOP), where degenerated MaOP refers to the optimization problem with a degenerated PF in a subspace of the objective space. In this framework, a complex MaOP can be adaptively decomposed into a number of many-objective optimization subproblems, which is realized by the adaptively direction vectors design according to the present population's distribution. A new adaptive weight vectors design method based on this adaptive region decomposition is also proposed for selection in MOEA/D-AM2M. This strategy can timely adjust the regions and weights according to the population's tendency in the evolutionary process, which serves as a remedy for the inefficiency of fixed and evenly distributed weights when solving MaOP with a degenerated PF. Five degenerated MaOPs with disconnected PFs are generated to identify the effectiveness of proposed MOEA/D-AM2M. Contrast experiments are conducted by optimizing those MaOPs using MOEA/D-AM2M, MOEA/D-DE and MOEA/D-M2M. Simulation results have shown that the proposed MOEA/D-AM2M outperforms MOEA/D-DE and MOEA/D-M2M.**

I. Introduction

Many objective optimisation problems (MaOPs) have recently attracted much attention from the evolutionary computation community due to its great importance in real world applications [?]. Nowadays, more and more EMO algorithms have been specially designed for MaOP optimization. Some use modified Pareto domination relationship to decrease the proportion of nondominance solutions in a population in MaOP optimization, such as ϵ dominance [?], [?], subspace dominance [?], fuzzy dominance [?], L-optimality [?], Grid method [?] and preference order ranking [?]. Some use linear dimensionality or nonlinear dimensionality reduction algorithms to reduce the dimension of search space. Many dimensionality reduction algorithms in machine learning, such as PCA [?] have been successfully used for MaOP with the so-called redundant objectives. Saxena and Deb present a framework for both linear (L-PCA) and nonlinear objective (NL-MVU-PCA) reduction algorithms in [?]. Some design a indicator to measure the quality of population, such as hypervolume measure based EMO algorithms [?], [?], [?]. Although studies have shown that Hypervolume based EMO algorithms might be more suitable for MaOP, its huge computational complexity hampers its practical applications. The most recent studies about Hypervolume indicator are almost all about how to fast calculate the Hypervolume [?], [?]. The others are weights aided EMO algorithms, and weight vectors are used to either decompose (MOEA/D) [?] or enhance population diversity and convergence (NSGA-III) [?].

Many real-life MaOPs may have redundant objectives and its optimal solution set is of very low dimensionality. We call these problems degenerated. For these problems, one can find its essential objectives in an offline manner and then reduce it to a problem with two or three objectives as conducted in [?]. Another possible way to deal with degenerated problems is to dynamically identify important objectives and promising search region for guiding the allocation of search effort to the search space [?], [?]. This paper also attempts to address this kind of problems along this direction.

MOEA/D-M2M decomposes a multiobjective optimisation problem into a number of simple multiobjective optimisation subproblems [?]. In MOEA/D-M2M, all the subproblems have the same objectives as the original problem and the feasible region of each subproblem is different. The Pareto sets of all the subproblems collectively form the Pareto set of the original problem. The feasible region of each subproblem is defined by a direction vector (or weight vector) in the objective space. Like other MOEA/D variants, each search procedure optimizes a different suproblem and these procedures exchange information for improving their search efficiency.

Under the MOEA/D-M2M framework, this paper proposes an adaptive decomposition method, called MOEA/D-AM2M, for solving degenerated problems. During each generation, MOEA/D-AM2M selects from the current population representative solutions and then uses them to define N subregions, each of which will be as the feasible region for a different subproblem. In such a way, the search can focus on more promising areas.

Five degenerated MaOP test instances are constructed and used to study the algorithm performance. We have compared MOEA/D-AM2M with other MOEA/D variants on these instances.

The remainder of paper is organized as follows: Section II gives the definition of the degenerated MaOP, and discusses the motivation behind MOEA/D-AM2M. Section III describes the main framework of MOEA/D-AM2M. In section IV, the contrast experiments are conducted, and the simulation results are shown and analyzed. Section V concludes this paper.

II. Definition and Motivation

In general, a continuous multiobjective optimization problem (MOP) has the PF [?] with the dimensionality one less than the number of objectives. If the dimension of a MaOP with m objectives is less than $m-1$, we call this optimization problem a degenerated MaOP. Under mild conditions, degenerated MaOP can be regarded as MaOP with high dimensional PSs and low dimension PFs. This paper focuses on the optimization of degenerated MaOPs. In such problems, it is possible to use a small set of solutions to approximate their optimal set. MOEA/D-M2M is a population decomposition strategy which can decompose a MOP into a set of simple multiobjective optimization subproblems. In this part, we show if population are divided in a proper way and weight vectors are selected in a right way, then MOEA/D-M2M can solve the problem we want to solve. Otherwise, it may not work well.

A. Behavior of MOEA/D-M2M

In MOEA/D-M2M, a set of direction vectors need to be defined in the initialization, and after the definition of the direction vectors, subregions can be determined and fixed. A set of evenly distributed weight vectors also need to be initialized for Tchebycheff selection in MOEA/D-M2M. This method can be effective for MOP with a PF full of the objective space. However, when it comes to MOPs with complex distributions PFs in the objective space, evenly distributed direction vectors can be unreasonable.

If the PF's distribution of MOP can be estimated in advance, we can design direction vectors according to its distribution and thus make reasonable region decomposition. Likewise, a pre-known PF can also facilitate the design of weight vectors for selection. A test instance with a disconnected PF is used to illustrate this idea.

Test instance 1:

$$\text{Min:} \begin{cases} f_1(\mathbf{x}) = (1 + g(x_3, x_4, ..., x_{10}) + h(x_2))x_1x_2 \\ f_2(\mathbf{x}) = (1 + g(x_3, x_4, ..., x_{10}) + h(x_2))(1 - x_1)x_2 \\ f_3(\mathbf{x}) = (1 + g(x_3, x_4, ..., x_{10}) + h(x_2))(1 - x_2) \end{cases}$$

where

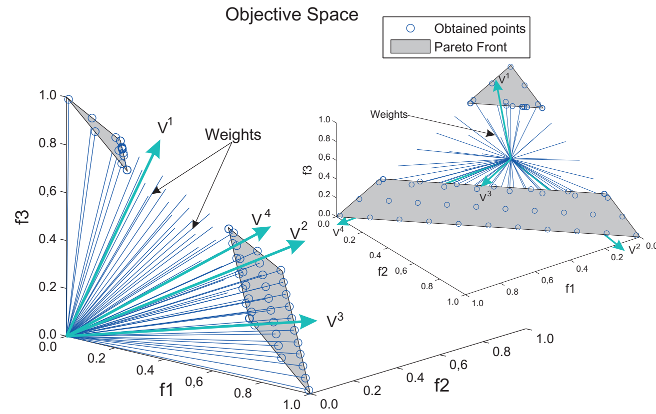
$\mathbf{x} = (x_1, x_2, ..., x_{10}), x_i \in [0, 1], i = 1, \ldots, 10,$

$g(x_3, x_4, ..., x_{10}) = \sum_{i=3}^{10} (x_i - 0.5)^2,$

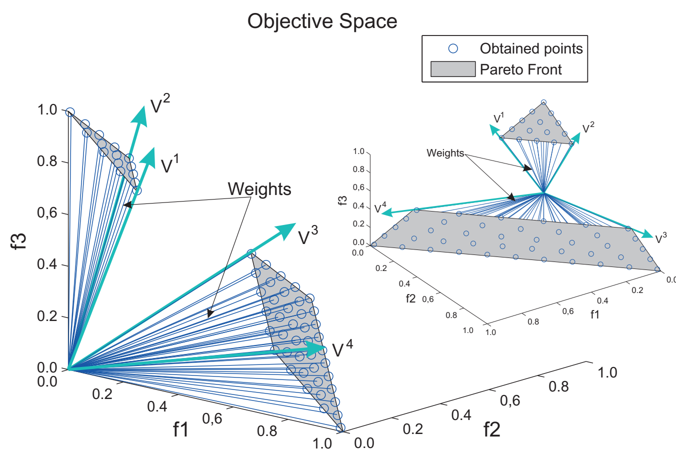
$h(x_2) = \max(0, -\cos(2\pi x_2)).$

Function $h(x_2)$ pushes all the objective values to have larger values for $x_2 \in [0.25, 0.75]$, thereby making these x_2 value to be dominated by $x_2 = 0.25$ and $x_2 = 0.75$ solutions. The above causes a gap within the original linear Pareto Front.

Figure **??**(a) shows the results obtained by MOEA/D-M2M with evenly distributed direction vectors and weight vectors, while Figure **??**(b) shows the results obtained by MOEA/D-M2M with designed direction vectors and weight vectors according to the pre-known PF. From the figure, we can see that MOEA/D-M2M with direction and weight vectors designed according to the PF's distribution has better diversity across the PF than that with evenly distributed direction and weight vectors. It is mainly because the almost every weight corresponds to a PF point in Figure **??**(b), while, in Figure **??**(a), the weight vectors with no intersection are more likely to get solutions on the PF boundary. Weights are designed to maintain the population diversity in weights aided EMO algorithms, which means that some weights are wasted in terms of diversing the population. For MaOP with a degenerated PF,



(a) Simulation results obtained by MOEA/D-M2M with evenly distributed direction/weight vectors for test instance 1



(b) Simulation results obtained by MOEA/D-M2M with designed direction/weight vectors for test instance 1

Fig. 1. Illustration of the weight vectors' influence to the final obtained solutions.

this waste can be very serious if evenly distributed weights are used because of its huge search space. The waste of weights should be avoided for it wastes the precious computing resources, and we can avoid it by reasonable weights and direction vector design. It seems that generating direction and weight vectors according to the PF's distribution is a very efficient remedy for the EMO algorithms' failures caused by the even and fixed direction and weight vectors. However, the truth is that a MOP's PF may be unknown before optimizing it. Most of the weight vectors aided EMO algorithms have no alternatives but to adopt the evenly distributed weight vectors design.

B. Information Extraction From the Evolving Population

When optimizing a MaOP, the evolutionary process can be seen as an approximation to the PF. This approximation will become more and more accurate with the evolution of population. The information contained in the population can be used to estimate the PF and thus, aid to adaptively decompose the region and design the weight vectors. Motivated by this, we propose an adaptive regions decomposition framework:

MOEA/D-AM2M for MaOPs optimization, where adaptive weight vectors can be designed by extracting information from the population. In the AM2M framework, the evolution of population can be divided into two process: evolution and adjustment.

- **Evolution:** Use the designed direction and weight vectors to adaptively decompose the region and select the next generation population.
- **Adjustment:** Adaptively adjust the subregions and weight vectors according to the distribution of present population.

The two processes will be carried out alternately to generate a circle in the evolutionary process. It's worth noting that, in AM2M framework, we do not even need to distinguish the beginning and ending of the circle. They will be mutually beneficial and supportive to guide the population search no matter who is first. On the one hand, evolved population will make the direction and weight vectors design more and more accurate; On the other hand, more and more accurate direction and weight vectors will, in turn, enhance the search efficiency by guiding the population to the more promising areas.

A so-called **Max-Min** method is proposed to adaptively decompose the region and design the weights by extracting PF information from the present population. **Max-Min** method works as follows: Randomly select a point from a set of points at first; And then select the first point in the remaining points and make it have the smallest cosine similarity to first selected point; Select the second point in the remaining points (except for the randomly selected point) and make it have the smallest similarity with the randomly selected point; The next points is selected to make it have the smallest similarity with the set of selected points. Continually select points until the requirement is satisfied.

The population search based **Max-Min** method can also effectively distinguish which areas is worthy searching and which is not. If there are no solutions in an area, no weights would be assigned for this area according to **Max-Min** method. Generally, population will change at every generation, however it will not change a lot until after a certain generation's evolution. Therefore, extracting PF information from population to adjust subregions and weights at every generation is unnecessary, what is more it will cost the precious computing resources. In this paper, we adjust subregions and weights by **Max-Min** method every certain generations.

III. Main Framework of MOEA/D-AM2M

The basic framework of MOEA/D-AM2M mainly includes two aspects: (1) adaptive region decomposition; (2) adaptive weights design. We will discuss the two aspects in details in this section.

A. Adaptive Region Decomposition

This part shows how to do adaptive region decomposition from a set of $2N$ solutions in the objective space, where N is the population size. As we have discussed above, once the direction vectors are generated, the entire objective space can be decomposed into a series subregions, and direction vectors

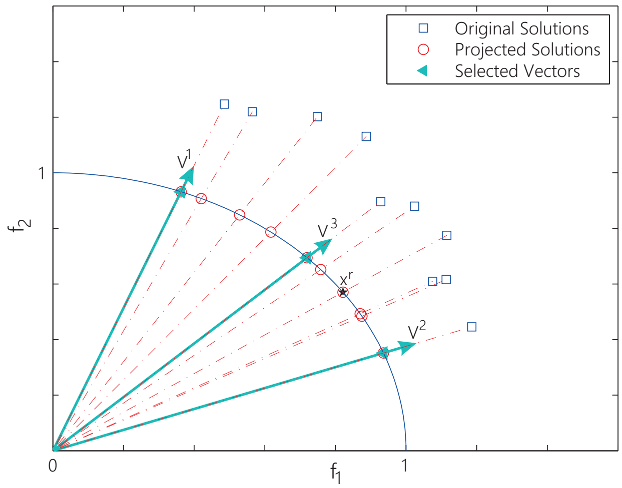


Fig. 2. Illustration of direction vectors design by the **Max-Min** method.

determine the location and size of the subregions. Therefore, adaptive region decomposition is realized by adaptive direction vectors design shown by the following Algorithm **??**.

Algorithm 1: Adaptive Direction Vectors Design

Input:
$\widetilde{\mathbf{P}}$: a set of $2N$ solutions in the objective space.
Output:
$K(K \ll N)$ direction vectors: $\mathbf{v}^1, \ldots, \mathbf{v}^K$.
Step 1: Project $\widetilde{\mathbf{P}}$ to unit sphere and get $\widehat{\mathbf{P}}$. Set $\mathbf{V} = \varnothing$, randomly select a vector $\mathbf{x}_r$ from set $\widehat{\mathbf{P}}$, and select the first vector $\mathbf{v}$ from $\widehat{\mathbf{P}}$ to make it have the largest cosine similarity with $\mathbf{x}^r$. let $\mathbf{V} = \mathbf{V} \cup \{\mathbf{v}\}$ and $\widehat{\mathbf{P}} = \widehat{\mathbf{P}} \backslash \{\mathbf{v}\}$.
Step 2: Select the vector $\mathbf{v}$ in $\widehat{\mathbf{P}}$ to make it have the smallest cosine similarity with the set of selected vectors in $\mathbf{V}$. let $\mathbf{V} = \mathbf{V} \cup \{\mathbf{v}\}$ and $\widehat{\mathbf{P}} = \widehat{\mathbf{P}} \backslash \{\mathbf{v}\}$.
Step 3: If K vectors have been selected, stop; Else, go back to **Step 2**.

The method to determine direction vectors can make best use of the distribution information, thereby making the adaptive region decomposition more reasonable. An illustration of direction vectors design by the **Max-Min** method can be found in Figure **??**.

In Figure **??**, the original solutions are firstly projected to the unit sphere in the first quadrant, where blue squares represent the original solutions, the red circles represent the projected solutions and the triangles represent the selected vectors. We first randomly select a projected solution $\mathbf{x}^r$ represented by the black pentagram in Figure **??**. The first direction vector $\mathbf{v}^1$ having the smallest cosine similarity with $\mathbf{x}^r$ is selected and then the direction vector having the smallest cosine similarity with $\mathbf{v}^1$ is selected as $\mathbf{v}^2$. Vector $\mathbf{v}^3$ is selected to make it such a direction which has the minimum of the maximum cosine similarity between $\mathbf{v}^3$ and $\mathbf{v}^1$ and $\mathbf{v}^3$ and $\mathbf{x}^2$. It is noteworthy that many other methods can also be used to generate direction vectors for adaptive regions decomposition in MOEA/D-AM2M framework, and the **Max-Min** method only illustrates an viable approach.

B. Adaptive Weights Design

Weight vectors are very essential for weights aided EMO algorithm. Because the distribution of weight vectors can determine the distribution of final solutions obtained by these weights to a large extent. Under the framework of adaptive region decomposition, weight vectors should be designed according to the adaptive change of the subregions. And the change of subregions can be reflected by the change of subpopulation in this subregion. Therefore, we design the weight vectors in each subregion using the distribution information of solutions in each subregion. The weight vectors are generated by Algorithm **??**:

Algorithm 2: Adaptive Weight Vectors Design

Input:

$\widetilde{\mathbf{P}}$: a set of $2N$ solutions in objective space,
K direction vectors: $\mathbf{v}^1, \ldots, \mathbf{v}^K$.

Output:

A set of weight vectors: $\mathbf{w}^{11}, \ldots, \mathbf{w}^{1S_1}, \ldots, \mathbf{w}^{K1}, \ldots, \mathbf{w}^{KS_K}$.

Step 1:Decompose the search space $\mathbf{\Omega}$ into K subregions: $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_K$, and assign the individuals of the whole population to each subregions to get $\widetilde{\mathbf{P}}_k = \{\mathbf{x} \in \mathbf{\Omega}_k | \mathbf{x} \in \widetilde{\mathbf{P}}\}$, $k = 1, \ldots, K$.

Step 2: Select individuals by **Max-Min** method from $\widetilde{\mathbf{P}}_k$ and use them to design the weights $\mathbf{w}^{k1}, \ldots, \mathbf{w}^{kS_k}$ for subregion $\mathbf{\Omega}_k$, $k = 1, \ldots, K$.

C. The Framework of MOEA/D-AM2M

Based on the adaptive direction and weight vectors design, we can get the basic framework of MOEA/D-AM2M, and it works as follows (Algorithm **??**):

IV. EXPERIMENTAL STUDIES

We have compared MOEA/D-AM2M with MOEA/D-DE in our experimental studies. MOEA/D-DE is an efficient and effective implementation of MOEA/D for continuous MOPs proposed in [**?**].

A. Test Instances

The following five degenerated MaOP test instances are generated for simulation, where $g(\hat{\mathbf{x}})$ functions are used to control the PSs. Their search space is $[0, 1]^n$. $n = 10$, and the dimension of objective space is $m = 10$. r_{i1}, r_{i2} and r_{i3} $4 \le i \le m$ are constant, and they can be generated randomly in [0,1]. In our study, we set: (r_{41}, r_{42}, r_{43})=$(0.1, 0.1, 0.3)$; (r_{51}, r_{52}, r_{53})=$(0.4, 0.1, 0.3)$; (r_{61}, r_{62}, r_{63})=$(0.3, 0.2, 0.4)$; (r_{71}, r_{72}, r_{73})=$(0.3, 0.1, 0.1)$; (r_{81}, r_{82}, r_{83})=$(0.4, 0.3, 0.1)$; (r_{91}, r_{92}, r_{93})=$(0.3, 0.2, 0.3)$; $(r_{10,1}, r_{10,2}, r_{10,3})$=$(0.1, 0.3, 0.1)$. All these instances are for minimization.

Algorithm 3: MOEA/D-AM2M

Input:

- MaOP;
- A stopping criterion;
- The probability parameter for crossover and mutation α;
- The population size N;
- Genetic operators and their associated parameters;
- The frequency of update G.

Output:

Ψ: a set of nondominated solutions.

Step 1: Initial population $\widetilde{\mathbf{P}}$ by uniformly randomly generating N points from $[a, b]^n$, compute their F-values; Uniformly initial direction vectors and weight vectors and then use them to set the subpopulation size of each subregions $S_1, \ldots, S_K$ by assigning weight vectors to each subregions.

Step 2: Decompose the population into each subregion, and use them to initialize subpopulation $\mathbf{P}_k (k = 1, \ldots, K)$, set $\mathbf{P} = \bigcup_{i=1}^{K} \mathbf{P}_i$.

Step 3:

while the stopping criterion is not met **do**
  Generation of New Solutions: Set $R = \emptyset$;
  for $K = 1$; $k < K$; $K{+}{+}$ **do**
    for each $x \in P_k$ **do**
      Generate a random number r in $[0, 1]$;
      if $r < \alpha$ **then**
        Randomly choose $\mathbf{y}$ from $\mathbf{P}_k$;
      else
        Randomly choose $\mathbf{y}$ from $\mathbf{P} \backslash \mathbf{P}_k$;
      end if
      Apply genetic operators on $\mathbf{x}$ and $\mathbf{y}$ to generate a new solution $\mathbf{z}$ and compute $F(\mathbf{z})$;
      $\mathbf{R} := \mathbf{R} \cup \{\mathbf{z}\}$;
    end for
    $\widetilde{\mathbf{P}} := \mathbf{R} \cup (\cup_{k=1}^{K} \mathbf{P}_k)$;
    use $\widetilde{\mathbf{P}}$ to set $\mathbf{P}_1, \ldots, \mathbf{P}_K$ by **Step 2**.
  end for
  if $mod(gen, G) == 0$ **then**
    Redesign direction and weight vectors by **Algorithm 1** and **Algorithm 2**;
  end if
  $gen = gen + 1$;
  Find all the nondominated solutions in $\cup_{k=1}^{K} \mathbf{P}_k$ and output them.
end while

$$\text{MaOP1:} \begin{cases} f_1(\mathbf{x}) = (1 + g(\hat{\mathbf{x}}))\frac{\sqrt{2}x_1}{2} \\ f_2(\mathbf{x}) = (1 + g(\hat{\mathbf{x}}))\frac{\sqrt{2}x_1}{2} \\ f_3(\mathbf{x}) = (1 + g(\hat{\mathbf{x}}))(2 - x_1^2 - sign(\cos(2\pi x_1))) \\ f_i(\mathbf{x}) = G_i(\mathbf{x})(\exp(|x_i - \sin(\frac{\pi x_1}{2})|^2) - 1) \end{cases}$$

where

$$G_i(\mathbf{x}) = (r_{i1} f_1(\mathbf{x}) + r_{i2} f_2(\mathbf{x}) + r_{i3} f_3(\mathbf{x})), 4 \le i \le m,$$
$$\mathbf{x} = (x_1, x_2, ..., x_n), \hat{\mathbf{x}} = (x_2, x_3, x_4, ..., x_n),$$
$$g(\hat{\mathbf{x}}) = \sum_{i=2}^{n} (x_i - \sin(\frac{\pi x_1}{2}))^2.$$

MaOP2: $$\begin{cases} f_1(\mathbf{x}) = (1+g(\hat{\mathbf{x}}))\frac{\sqrt{2}x_1}{2} \\ f_2(\mathbf{x}) = (1+g(\hat{\mathbf{x}}))\frac{\sqrt{2}x_1}{2} \\ f_3(\mathbf{x}) = (1+g(\hat{\mathbf{x}}))(2 - x_1^{0.5} - sign(\cos(2\pi x_1))) \\ f_i(\mathbf{x}) = G_i(\mathbf{x})(\exp(|x_i - \sin(\frac{\pi x_1}{2})|^2) - 1) \end{cases}$$

where

$$G_i(\mathbf{x}) = (r_{i1}f_1(\mathbf{x}) + r_{i2}f_2(\mathbf{x}) + r_{i3}f_3(\mathbf{x})), 4 \le i \le m,$$
$$\mathbf{x} = (x_1, x_2, ..., x_n), \hat{\mathbf{x}} = (x_2, x_3, x_4, ..., x_n),$$
$$g(\hat{\mathbf{x}}) = \sum_{i=2}^{n} (x_i - \sin(\frac{\pi x_1}{2}))^2.$$

MaOP3: $$\begin{cases} f_1(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_1))\frac{\sqrt{2}x_1}{2} \\ f_2(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_1))\frac{\sqrt{2}x_1}{2} \\ f_3(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_1))(1-x_1^2) \\ f_i(\mathbf{x}) = G_i(\mathbf{x})(\exp(|x_i - \sin(\frac{\pi x_1}{2})|^2) - 1) \end{cases}$$

where

$$G_i(\mathbf{x}) = (r_{i1}f_1(\mathbf{x}) + r_{i2}f_2(\mathbf{x}) + r_{i3}f_3(\mathbf{x})), 4 \le i \le m,$$
$$\mathbf{x} = (x_1, x_2, ..., x_n), \hat{\mathbf{x}} = (x_2, x_3, ..., x_n),$$
$$g(\hat{\mathbf{x}}) = \sum_{i=2}^{n} (x_i - \sin(\frac{x_1 \pi}{2}))^2,$$
$$h(x_1) = \max(0, 2\sin(4\pi x_1)).$$

MaOP4: $$\begin{cases} f_1(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))x_1x_2 \\ f_2(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))(1-x_1)x_2 \\ f_3(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))(1-x_2) \\ f_i(\mathbf{x}) = G_i(\mathbf{x})(\exp(|x_i - \sin(\frac{\pi x_1}{2})|^2) - 1) \end{cases}$$

where

$$G_i(\mathbf{x}) = (r_{i1}f_1(\mathbf{x}) + r_{i2}f_2(\mathbf{x}) + r_{i3}f_3(\mathbf{x})), 4 \le i \le m,$$
$$\mathbf{x} = (x_1, x_2, ..., x_n), \hat{\mathbf{x}} = (x_3, x_4, ..., x_n),$$
$$g(\hat{\mathbf{x}}) = \sum_{i=3}^{n} (x_i - \sin(\frac{\pi x_1}{2}))^2,$$
$$h(x_2) = \max(0, -\cos(2\pi x_2)).$$

MaOP5: $$\begin{cases} f_1(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))\cos(\frac{\pi x_1}{2})\cos(\frac{\pi x_2}{2}) \\ f_2(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))\cos(\frac{\pi x_1}{2})\sin(\frac{\pi x_2}{2}) \\ f_3(\mathbf{x}) = (1+g(\hat{\mathbf{x}})+h(x_2))\sin(\frac{\pi x_1}{2}) \\ f_i(\mathbf{x}) = G_i(\mathbf{x})(\exp(|x_i - \sin(\frac{\pi x_1}{2})|^2) - 1) \end{cases}$$

where

$$G_i(\mathbf{x}) = (r_{i1}f_1(\mathbf{x}) + r_{i2}f_2(\mathbf{x}) + r_{i3}f_3(\mathbf{x})), 4 \le i \le m,$$
$$\mathbf{x} = (x_1, x_2, ..., x_n), \hat{\mathbf{x}} = (x_3, x_4, ..., x_n),$$
$$g(\hat{\mathbf{x}}) = \sum_{i=3}^{n} (x_i - \sin(\frac{\pi x_1}{2}))^2,$$
$$h(x_2) = \max(0, -1.4(\cos(2\pi x_1) - 0.5)).$$

B. Simulation Results and Analysis

In this section, five degenerated test instances are tested by MOEA/D-AM2M, MOEA/D-DE and MOEA/D-M2M. The genetic operators of MOEA/D-DE and MOEA/D-M2M are kept the same with their original version in [**?**] and [**?**]. The population of the three algorithms is 110 for MaOP1-MaOP3, and 275 for MaOP4-MaOP5. In MOEA/D-AM2M, direction and weight vectors are adjusted every $G = 100$. All algorithm stop after 1000 generation for MaOP1-3, and 2000 for MaOP4-5. The direction and weight vectors used in the three algorithms are all initialized by latin Square method. The characteristic of those test instances is that their PFs are all determined by the first three objectives and thereby are manifolds in the objective subspace. Since these two objectives of MaOP1-MaOP3 are correlated to each other, we obtain a Pareto curve for the three MaOPs and Pareto-optimal surface for the last two MaOPs. HV-metric [**?**] is used to measure the quality of obtained solutions by the two algorithms. Considering that the PFs of these test instances are all located in a subsapce of the objective space, the HV-metric is calculated in the subspace for the sake of simplicity and we call it approximate HV-metric (AHV-metric) where the reference point is set as the nadir point+0.001 in this paper. Table **??** shows the best, mean and the worst of the AHV-metric values of the two algorithms for each test instance in 20 independent runs.

TABLE I
THE BEST AND MEAN OF APPROXIMATE HV-METRIC VALUES OF MOEA/D-AM2M, MOEA/D-DE AND MOEA/D-M2M IN 20 INDEPENDENT RUNS FOR EACH TEST INSTANCE. THE BEST AVERAGE AHV VALUE AMONG THE THREE ALGORITHMS IS BOLDED

AHV-metric	MOEA/D-AM2M		MOEA/D-DE		MOEA/D-M2M	
Instance	best	mean	best	mean	best	mean
MaOP1	**0.420469**	**0.412386**	0.397334	0.388025	0.222989	0.212537
MaOP2	**0.790441**	**0.786794**	0.755905	0.752269	0.544341	0.523191
MaOP3	**0.742017**	**0.732719**	0.731753	0.722794	0.273794	0.256391
MaOP4	**0.412715**	**0.406265**	0.394445	0.383742	0.742361	0.677383
MaOP5	**0.382389**	**0.363430**	0.374125	0.361364	0.325041	0.282363

Figure 4 plots, in the subspace of objective space, the distribution of the final solutions obtained by MOEA/D-AM2M, MOEA/D-DE and MOEA/D-M2M in the run with the median AHV-metric value of each algorithm for the five test instances. From the table and figure, we can conclude that MOEA/D-AM2M outperforms MOEA/D-DE and MOEA/D-M2M in solving those degenerated MaOPs. The reason of better achievement to MOEA/D-DE and MOEA/D-M2M is obvious, because adaptive weight design can effectively adjust the weights according to the population's distribution, while the fixed weights generated by latin Square method in MOEA/D can not yield to good population distribution along the PF.

V. CONCLUSION

In this paper, we have proposed an adaptive region decomposition framework called MOEA/D-AM2M for MaOPs. In this framework, objective space can be adaptively adjust by dynamic direction vectors design. Based on this framework, an effective weights design method is proposed for Tchebycheff selection. Both adaptive region decomposition and weight vectors design are realized by the proposed **Max-Min** method which can extract useful information from the evolving population. Results from our proposed method have been compared with MOEA/D-DE method and simulation results show that the proposed MOEA/D-AM2M has better performance than MOEA/D in solving degenerated MaOPs.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of Guangdong Province (2014A030313507,

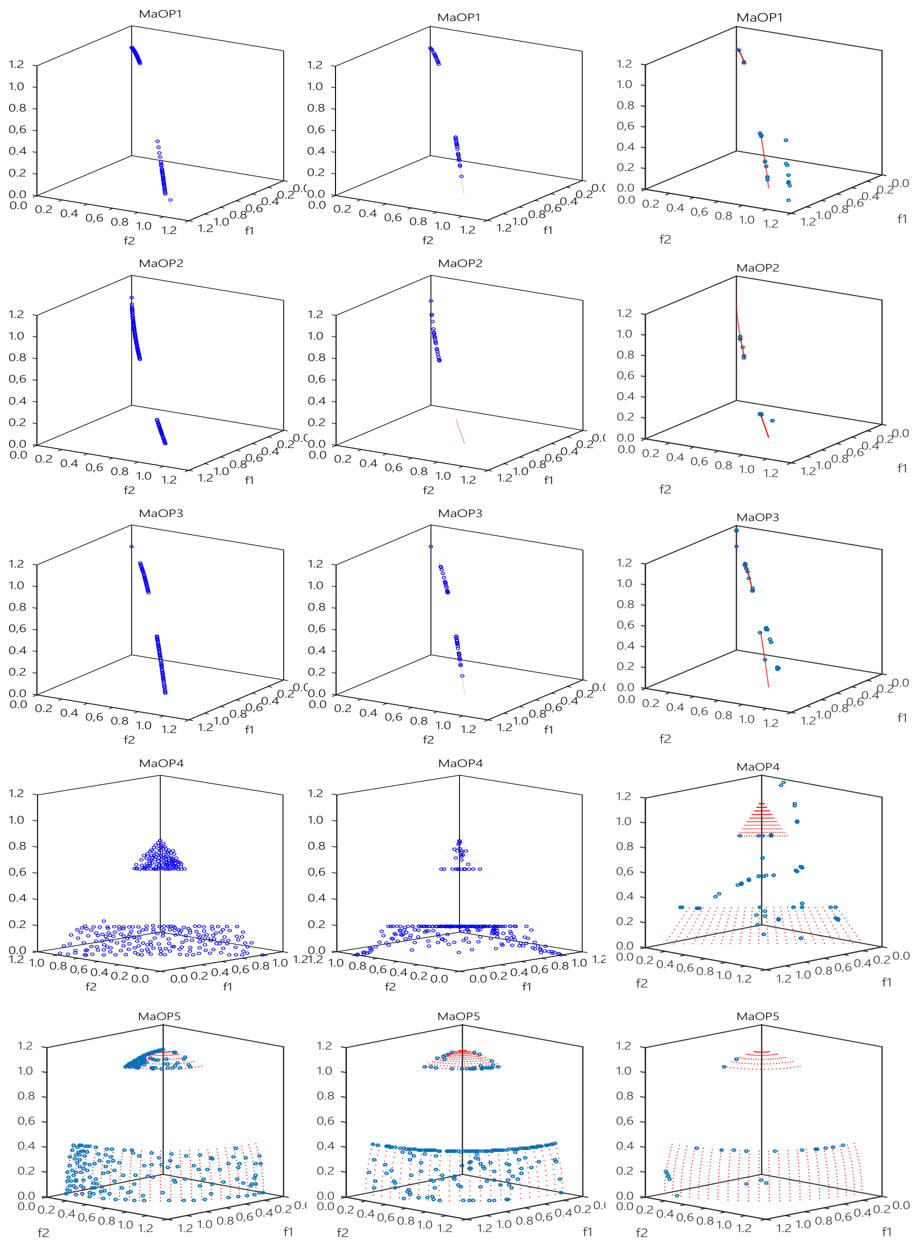


Fig. 3. Plot of the nondominated front in the objective subspace with the median IGD-metric value found by MOEA/D-AM2M MOEA/D-DE and MOEA/D-M2M.

2014A030310257), and in part by the Projects of Science and Technology of Guangzhou (2014J4100209, 201508010008).

References

[1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: a short review," *in Proceedings of IEEE Congress on Evolutionary Computation*, Hong Kong, China, pp. 2424-2431, 2008.

[2] Q. Zhang, H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712-731, 2007.

[3] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, 2015.

[4] K. Miettinen, *"Nonlinear Multiobjective Optimization. Boston,"* MAA: Kluwer, 1999.

[5] H-L. Liu, F. Gu and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, 2013.

[6] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *in IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284-302, 2009.

[7] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multiobjective evolutionary algorithm for a quick computation of Paretooptimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501-525, 2005.

[8] Hernandez Diaz A, Santana Quintero L, Coello Coello C, et al, "Pareto adaptive ϵ-dominance," *Evolutionary Computation*, vol. 15, no. 4, pp. 493-517, 2007.

[9] A. L. Jaimes, C. A. Coello Coello, H. Aguirre, and K. Tanaka, "Adaptive objective space partitioning using conflict information for manyobjective optimization," *in Proc. Evol. Multi-Criterion Optimization*, 2011, pp. 151-165.

[10] M. Koppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," *in Proc. Evol. Multi-Criterion Optimization*, 2007, pp. 727-741.

[11] M. Farina and P. Amato, “A fuzzy definition of ’optimality’ for many-criteria optimization problems,” *IEEE Tran. Syst. Man, Cybern., Part A: Syst. Hum.*, vol. 34, no. 3, pp. 315-326, May 2004.

[12] Yang S, Li M, Liu X., “ A Grid-Based Evolutionary Algorithm for Many-Objective Optimization,” *IEEE Trans. On Evolutionary Computation* vol. 17, no. 5, pp. 721-736, 2013.

[13] F. di Pierro, S.-T. Khu, and D. A. Savic, “An investigation on preference order ranking scheme for multiobjective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17-45, Jan. 2007.

[14] D. K. Saxena and K. Deb, “ Non-linear dimensionality reduction procedures for certain large-dimensional multiobjective optimization problems: Employing correntropy and a novel maximum variance unfolding,” *in Proc. EMO*, pp. 772-787, 2007.

[15] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, “Objective jective reduction in many-objective optimization: Linear and nonlinear algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77-99, 2013.

[16] Bader, J., and Zitzler, E., “Hype: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45-76, 2011.

[17] N. Beume, B. Naujoks, and M. Emmerich, “SMS-EMOA: Multiobjective selection based on dominated hypervolume,” *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653-1669, 2007.

[18] Bringmann K, Friedrich T, Igel C, “Speeding Up Many-Objective Optimization by Monte Carlo Approximations,” *Artificial Intelligence*, 2013.

[19] Lyndon While, Lucas Bradstreet, and Luigi Barone, “ A fast way of calculating exact Hypervolumes,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86-95, February 2012.

[20] Russo, L. M. S., Francisco, A. P., “Quick Hypervolume,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 481-502, 2013.

[21] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun and J. Wu, “ MOEA/D with Adaptive Weight Adjustment,” *Evolutionary Computation*, vol. 22, no. 2, pp. 231-264, 2014.

[22] Gu F, Liu H, “ An adaptive multiobjective differential evolution algorithm,” *Journal of Computers*, vol. 8, no. 2, pp. 94-301, 2013.

[23] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, Apr. 2003.