A Decomposition-Based Evolutionary Algorithm with Correlative Selection Mechanism for Many-Objective Optimization

Ruochen Liu  ruochenliu@xidian.edu.cn
Ruinan Wang  1617171590@qq.com
Renyu Bian  1135119460@qq.com
Jing Liu  neouma@mail.xidian.edu.cn
Licheng Jiao  lcjiao@mail.xidian.edu.cn

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, International Center of Intelligent Perception and Computation, Xidian University, Xi’an, 710071, China

https://doi.org/10.1162/evco_a_00279

Abstract
 Decomposition-based evolutionary algorithms have been quite successful in dealing with multiobjective optimization problems. Recently, more and more researchers attempt to apply the decomposition approach to solve many-objective optimization problems. A many-objective evolutionary algorithm based on decomposition with correlative selection mechanism (MOEA/D-CSM) is also proposed to solve many-objective optimization problems in this article. Since MOEA/D-SCM is based on a decomposition approach which adopts penalty boundary intersection (PBI), a set of reference points must be generated in advance. Thus, a new concept related to the set of reference points is introduced first, namely, the correlation between an individual and a reference point. Thereafter, a new selection mechanism based on the correlation is designed and called correlative selection mechanism. The correlative selection mechanism finds its correlative individuals for each reference point as soon as possible so that the diversity among population members is maintained. However, when a reference point has two or more correlative individuals, the worse correlative individuals may be removed from a population so that the solutions can be ensured to move toward the Pareto-optimal front. In a comprehensive experimental study, we apply MOEA/D-CSM to a number of many-objective test problems with 3 to 15 objectives and make a comparison with three state-of-the-art many-objective evolutionary algorithms, namely, NSGA-III, MOEA/D, and RVEA. Experimental results show that the proposed MOEA/D-CSM can produce competitive results on most of the problems considered in this study.

Keywords
 Many-objective optimization, evolutionary computation, decomposition, correlative selection mechanism.

1 Introduction

Many researchers have paid a lot of attention to multiobjective optimization problems since the beginning of the 1990s, and up to now, they have proposed many excellent multiobjective evolutionary algorithms (MOEAs) to deal well with multiobjective...
optimization problems. PAES (Knowles and Corne, 1999), SPEA2 (Zitzler et al., 2001), and NSGA-II (Deb et al., 2002) are representative of traditional Pareto-based MOEAs and their performance was better than that of other MOEAs proposed at that time. However, in the real world, many multiobjective optimization problems often involve more than two or three contradictory objectives, sometimes demanding to have 10 to 15 objectives (Chikumbo et al., 2012; Coello and Lamont, 2004). Generally, we consider optimization problems with more than three objectives to be many-objective problems. Nevertheless, if these current MOEAs are directly used to solve many-objective problems, we find that their performance will degrade sharply or some of them cannot even deal with these kinds of problems. Many-objective problems have attracted increasing attention in recent years, and consequently, a number of many-objective evolutionary algorithms have been developed. Also, several good reviews on this topic are currently available (Ishibuchi et al., 2008; Li, Li et al., 2015; von Lück et al., 2014).

Recent studies show that decomposition approaches are helpful for solving many-objective problems. Therefore, this article also proposes a new algorithm based on decomposition called MOEA/D-CSM. MOEA/D-CSM introduces a novel selection mechanism which can select an individual according to the individual’s correlative condition rather than using the individual’s fitness values based on the hypervolume indicator (Bader and Zitzler, 2011) or on the crowded distance (Deb et al., 2002). Since MOEA/D-SCM also adopts the decomposition approach of MOEA/D (Wang et al., 2017; Zhang and Li, 2007; Wang et al., 2016), a set of reference points must be generated in advance. MOEA/D-CSM proposes two important concepts related to the reference points: 1) correlation between an individual and a reference point, and 2) neighboring reference points of each reference point. Generally, in the decomposition approach, we know that a reference point defines a reference line along with an ideal point. If an individual is closer to the reference line than other reference ones, the individual and the corresponding reference point are considered to be correlative. If an individual and a reference point are correlative, the individual is called a correlative individual of the reference point and the reference point is also called a correlative reference point of the individual. According to the definition of correlation, it is easy to know that an individual must have a correlative reference point while a reference point maybe has any (zero, one, or more) number of correlative individuals. The proposed algorithm aims to find their correlative individuals for each reference point so that the diversity among population members can be maintained. When two solutions have the same correlative reference point, the individual with a lower scalarization function value using the penalty boundary intersection (PBI) approach (Zhang and Li, 2007) is better than the other approaches (if minimization problems are considered where all objectives are to be minimized). If a reference point has two or more correlative individuals, we choose a better correlative individual so that the solution can move toward the Pareto-optimal front. Since the proposed algorithm does not adopt nondominated sorting, its computation cost can be reduced to some degree.

In the remainder of the article, we first review a number of existing many-objective evolutionary algorithms in Section 2. Thereafter, in Section 3, we describe the proposed MOEA/D-CSM algorithm in detail. Section 4 presents experimental results of MOEA/D-CSM and compares it with three other elitist many-objective evolutionary algorithms; that is, NSGA-III (Deb and Jain, 2014), MOEA/D (Zhang and Li, 2007), and RVEA (Cheng et al., 2016). Section 5 gives a comparison between MOEA/D-CSM and MOAE/D-STM (Li et al., 2014). Finally, conclusions are drawn in Section 6.
2 Related Work

As the number of objectives in optimization problems increases, the main difficulties to solve such problems using MOEAs are the following:

1. With the increase in the number of objectives, the number of nondominated individuals will be expanded sharply due to the Pareto-based domination relation, which is adopted by most MOEAs. Therefore, when these MOEAs are used to deal with many-objective problems, the probability of generating new individuals better than their parents at each generation, is quite small. This phenomenon will make the execution of these algorithms slow down or some of them will simply be unable to obtain satisfactory results.

2. It is difficult to maintain the diversity of the population. To determine the extent of crowding of solutions in a population, the identification of neighbors becomes computationally expensive when an optimization problem has many objectives (Deb and Jain, 2014).

3. The representation of the Pareto-optimal front becomes difficult in high-dimensional spaces. In order to properly represent the Pareto-optimal front, as a problem has more objectives, a larger population size is required. Nevertheless, the increase in the population size not only considerably increases the running time of the algorithm but also brings difficulties to make a proper decision for a decision maker (Ishibuchi et al., 2008; Deb and Jain, 2014).

For the three problems mentioned previously, recent research directions can be simply divided into three aspects besides preference-based methods and dimensional reduction methods. First, the hypervolume indicator has been adopted to assign a fitness value to each individual so that the quality of each solution can be evaluated. A representative of hypervolume-based MOEAs is HypE (Bader and Zitzler, 2011). Second, GrEA (Yang et al., 2013) was proposed as a grid-based MOEA that adopts a relaxed form of Pareto dominance called ε-dominance (Laumanns et al., 2002). Moreover, a new decomposition-based (or reference-point-based) algorithm, NSGA III, was proposed by Deb and Jain (2014), and many researchers have paid attention to this new algorithm and developed similar algorithms; for example, MOEA/D-DD (Li, Deb et al., 2015) and RVEA (Cheng et al., 2016). The main idea of NSGA-III is to combine the decomposition strategy of MOEA/D (Zhang and Li, 2007) with the nondominated sorting approach from NSGA-II (Deb et al., 2002). NSGA-III can successfully generate well converged and well diversified sets of solutions to many-objective optimization problems.

These Pareto-based MOEAs work well for low-dimensional objective optimization problems. But for many-objective problems, the proper balance between convergence and diversity for these algorithms becomes very difficult. Therefore, some scholars proposed the hypervolume indicator, which is strictly monotonic with the Pareto dominance and can act as an alternative mechanism to the Pareto dominance. Bader and Zitzler (2011) put forward a fast search algorithm that uses Monte Carlo simulation to approximate exact hypervolume values. Jiang et al. (2015) proposed a simple and fast method to update the exact hypervolume contributions of different solutions. Menchaca-Méndez et al. (2018) developed an adaptative control strategy to reduce the number of hypervolume contributions per iteration. Zapotecas-Martínez et al. (2019) put forward a Lebesgue indicator-based evolutionary algorithm to solve continuous and box-constrained multiobjective optimization problems. However, the
hypervolume indicator has a high computational cost. As the number of objectives increases, hypervolume-based algorithms become inapplicable. Because of this, some methods to approximate the hypervolume contribution have been proposed in order to reduce its computational cost. For example, HypE uses the Monte Carlo simulations (Everson et al., 2002; Bader et al., 2010). However, experimental results indicate that regardless of the approximation method adopted, hypervolume-based MOEAs that incorporate such methods have poor performance. Therefore, the use of hypervolume-based MOEAs for solving many-objective optimization problems is still an open research area.

$\varepsilon$-dominance (Laumanns et al., 2002) is a relaxed form of the Pareto dominance that was proposed as an archiving technique, but that can be adopted to dilute the selection pressure generated in many-objective optimization problems. This new relation can not only make the number of nondominated solutions in a population decrease to some degree but also play a crucial role in diversity maintenance. These properties of $\varepsilon$-dominance inspired Yang et al. (2013) to propose a new evolutionary algorithm called GrEA which is very competitive with respect to some state-of-the-art MOEAs such as HypE, MOEA/D and $\varepsilon$-MOEA (Deb, Mohan et al., 2005). Nevertheless, the performance of GrEA is affected by its many parameters. For the better use of this algorithm, it is necessary to understand the specific effect of these parameters on its performance.

It is well known by the researchers in the multiobjective optimization field that MOEA/D plays a very important role in solving multiobjective optimization problems. Early decomposition algorithms (Jin et al., 2001) applied dynamic aggregation-based methods to archive the Pareto solutions of each generation. Decomposition-based MOEAs require a set of weighted vectors or reference points. In essence, these vectors or points play a similar role of decomposing the objective space into a number of subspaces to ensure diversity. As we know, a large number of decomposition-based MOEAs have been developed such as MOEA/D-STM (Li et al., 2014), MOEA/D-DE (Li and Zhang, 2009), MOEA/D-M2M (Liu et al., 2014), MOEA/DD (Li, Deb et al., 2015), IM-MOEA (Cheng et al., 2015), RVEA (Cheng et al., 2016), NSGA-III (Deb and Jain, 2014), etc., since Zhang and Li proposed MOEA based on decomposition in 2007. Here, we just give a simple comparison between the proposed algorithm and other decomposition-based MOEAs that adopt a similar algorithmic framework.

The algorithm proposed here is a decomposition-based MOEA designed for solving many-objective optimization problems. Regarding decomposition-based MOEAs for multiobjective optimization problems, there are four important aspects that need to be considered by their designers (Trivedi et al., 2016): 1) the way of generating the weight vectors; 2) the decomposition method; 3) the reproduction operators; and 4) the mating selection and the replacement strategy. For the four aspects just mentioned, we first explain the proposed algorithm in our article item by item, and then we make a comparison among those similar decomposition-based methods like MOEA/D, NSGA-III, MOEA/D-STM, MOEA/DD, and RVEA.

1. When the number of objectives is more than 8, the proposed algorithm adopts the two-layered generation method (Deb and Jain, 2014) to generate the weight vectors which is also used in NSGA-III, MOEA/DD, and RVEA. MOEA/D-STM adopts the simplex lattice design method. In the original version of MOEA/D, it also employs the simplex lattice design method to generate the weight vectors.
2. As we know, there are three commonly used decomposition methods in the evolutionary multiobjective optimization community, that is, the weighted sum (WS), the weighted Tchebycheff (TCH), and PBI. MOEA/D-STM utilizes the TCH method. NSGA-III, MOEA/DD, MOEA/D, and RVEA essentially use the PBI method, although their authors indicated in their original references that they employed a set of reference vectors that spread over the objective space to divide the objective space into multiple subspaces. The proposed MOEA/D-CSM also adopts the PBI method.

3. In the original MOEA/D, simulated binary crossover (SBX) and polynomial-based mutation (PM) operators are incorporated as the genetic operators. Up to now, many reproduction operators have been developed in different decomposition-based MOEAs such as differential evolution (DE), particle swarm optimization, ant colony optimization, etc. In the proposed algorithm (MOEA/D-CSM), we also use SBX and PM, the same as NSGA-III, MOEA/DD, MOEA/D, and RVEA. MOEA/D-STM adopts DE and SBX as the reproduction operator.

4. As we know, the mating selection and the replacement strategy play important roles in decomposition-based MOEAs. NSGA-III is an extension of the NSGA-II framework and it utilizes a set of reference points that spread over the objective space and decompose the objective space into multiple small subspaces. The hybrid population is classified into different nondominated levels, and solutions in the first level have the highest priority to be selected. Solutions in the last acceptable level are selected based on a niche-preservation operator, in which a solution with a less crowded reference line has a higher probability of being selected. MOEA/D-DD combines dominance and decomposition-based approaches for many-objective optimization and the update of the population is done in a hierarchical manner, adopting Pareto dominance, local density estimation, and scalarization functions, considered in a sequential manner. In MOEA/D-STM, a stable matching model coordinates the selection process of MOEA/D to select the most promising solutions for each subproblem. A subproblem prefers solutions that can lower its aggregation function value, while a solution prefers subproblems whose direction vectors are close to it. As a newly proposed decomposition-based MOEA, RVEA uses the reference vectors to decompose the objective space into multiple small subspaces, and it inherits an elitism strategy similar to NSGA-II where the parents population and the offspring population are combined at every generation to undergo an elitist selection. A new angle-penalized distance (APD) is used to select the solution from each subpopulation to enter the next generation. Experimental results show that it is highly competitive in comparison with MOEA/DD and NSGA-III in terms of the hypervolume indicator. In our algorithm, a new correlation between the reference points and solutions is proposed. The mating selection mechanism based on correlation is based on three entities of a solution and aims to find one correlative solution for all reference points, at least according to the second entity of solutions so that population diversity can be maintained. By computing the scalarization functions and the distance between the objective vector of a solution and the reference line decided by its correlative reference point, the proposed algorithm balances the diversity and the convergence. Moreover, the replacement strategies are also executed on the combination of offspring.
population and parents population based on three entities of reference points. It
is mainly to ensure for a reference point with the smallest number of correlative
solutions to save those solutions with better scalarization function values. Thus
NSGA-III, MOEA/DD, and RVEA were developed to solve the many-objective
problems. MOEA/D was found to perform well on the many-objective problems
in the original paper of NSGA-III. While MOEA/D-STM was proposed to solve
complex multiobjective problems like UF, RVEA was proposed in 2016 and from
the original paper of RVEA (Cheng et al., 2016). We found this algorithm to be
very competitive with respect to some outstanding algorithms like MOEA/DD.
This is why we make a comparison between MOEA/D, NSGA-III, RVEA, and
the proposed algorithm which is also designed for the many-objective optimiza-
tion problems at the beginning of the experimental section and then, we make a
separate comparison of MOEA/D-STM and MOEA/D-CSM in the latter section.

In the following section, the proposed algorithm, that is, a decomposition-based
evolutionary algorithm with correlative selection mechanism is suggested, investi-
gated, and discussed in detail.

3 The Proposed Algorithm: MOEA/D-CSM

3.1 Definition

Generally, a minimization multiobjective optimization problem can be defined as fol-
lows (Jain and Deb, 2014):

Minimize $F(x) = (f_1(x), f_2(x), f_3(x), \ldots, f_M(x)), x \in \Omega, F \in \mathbb{Z}$

$$g_j(x) \geq 0, \quad j = 1, 2, \ldots, p,$$

Subject to $h_k(x) = 0, \quad k = 1, 2, \ldots, q,$

$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n.$ (1)

where $f_1(x), f_2(x), f_3(x), \ldots, f_M(x)$ are the $M$ objective functions, $p$ is the number of
inequalities, and $q$ is the number of equalities. In this article, $p$ and $q$ are set as zero, and
$\mathbb{Z}$ is the feasible region of $F$. Therefore, the algorithm proposed by this article is used to
handle problems with box constraints.

In order to clarify the proposed algorithm, in this section, we first describe the de-
composition approach used in MOEA/D-CSM, and then we introduce a new type of
relationship between an individual and a reference point.

3.2 Generating Reference Vectors

The proposed MOEA/D-CSM adopts the PBI decomposition approach. For the PBI de-
composition approach, the most important issue is how to generate the reference vec-
tors, that is to say, how to uniformly segment the objective space. In earlier MOEA/Ds,
Das and Dennis’s (1998) systematic approach was a commonly used way to generate
the reference vectors. In fact, there are also many other ways to generate the reference
vectors. Zapotecas-Martínez et al. (2015) introduced a new methodology based on low
discrepancy sequences to generate the weighted vectors. He et al. (2016) proposed a
reference point sampling approach by taking the specific shape of the Pareto optimal
front into account for measuring the performance of MOEAs. Jiang and Yang (2017)
presented a k-layer reference direction generation approach. In this article, we use Das
and Dennis’s systematic approach. The total number of reference points ($H$) is con-
trolled by a parameter $D$ which represents the number of divisions along each objective.
A Decomposition-Based Evolutionary Algorithm

Figure 1: An example of 10 reference points distributed on a normalized hyper-plane by using the one-layered method when $D = 3, M = 3$.

So $r^1, r^2, \ldots, r^H$ denote all reference points in which each item of $r^i (i = 1, \ldots, H)$ takes a value from $\{0, \frac{1}{D}, \ldots, \frac{D}{D}\}$ and the sum of all items of $r^i (i = 1, \ldots, H)$ is 1. Therefore, the total number of reference points $H$ and the parameter $D$ must meet the following equation:

$$H = C_{D+M-1}^{M-1}$$  \hspace{1cm} (2)

Here, $M$ is the number of objectives. For example, in a three-objective problem ($M = 3$), the reference points are placed on a triangle with apex at (1, 0, 0), (0, 1, 0), and (0, 0, 1). If the parameter $D$ is set to 3, the total number of reference points $H$ is equal to 10. The distribution of 10 reference points is shown in Figure 1.

According to Eq. (2), for an eight-objective problem, if $D = 8$, it is easy to get the total number of reference points $H = 5,040$, which results in a large and impractical value since the population size is set as the number of reference points or the number of reference points plus 1 in the proposed algorithm. Moreover, if the parameter $D$ or the number of objectives, $M$, increases a bit, the total number of reference points $H$ will dramatically increase. Therefore, it is necessary to apply some advanced experimental design methods (Zhang and Leung, 1999; Santner et al., 2013) to generate reference points. In this article, when the number of objectives is large, two layers of reference points with small values of $D$ are used ( Deb and Jain, 2014). For example, for an eight-objective problem, on the first layer, if $D_1 = 3$, 120 reference points are created according to Eq. (2); on the second layer, if $D_2 = 2$, 36 reference points are created. So 156 ($H = 120 + 36 = 156$) reference points are generated which is much less than the 5,040 reference points created by the one-layered method. Figure 2 gives an example of two-layered reference points generated for a three-objective problem. In this article, when $M < 8$, the one-layered method is used to generate the reference points, and when $M \geq 8$, the two-layered method can be implemented in the study in order to reduce the computational complexity.

As shown in Figure 2, 9 reference points are generated by using the two-layered method when $D_1 = 2$ and $D_2 = 1$. In order to give a more common example, Figure 3 shows a distribution of more than 90 reference points generated by using the two-layered method. All of these reference points are obtained when $D_1, D_2$ are set as 7, 4,
Figure 2: An example of 9 reference points generated by using the two-layered reference points. There are six reference points on the first layer ($D_1 = 2$) and three reference points on the second layer ($D_2 = 1$).

Figure 3: A common example for reference points generated by using the two-layered reference points when $M = 3$, $D_1 = 7$, and $D_2 = 4$. The black points are the reference points in the first layer. The blue points represent reference points in the second layer.

respectively. In fact, the two-layered method used here is the same as in NSGA-III, in which the authors did not present the detailed procedure of the method. Here, in order to make the two-layered method easily reproducible, the procedures of the one-layered method and two-layered method are presented in Algorithms 1, 2, and 3.

In Algorithm 1, the set $R_0$ is used to store the reference points. Then the algorithm would generate one-layer reference points by calling the sub-function presented in Algorithm 3. In Algorithm 3, the parameter $K$ represents the time of the function $Recursion$.
Algorithm 1 The one-layered method

Require: number of objectives $M$, divisions along each objective $D_0$
1: Procedure One_layer($M, D_0$)
2: $R_0 = \Phi$ ($R_0$ is a point set that stores reference points.)
3: $r = 0$ ($r$ denotes an $M$-dimensional vector)
4: $R_0 = \text{Recursion}(R_0, 0, 0, r, D_0, M)$
Ensure: $R_0$

Algorithm 2 The two-layered method

Require: number of objectives $M$, divisions on the first layer $D_1$, divisions on the
second layer $D_2$
1: Procedure Two_layer($M, D_1, D_2$)
2: $R_1 = \Phi$ ($R_1$ is a point set that stores reference points.)
3: $R_2 = \Phi$
4: $R_1 = \text{Recursion}(R_1, 0, 0, r, D_1, M)$
5: $R_2 = \text{Recursion}(R_2, 0, 0, r, D_2, M)$
6: for each $q \in R_2$ ($q$ denotes a $M$-dimension vector.) do
7: \[ q' = ((D_2/M + q \ast D_2)/2) / D_2 \] ($q'$ denotes each dimension of the vector $q$.)
8: $R_1 = R_1 \cup \{q\}$
9: end for
Ensure: $R_1$

Algorithm 3 Recursion($R, k, total, r', D, M$)

Require: reference points set: $R$, a reference point: $r'$, the recursion level: $k$, the number of objectives: $M$, the number of divisions along each objective: $D$, the sum of the first $k$ divisions of a reference point $r'$: total, the $k$th division of a reference point $r'$: $r_k$
1: Procedure Recursion($R, k, total, r', D, M$)
2: if $k = M - 1$ then
3: \[ r'_k = D - \text{total} \] ($r'_k$ denotes the $k$th dimension of a reference point $r'$.)
4: $R = R \cup \{r'/D\}$ ($r'/D$ represents each dimension of a reference point $r'$ divided by divisions $D$.)
5: else
6: for $j = 0, 1, ..., D - \text{total}$ do
7: \[ r'_k = j \]
8: \[ \text{Recursion}(R, k + 1, \text{total} + j, r', D, M) \]
9: end for
10: end if
Ensure: $R$

calling itself. The set $R$ is used to store the reference points in the process of Recursion. By using $r'/D$, the reference points would be added to the set $R$ until the terminal conditions are satisfied.
In Algorithm 2, $R_1$ and $R_2$ are used to store the reference points in the first-layer and the second-layer, respectively. Then the algorithm would call the function *Recursion* to generate the reference points. Since the above reference points are generated respectively, it is indeed necessary to integrate the reference points in $R_2$ into $R_1$ based on $q' = ((D_2 / M + q * D_2) / 2) / D_2$. Thus, $R_1$ stores all reference points.

### 3.3 Correlation between a Solution and a Reference Point

A reference line can be obtained by linking an ideal point with a reference point. So, when all reference points are generated, all reference lines can also be obtained. These reference lines can be uniformly distributed in the hyperspace. The purpose of a decomposition-based algorithm is to try to conduct the search toward the best and the closest solutions from these reference lines in this article. To explain the idea more clearly, we first propose a new concept of correlation between an individual and a reference point.

In Figure 4, $z^*$ and $r$ are an ideal point and a reference point, respectively. Since the exact ideal objective vector is usually unknown a priori, it is generally set as the best value of each objective in the current population; that is, $z^*_i = \min\{f_i(x)|x \in S\}$, $i \in \{1, \ldots, M\}$, where $S$ denotes a set of solutions in the current generation. In this article, the ideal point is always the original one considering the minimization problem in Eq. (1). $\lambda = r - z^*$ is a reference line across the reference point $r$. The distance $d_1$ is the projection of line segment with apex at $F(x)$ and $z^*$ on the line $\lambda$. The distance $d_2$ denotes the perpendicular distance between $F(x)$ and the reference line $\lambda$. $d_1$ and $d_2$ are computed based on the PBI function which is shown as follows:

$$d_1 = \frac{|(F(x) - z^*) \cdot \lambda|}{\|\lambda\|}$$  \hspace{1cm} (3)

$$d_2 = \left\| F(x) - \left( z^* + d_1 \frac{\lambda}{\|\lambda\|} \right) \right\|$$  \hspace{1cm} (4)
If the distance $d_2$ between $F(x)$ and the reference line $\lambda$ is not larger than that between $F(x)$ and all other reference lines, the solution $x$ and the reference point $r$ are correlative. $x$ is called correlative solution of the reference point $r$ and the reference point $r$ is called correlative reference point of the solution $x$. Therefore, a solution should only have one correlative reference point while a reference point can have zero, one or more correlative solutions.

3.4 Mating Selection Based on Correlation

A novel selected mechanism is proposed in MOEA/D-CSM. It includes two parts: mating selection and environmental selection based on correlation. Most traditional MOEAs select solutions from the parents population according to fitness or nondomination rank, while MOEA/D-CSM chooses solutions and updates parent solutions according to related entities of reference points.

Before we introduce the mating selection based on correlation, three entities of each parent solution and each reference point should be given.

For each solution $x$, we calculate three entities: 1) its correlative reference point $r_x$; 2) $d_{1x}$, the distance between objective vector $F(x)$ and the reference line decided by the correlative reference point $r_x$, that is to say, $d_{1x} = d_2$ as shown in Figure 3; and 3) $d_{2x}$, the penalty distance between objective vector $F(x)$ and the reference line decided by the correlative reference point $r_x$, that is to say, $d_{2x} = d_1 + \theta d_2$ where $d_1, d_2$ are also shown in Figure 3 and $\theta = 5$ is set in this article which is suggested in Zhang and Li (2007). So, three entities of a solution can be presented as $(r_x, d_{1x}, d_{2x})$.

For each reference point $r$, we also calculate three entities: 1) $U_r$, a set of solutions which are correlative to the reference point $r$ in the parents population, 2) $n_r$, the number of solutions which are correlative solutions of the reference point $r$ in the parents population, and 3) $V_r$, a set of reference points which are the $T$ closest reference points to the reference point $r$ ($T = 8$ was adopted in this article). Therefore, three entities of the reference point $r$, can be presented as $(U_r, n_r, V_r)$.

In fact, the mating selection mechanism based on correlation mainly aims to find one correlative solution for all reference points at least according to the second entity of solutions so that population diversity can be maintained. As mentioned previously, whether a solution is relevant to a reference point is based on the second entity of a solution. There possibly exists a case such that a solution has more than one reference points. Here, we just select one as its reference point. When a reference point has more than one correlative solution in the search process, the solution with better convergence according to its third entity is saved. From the definition of the third entity of a solution, it is easy to see that it can denote the convergence of the solution. In other words, the second entity of solutions is used to maintain population diversity and the third entity of solutions is used to maintain population convergence.

Before we execute the mating selection operators based on correlation, entities of each solution and reference point must be calculated in advance. However, we need to point out that these entities are calculated according to the current parents population $P_t$ in Algorithm 4. Next, we will explain the mating selection mechanism based on correlation in detail.

First, two solutions $x_1$ and $x_2$ are randomly selected from a parents population. $r_1$ and $r_2$ are their correlative reference points, respectively. If $n_1 \leq n_2$ (where $n_1$ and $n_2$ denote the numbers of correlative solutions of reference points $r_1$ and $r_2$), the solution $x_1$ is saved; otherwise, the solution $x_2$ is saved. In the following explanation, we suppose that $n_1 \leq n_2$, that is, the solution $x_1$ is reserved.
Next, we randomly choose a reference point \( r_3 \) from the set \( V_{r_1} \) which is a set of \( T \) closest reference points to the reference point \( r_1 \). Then, a solution \( x_3 \) is randomly selected from a set \( U_{r_3} \) which is a set of correlative solutions of the reference point \( r_3 \).

Finally, \( x_1 \) and \( x_3 \) are considered as two parent solutions. Two offspring solutions are generated from the two parent solutions by using crossover and mutation. In this article, polynomial-based mutation (PM) (Deb et al., 2002) and simulated binary crossover (SBX) (Deb and Agrawal, 1994) are adopted within the mating selection mechanism.

Population size must be even in this article, because offspring solutions are always created in pairs according to mating selection based on correlation. Moreover, in decomposition-based evolutionary algorithms like NSGA-III, population size generally equals the number of reference points. Hence, in this article, if the number of reference points \( H \) is even, the population size, \( \text{popsize} \), is equal to \( H \); otherwise, \( \text{popsize} \) is equal to \( H + 1 \).

### 3.5 Environmental Selection Based on Correlation

Suppose the offspring population is \( Q_t \) after the mating selection operation is performed on the parents population \( P_t \). Then, the update of the population \( P_{t+1} \) using \( Q_t \) and \( P_t \) will be done by the proposed environmental selection based on correlation. So, the environmental selection operator can also be called population update operator, which is given in Algorithm 5.

Suppose that a solution \( x_k \) is the \( k \)th solution of the offspring population \( Q_t \), and three entities of the solution \( x_k \) are calculated and denoted as \( (r_k, d_{1k}, d_{2k}) \). Then three entities of the reference point \( r_k \) can also be obtained and denoted as \( (U_{r_k}, n_{r_k}, V_{r_k}) \).
The offspring individual \( x_k \) is going to update the current parents population \( P_t \). According to its second entity, it will find its correlative reference point, and then by comparing the values of \( n_r \) (\( r_i = 1, \ldots, H \)), the reference point with the largest number \( n_{r_{\text{max}}} \) of the correlative solutions among all reference points (denoted as \( r_{\text{max}} \)) is found. In all correlative solutions of \( r_{\text{max}} \), the solution \( x_{\text{max}} \) with the largest third entity will be computed and removed from the set \( U_{r_{\text{max}}} \). Finally, three entities of \( r_{\text{max}} \) are updated.

Furthermore, in order to provide a more clear explanation for the environmental selection process, Figures 5 and 6 show the selection process in detail when considering a multiobjective optimization problem with all the objectives being minimized, in which, we show how the current parents population is updated with the current offspring population in different situations. Here, we assume that the objective space has two dimensions and the population consists of six solutions as shown in Figures 5 and 6.

As shown in Figure 5a, in the current parents population, each reference vector has one correlative solution. After obtaining the current offspring population, the current parents population will be updated by using the proposed environmental selection. For example, when a solution \( c_1 \) in the current offspring population is produced, first, we need to find its correlative reference vector (\( r_1 \) in Figure 5b) according to its second entity (\( d_{1c_1} \)), so that the reference vector \( r_1 \) now has two correlative solutions. The number of correlative solutions of \( r_1 \) is the largest and its worst solution (\( p_1 \)) which has the largest value of \( d_{2p_1} \) in \( U_{r_1} \), will be discarded as shown in Figure 5c. The other solutions in the current offspring population will successively update the population until all solutions in the current offspring population participate in the updating process.

As shown in Figure 6a, in the current parents population, reference vectors (\( r_2 \) and \( r_3 \)) have no correlative solution and a reference vector (\( r_1 \)) has three correlative solutions (\( p_1 \), \( p_2 \), and \( p_3 \)). In the selection process, by computing the three entities of solution \( c_1 \), \( c_1 \) will be added to \( U_{r_2} \), the set of solutions which are correlative to \( r_2 \). Then

---

**Algorithm 5 Environmental selection based on correlation**

**Require:** current parents population: \( P_t \in \Omega \), current offspring population: \( Q_t \in \Omega \), current generation: \( t \), set of reference points: \( R \)

1. Procedure \( \text{Environmental\_selection}(P_t, Q_t, R) \)
2. **for each** \( x_k \in Q_t \) **do**
3. \( r_k, d_{1k}, d_{2k} \) \( (r_k, d_{1k}, d_{2k}) \) are three entities of the solution \( x_k \).)
4. \( P_t = P_t \cup \{x_k\} \)
5. \( U_{r_k} = U_{r_k} \cup \{x_k\} \) \( (U_{r_k} \) is the first entity of the reference point \( r_k \).)
6. \( n_{r_k} = n_{r_k} + 1 \) \( (n_{r_k} \) is the second entity of the reference point \( r_k \).)
7. \( \forall r_{\text{max}} \in \{r_m|\forall r \in R: n_{r_m} \geq n_r\} \) \( (n_{r_{\text{max}}} \) denotes the second entity of a reference point \( r_{\text{max}} \).)
8. \( \forall x_{\text{max}} \in \{x_m|\forall x \in U_{r_{\text{max}}}: d_{2x_m} \geq d_{2x}\} \) \( (U_{r_{\text{max}}} \) denotes the first entity of the reference point \( r_{\text{max}} \) and \( d_{2x_m} \) denotes the third entity of a solution \( x_m \).)
9. Remove \( x_{\text{max}} \) from \( P_t \)
10. Remove \( x_{\text{max}} \) from \( U_{r_{\text{max}}} \)
11. \( n_{r_{\text{max}}} = n_{r_{\text{max}}} - 1 \)
12. **end for**

**Ensure:** \( P_t \)
the worst solution ($p_1$) will be removed. So far, the current parents population has been updated by the solution $c_1$.

Here, we need to emphasize three points for the environmental selection based on correlation in the proposed algorithm:

1. We try to find a correlative solution at least for each reference point as soon as possible at the start of the evolutionary process.

2. Once a reference point finds a correlative solution, this means that it has at least one correlative solution in the previous iteration.

3. If a reference point has only one correlative solution, we just remove this solution by replacing it with the other correlative solution with a better third entity.

According to the above explanations, the maintenance of diversity among population members in MOEA/D-CSM is aided by finding at least one related solution for each reference point as far as possible. Nevertheless, the sum of correlative solutions of all reference points must equal the population size. Hence, during the search for the

Figure 5: (a) Current parents population (six red solid dots) and their correlative reference vectors (dotted lines); (b) Distribution of the current population when an offspring $c_1$ is produced before the environmental selection; (c) The distribution of the current population after the environmental selection.
correlative solutions, if there are many correlative solutions for a reference point, we should remove some of the worst solutions from these correlative solutions of the reference point. Here, the worst solution is the one whose third entity is larger than that of others in the correlative solutions of the same reference point. Removing these worst solutions can guide the population closer to the Pareto-optimal front.

3.6 Main Loop

Algorithm 6 provides the main procedure of MOEA/D-CSM. The computation complexity of one generation of MOEA/D-CSM is analyzed here. In Algorithm 4, because there is nothing but a repetition of line 15, reproducing the offspring population by the mating selection operator requires $O(\text{popsize}/2)$ computations. However, in Algorithm 5, firstly, determining three entities of an offspring solution (line 2) requires $O(H \times M)$ computations; secondly, finding a reference point with the largest second entity in all reference points (line 7) requires $O(H)$ computations; finally, determining a solution whose third entity is the largest in all solutions having the same correlative reference point requires $O(\text{popsize})$ (line 8) in the worst case. Because $\text{popsize} \geq H$ is
used in all our simulations, the complexity of our environmental selection operator is $O(popsize \times M)$. Considering all the above considerations, the overall worst-case complexity of one generation of MOEA/D-CSM is $O(popsize^2 \times M)$ which is not worse than that of NSGA-III.

4 Simulation Results

4.1 Experimental Setup

Since MOEA/D-CSM, NSGA-III, RVEA, and MOEA/D are based on the same decomposition approach, we used 3- to 15-objective DTLZ1, DTLZ2, DTLZ3, and DTLZ4 problems (Deb, Thiele et al., 2005; Wang et al., 2019) and UF8, UF9, UF10 (Zhang et al., 2009; Wang et al., 2019) to assess the performance of four algorithms. The implementation of NSGA-III adopted in this article was taken from http://web.ntnu.edu.tw/~tcchiang/publications/nsga3cpp/nsga3cpp.htm; the code of MOEA/D is from http://dces.essex.ac.uk/staff/zhang/webofmoead.htm; and the code of RVEA is from http://www.surrey.ac.uk/cs/people/yaochu_jin/. The number of variables are $(M + k - 1)$, where $k = 10$ for DTLZ2, DTLZ3, and DTLZ4. For UF8-UF10, the number of variables is 30.

Table 1 shows the number of reference points ($H$) for problems with different numbers of objectives. The population size and the number of the reference points in NSGA-III and MOEA/D are suggested in Zhang and Li (2007) and Deb and Jain (2014), respectively, and the population size of RVEA is set as in MOEA/D. In MOEA/D-CSM, the population size is a bit different from that of the other three algorithms. The reason is that the population size of MOEA/D-CSM ($popsize$) should be even in the mating selection section which we have explained in Subsection 3.4, so $popsize$ is equal to $H$ or
Table 1: The number of reference points and population sizes used in MOEA/D-CSM, NSGA-III, MOEA/D, and RVEA.

<table>
<thead>
<tr>
<th>The number of objectives ($M$)</th>
<th>The number of reference points ($H$)</th>
<th>MOEA/D-CSM population size ($popsize$)</th>
<th>NSGA-III population size ($popsize$)</th>
<th>MOEA/D and RVEA population size ($popsize$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>91</td>
<td>92</td>
<td>91</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>210</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td>8</td>
<td>(3, 2)</td>
<td>156</td>
<td>156</td>
<td>156</td>
</tr>
<tr>
<td>10</td>
<td>(3, 2)</td>
<td>275</td>
<td>276</td>
<td>275</td>
</tr>
<tr>
<td>15</td>
<td>(2, 1)</td>
<td>135</td>
<td>136</td>
<td>135</td>
</tr>
</tbody>
</table>

Table 2: Parameter values used in MOEA/D-CSM, NSGA-III, MOEA/D, and RVEA ($n$ is the number of variables).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MOEA/D-CSM</th>
<th>NSGA-III</th>
<th>MOEA/D</th>
<th>RVEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover probability $p_c$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mutation probability $p_m$</td>
<td>$1/n$</td>
<td>$1/n$</td>
<td>$1/n$</td>
<td>$1/n$</td>
</tr>
<tr>
<td>$\eta_c$</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>$\eta_m$</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

$H + 1$ in MOEA/D-CSM. As shown in Table 1, the difference between the population size of the proposed algorithm and that of other algorithms cannot be higher than 2.

Table 2 presents other parameters of the four algorithms used in this study. The neighborhood size $T$ and the penalty parameter are set as 8 and 5 for both MOEA/D-CSM and MOEA/D, respectively. The remaining special parameters in NSGA-III, MOEA/D, and RVEA are recommended by the authors in the respective articles. These common parameters in Table 2 are suggested in Deb and Jain (2014).

4.2 Performance Measures

The inverted generational distance (IGD) (Coello and Cortés, 2005) is used to compare the performance of four algorithms which can provide combined information about the convergence and diversity of the obtained solutions. First, since the true Pareto-optimal surface for all test problems used in this study is known, we can find a set of targeted Pareto-optimal points which are distributed in the true Pareto-optimal surface uniformly or approximately uniformly. Here, the method for generating these Pareto-optimal points is the same used in NSGA-III. We call these targeted Pareto-optimal points set $A$. In addition, we can obtain the final nondominated points in the objective space by using any of the four algorithms and we call them the set $B$. Now, IGD is computed as follows:

$$ IG D(A, B) = \frac{1}{|A|} \sum_{i=1}^{|A|} \min_{j=1}^{|B|} d(a_i, b_j) $$

where $d(a_i, b_j) = ||a_i - b_j||$ is the Euclidean distance between $a_i$ and $b_j$. In our study, $|A|$ is set as the number of reference points $H$. It is quite obvious that the smaller the
IGD metric value, the better the performance of the algorithm. For each test problem, 20 different runs from different initial populations are performed and mean and standard deviation of the IGD values are reported.

4.3 Experimental Results of DTLZ and UF8-U10 Problems and Discussion

In this section, we will present the comparative results of MOEA/D-CSM, NSGA-III, MOEA/D, and RVEA on DTLZ1-4 problems having 3 to 15 objectives and UF8-UF10 with 3 objectives. Table 3 gives the mean IGD and its standard deviation of 20 independent runs.

From Table 3, we can see that the proposed MOEA/D-CSM performs better than NSGA-III, RVEA, and MOEA/D on DTLZ3 having from 3 to 15 objectives. Moreover, because DTLZ3 is used to investigate an algorithm’s ability to converge to the global Pareto-optimal front, it is obvious that the convergence of MOEA/D-CSM is better than that of the three other algorithms considering DTLZ3. IGD results obtained by the proposed algorithm are much better than those of the other algorithms on DTLZ4 with 3 and 5 objectives. However, when the number of objectives of DTLZ4 is larger than 5, IGD results obtained by the proposed algorithm are gradually worse than those of NSGA-III; however, they are still better than those of MOEA/D and RVEA when the number of objectives is 8 or 15. Therefore, when DTLZ4 has fewer than 8 objectives, MOEA/D-CSM performs much better than the other three algorithms. For DTLZ2 having 3 to 19 objectives, IGD results obtained by MOEA/D-CSM are better than those of NSGA-III and RVEA, but they are worse than MOEA/D and for DTLZ2 having 15 objectives. The IGD results of MOEA/D-CSM and MOEA/D are similar. For DTLZ1 having 5 and 8 objectives, the IGD results obtained by MOEA/D-CSM are better than those obtained by the two other algorithms. RVEA is much better than other three algorithms on DTLZ1 with 3 objectives. For DTLZ1 with 10 and 15 objectives, NSGA-III performs better than the three other algorithms. For three UF problems, the performance of MOEA/D is the best.

Out of 23 cases, there are 14 cases in which MOEA/D-CSM performs better than NSGA-III, and there are 17 cases in which MOEA/D-CSM performs better than MOEA/D. Additionally, there are 20 cases in which MOEA/D-CSM performs better than RVEA. Moreover, MOEA/D-CSM performs much better than NSGA-III in 5 cases including DTLZ2 and DTLZ4 with 3 or 5 objectives, and DTLZ3 with 8 objectives. In these cases, MOEA/D-CSM has an index-level-based improvement. MOEA/D-CSM also performs much better than MOEA/D in 8 cases including five-objective DTLZ2, DTLZ3 with 8 or 15 objectives, DTLZ4 with from 3 to 15 objectives. Except for DTLZ1 with 3 objectives and DTLZ4 with 10 objectives, MOEA/D-CSM is better than RVEA in all the other test problems. So, we can conclude that MOEA/D-CSM proposed in this article can be used to deal with many-objective optimization problems and MOEA/D-CSM outperforms NSGA-III, MOEA/D, and RVEA in most cases. For three UF problems with 3 objectives, MOEA/D is the best among the four algorithms.

In order to perform a statistical analysis of results, we adopted the Wilcoxon signed-rank test (Hollander and Wolfe, 1999; Sheskin, 2004) to analyze the IGD results of 20 runs obtained by two algorithms (rank = 1 and rank = 2) under the significance level \( \alpha = 0.05 \); all \( p \)-value results are given in the last column in Table 3. As we know, if the \( p \)-value \( \leq 0.05 \), then it represents that we reject the hypothesis \( H_0 \) at a significance level \( \alpha = 0.05 \), and this means that there exists a significant difference between the best-performance algorithm (rank = 1) and the second best-performance algorithm (rank = 2). Otherwise, if the \( p \)-value \( > 0.05 \), then it denotes that we accept the hypothesis \( H_0 \)
Table 3: Mean IGD and its standard deviation obtained by four algorithms on test problems, where the best result for each problem is shown in boldface and the numbers correspond to rankings.

<table>
<thead>
<tr>
<th>Problem</th>
<th>(M)</th>
<th>(\text{MaxGen})</th>
<th>MOEA/D-CSM</th>
<th>NSGA-III</th>
<th>MOEA/D</th>
<th>RVEA</th>
<th>(p)-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>400</td>
<td>1.075E-3 (4.748E-4)</td>
<td>2</td>
<td>1.826E-3 (1.086E-3)</td>
<td>3</td>
<td>1.990E-3 (1.208E-3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>600</td>
<td><strong>2.780E-4</strong> (1.113E-4)</td>
<td>1</td>
<td>8.923E-4 (3.420E-4)</td>
<td>3</td>
<td>8.328E-4 (3.225E-4)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>750</td>
<td><strong>4.153E-3</strong> (7.305E-3)</td>
<td>1</td>
<td>4.960E-3 (3.863E-3)</td>
<td>3</td>
<td>7.204E-3 (7.363E-4)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1000</td>
<td>5.709E-3 (6.735E-4)</td>
<td>2</td>
<td><strong>3.667E-3</strong> (7.816E-4)</td>
<td>1</td>
<td>6.907E-3 (5.084E-4)</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1500</td>
<td>6.068E-2 (2.456E-2)</td>
<td>4</td>
<td><strong>4.931E-3</strong> (2.113E-3)</td>
<td>1</td>
<td>5.491E-2 (3.271E-3)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>250</td>
<td><strong>6.553E-4</strong> (9.193E-5)</td>
<td>1</td>
<td>1.250E-3 (1.643E-4)</td>
<td>3</td>
<td>7.371E-4 (6.458E-5)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>350</td>
<td><strong>7.259E-4</strong> (1.478E-4)</td>
<td>1</td>
<td><strong>4.507E-3</strong> (3.920E-4)</td>
<td>3</td>
<td>1.652E-3 (1.371E-4)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>500</td>
<td>5.932E-3 (8.043E-4)</td>
<td>2</td>
<td>1.615E-2 (2.269E-3)</td>
<td>4</td>
<td><strong>4.475E-3</strong> (6.474E-4)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>750</td>
<td>1.167E-2 (7.958E-4)</td>
<td>4</td>
<td>1.585E-2 (1.048E-3)</td>
<td>3</td>
<td><strong>4.763E-3</strong> (5.174E-4)</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1000</td>
<td>2.945E-2 (2.579E-2)</td>
<td>2</td>
<td><strong>2.036E-2</strong> (2.438E-3)</td>
<td>1</td>
<td>5.168E-2 (1.166E-2)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>1000</td>
<td><strong>3.175E-3</strong> (2.866E-3)</td>
<td>1</td>
<td>3.855E-3 (1.709E-3)</td>
<td>2</td>
<td>3.988E-3 (2.757E-3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1000</td>
<td><strong>1.059E-3</strong> (6.656E-4)</td>
<td>1</td>
<td>4.998E-3 (2.713E-3)</td>
<td>3</td>
<td>2.043E-3 (6.600E-4)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1000</td>
<td><strong>1.003E-3</strong> (1.695E-3)</td>
<td>1</td>
<td>6.530E-2 (2.149E-3)</td>
<td>3</td>
<td>6.945E-2 (2.580E-1)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1500</td>
<td><strong>1.375E-2</strong> (1.686E-3)</td>
<td>1</td>
<td>1.508E-2 (3.669E-3)</td>
<td>3</td>
<td>5.614E-2 (2.071E-1)</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>2000</td>
<td><strong>1.369E-2</strong> (1.286E-2)</td>
<td>1</td>
<td>4.404E-2 (3.375E-2)</td>
<td>2</td>
<td>6.338E-1 (5.919E-1)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>600</td>
<td><strong>8.604E-5</strong> (1.079E-5)</td>
<td>1</td>
<td>1.065E-1 (2.177E-1)</td>
<td>3</td>
<td>3.228E-1 (3.931E-1)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1000</td>
<td><strong>8.130E-5</strong> (1.373E-5)</td>
<td>1</td>
<td>5.911E-1 (1.313E-1)</td>
<td>2</td>
<td>2.080E-1 (2.790E-1)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1250</td>
<td>4.233E-3 (7.382E-4)</td>
<td>2</td>
<td><strong>4.132E-3</strong> (6.607E-4)</td>
<td>4</td>
<td>3.837E-1 (1.795E-1)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2000</td>
<td>2.454E-2 (1.317E-3)</td>
<td>3</td>
<td><strong>4.380E-3</strong> (4.407E-4)</td>
<td>1</td>
<td>2.279E-1 (1.641E-1)</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>3000</td>
<td>2.527E-2 (3.067E-2)</td>
<td>2</td>
<td><strong>7.215E-3</strong> (1.123E-2)</td>
<td>1</td>
<td>4.425E-1 (1.273E-1)</td>
</tr>
<tr>
<td>UF8</td>
<td>3</td>
<td>300</td>
<td>1.704E-1 (7.102E-5)</td>
<td>3</td>
<td>1.240E-1 (9.771E-5)</td>
<td>2</td>
<td>3.310E-2 (4.432E-5)</td>
</tr>
<tr>
<td>UF9</td>
<td>3</td>
<td>300</td>
<td>2.133E-1 (6.225E-4)</td>
<td>3</td>
<td>1.187E-1 (7.918E-4)</td>
<td>2</td>
<td><strong>2.867E-2</strong> (2.713E-5)</td>
</tr>
<tr>
<td>UF10</td>
<td>3</td>
<td>300</td>
<td>1.291E-1 (4.123E-5)</td>
<td>3</td>
<td>7.497E-2 (2.971E-6)</td>
<td>2</td>
<td><strong>7.042E-2</strong> (4.254E-4)</td>
</tr>
</tbody>
</table>
at the significance level $\alpha = 0.05$, there does not exist significant difference between the two algorithms (rank = 1 and rank = 2). For example, for DTLZ1 (5), $p$-value = 0 as shown in Table 3 and $p$-value < 0.05, it means that MOEA/D-CSM (rank = 1) is significantly better than MOEA/D (rank = 2); and for DTLZ3 (3), $p$-value = 0.1560 (in Table 3) and $p$-value > 0.05, it represents that it is not significantly better than NSGA-III although MOEA/D-CSM (rank = 1) is better than the NSGA-III (rank = 2).

In summary, when its rank is 1, the proposed algorithm MOEA/D-CSM is significantly better than those algorithms with rank = 2 on 9 test problems and not significantly better than comparative algorithms on 2 test problems out of 23 test problems according to the obtained $p$-value. As for NSGA-III, when its rank = 1, there are only 3 test problems in which NSGA-III is significantly better than those algorithms with rank = 2. Similarly, MOEA/D is significantly better than the second rank algorithms on DTLZ2 (8), DTLZ2 (10), and UF8-UF10. RVEA performs only significantly better than the second rank algorithm on DTLZ1 (3). The statistical results also show the superiority of the proposed algorithm.

Since Table 3 shows the mean and standard deviations of the IGD values, we cannot investigate the stability of these algorithms. In this section, Figures 7–11 give boxplots of IGD results on DTLZ1-4 and UF8-UF10 obtained by the four algorithms to show the stability of the four algorithms. From Figure 7, we can see that MOEA/D-CSM is more stable than the three other algorithms when DTLZ1 has five objectives. But for DTLZ1 having 15 objectives, the stability of MOEA/D-CSM is worse. From Figure 8, we can also find that the stability of MOEA/D-CSM is better than NSGA-III for DTLZ2 having three to ten objectives. From Figure 9, it is known that the stability of MOEA/D-CSM is better than that of the other algorithms when DTLZ3 has more objectives. From Figure 10, we can find that the stability of MOEA/D-CSM is good when DTLZ4 has 8 to 15 objectives. For UF test problems with 3 objectives, MOEA/D is presented in Figure 11.

4.4 Discussion on the Setting of the Ideal Point

In all of the above experiments, we set the original point as the ideal point. One reason is that it can reduce the computational cost and we do not need to update it during the evolutionary process; the other reason is that doing so can magnify the selection pressure toward the optimal solution when we apply algorithms to solve a minimization problem and it can speed up convergence to the optimal PF.

Here, we make a comparison between two cases: case 1: the original point is set as the ideal point; case 2: the ideal point is generally set as the best value of each objective in the current population, that is, $z^* = \min \{ f_i(x) | x \in S \}$, which denotes a set of solutions in the current generation. We provide the mean IGD values and the running times of 20 independent runs of MOEA/D-CSM in Table 4.

Table 4 clearly shows that the IGD values obtained at the two cases are very similar. As expected, the algorithm takes a relatively short span of time to converge to the optimal PF when setting the original point as the ideal point.

5 Comparison of MOEA/D-CSM with MOEA/D-STM

In this section, we first introduce the stable matching-based selection in evolutionary multiobjective optimization (MOEA/D-STM) proposed by Li et al. (2014). Then, the differences between MOEA/D-CSM and MOEA/D-STM are presented. Finally, MOEA/D-CSM and MOEA/D-STM are compared through many experiments.
Figure 7: Boxplots of IGD results obtained by the four algorithms for DTLZ1 having 3 to 15 objectives in 20 independent runs.
Figure 8: Boxplots of IGD results obtained by the four algorithms for DTLZ2 having 3 to 15 objectives in 20 independent runs.
Figure 9: Boxplots of IGD results obtained by the four algorithms for DTLZ3 having 3 to 15 objectives in 20 independent runs.
Figure 10: Boxplots of IGD results obtained by the four algorithms for DTLZ4 having 3 to 15 objectives in 20 independent runs.
Figure 11: Boxplots of IGD results obtained by the four algorithms for UF8, UF9, UF10 in 20 independent runs.

5.1 Introduction of MOEA/D-STM

MOEA/D-STM mainly proposed a new selection mechanism, called the STM model, which is based on MOEA/D. This section will explain the STM model in detail. As we know, MOEA/D decomposes a multiobjective optimization problem into a set of scalar optimization subproblems and optimizes them in a collaborative manner, so there always exist two sets; that is, subproblems and solutions existed in MOEA/D in a natural fashion. In some versions of MOEA/D, each subproblem indirectly chooses its best solution according to a scalar function at each generation. Therefore, a solution may be chosen by more than one subproblem so that the diversity of the current population may be influenced. In MOEA/D-STM, STM proposes a bidirectional selection model between these subproblems and solutions. Specific details are shown in Figure 12. First, according to the scalar optimization, a subproblem $P_0$ looks for its best solution which exists in the following two cases: one case is that this solution has not been chosen by other subproblems; another is that this solution has been chosen as its best solution by another subproblem $P_1$. If this solution has been chosen as the best solution of another subproblem $P_1$, this solution has the opportunity of choosing a subproblem.
Table 4: Mean IGD results and mean running times obtained for two cases on some UF and DTLZ test problems with three objectives, where the best results are shown in boldface.

<table>
<thead>
<tr>
<th>Test problem</th>
<th>Case 1: Set the original point as the ideal point</th>
<th>Case 2: Set the best objective function value as the ideal point</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>IGD</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>1.075E-3</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>6.553E-4</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>3.175E-3</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>8.604E-5</td>
</tr>
<tr>
<td>UF8</td>
<td>3</td>
<td>1.704E-1</td>
</tr>
<tr>
<td>UF9</td>
<td>3</td>
<td>2.133E-1</td>
</tr>
<tr>
<td>UF10</td>
<td>3</td>
<td>1.291E-1</td>
</tr>
</tbody>
</table>

Figure 12: Simple example of STM where \( g(x|\lambda, z^*) \) denotes the scalar function and \( d \) denotes the distance between a solution and a subproblem. \( \{S_2, S_3, S_4, S_5, S_1\} \) denotes the ascending order of scalar function \( g(x|\lambda, z^*) \), that is to say, \( S_2 \) is the best solution of the subproblem, \( P_1 \) and \( S_1 \) is the second and so on. \( \{P_2, P_4, P_3, P_1, P_5\} \) denotes the ascending order of the distance between the solution \( S_1 \) and all subproblems.

According to the distance \( d_0 \) (denoting the distance between the solution and the subproblem \( P_0 \)) and \( d_1 \) (denoting the distance between the solution and the subproblem \( P_1 \)), if \( d_0 \) is smaller than \( d_1 \), the solution is chosen as the best solution of the subproblem \( P_0 \), and then the subproblem \( P_1 \) gives up the solution and looks for the second good solution as its best solution according to the scalar function. But the second good solution may exist in the two above mentioned cases, and these cases are processed in the same way.

Although two algorithms are essentially decomposition-based evolutionary multiobjective optimization algorithms, they are different from each other including the
mechanism for generating reference points, the decomposition method, the solution selection mechanism and the solution updating strategy, as discussed next:

1. When the number of objectives is more than 8, the proposed algorithm adopts the two-layered method to generate the reference vectors while MOEA/D-STM adopts the simplex lattice design method.

2. The decomposition method in MOEA/D-STM is the weighted TCH approach while MOEA/D-CSM uses PBI.

3. The selection of MOEA/D-CSM is based on the correlation between the reference points and solutions, and MOEA/D-STM is based on the relationship between the subproblems and solutions.

4. The genetic operators in MOEA/D-STM include the differential evolution (DE) operator and the polynomial-based mutation (PM), and in MOEA/D-CSM, we use PM and SBX to update the population and by using the proposed mating selection based on correlation, the parents are selected to execute the genetic operators, while MOEA/D-STM randomly selects parent solutions to carry out DE and PM.

5. In the proposed algorithm, after we got an offspring population, we select a set of solutions by using the environmental selection operator to generate the new population in the next generation, while in MOEA/D-STM, for each subproblem, it first selects some solutions from the parents population and from the offspring population according to nondominated sorting. Then, the subproblem and its corresponding solutions are updated by using the STM model.

5.2 Comparison of Performance of MOEA/D-STM Using Two Different Crossover Operators

MOEA/D-STM introduced the STM mechanism into MOEA/D-DRA (Zhang et al., 2009) framework—because the differential evolution (DE) operator, which is proposed for those continuous multiobjective optimization test instances with arbitrary prescribed PS shapes, is used in MOEA/D-DRA. MOEA/D-STM, which also uses the DE operator may be appropriate for UF (Zhang et al., 2008) test problems but not appropriate for the DTLZ test problems. In this subsection, to demonstrate this statement that the DE operator is suitable for UF test problems and that the SBX operator is suitable for DTLZ test problems, we give a performance demonstration when the SBX operator is incorporated into MOEA/D-STM.

All parameter settings of MOEA/D-STM when using the DE operator are the same as the original paper (Li et al., 2014). But the original paper did not apply the SBX operator into MOEA/D-STM and also did not test on the DTLZ problems. Hence, when the SBX operator is used into MOEA/D-STM, the crossover probability $p_c = 1.0$ and its distribution index is set to be 20. When MOEA/D-STM is applied into the DTLZ test problems, the population size, the number of reference points, and the maximum number of generations are shown in Section 4. Table 5 shows the mean IGD values and their standard deviations over 20 independent runs on the UF test problems.

From Table 5, we can see that MOEA/D-STM using the DE operator is better than using the SBX operator except for UF1, UF4, and UF7 on ten UF test problems. While for four DTLZ problems with 3 objectives, MOEA/D-STM using the SBX operator outperforms the use of the DE operator on three cases including DTLZ1, DTLZ2, and DTLZ4.
Table 5: Mean IGD values and its standard deviation obtained by MOEA/D-STM + SBX and MOEA/D-STM + DE on the UF and DTLZ test problems, where the best results for each problem are shown in boldface and the values in the fourth and sixth columns correspond to the ranking of the two algorithms.

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>MOEA/D-STM + SBX</th>
<th>MOEA/D-STM + DE</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>2</td>
<td>4.288E-02(2.446E-04)</td>
<td>4.686E-02(9.617E-03)</td>
<td>2 0.0571</td>
</tr>
<tr>
<td>UF2</td>
<td>2</td>
<td>1.945E-02(1.426E-05)</td>
<td>3.321E-03(1.369E-07)</td>
<td>1 0</td>
</tr>
<tr>
<td>UF3</td>
<td>2</td>
<td>7.793E-02(1.405E-04)</td>
<td>4.887E-02(9.445E-03)</td>
<td>1 0.0014</td>
</tr>
<tr>
<td>UF4</td>
<td>2</td>
<td>3.713E-02(2.437E-07)</td>
<td>4.532E-02(5.474E-04)</td>
<td>2 0.2536</td>
</tr>
<tr>
<td>UF5</td>
<td>2</td>
<td>2.533E-01(2.729E-06)</td>
<td>2.331E-01(1.475E-03)</td>
<td>1 0</td>
</tr>
<tr>
<td>UF6</td>
<td>2</td>
<td>9.758E-02(8.681E-04)</td>
<td>8.706E-02(1.384E-03)</td>
<td>1 0.0545</td>
</tr>
<tr>
<td>UF7</td>
<td>2</td>
<td>1.847E-02(8.603E-06)</td>
<td>2.870E-02(3.884E-03)</td>
<td>3 0.0326</td>
</tr>
<tr>
<td>UF8</td>
<td>2</td>
<td>7.656E-02(1.423E-05)</td>
<td>7.641E-02(1.629E-05)</td>
<td>2 0.2878</td>
</tr>
<tr>
<td>UF9</td>
<td>3</td>
<td>8.753E-02(3.915E-05)</td>
<td>6.661E-02(6.823E-05)</td>
<td>1 0</td>
</tr>
<tr>
<td>UF10</td>
<td>3</td>
<td>2.246E+00(1.140E-02)</td>
<td>2.035E+00(1.228E-02)</td>
<td>2 0</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>3.799E-03(5.926E-06)</td>
<td>1.866E-02(9.141E-05)</td>
<td>1 0</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>1.346E-04(8.323E-05)</td>
<td>5.124E-02(1.476E-04)</td>
<td>2 0</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>1.477E+00(4.082E+00)</td>
<td>5.220E-02(3.714E-06)</td>
<td>1 0</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>1.346E-02(1.161E-05)</td>
<td>5.127E-02(1.150E-05)</td>
<td>2 0.0025</td>
</tr>
</tbody>
</table>

By these simple comparisons, it is concluded that MOEA/D-STM using the DE operator is more appropriate for the UF problems than for the DTLZ problems.

Furthermore, in order to have a statistical analysis of results, we adopt the Wilcoxon signed-rank test to analyze the IGD results of 20 runs obtained by two algorithms (rank = 1 and rank = 2) under a significance level $\alpha = 0.05$; all $p$-value results are given in the last column in Table 5.

It is easy to see that MOEA/D-STM + DE (when its rank = 1) is significantly better than MOEA/D-STM + SBX on 5 out of 10 UF test problems while there are 3 out of 4 test DTLZ problems in which MOEA/D-STM + SBX performs significantly better than MOEA/D-STM + DE. By analyzing the statistical results, we can conclude that MOEA/D-STM + SBX is more suitable for solving the DTLZ problems as we expected.

Since most of the parameters used in MOEA/D-STM are from the original paper, in which the ideal point is not the original point, in order to make a fair comparison in this section, MOEA/D-CSM also applies the same method of creating an ideal point as MOEA/D-STM. MOEA/D-STM used the minimum function values on each objective in the current population as the ideal point; that is, $z^*_i = \min\{f_i(x)|x \in P\}$, for all $i \in \{1, \ldots, M\}$, where $P$ is a set of solutions that have been created. In this section, population size, the number of reference points and the maximum number of generations required by all experiments are given in Tables 1 and 3.

To compare the performance of the proposed algorithm and MOEA/D-STM + SBX, we applied them to the DTLZ problems with 3, 5, and 8 objectives, respectively. Table 6 shows the mean IGD values and their standard deviation over 20 independent runs. Ranks 1 and 2, according to the average IGD values, are analyzed by using the Wilcoxon signed-rank test and the significance level $\alpha$ is set as 0.05.

From Table 6, we can see that there are eleven out of twelve cases in which MOEA/D-CSM performs better than MOEA/D-STM for DTLZ1-4 problems, especially...
Table 6: Mean IGD values and the standard deviations obtained by MOEA/D-CSM and MOEA/D-STM + SBX on the DTLZ test problems, where the best results for each problem are shown in boldface and the values in the fourth and sixth columns correspond to the ranking of the two algorithms compared.

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>MOEA/D-CSM</th>
<th>MOEA/D-STM + SBX</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>1.074E-03(2.141E-03)</td>
<td>3.799E-03(5.926E-06)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2.779E-04(1.177E-05)</td>
<td>1.349E-02(1.182E-06)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4.153E-03(5.070E-07)</td>
<td>4.684E-03(2.105E-06)</td>
<td>2 0.0793</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>6.553E-04(9.866E-08)</td>
<td>1.346E-04(8.323E-05)</td>
<td>1 0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7.259E-04(2.074E-04)</td>
<td>1.355E-03(2.984E-06)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>5.932E-03(6.145E-05)</td>
<td>1.452E-02(8.937E-06)</td>
<td>1 0.0026</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>3.175E-03(2.866E-03)</td>
<td>1.477E+00(4.082E+00)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.059E-03(6.656E-04)</td>
<td>1.355E-03(5.873E-05)</td>
<td>2 0.0017</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.003E-03(1.695E-03)</td>
<td>4.606E-03(9.523E-06)</td>
<td>2 0</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>8.604E-05(1.079E-05)</td>
<td>1.346E+00(1.161E-05)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8.130E-05(1.373E-05)</td>
<td>1.527E-01(5.008E-05)</td>
<td>2 0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4.233E-03(7.382E-04)</td>
<td>2.649E-01(1.132E-05)</td>
<td>2 0.0031</td>
</tr>
</tbody>
</table>

on seven test problems including DTLZ1 with 5 objectives, DTLZ2 with 5 and 8 objectives, DTLZ3 with 3 objectives, and DTLZ with 3, 5, and 8 objectives; MOEA/D-CSM largely outperforms MOEA/D-STM. There are five test problems in which MOEA/D-CSM obtains a lower standard deviation. But this does not mean that MOEA/D-STM has a poor performance. And we can see that it is more suitable to solve UF test problems as mentioned in its original paper. That is, MOEA/D-STM is appropriate for multiobjective optimization problems with arbitrary prescribed PS shapes, i.e. UF test problems, but it may not be appropriate for many-objective optimization problems with simple PS shapes.

Furthermore, to perform a statistical analysis of results, we adopt the Wilcoxon signed-rank test to analyze the IGD results of 20 runs obtained by two algorithms (rank = 1 and rank = 2) under significance level $\alpha = 0.05$; all $p$-value results are given in the last column in Table 6.

In summary, our proposed MOEA/D-CSM is significantly better than MOEA/D-STM + SBX on 10 test problems and not significantly better than comparative algorithms on a test problem out of 12 test problems according to the obtained $p$-value. As for MOEA/D-STM + SBX, when its rank = 1, there is a test problem in which MOEA/D-STM + SBX is significantly better than MOEA/D-CSM. The statistical results also show the superiority of the proposed algorithm MOEA/D-CSM.

Furthermore, to make a visual comparison between MOEA/D-CSM, MOEA/D-STM + SBX, and MOEA/D-STM + DE, we show the final obtained PF by the three algorithms on all the 3-objective test problems [UF8-UF10, DTLZ1 (3), DTLZ2 (3), DTLZ3 (3), DTLZ4 (3)] in Figures 13–19.

Figures 13–19 show that the proposed MOEA/D-CSM has good convergence performance on 4 test problems [UF10, DTLZ1(3), DTLZ2(3), and DTLZ4 (3)] and it can obtain a set of reasonable and uniformly distributed solutions very close to the ideal PF when solving the DTLZ test problems, and MOEA/D-STM + DE is more suitable to deal with UF problems.
Figure 13: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on UF8.

Figure 14: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on UF9.
Figure 15: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on UF10.

Figure 16: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on DTLZ1(3).
Figure 17: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on DTLZ2(3).

Figure 18: The final PFs obtained by (a) MOEA/D-CSM, (b) MOEA/D-STM + SBX, (c) MOEA/D-STM + DE on DTLZ3(3).
The proposed correlative selection mechanism is based on PBI decomposition. Therefore, this mechanism could be embedded into any PBI decomposition-based MOEA/Ds. For example, MOEA/D-RBF (Zapotecas-Martínez and Coello, 2013), which uses the PBI decomposition mechanism, can adopt the proposed correlative selection mechanism, while for MOEA/D-EGO (Zhang et al., 2010), it may be required to adjust the proposed correlative selection mechanism since it adopts the Tchebycheff decomposition mechanism.

6 Conclusion

This article proposes a decomposition-based evolutionary algorithm with a correlative selection mechanism (MOEA/D-CSM) to solve many-objective optimization problems. The new selection mechanism includes the mating selection and environmental selection mechanisms, which are based on correlation. In order to improve the selection mechanism, the correlation between a reference point and a solution, three entities of a reference point and three entities of a solution are introduced. Based on these entities, the mating selection and the environmental selection are used to evolve and update the population. We have carried out systematic experiments to compare our proposed algorithm with three other elitist many-objective evolutionary algorithms (NSGA-III, MOEA/D, and RVEA). Results show that the proposed MOEA/D-CSM is found to produce satisfactory results on most problems considered in this study.

However, in this article, all the test problems adopted have an identical range of values for each objective. According to the distribution of the reference points, we found out that the decomposition-based approach used in this article is very suitable for
solving these test problems. As part of our future work, we will improve our algorithm so that it can properly solve more many-objective optimization problems with different characteristics.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61876141 and 61373111), and the Provincial Natural Science Foundation of Shaanxi of China (No. 2019JZ-26), and the Opening Project of Science and Technology on Reliability Physics and Application Technology of Electronic Component Laboratory (No. 614280620190403-1).

References


A Decomposition-Based Evolutionary Algorithm


