2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

An enhanced MOEA/D using uniform directions and a pre-organization procedure

Rui Wang, Tao Zhang and Bo Guo
Department of Systems Engineering
College of Information Systems and Management
National University of Defense Technology
ChangSha, China, 410073

***Abstract*—Multi-objective evolutionary algorithm based on decomposition (MOEA/D) has become increasingly popular in solving multi-objective problems (MOPs). In MOEA/D, weight vectors are responsible for maintaining a nice distribution of Pareto optimal solutions. Often, we expect to obtain a set of uniformly distributed solutions by applying a set of uniformly distributed weight vectors in MOEA/D. In this paper, we argue that uniformly distributed weights do not produce uniformly distributed solutions, however, uniformly distributed search directions do. Moreover, we propose to perform a pre-*organization* procedure before running MOEA/D. The procedure matches each weight to its closet candidate solution. Experimental results show (i) MOEA/D with uniformly distributed search directions would exhibit a better diversity performance, and (ii) MOEA/D with the pre-*organization* procedure performs better, especially for the convergence performance.**

I. Introduction

MUlti-objective problems (MOPs) regularly arise in real-world design scenarios, where two or more objectives are required to be optimized simultaneously. As such objectives are often in competition with one another, the optimal solution of MOPs is a set of trade-off solutions, rather than a single solution. The population-based approach: multi-objective evolutionary algorithms (MOEAs) are well suited for solving MOPs since they lead naturally to the generation of an approximate trade-off surface (or Pareto front) in a run [1].

A variety of MOEA approaches has been proposed among which Pareto dominance based MOEAs (e.g. MOGA [2], [3], NSGA-II [4], SPEA2 [5]) were one of the earliest approaches. It is accepted that they perform well on MOPs with 2 and 3 objectives. However, their search capability often degrades significantly as the number of objectives increase [6]. This is because the proportion of Pareto-optimal (or non-dominated) objective vectors in the population grows large when MOPs have more than 3 objectives i.e. many-objective problems. As a result, insufficient selection pressure can be generated toward the Pareto front [7], [8], [9].

Multi-objective evolutionary algorithms based on decomposition, i.e. MOEA/D [10], has a number of advantages, compared with Pareto dominance based MOEAs, such as the scalability to many-objective problems (MOPs with more than 3 objectives)[1], high search ability for combinatorial optimization, computational efficiency of fitness evaluation, high compatibility with local search and good performance on problems with complex Pareto sets [18]. MOEA/D, the winner of the "Unconstrained multi-objective evolutionary algorithm" competition at the 2009 Congress on Evolutionary Computation [19], has been used in many real-world applications[2].

Many work has been done to further improve the performance of MOEA/D, among which a large body of work is about the generation of weights. The method used in Zhang and Li's MOEA/D is not flexible enough to generate an arbitrary number of weights which implicitly brings a problem for the setting of population size as in their approach populations size should be equal to the number of weights. To solve this issue, Tan et al [20] analyse three different uniform design methods, i.e. simplex-lattice design (the method used in MOEA/D [10], [18], [19]), simplex-centroid design and axial design, based on the analysis, a new uniform design method is developed. Their method is able to generate an arbitrary number of uniformly distributed weights. However, they did not consider the issue that weight vectors (used in MOEA/D) should be different for problems having different Pareto optimal fronts. In the paper of Jiang et al. [21], a method, called Pareto-adaptive weight vector, is proposed. This method not only can generate an arbitrary number of weights but also can adjust the distribution of weights according to the shape of Pareto optimal front. However, this method is only shown to be effective on sphere-like Pareto front.

In this paper, we study how an optimal weight $\mathbf{w}$ is identified for obtaining a required solution $\mathbf{s}$ by MOEA/D. As a preliminary study, this issue is only studied when using Chebyshev scalarizing function (g^{te}) based MOEA/D. It is found that the optimal $\mathbf{w}$ for a required $\mathbf{s}$ is equal to the normalized reciprocal of the objective values of $\mathbf{s}$. In other words, by using g^{te} based MOEA/D [10], the optimal solution corresponds to $\mathbf{w}$ is actually along the vector $1/\mathbf{w}$ (denoted as $\boldsymbol{\lambda}$). This suggests an easy way to find the optimal

The work of Rui Wang (ruiwangnudt@gmail.com) was supported by the China Scholarship Council and the National Science Foundation of China (No. 70971132)

[1] In addition to MOEA/D, some other MOEAs also have good performance on many-objective problems, e.g. PICEA-g [11], [12], [13], [14], [15], HypE[16] and MSOPS[17].

[2] Professor Qingfu Zhang maintains a website where most of the related research and applications of MOEA/D are recorded: http://dces.essex.ac.uk/staff/zhang/webofmoead.htm

weights for obtaining uniformly distributed solutions. First we get the search directions $\boldsymbol{\lambda}$ which starts from the *ideal* point to those required solutions. Second we transform these search directions into the required weights according to the normalized reciprocal relation.

Additionally, in Zhang and Li's MOEA/D [10], the initial candidate solutions are randomly assigned to the uniformly distributed weights. We argue that such a random assignment may slow down the search speed of MOEA/D. Imagining candidate solution $\mathbf{s}_1$ is assigned a weight vector $\mathbf{w}_1$, as shown Figure 1. It is obvious that $\mathbf{s}_1$ is not as good as $\mathbf{s}_2$ for obtaining the optimal solution $\mathbf{s}_0$ when using the weight $\mathbf{w}_1$. This is because $\mathbf{s}_2$ is right along the direction $\boldsymbol{\lambda}_1$ (where $\boldsymbol{\lambda}_1$ is equal to $1/\mathbf{w}_1$) while $\mathbf{s}_1$ is not, which means more effort is required to guide $\mathbf{s}_1$ toward $\mathbf{s}_0$. Therefore, we suggest to perform a pre-*organization* procedure which assigns each weight with an adjacent candidate solution before the search process.

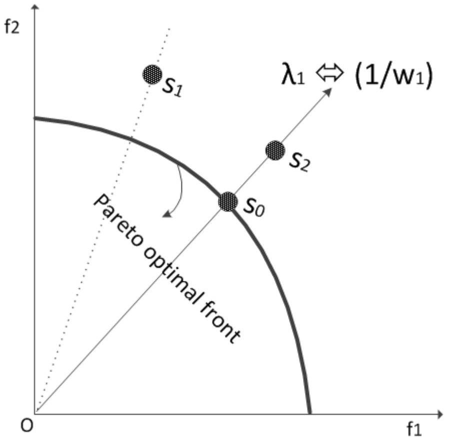


Fig. 1. Illustration of the unmatched candidate solution and weight ($\boldsymbol{\lambda_1}$ is the reciprocal of $\mathbf{w}_1$)

The reminder of the paper is organized as follows: in Section II the original MOEA/D [10] is briefly reviewed. This is followed, in Section III, by the analysis of the normalized reciprocal relation and the description of the pre-*organization* procedure. The effectiveness of these two strategies are studied in Section IV. Section V concludes and discusses the future research.

II. MOEA/D ALGORITHM

Without loss of generality, let us consider the following multi-objective minimization problem:

$$Minimize\ \ F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_M(\mathbf{x})\} \quad (1)$$

where M is the number of objectives, $F(\mathbf{x})$ is the M-dimensional objective vector, $f_i(\mathbf{x})$ is the ith objective to be minimized, and $\mathbf{x}$ is the decision vector.

MOEA/D requires a decomposition approach for converting approximation of the Pareto front of a MOP into a number of single objective optimization problems (SOPs). Each SOP is defined by a scalarizing function with a different weight vector. In principle, any scalarizing function can be used. The widely used ones are weighted-sum and weighted Chebyshev.

(1) the weighted sum scalarizing function with a non-negative weight vector $\mathbf{w}$ is written as follows:

$$g^{ws}(\mathbf{x}|\mathbf{w}) = \sum_{i=1,2,\cdots,M} w_i f_i(\mathbf{x}) \quad (2)$$

where $\mathbf{w}_i \geq 0$ and $\sum_{i=1}^{M} w_i = 1$. $g^{ws}(\mathbf{x}|\mathbf{w})$ is minimized.

(2) the weighted Chebyshev scalarizing function with a reference point $\mathbf{z}^* = \{z_1^*, z_1^*, \cdots, z_M^*\}$ is written as follows:

$$g^{te}(\mathbf{x}|\mathbf{w}) = \max_{i=1,2,\cdots,M} \{w_i(f_i(\mathbf{x}) - z_i^*)\} \quad (3)$$

where $\mathbf{z}^*$ is updated once a better (smaller) value of f_i is found during the execution of MOEA/D. See Equation 4

$$z_i^* = \min\{f_i(\mathbf{x})|\mathbf{x} \in \mathbf{\Omega}\} \quad (4)$$

where $\mathbf{\Omega}$ shows all the examined solutions during the execution of MOEA/D.

In MOEA/D, each weight vector has a candidate solution in the current population, i.e. the population size is the same as the number of weight vectors. Each weighted scalarizing fitness function $g(\mathbf{x}|\mathbf{w}_i)$ identifies a single solution which is the best with respect to that scalarizing fitness function. For each SOP, a new solution is generated by performing genetic operators on several solutions from amongst its neighbours. Neighbours are defined based on the Euclidean distance between the weight vectors. A SOP i is a neighbour of SOP j if the weight vector of SOP i is close to that of SOP j. The newly generated solution is compared with all of its neighbours. If the new solution is better, then some (or all) of its corresponding neighbours are replaced by the new solution. In the first MOEA/D version [10], the new solution will replace all the neighbours that are worse than itself. However, in order to maintain a better diversity, in [18], an upper bound (nr) is defined to limit the maximum number of replacements. Principally, the diversity of solutions is maintained by a number of uniformly distributed weight vectors in MOEA/D. As mentioned in Section I, many methods can be used to generate uniformly distributed weights. MOEA/D adopts the simplex-lattice design method. That is, $w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \cdots \frac{H}{H}\right\}, \quad i = 1, 2, \cdots, M$ and $w_1 + w_2 + \cdots + w_M = 1$. where H is a user-definable positive integer. The number of weight vectors is calculated as $Nw = C_{H+M-1}^{M-1}$ [10], where C stands for the combination formula. For example, for 2-objective problems, if H is specified as 100, then we can generate $C_{101}^{1} = 101$ groups of weight vectors (0, 1), (0.01, 0.99), ... , (1, 0). The pseudo-code of MOEA/D is presented in Algorithm 1.

III. TWO ISSUES OF MOEA/D

A. Uniform directions or uniform weights

Despite the Pareto front geometry, it is expected that by using uniformly distributed weight vectors, MOEA/D can identify uniformly distributed solutions. However, this is not

Algorithm 1: MOEA/D

```
Input: initial candidate solutions, $\mathbf{s}$ of size $N$ , initial
       weight vectors, $\mathbf{w}$ of size $Nw$, selection
       neighbourhood size, $T$, replacement
       neighbourhood size, $nr$, an external population
       ($EP$)
Output: $\mathbf{s}$
(1) Initialize the weight vectors and candidate solutions ;
(2) Randomly assign each weight, $\mathbf{w}_i$ with a candidate
    solution, $\mathbf{s}_i$ ;
(3) Calculate the Euclidean distance among the weights ;
while each $\mathbf{w}_i$ do
    (4) Find the $T$ neighbouring weights $B(\mathbf{w}_i)$ of $\mathbf{w}_i$
    based on the distance of weights and identify the
    related neighbouring solutions $B(\mathbf{s}_i)$ of $\mathbf{s}_i$ ;
    (5) Randomly select two indexes, $k, l$ from $T$
    neighbourhood ;
    (6) Generate a new offspring $\mathbf{s}'$ by performing
    genetic variation operators on the $\mathbf{s}^k, \mathbf{s}^l$ ;
    (7) If the Chebyshev scalarizing function is applied,
    the reference point $z_i^*$ should also be updated when
    $f_i(\mathbf{s}') < z_i^*$ ;
    (8) Randomly select $nr$ indexes from $T$
    neighbourhood ;
    (9) Update the selected neighbouring solutions
    $B(\mathbf{s}_{nr})$ if $g(\mathbf{s}'|w_i)$ is better than
    $g(\mathbf{s}_j|\mathbf{w}_i), \mathbf{s}_j \in B(\mathbf{s}_{nr})$ ;
    (10) Store the non-dominated solutions into $EP$;
end
```

the case even for a problem whose Pareto front is an unit hyperplane. In this section we explain why cannot the uniformly distributed weights produce uniformly distributed solutions by g^{te} based MOEA/D.

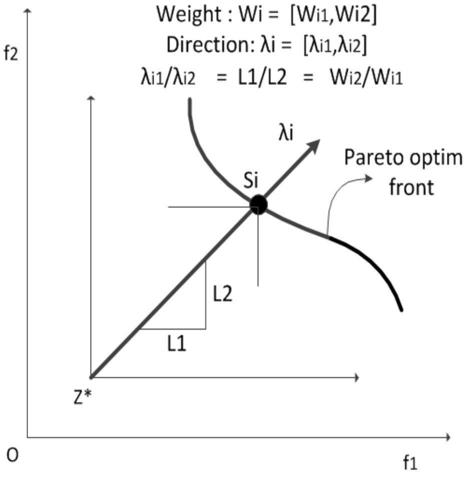


Fig. 2. Illustration of the rationale for Chebyshev scalarzing function

Figure 2 illustrates how an optimal solution is obtained based on $g^{te}(\mathbf{x}|\mathbf{w}_i)$, where $\mathbf{w}_i = [w_{i1}, w_{i2}]$. First, we subtract the objective values by $\mathbf{z}^*$, i.e. transforming the coordinate origin into $\mathbf{z}^*$. Then, the optimal solution for $\mathbf{w}_i$ is identified once the negative orthant cone is shifted along the direction $\lambda_{ij} = (1/w_{ij})/\sum_{j=1}^{j=M}(1/w_{ij})$, i.e. the cone touches the Pareto optimal front. The optimal solution is the intersection point, $\mathbf{s}_i$ shown in Figure 2.

However, it is always neglected that $\frac{w_{i1}}{w_{i2}}$ is NOT equal to $\frac{L_1}{L_2}$ but is equal to $\frac{L_2}{L_1}$. Also, $\frac{L_2}{L_1} = \frac{f_1(\mathbf{s}_i)}{f_2(\mathbf{s}_i)} = \frac{\lambda_{i1}}{\lambda_{i2}}$ [22]. Here, $\boldsymbol{\lambda}_i = [\lambda_{i1}, \lambda_{i2}]$ represents search direction which starts from $\mathbf{z}^*$ to the required solution ($\mathbf{s}_i$) in the objective space, $\sum_{j=1}^{j=M} \lambda_{ij} = 1$.

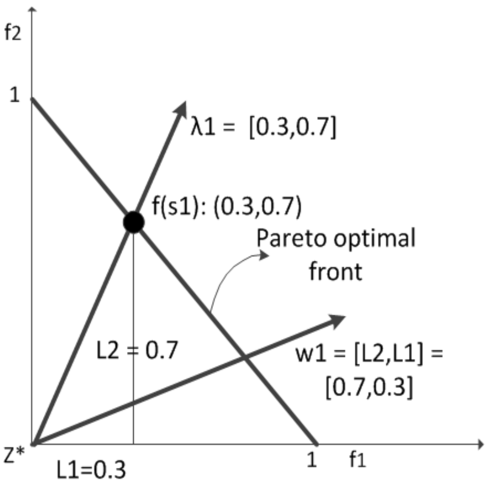


Fig. 3. The relation between search direction ($\boldsymbol{\lambda}$) and weight vector ($\mathbf{w}$): a bi-objective problem

To further explain the rationale of g^{te}, let us consider a bi-objective minimization problem as shown in Figure 3. Assuming the Pareto optimal front is a straight line $f_1 + f_2 = 1$. Given a weight vector $\mathbf{w}_1 = [0.7, 0.3]$, we have $\frac{L_2}{L_1} = \frac{w_{11}}{w_{12}} = \frac{7}{3}$. Thus, according to Figure 2 (as well as the definition of $g^{te}(\mathbf{x}|\mathbf{w})$, Equation 3), the optimal solution identified by $g^{te}(\mathbf{x}|\mathbf{w}_1)$ is $f(\mathbf{s}_1)$: $(0.3, 0.7)$. Give a closer examination, we could find that $(\frac{1}{w_{11}})/(\frac{1}{w_{12}}) = f_1(s_1)/f_1(s_2) = \lambda_{11}/\lambda_{12}$. In other words, the solution identified by $g^{te}(\mathbf{x}|\mathbf{w}_i)$ is not along the direction of $\mathbf{w}_i$ but the direction of $1/\mathbf{w}_i$. It is known that when $\mathbf{w}$ is uniformly distributed, $1/\mathbf{w}$ are often not uniformly distributed. Therefore, the obtained solutions are not uniformly distributed. This is the reason that uniformly distributed weights $\mathbf{w}$ do not lead to uniformly distributed solutions. Moreover, based on the analysis, in some sense, we could say that uniformly distributed search directions ($\boldsymbol{\lambda}$) lead to uniformly distributed solutions.

An interesting thing is that (from the empirical results shown in [10]) for bi-objective cases, we can still find uniformly distributed solutions by uniformly distributed weights. The reason could be explained as that for bi-objective problems, the Pareto front is one-dimension in general. Although the obtained solution is not along the provided weight vector, still one solution matches one weight vector only. Therefore, the performance of MOEA/D is not much influenced. However, this problem becomes evident when the dimension of Pareto front is more than 2. Let us consider a 3-objective minimization problem. The Pareto optimal front is assumed as a unit plane (intercept is 1), see Figure 4. Assume a Pareto optimal solution $f(\mathbf{s}_1) : (0, 0, 1)$ is expected to be obtained. According to the definition of $g^{te}(\mathbf{x}|\mathbf{w})$, the optimal weight can be any value from the set $argmin(\max(g^{te}(\mathbf{s}_1|\mathbf{w})))$. Therefore, $\mathbf{w_1} = [0.5, 0.5, 0]$ is an optimal weight as $\max(g^{te}(\mathbf{s}_1|\mathbf{w})) = 0$. Give a close examination, we find all the weight vectors such as $\mathbf{w_2}$ and $\mathbf{w_3}$ on the line BC satisfy the Equation $\max(g^{te}(\mathbf{s}_1|\mathbf{w})) = 0$, which indicates that all these $\mathbf{w}$ are optimal weights for obtaining the Pareto optimal solution $f(\mathbf{s}_1) : (0, 0, 1)$. In this sense, one optimal solution matches many optimal weight vectors. Therefore, it is difficult to obtain a set of uniformly distributed solutions by uniformly distributed weights.

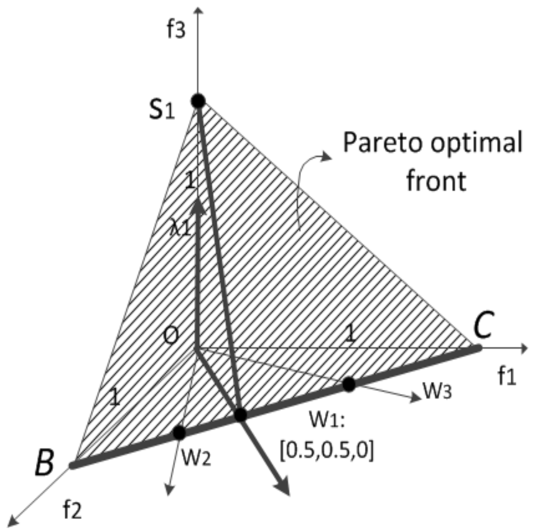


Fig. 4. The relation between search direction ($\boldsymbol{\lambda}$) and weight vector ($\mathbf{w}$): a 3-objective problem

To further illustrate this issue, we present the Pareto fronts of 3-objective DTLZ1 and DTLZ2 [23] obtained by g^{te} based MOEA/D using 190 ($H = 18$) uniformly distributed weights. MOEA/D is run 500 generations and 50 runs are performed. Plots of the non-dominated fronts with the median hypervolume value [24] (the larger of the hypervolume, the better performance of the algorithm) are shown in Figures 5(a) and 5(b). The used weights are uniformly distributed shown as Figure 5(c). From the results, it is evident that uniformly distributed weights have not produced uniformly distributed solutions.

Can we obtain a set of uniformly distributed solutions by g^{te} based MOEA/D? Yes. As shown in Figure 2, to obtain the solution $\mathbf{s}_i$, we need a search direction (i.e. $\boldsymbol{\lambda}_i$) to guide candidate solutions approach to this solution. Such a direction starts from $\mathbf{z}^*$ to $\mathbf{s}_i$. Therefore, uniformly distributed search directions lead to uniformly distributed solutions.

Next, we will study the relation between the search direction and weights. The aim is to identify a suitable distribution of weights which can be applied in $g^{te}(\mathbf{x}|\mathbf{w})$ based MOEA/D so as to find a set of uniformly distributed solutions.

Refer to Figure 2, we have $\frac{\lambda_{i1}}{\lambda_{i2}} = \frac{L_1}{l_2} = \frac{1/w_{i1}}{1/w_{i2}}$, and also $[\lambda_{i1}, \lambda_{i2}] = [\frac{f_1}{f_1+f_2}, \frac{f_2}{f_1+f_2}]$. According to [25], $\frac{w_{i1}}{w_{i2}} = [\frac{1/f_1}{1/f_1+1/f_2}, \frac{1/f_2}{1/f_1+1/f_2}]$. Therefore, the optimal weight for searching for a solution $\mathbf{s}_i$ is the norm of the reciprocal of the objective values of this solution $f_i(\mathbf{s}_i)$. It is numerically equal to the search direction. Figure 3 can serve as an example, the weight for obtaining solution $\mathbf{s}_1$ is [0.7,0.3] which is equal to the norm of the reciprocal of $f(\mathbf{s}_1) : (0.3, 0.7)$. According to the definition of the search direction, $\boldsymbol{\lambda}_1 = [0.3, 0.7]$.

Having known this, we can easily generate a suitable distribution of weights which can achieve a set of uniformly distributed solutions.

(i) According to the required distribution of solutions, we generate the related search directions;

(ii) According to the relation between search direction $\boldsymbol{\lambda}_i$; and weight $\mathbf{w}_i$, we obtain the corresponding weight by transforming $\boldsymbol{\lambda}_i$ with Equation 5:

$$w_{ij} = \frac{1}{\lambda_{ij} + \epsilon}, \quad w_{ij} = \frac{w_{ij}}{\sum_{j=1}^{j=M}(w_{ij})} \tag{5}$$

where ϵ is a small positive number, $\epsilon = 0.0001$ is used in this study.

To prove our statement, we again solve the two 3-objective DTLZ problems by using "uniformly distributed search directions". Specifically, we firstly generate 190 ($H = 18$) uniformly distributed search directions with the simplex-lattice design method. Then we transform these search directions into the essential weights, according to Equation 5

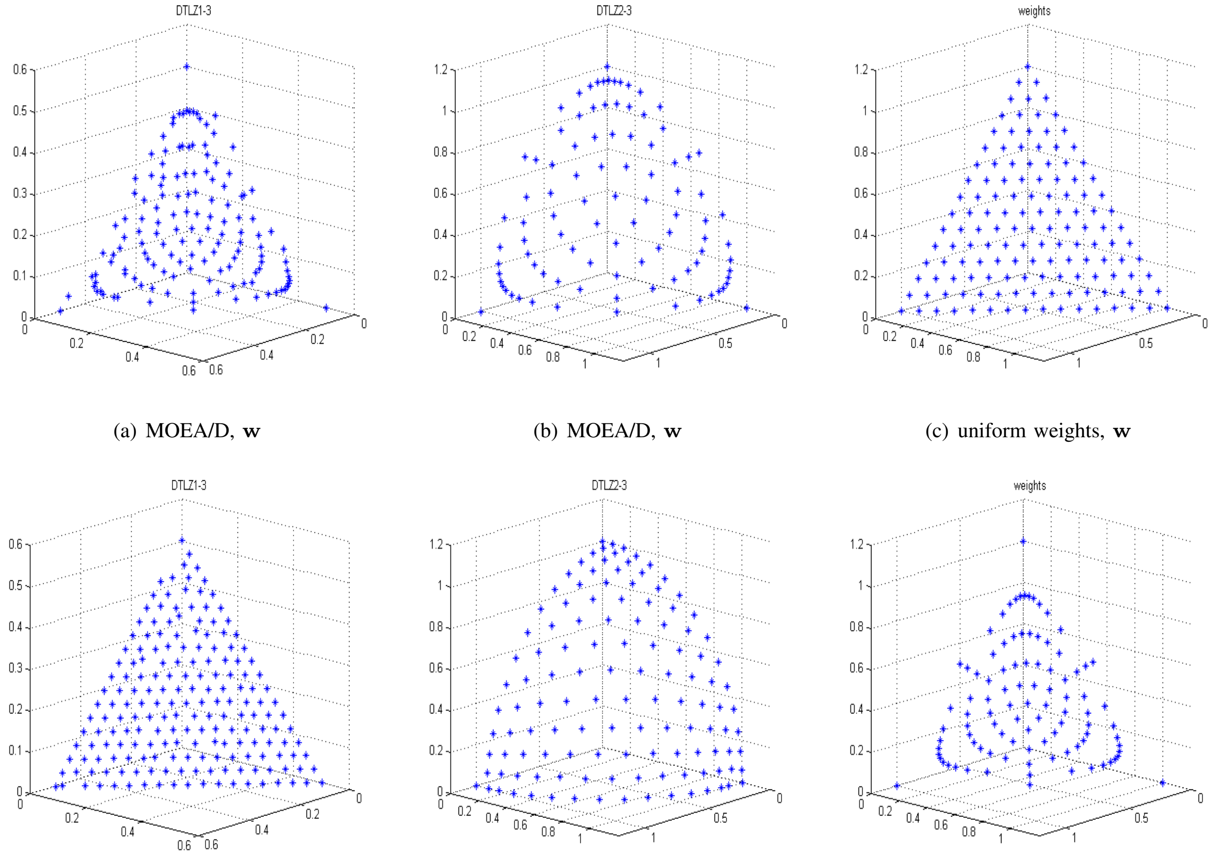
Similarly, run g^{te} based MOEA/D for 500 generations and 50 runs. Plots of the non-dominated fronts with the median hypervolume value are shown in Figures 5(d) and 5(e), respectively. The transformed weights are also shown in Figure 5(f). Compared to the results shown in Figures 5(a) and 5(b), we can clearly observe that the solutions obtained by using the 'uniformly distributed search directions'are distributed much more uniformly.

We have illustratively shown the difference of solutions obtained by using uniformly distributed $\boldsymbol{\lambda}$ and $\mathbf{w}$. In Section IV, the quality of the solutions would be quantitatively compared.

B. A pre-organization procedure

In MOEA/D, search directions are uniformly initialized while the candidate solutions are randomly initialized. MOEA/D requires every candidate solution to match one search direction. That is to say, each search direction $\boldsymbol{\lambda}_i$ is randomly assigned a candidate solution $\mathbf{s}_i$ at the beginning. More specifically, for each $g(\mathbf{x}|\boldsymbol{\lambda}_i)$, search process starts with a candidate solution that is randomly initialized for this $g(\mathbf{x}|\boldsymbol{\lambda}_i)$[3]. We argue that such a random assignment strategy

[3] Note that in the original MOEA/D, weights are uniformly initialized. However, as discussed in the Section III-A we need uniformly distributed search directions. For clarity, in the following sections we describe MOEA/D with search directions instead of weights



(a) MOEA/D, $\mathbf{w}$

(b) MOEA/D, $\mathbf{w}$

(c) uniform weights, $\mathbf{w}$

(d) MOEA/D, $\boldsymbol{\lambda}$

(e) MOEA/D, $\boldsymbol{\lambda}$

(f) uniform directions, $\boldsymbol{\lambda}$

Fig. 5. Illustration of using uniform weights and uniform directions

may slow down the search speed and so influence the performance of MOEA/D.

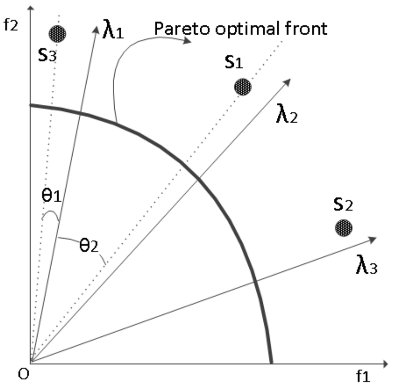


Fig. 6. Illustration of the pre-*organization* procedure

This issue is illustratively shown in Figure 6. Assuming there are 3 uniformly distributed search directions, $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}_3$ and 3 randomly initialized candidate solutions, $\mathrm{s}_1, \mathrm{s}_2$ and s_3. In MOEA/D, these candidate solutions are randomly assigned to the search directions. For example, $\boldsymbol{\lambda}_1$ is assigned to s_1, $\boldsymbol{\lambda}_2$ is assigned to s_2 and $\boldsymbol{\lambda}_3$ is assigned to s_3. However, from Figure 6, s_3 seems a better candidate solution for $g(\mathbf{x}|\boldsymbol{\lambda}_1)$. This is because s_3 is closer to the optimal solution of $g(\mathbf{x}|\boldsymbol{\lambda}_1)$ than s_1, i.e. $\theta_1 < \theta_2$. Therefore, s_3 is more likely to carry more information for searching the optimal solution of $g(\mathbf{x}|\boldsymbol{\lambda}_1)$.

Inspired by this we propose to perform a pre-*organization* procedure after the initialization of candidate solutions and search directions. The aim is to match each pair of candidate solution and search direction appropriately, i.e. assigning each search direction with an adjacent candidate solution.

The pseudo-code of this pre-*organization* procedure is shown in Algorithm 2.

Algorithm 2: A pre-*organization* procedure for MOEA/D

```
Input: initial candidate solutions, s of size $N$ , initial
       search directions, $\boldsymbol{\lambda}$ of size $N_w$
Output: $\{\mathrm{s}, \boldsymbol{\lambda}\}$
$\theta_{[\mathbf{s}_i, \boldsymbol{\lambda}_j]} \leftarrow$ angle($\mathbf{s}_i, \boldsymbol{\lambda}_j$ );
while each $\boldsymbol{\lambda}_i$ do
    $\{\mathbf{s}_i, \boldsymbol{\lambda}_i\} \leftarrow$ match($\boldsymbol{\lambda}_i, \theta_{[\mathbf{s},\boldsymbol{\lambda}]}$);
    $\theta_{[\mathbf{s},\boldsymbol{\lambda}]} \leftarrow$ modify ($\theta_{[\mathbf{s},\boldsymbol{\lambda}]}, i$);
end
```

The function `angle` compute the angle $\theta_{[\mathbf{s}_i,\boldsymbol{\lambda}_j]}$ for $\mathbf{s}_i$ and $\boldsymbol{\lambda}_j$. For each search direction (e.g. $\boldsymbol{\lambda}_1$), function `match` selects the candidate solution where $\theta_{[\mathbf{s}_i,\boldsymbol{\lambda}_1]}$ is the smallest. The function `modify` set the i-th row of $\theta_{[\mathbf{s},\boldsymbol{\lambda}]}$ as the largest value among all the $\theta_{[\mathbf{s}_i,\boldsymbol{\lambda}_j]}$. This pre-*organization* procedure is executed after the initialization of $\boldsymbol{\lambda}$ and $\mathbf{s}$, that is, replace step (2) in Algorithm 1 by this pre-*organization* procedure.

Here we suggest to perform the pre-*organization* according to the angles between the search directions and candidate solutions. We have also considered to perform the pre-*organization* based on the result of $g(\mathbf{x}|\boldsymbol{\lambda})$. That is, selecting the candidate solution with the smallest $g(\mathbf{x}|\boldsymbol{\lambda}_i)$ for each $\boldsymbol{\lambda}_i$. However, the experimental results show that in this case MOEA/D fails to explore new areas and results in a poor diversity performance.

IV. Comparative study

To demonstrate the effectiveness of the proposed ideas, we compare MOEA/D with its two variants: MOEA/D-a and MOEA/D-b on some benchmark problems. MOEA/D-a applies uniform directions on MOEA/D. MOEA/D-b applies uniform directions and also the pre-*organization* procedure.

A. Comparative study: experimental setup

The test problems are selected from the WFG test suite [26]. Problems 2 to 9 are invoked in 2 and 7-objective instances. In each case, the WFG position parameter and distance parameter are set to 18 and 14, providing a constant number of decision variables ($nvar = 18 + 14 = 32$) for each problem instance. Problem attributes include separability/non-separability, unimodality/multimodality, unbiased/biased parameters and convex/concave geometries. Note that in this study WFG*n*-Y refers to problem WFG*n* with Y objectives.

For each test problem, 50 runs of each algorithm test are performed to facilitate a robust statistical analysis. Some general parameter settings are listed in Table I and are fixed across all algorithm runs.

TABLE I
General parameter settings

Objectives,M	2, 7
Maximum generations,$maxGen$	250
The No. of weights, N_w ($M = 2$)	$H = 99 \longrightarrow Nw = 100$
The No. of weights, N_w ($M = 7$)	$H = 5 \longrightarrow Nw = 462$
Population size, N	$N = Nw$
Selection neighbourhood, T	10
Replacement neighbourhood, nr	2
Simulated binary crossover (SBX [4])	$(p_c = 1, \eta_c = 15)$
Polynomial mutation (PM [4])	$(p_m = \frac{1}{nvar}, \eta_m = 20)$

For the performance assessment, the generational distance metric (GD), the spread metric (Δ) and the hypervolume metric (HV) are used to measure the convergence, diversity and overall performance, respectively. A favourable (smaller) GD value implies good proximity and also a favourable (smaller) spread metric value implies good diversity. A favourable hypervolume (larger, for a minimization problem) implies good proximity with diversity.

The approximation sets used in the comparisons are the non-dominated solutions in the final stage. For each problem instance, performance metric values for each algorithm are calculated for each approximation set. Note that prior to calculating the performance metrics, we normalise all objective values to be within the range $[0, 1]$ by the *nadir* point [27] (which assumes equal relative importance of all the objectives). The *nadir* point of these problems can be obtained from [26]. For the hypervolume metric, the reference point is set as $r_i = 1.2, i = 1, 2, \cdots, M$ and the software developed by [28] is used to calculate the hypervolume. For all the three metrics, the null hypotheses used are those of equality of median values, against the two-sided alternative hypotheses. The non-parametric Wilcoxon-ranksum two-sided comparison [29] procedure at the 95% confidence level is employed to test the working hypotheses.

B. Comparative study: experimental results

The comparison results of MOEA/D, MOEA/D-a and MOEA/D-b in terms of the three performance metrics GD, Δ and HV are presented in Tables II to VII. In each table, the mean/deviation values of each performance metric across the 50 independent runs are shown. Superior results (in terms of the mean value) are marked in **boldface**. The symbol '$<$', '$=$' or '$>$' between two algorithms indicates that the first algorithm is statistically worse, equal or better than the second algorithm at 95% confidence level. Note the statistical comparison is conducted for MOEA/D and MOEA/D-a (assessing the difference of using uniformly distributed of search directions and weights), MOEA/D-a and MOEA/D-b (assessing the effect of the pre-*organization* procedure), separately.

1) Comparison results for 2-objective problems: Seen from Table II, MOEA/D-b gives the largest mean HV values for all the eight 2-objective problems. Refer to the non-parametric statistical test results, (1) MOEA/D-a performs comparably to MOEA/D for all the problems; and (2) MOEA/D-b performs statistically better than MOEA/D-a for all the problems except for WFG2, where the two algorithms show comparable performance.

TABLE II
The mean/deviation values of HV metric for 2-objective problems

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG2-2	0.5598/0.0100	=	0.5616/0.0124	=	**0.5630**/0.0131
WFG3-2	0.5909/0.0102	=	0.6017/0.0090	<	**0.6114**/0.0106
WFG4-2	0.4101/0.0087	=	0.4088/0.0084	<	**0.4273**/0.0084
WFG5-2	0.3846/0.0077	=	0.3872/0.0058	<	**0.3945**/0.0076
WFG6-2	0.3978/0.0062	=	0.3961/0.0124	<	**0.4012**/0.0131
WFG7-2	0.3584/0.0064	=	0.3593/0.0124	<	**0.3620**/0.0131
WFG8-2	0.2663/0.0068	=	0.2651/0.0124	<	**0.2791**/0.0131
WFG9-2	0.3845/0.0203	=	0.3869/0.0211	<	**0.3937**/0.0210

The comparison results of the convergence performance are shown in Table III, MOEA/D-b offers the smallest mean GD values for all the problems except for WFG6-2, where

TABLE III
THE MEAN/DEVIATION VALUES OF GD METRIC FOR 2-OBJECTIVE PROBLEMS

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG3-2	0.0029/0.0204	=	0.0029/0.0090	=	**0.0029**/0.0032
WFG4-2	0.0741/0.0019	=	0.0739/0.0012	=	**0.0722**/0.0018
WFG5-2	0.0714/0.0013	=	0.0719/0.0012	<	**0.0706**/0.0011
WFG6-2	**0.0699**/0.0009	=	0.0714/0.0013	=	0.0710/0.0017
WFG7-2	0.0806/0.0018	=	0.0807/0.0017	<	**0.0781**/0.0013
WFG8-2	0.1010/0.0037	=	0.1016/0.0051	<	**0.0952**/0.0054
WFG9-2	0.0725/0.0026	=	0.0732/0.0023	<	**0.0701**/0.0021

MOEA/D gives the smallest value. According to the non-parametric tests, the convergence difference between MOEA/D and MOEA/D-a is not significant for any problem. With respect to MOEA/D-a and MOEA/D-b, MOEA/D-b performs statistically better than MOEA/D-a for 4 out of 7 problems (WFG5-2, WFG7-2, WFG8-2 and WFG9-2). The difference is not significant for the other 3 problems.

TABLE IV
THE MEAN/DEVIATION VALUES OF Δ METRIC FOR 2-OBJECTIVE PROBLEMS

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG3-2	0.2228/0.0228	=	**0.2214**/0.0182	=	0.2281/0.0237
WFG4-2	1.2750/0.0121	=	**1.2673**/0.0096	>	1.2863/0.0035
WFG5-2	1.3117/0.0110	<	**1.3012**/0.0064	=	1.3187/0.0052
WFG6-2	**1.3048**/0.0084	=	1.3054/0.0103	=	1.3074/0.0101
WFG7-2	**1.3601**/0.0169	=	1.3765/0.0173	=	1.3797/0.0197
WFG8-2	1.5694/0.0143	<	**1.5408**/0.0119	>	1.5632/0.0289
WFG9-2	1.3253/0.0297	=	**1.3147**/0.0290	=	1.3136/0.0273

Table IV presents the diversity comparison for the MOEA/D and its two variants. MOEA/D-a gives the smallest mean Δ values for all the problems with the exceptions of WFG6-2 and WFG7-2, where MOEA/D provides the smallest values. From the non-parametric tests results, MOEA/D-a performs better than (WFG5-2 and WFG8-2) or at least comparably (WFG3-2, 4-2, 6-2, 7-2 and 9-2) to MOEA/D. As for the results of MOEA/D-a and MOEA/D-b, MOEA/D-a performs better than MOEA/D-b on WFG4-2 and WFG8-2; comparably on the rest of 5 problems.

Overall, for 2-objective problems, (i) the difference between MOEA/D and MOEA/D-a is not significant. The reason is that for all the considered 2-objectives problem, the Pareto optimal front is one-dimension. Therefore, the uniformly distributed weights are luckily to be coincide with the uniformly distributed search directions (the transformed weights). (ii) MOEA/D-b shows a better convergence performance and a comparable diversity performance, compared to MOEA/D-a for most of the problems. These results indicate that the pre-*organization* procedure helps MOEA/D-a improve the convergence without decreasing its diversity performance.

2) Comparison results for 7-objective problems: Table V presents the HV comparison results for 7-obejctive problems. MOEA/D-b again gives the largest average HV value for all the problems except for WFG2-7, where MOEA/D-a provides the largest value. With respect to the non-parametric test results, MOEA/D-a performs statistically better than MOEA/D on 5 out of 8 problems. For the rest of 3 problems (WFG2-7,3-7 and 8-7), the two algorithms performs comparably. Simultaneously, MOEA/D-b performs better than MOEA/D-a on all the problems except for WFG2-7 and WFG7-7, where the two algorithms show comparable performance.

TABLE V
THE MEAN/DEVIATION VALUES OF HV METRIC FOR 7-OBJECTIVE PROBLEMS

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG2-7	0.8838/0.0787	=	**0.8850**/0.0903	=	0.8876/0.0914
WFG3-7	0.9586/0.0204	=	0.9606/0.0090	<	**0.9706**/0.0032
WFG4-7	0.8461/0.0204	<	0.8665/0.0199	<	**0.8812**/0.0187
WFG5-7	0.8171/0.0032	<	0.8227/0.0067	<	**0.8433**/0.0093
WFG6-7	0.7760/0.0337	<	0.8118/0.0231	<	**0.8343**/0.0196
WFG7-7	0.8383/0.0185	<	0.8674/0.0125	=	**0.8640**/0.0119
WFG8-7	0.7188/0.0120	=	0.7218/0.0158	<	**0.7524**/0.0107
WFG9-7	0.6080/0.0182	<	0.6391/0.0295	<	**0.6640**/0.0325

Comparing the GD values for 7-objective problems, the superiority of MOEA/D-b is more evident. It provides the smallest mean GD values for all the problems. Moreover, the difference between MOEA/D-b and MOEA/D-a is significant at 95% confidence level for all the problems except for WFG2-7. Considering the non-parametric comparison results for MOEA/D-a and MOEA/D, we find that the two algorithms show equivalent convergence performance for all the problems with exceptions of WFG3-7 and WFG8-7 on which MOEA/D performs better.

TABLE VI
THE MEAN/DEVIATION VALUES OF GD METRIC FOR 7-OBJECTIVE PROBLEMS

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG3-7	0.0438/0.0059	>	0.0573/0.0052	=	**0.0439**/0.0052
WFG4-7	0.0708/0.0103	=	0.0793/0.0115	<	**0.0536**/0.0136
WFG5-7	0.0483/0.0004	=	0.0503/0.0005	<	**0.0456**/0.0004
WFG6-7	0.0653/0.0101	=	0.0615/0.0077	<	**0.0581**/0.0127
WFG7-7	0.0853/0.0100	=	0.0882/0.0153	<	**0.0731**/0.0148
WFG8-7	0.1165/0.0206	>	0.1431/0.0122	<	**0.1052**/0.0137
WFG9-7	0.0519/0.0026	=	0.0515/0.0011	<	**0.0413**/0.0023

TABLE VII
THE MEAN/DEVIATION VALUES OF Δ METRIC FOR 7-OBJECTIVE PROBLEMS

Problem	MOEA/D		MOEA/D-a		MOEA/D-b
WFG3-7	1.1934/0.0421	<	**0.8840**/0.0431	=	0.9134/0.0401
WFG4-7	7.6241/0.1050	<	**7.1241**/0.1606	>	7.4241/0.1062
WFG5-7	7.3733/0.0267	<	**7.0143**/0.0361	>	7.2143/0.0411
WFG6-7	7.4356/0.0523	<	**7.1427**/0.0660	=	7.2153/0.0725
WFG7-7	7.2327/0.0548	<	**6.9854**/0.0482	=	7.1012/0.0518
WFG8-7	7.9674/0.0423	<	**7.3661**/0.0550	>	7.7822/0.0490
WFG9-7	7.5582/0.1082	<	**7.0143**/0.1251	=	7.1033/0.1331

As shown in Table VII, MOEA/D-a provides the best diversity performance for all the problems in terms of the average Δ. According to the non-parametric test results, the diversity performance of MOEA/D-a is statistically better than MOEA/D for all the problems. MOEA/D-a also performs statistically better than MOEA/D-b for WFG4-7, 5-7 and 8-7, and equivalently for the rest of 4 test problems.

Overall, when the number of objectives increases, MOEA/D-a tends to show better diversity performance and comparable convergence performance to MOEA/D for most of problems. This indicates that using uniformly distributed search directions is effective. Moreover, the pre-*organization* procedure is effective, especially on improving the convergence performance of MOEA/D-a.

V. CONCLUSION

In this paper, two novel strategies are presented to further improve the performance of MOEA/D. The first strategy improves the diversity performance (uniformity) of MOEA/D, which is to apply uniformly distributed search directions, rather than weights, to search for uniformly distributed solutions. To the best of our knowledge, this is the first time to explain why cannot uniformly distributed weights produce uniformly distributed solutions by Chebyshev scalarizaing function based MOEA/D [10]. The second strategy improves the convergence performance of MOEA/D, which is to perform a pre-*organization* procedure to match the candidate solutions and search directions in an appropriate order. The effectiveness of these two strategies are tested on 2 and 7-objective WFG benchmark problems and are shown to be good for most of the problems.

With respect to further research, the first is to extend this study to other scalarizing function based MOEA/D. The second is to look at how to adaptively adjust the weight vectors for different problems, e.g., co-evolving weights and candidate solutions. Thirdly, parameters T and nr greatly influence the performance of MOEA/D [30], and so some parameter control methods are needed.

REFERENCES

[1] C. Coello, G. Lamont, and D. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[2] C. Fonseca and P. Fleming, “Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization,” in *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1993, pp. 416–423.

[3] ——, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 1, pp. 26–37, 1998.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[5] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization,” *Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems EUROGEN 2001*, vol. 3242, no. 103, pp. 95–100, 2002.

[6] R. Purshouse and P. Fleming, “Evolutionary many-objective optimisation: an exploratory analysis,” in *Evolutionary Computation, 2003. The 2003 IEEE Congress on*, vol. 3. IEEE, 2003, pp. 2066–2073.

[7] ——, “On the Evolutionary Optimization of Many Conflicting Objectives,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[8] D. Corne and J. Knowles, “Techniques for highly multiobjective optimisation: some nondominated points are better than others,” in *GECCO 2007: Proceedings of the Genetic and Evolutionary Computation Conference*. London,England,United Kingdom: ACM, 2007, pp. 773–780.

[9] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” in *Evolutionary Computation, 2008. The 2008 IEEE Congress on*. IEEE, 2008, pp. 2419–2426.

[10] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[11] R. Wang, R. Purshouse, and P. Fleming, “Local preference-inspired co-evolutionary algorithms,” in *GECCO 2012: Proceedings of the Genetic and Evolutionary Computation Conference*, Philadelphia, USA.

[12] ——, “Preference-inspired Co-evolutionary Algorithms for Many-objective Optimisation,” *Evolutionary Computation, IEEE Transactions on - to appear*, 2013.

[13] R. Purshouse, C. Jalbă, and P. Fleming, “Preference-driven co-evolutionary algorithms show promise for many-objective optimisation,” in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 136–150.

[14] R. Wang, R. Purshouse, and P. Fleming, “Preference-inspired co-evolutionary algorithm using adaptively generated goal vectors- to appear,” in *Evolutionary Computation, 2013. The 2013 IEEE Congress on*. Cancun, Mexico: IEEE, 2013.

[15] ——, “On finding well-spread pareto optimal solutions by preference-inspired co-evolutionary algorithm,” in *GECCO 2013: Proceedings of the Genetic and Evolutionary Computation Conference - to appear*. Amsterdam,Netherlands: ACM, 2013.

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

[17] E. Hughes, “Multiple single objective pareto sampling,” in *Evolutionary Computation, 2003. The 2003 IEEE Congress on*. IEEE, 2003, pp. 2678–2684.

[18] H. Li and Q. Zhang, “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 284–302, 2009.

[19] Q. Zhang, W. Liu, and H. Li, in *Evolutionary Computation, 2009. The 2009 IEEE Congress on*.

[20] Y. Tan, Y. Jiao, H. Li, and X. Wang, “A modification to moea/d-de for multiobjective optimization problems with complicated pareto sets,” *Information Sciences*, vol. 213, no. 2012, pp. 14–38, 2012.

[21] S. Jiang, Z. Cai, J. Zhang, and Y. Ong, “Multiobjective optimization by decomposition with pareto-adaptive weight vectors,” in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 3. IEEE, 2011, pp. 1260–1264.

[22] A. Wierzbicki, “The use of reference objectives in multiobjective optimization-theoretical implications and practical experiences,” in *Multiple criteria decision making: theory and application: proceedings of the third conference*, vol. 1979. Hagen Konigswinter, West Germany: Springer, 1979, p. 468.

[23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Evolutionary Computation, 2002. The 2002 IEEE Congress on*. IEEE, 2002, pp. 825–830.

[24] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach,” *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, 1999.

[25] K. Miettinen and M. Mäkelä, “On scalarizing functions in multiobjective optimization,” *OR spectrum*, vol. 24, no. 2, pp. 193–213, 2002.

[26] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[27] K. Deb, K. Miettinen, and S. Chaudhuri, “Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 821–841, 2010.

[28] C. Fonseca, L. Paquete, and M. López-Ibáñez, “An improved dimension-sweep algorithm for the hypervolume indicator,” in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 1157–1163.

[29] M. Hollander and D. Wolfe, *Nonparametric statistical methods*. Wiley-Interscience, 1999.

[30] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, “Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization,” in *Evolutionary Computation, 2009. The 2009 IEEE Congress on*. IEEE, May 2009, pp. 2508–2515.