Pareto adaptive scalarising functions for decomposition based algorithms

Rui Wang[1], Qingfu Zhang[2], and Tao Zhang[1]

[1] College of Information Systems and Management, National University of Defense Technology, Changsha, Hunan, P.R. China, 410073
ruiwangnudt@gmail.com
[2] Department of Computer Science, City University of Hong Kong, and the University of Essex, UK

Abstract. Decomposition based algorithms have become increasingly popular for solving multi-objective problems. However, the effect of s-calarising functions in decomposition based algorithms is under-explored. This study analyses the search behaviour of a family of frequently used s-calarising functions— the L_p weighted approaches, and identifies that the p value corresponds to a trade-off between the L_p approach's search ability and its robustness on Pareto front geometries. That is, as the p value increases, the search ability of the L_p approach decreases whereas its robustness on Pareto front geometry increases. Based on this observation, we propose to use Pareto adaptive scalarising functions in decomposition based algorithms, where the p value is adaptively fine-tuned based on an estimation of the Pareto front shape. MOEA/D using Pareto adaptive scalarising functions (MOEA/D-par) is tested on a set of problems (with up to seven objectives) encompassing three basic Pareto front geometries, i.e., convex, concave and linear, and is shown to outperform MOEA/D using Chebyshev function on all the test problems.

Keywords: Multi-objective optimization, Evolutionary computation, Decomposition, Scalarising function. Pareto adaptive

1 Introduction

Multi-objective optimisation problems (MOPs) arise in many disciplines such as engineering, finance, logistics and control systems [1], where multiple objectives must be simultaneously optimised. Often objectives in a MOP are in competition with each other, and thus, the optimal solution set of MOPs is not a single solution but comprises of a set of trade-off solutions. Multi-objective evolutionary algorithms (MOEAs) are well suited for solving MOPs since (i) their population-based nature leads naturally to the generation of an approximate trade-off surface in a single run; and (ii) they tend to be robust to underlying objective function characteristics.

During the last two decades, a variety of MOEA approaches has been proposed. These approaches can be categorised into three main classes: Pareto-dominance or modified dominance based algorithms, e.g., MOGA [2], NSGA-II

[3], PICEA-g [4]; Performance indicator based algorithms, e.g., IBEA[5], HypE [6]; and decomposition based algorithms, e.g., CMOGA [7], MSOPS [8], MOEA/D [9]. Amongst these approaches, decomposition based algorithms become increasingly popular recently. Decomposition based algorithms decompose a MOP into a set of single objective problems by means of weighted scalarising functions, or a set of simple MOPs [10, 11] and optimise them in a collaborative manner. Compared with the other two types of algorithms, decomposition based algorithms have a number of advantages such as high search ability for combinatorial optimisation, computational efficiency on fitness evaluation and high compatibility with local search [9, 12–14]. The seminal decomposition based MOEA, i.e., MOEA/D [9], that popularised this method, has been used in many real-world applications [15]. Despite these advantages, the performance of decomposition based algorithms is arguably dependent on the specification of weights[16] and scalarising functions [17]. The choice of suitable weights and scalarising functions is typically problem-dependent and therefore is difficult if no information about the problem characteristics is known before the search proceeds.

Regarding the choice of weights, we have known that when the Pareto front geometry of a MOP is known *a priori*, an optimal distribution of weights for certain scalarising function can be identified [16, 18]. Otherwise, a suitable set of weights can be configured adaptively. A number of effective methods have been proposed for this purpose, for example, co-evolving weights with solutions [19, 20], using Pareto adaptive weights [21], adjusting weights adaptively based on an estimation of Pareto front geometry [22–24]. Regarding the choice of scalarising functions, although we have known: for example, the weighted Chevbshev is able to find solutions on both convex and non-convex regions whereas the weighted sum cannot [25]; the weighted sum can obtain better results than the weighted Chevbshev on multi-objective knapsack problems [9], this is still far from being well understood. It is in general unclear what the relation is between different scalarising functions; and how an appropriate scalarising function can be identified for a new problem. Towards a better understanding of the effect of scalarising functions in decomposition based algorithms as well as unlocking the aforementioned issues, in this study we analyse a family of frequently used scalarising functions, i.e., the L_p weighted approaches in terms of their search ability and their robustness on the Pareto front geometry. Moreover, based on the analysis, we propose to use Pareto adaptive L_p scalarising functions in decomposition based algorithms so as to enhance the algorithm's performance.

The remainder of this paper is organised as follows: in Section 2 some background knowledge about decomposition based approaches, is provided. Section 3 elaborates the effect of L_p scalarising functions and how to choose a suitable L_p scalarising function. Experiments and discussions are provided in Section 4. Finally, Section 5 concludes the paper and identities future studies.

2 Decomposition approaches

Without loss of generality, a minimisation MOP is defined as,

$$\begin{aligned} \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})) \\ \text{subject to } & \mathbf{x} \in \boldsymbol{\Omega} \end{aligned} \tag{1}$$

where m is the number of objective functions (generally, $m > 2$); $\mathbf{x}$ is a *vector* in the *decision (variable) space* $\boldsymbol{\Omega}$. $\mathbb{R}^m$ is the *objective space*. $F : \boldsymbol{\Omega} \to \mathbb{R}^m$ consists of m real-valued objective functions that are to be minimised.

Decomposition based approaches decompose a MOP into a set of single objective problems defined by means of scalarising functions with different weights. The optimal solution of each single objective problem corresponds to one Pareto optimal solution of a MOP [26]. The weight vector defines a search direction for the scalarising function. Diversified solutions can be obtained by employing different search directions.

A variety of scalarising functions can be used in decomposition based algorithms [26]. The weighted sum and the weighted Chebyshev from the family of weighted L_p scalarising functions are two of the most popular ones. Mathematically, the weighted L_p scalarising function can be written as,

$$g^{wd}(\mathbf{x}|\mathbf{w}, p) = \left(\sum_{i=1}^{m} \lambda_i \left(f_i(\mathbf{x}) - z_i^* \right)^p \right)^{\frac{1}{p}}, \ p > 0 \quad \lambda_i = (1/w_i)^p \tag{2}$$

where $\mathbf{z}^* = (z_1, z_2, \cdots, z_m)$ is the *ideal* point; $\mathbf{w} = (w_1, w_2, \cdots, w_m)^T$ is a weighting vector and $\sum_{i=1}^{m} w_i = 0, \ w_i \geq 0$; The $\mathbf{w}$ determines the search direction of the scalarising function. Note that whether the obtained Pareto optimal solution is along the search direction or not is also influenced by the Pareto front geometry [16, 27]. The weighted sum and weighted Chebyshev are derived by setting $p = 1$ and $p \to \infty$, respectively.

In addition, decomposition based algorithms combine different objective function values into one scalar value. These objectives might have various units of measurement, and/or scaled disparately. It is therefore important to rescale different objectives to dimension-free units before aggregation. Typically, the normalisation procedure transforms an objective value f_i by

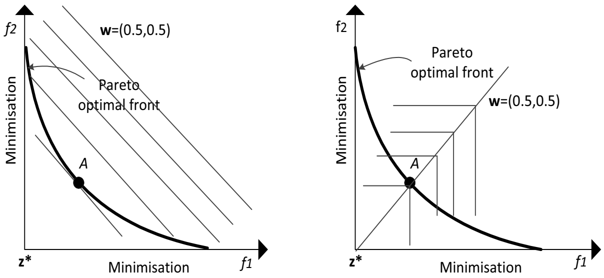
$$\overline{f}_i = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*} \tag{3}$$

If the z_i^* and z_i^{nad} (the *nadir* point) are not available, the smallest and largest f_i of all non-dominated solutions found so far could be used instead.

3 The choice of a suitable L_p scalarising function

3.1 Analysis: property of different L_p scalarising functions

This section analyses the property of different L_p scalarising functions, that is, the trade-off between their search ability and their robustness on Pareto front



(a) (b)

Fig. 1. Contour lines of the weighted sum (a) and weighted Chebyshev (b) scalarising functions

geometries [28]. Inspired by [29], we first look at two special cases, i.e., the weighted sum and the weighted Chebyshev. Fig. 1 shows contour lines of the two scalarising functions in a bi-objective case with *ideal* point at the origin and weight vector $\mathbf{w} = (0.5, 0.5)$. The objective space is divided into two sub-spaces by the contour line. Solutions in one sub-space are better than solutions on the contour line while solutions in the other sub-space are worse. Solutions that lie on the same contour line have the same scalar objective value. In Fig. 1, solution A is the optimal solution of $g^{wd}(\mathbf{x}|\,(0.5, 0.5), 1)$ and $g^{wd}(\mathbf{x}|\,(0.5, 0.5), \infty)$.

The contour line of the weighted sum approach is a line, and the contour line of the weighted Chebyshev approach is a polygonal line (with vertical angle). According to the shape of the contour line we can observe that for the weighted sum approach the size of a better region equals to half of the whole objective space regardless of the number of objectives. This indicates that the probability of replacement of an existing solution by a newly generated solution always decreases from $\frac{1}{2}$ to 0 as the search progresses. The maximal probability of replacement (i.e., $\frac{1}{2}$) is not influenced by the number of objectives. In this sense, the search ability of the weighted sum approach is not affected by an increase in the number of objectives. With respect to the weighted Chebyshev function, a better region roughly equals to $\left(\frac{1}{2}^m\right)$ of the m-dimensional objective space. This indicates that the maximal probability of replacement is $\left(\frac{1}{2}^m\right)$. Compared with the weighted sum approach, the maximal probability of replacement significantly decreases as the number of objective increases. In other words, the search ability of Chebyshev scalarising function deteriorates as the number of objectives increases [28, 30]. However, it is suspected that the search ability of the Chebysheve scalarising function is comparable to the Pareto-dominance relation as claimed in [28]. Our preliminary experiments show that compared with the Pareto dominance, solutions selected by the Chebyshev function are more likely to be closer to the *ideal* point [31]. Moreover, it has been widely demonstrated that decomposition based algorithms (even using random weighted Chebyshev functions) outperform Pareto-dominance based algorithms on many-objective problems [14, 19].

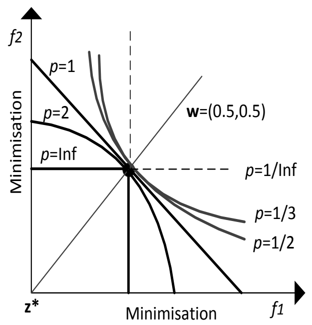


Fig. 2. Contour lines of the L_p scalarising function with different p values.

Contour lines of the L_p scalarising functions with different p values are shown in Fig. 2. We can observe from the figure that the volume enclosed by the contour line and the *ideal* point decreases as p increases (a calculation of the volume can be referred to [28]). This indicates that as p increases, the probability of finding a better solution (measured by the L_p approach) decreases, that is, the search ability of the L_p scalarising function decreases. This observation is also experimentally demonstrated by applying MOEA/D with L_3, L_7 and L_∞ scalarising functions to solve the 4 objective WFG4 [32] problem whose Pareto optimal front is a hyper-sphere. Each of the algorithm instantiations is run for 31 independent runs. The mean hypervolume (HV) values and the generation distance (GD) values over generations are plotted in Fig. 3. We can clearly observe from Fig. 3 that MOEA/D with $p = 3$ performs the best, followed by $p = 7$, and then $p \to \infty$, i.e., the weighted Chebyshev function.

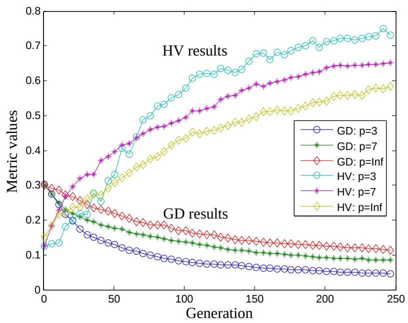
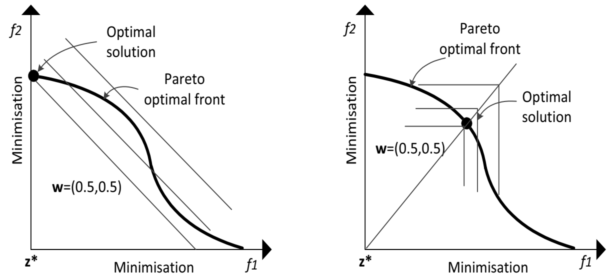


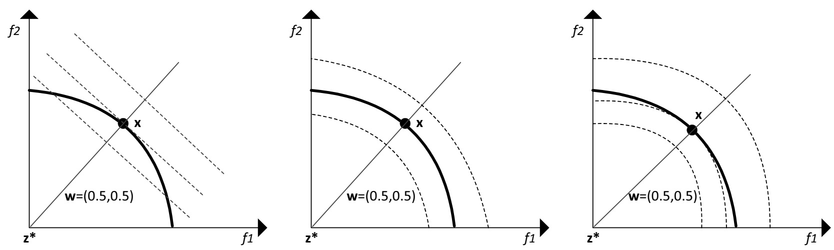
Fig. 3. (Colour online) The performance of MOEA/D using the $p = 3$, $p = 7$ weighted L_p scalsrising functions and the weighted Chebyshev function on the 4-objective WFG4 problem: the mean HV and GD values of over generations.



(a) weighted sum (b) weighted Chebyshev

Fig. 4. Behaviours of the weighted sum (a) and Chebyshev (b) scalarising function on non-convex Pareto front.

As previously mentioned, the weighted sum function may not be able to find all the Pareto optimal solutions in the case of non-convex PFs [26, p. 79], whereas the Chebyshev scalarising function can find solutions in both convex and non-convex regions, see Fig. 4. Upon closer examination, we can imagine that all the L_p scalarising functions except for the Chebyshev, face difficulties in searching for solutions in a non-convex region. To be more specific, a weighted L_p scalarising function can find solutions along certain search direction in a non-convex region only if the curvature of its contour line is larger than the curvature of the PF shape. Otherwise the selected scalarising function suffers from the non-convex geometry issue. Since the curvature of the Chebyshev function is ∞, it is able to find Pareto optimal solutions for any type of geometries. For example, assuming that the PF is a circle (quadratic) in the first quadrant, see Fig. 5. In order to find the Pareto optimal solution $\mathbf{x}$ along the search direction $(0.5, 0.5)$, the L_p with $p > 2$ should be used, e.g, $p = 3$.



(a) $p = 1$ Failed (b) $p = 2$ Threshold (c) $p = 3$ Successful

Fig. 5. Searching the same solution using different L_p functions.

Overall the search ability of a L_p scalarising function and its robustness on Pareto front geometries are a trade-off— the higher the search ability, the lower the robustness. If the Pareto front geometry is known *a priori*, we will be able to determine a suitable L_p scalarising function by taking into account the curvature of the Pareto front. For example, for a search direction $\mathbf{w}^j$, we can set the p value being larger than the curvature of the segmented Pareto front along $\mathbf{w}^j$. However, if the Pareto front geometry is unknown, we could set the p value based on the estimated Pareto front geometry.

3.2 Methodology: estimation of the Pareto front geometry on line

We have analysed the property of different L_p scalarising functions, and have identified that the choice of a suitable p value is determined by the Pareto front geometry. By a suitable L_p scalarising function, we mean that its search ability is maximised, and simultaneously, it guarantees that any Pareto optimal solution can be obtained for a certain weight. This section describes in elaborate detail how a suitable L_p scalarising function is determined. The key issue here is to effectively estimate the Pareto front geometry.

A number of methods are available in the literature for estimating the Pareto front geometry. Here, we borrow the idea from [33, 21], that is, approximating the Pareto front using a family of reference curves:

$$\{(y_1)^\alpha + (y_2)^\alpha, \cdots, (y_m)^\alpha = 1; y_j \in (0, 1], \alpha \in (0, \infty)\} \quad (4)$$

The family of curves as shown in Fig. 6 possesses the following properties: i) if $\alpha > 1$, the curve is a concave; ii) if $\alpha < 1$, the curve is convex; and iii) if $\alpha = 1$, the curve is linear.

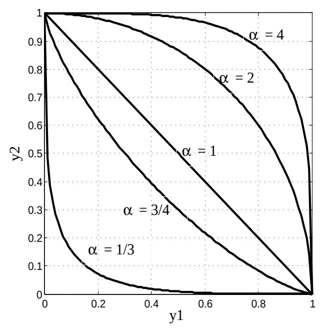


Fig. 6. Illustration of reference curves for $\alpha = \frac{1}{3}, \frac{3}{4}, 1, 2$ and 4 in 2-objective space.

Next we describe how the PF is associated with one of the curves. The pseudo-code is presented in Algorithm 1. First we initialise a set of candidate p values, and store them in a set P (line 1); Then we normalise solutions within the range $[0, 1]$ (line 2). Next for a search direction $\mathbf{w}^j$, we identify its T neighbouring solutions, denoted as Q (line 4). These neighbouring solutions are the current solutions of the neighbouring problems. The parameter T is the same as the selection neighbourhood size in MOEA/D [9]. We compute the Eq. (5) for each candidate p value. The smaller the $h(p, Q)$, the better the solutions Q match the reference curve. The p is determined as the value that produces the second smallest $h(p, Q)$ (lines 5 and 6). The reason for not choosing the p associated with the minimal $h(p, Q)$ is that the curvature of L_p function is required to be larger than the curvature of this segmented PF shape, i.e. $p > \alpha$, see Fig. 5. In addition, we include a pre-defined large number, e.g., 1000, in the set P. If $h(Q, 1000)$ is found to be the minimal, p is set to ∞, i.e., the Chebyshev function is used instead (lines 7-9).

$$h\left(p, Q\right) = \sum_{\forall \mathbf{x}^k \in Q} \left(\sum_{i=1,\ldots,m} \left(f_i\left(\mathbf{x}^k\right)\right)^p - 1 \right)^2, \quad p \in P \tag{5}$$

Algorithm 1: Selecting a suitable L_p scalarising function

Input: non-dominated solutions available Q, neighbourhood size, T
Output: p value.

1 Initialise the candidate L_p functions, e.g., $P = \{\frac{1}{2}, \frac{2}{3}, 1, 2, 3, ..., 10, 1000\}$;
2 Normalise solutions within the range $[0, 1]$;
3 **foreach** *search direction,* $\mathbf{w}^j$ **do**
4   Find the T neighbouring solutions, Q, of the search direction $\mathbf{w}^j$;
5   Compute the $h(p, Q)$ for each candidate p;
6   Find the second smallest $h(p, Q)$ and identify the corresponding p value;
7   **if** *p equals to a pre-defined large value in the P* **then**
8   using the Chebyshev function instead, i.e., $p \leftarrow \infty$;
9   **end**
10 **end**

4 Experiments and discussions

This section examines the effect of Pareto adaptive scalarising functions. We incorporate it into the state-of-the-art decomposition based algorithm, i.e., MOEA/D [9], and compare the derived algorithm, denoted as MOEA/D-par (see Algorithm 2), with MOEA/D using the Chebyshev scalarising functions.

4.1 Experimental descriptions

Test problems Test problems used in this study are constructed by applying different shape functions provided in the WFG toolkit to the standard WFG4 benchmark problem, please refer to [19] for more details. The WFG41 has a

Algorithm 2: MOEA/D using Pareto adaptive scalarising functions

Input: initial population, $S \leftarrow \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^N\}$, initial weights, $W \leftarrow \{\mathbf{w}^1, \mathbf{w}^2, \cdots, \mathbf{w}^N\}$, selection neighbourhood size, T, replacement neighbourhood size, nr

Output: S

Initialise the L_k^p as the weighted sum style, i.e., $p_k \leftarrow 1, i \in \{1, 2, \cdots, N\}$;
Evaluate the objective function values of the initial S;
Update the *ideal* and *nadir* vectors, $\mathbf{z}^*$ and $\mathbf{z}^{nad}$;
Randomly assign each weight, $\mathbf{w}^i$ with a candidate solution, $\mathbf{x}^i$;
Calculate the Euclidean distance between weights, $\mathbf{w}^i$ and $\mathbf{w}^j$, $i, j \in 1, 2, \cdots, N$;
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 Q of $\mathbf{x}^i$;
Set *iteration* $\leftarrow 0$, set $matingS \leftarrow \emptyset$;
while *the stopping criterion is not satisfied* **do**
    for $i \leftarrow 1$ **to** N **do**
        if $rand < \delta$ **then**
            $matingS \leftarrow Q$;
        else
            $matingS \leftarrow S$;
        end
        Randomly select three solutions $\mathbf{x}^{r1}, \mathbf{x}^{r2}$ and $\mathbf{x}^{r3}$ from the mating pool, $matingS$;
        Generate a new solution $\mathbf{x}^{new}$ by performing differential evolution (DE) and polynomial mutation (PM) operators;
        Evaluate the objective value of $\mathbf{x}^{new}$, and update the ideal and nadir vectors;
        for *each* $\mathbf{x}^k \in Q$ **do**
            Compare $g^{wd}(\mathbf{x}^{new}|\mathbf{w}^k, p^k)$ with $g^{wd}(\mathbf{x}^k|\mathbf{w}^k, p_k)$;
        end
        Replace no more than nr solutions in Q with $\mathbf{x}^{new}$ if $g^{wd}(\mathbf{x}^{new}|\mathbf{w}^k, p_k)$ is smaller;
    end
    Update the p_k value for each search direction using Algorithm 1;
end

concave Pareto optimal front. WFG42 has a convex Pareto optimal front. The Pareto optimal front of WFG43 is a hyperplane. The number decision variables of these problems is set to $n = 100$ wherein the WFG position parameter (k) and the distance parameter (l) are $\frac{m-1}{2}$ and $100 - k$, respectively. The Pareto optimal front of these problems has the same trade-off magnitudes, and it is within $[0, 2]$. These problems are invoked in 2-, 4- and 7-objective instances. Note that unless otherwise stated we use WFGn-Y to denote the problem WFGn with Y objectives.

General parameters The following parameters are set constant across all algorithm runs:

- *Algorithm runs and stopping criterion*: each algorithm is performed for 31 runs, each run for 25,000 function evaluations.
- *Population size*: $N = 200$ for bi-objective problems, 400 for 4-objective problems, and 700 for 7-objective problems.
- *DE and PM operators*: the DE control parameters are set as $F = 0.5$ and $CR = 0.9$. The mutation probability $pm = 1/n$ and its distribution index is set to be $\eta_m = 20$.
- *The initial candidate p values*: $p \in P = \{\frac{1}{2}, \frac{2}{3}, 1, 2, 3, 4, 5, 10, 1000\}$.
- *MOEA/D parameters*: the selection neighbourhood size is set to 10% of N, the replacement size (nr) is 10% of T.

4.2 Experimental results

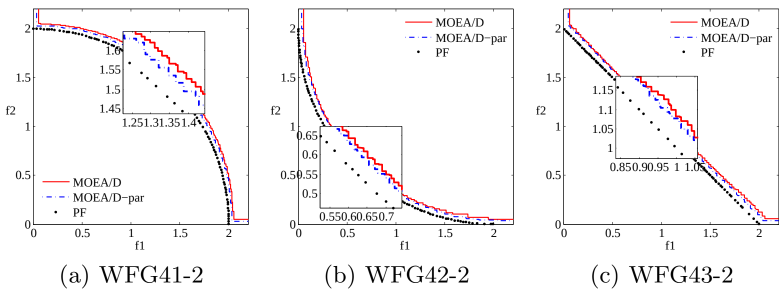


Fig. 7. (Color online) Attainment surfaces for the 2-objective WFG4X problems.

Median attainment surfaces Plots of median attainment surfaces across the 31 runs of each algorithm are shown in Fig. 7. These allow visual inspection of performance in terms of the dual aims of proximity to and diversity across the global trade-off surface. The PF of each problem serves as a reference. From inspection of Fig. 7, the two algorithms appear to have comparable diversity performance while the MOEA/D-par has a clear better convergence performance than MOEA/D for all the three problems.

Table 1. Comparison results of the HV and C metric values for the WFG4X problems. The symbol '<', '=' or '>' means MOEA/D is statistically worse, comparable or better than MOEA/D-par. A refers to MOEA/D, B refers to MOEA/D-par.

	HV(A)	HV(B)	C(A,B)	C(B,A)
WFG41-2	0.2982(0.0052)	< 0.3138(0.0031)	0.0052(0.0104)	< 0.9640(0.0426)
WFG42-2	0.7710(0.0047)	< 0.7887(0.0038)	0.0984(0.0668)	< 0.7846(0.0464)
WFG43-2	0.5286(0.0071)	< 0.5460(0.0021)	0.0438(0.0876)	< 0.7548(0.0404)
WFG41-4	0.4668(0.0097)	< 0.5861(0.0028)	0.0020(0.0041)	< 0.3495(0.0427)
WFG42-4	0.8828(0.0072)	< 0.9100(0.0079)	0(0)	< 0.1355(0.0642)
WFG43-4	0.7318(0.0218)	< 0.8147(0.0194)	0.0044(0.0051)	< 0.1118(0.0638)
WFG41-7	0.5319(0.0212)	< 0.7204(0.0340)	0(0)	< 0.0633(0.0368)
WFG42-7	0.9500(0.0069)	< 0.9556(0.0011)	0.0263(0.0137)	< 0.0551(0.0178)
WFG43-7	0.8112(0.0165)	< 0.8884(0.0084)	0(0)	< 0.0797(0.0239)

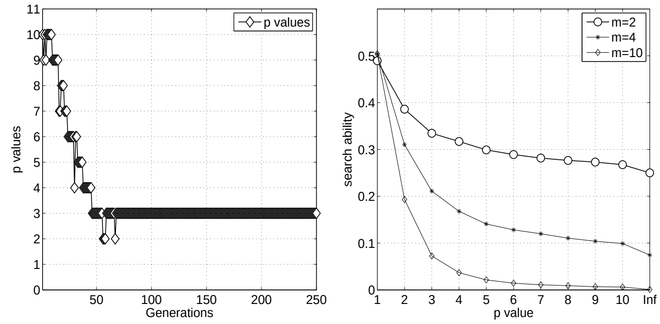
Comparison results in terms of the *HV* and *C* metrics Comparison results of MOEA/D-par with MOEA/D in terms of the HV and C metrics are presented in Table 1. A favourable HV value (larger, for a minimisation problem) implies good proximity with diversity. In our experimental studies, the reference point is set to $1.1 \times \mathbf{z}^{nad}$, i.e., $(2.2, 2.2, \cdots, 2.2)$. The C metric is a binary metric which provides information on convergence. For example, given two sets, A and B, $C(A, B)$ refers to the fraction of solutions in B that are dominated at least by one solution in A. $C(A, B) > C(B, A)$ indicates a better convergence of the A set. Moreover, the non-Parametric Wilcoxon-ranksum two-sided comparison procedure at the 95% confidence level is employed to compare the significance of difference between two algorithms.

From Table 1, we can clearly observe that MOEA/D-par performs better than MOEA/D for all problems in terms both the HV and C metrics. As the only difference between MOEA/D-par and MOEA/D lies in the use of Pareto adaptive scalarising functions, such results are able to confirm that provided a good estimation of the Pareto front geometry, the use of Pareto adaptive scalarising function is helpful, which can improve the performance of MOEA/D significantly for both bi- and many-objective problems (up to 7 objectives).

4.3 Experimental discussions

This section investigates two issues, as part of a wider discussion for the use of Pareto adaptive scalarising functions. First, we examine the obtained p values in MOEA/D-par; Second, the range of the candidate p values.

Observation of the obtained *p* values Empirical comparison results have demonstrated the benefits of using Pareto adaptive scalarising functions in MOEA/D.



(a) (b)

Fig. 8. (a)The obtained p values for WFG41-4 over generations; (b) The change of search ability of different L_p scalarising function in 2-, 4- and 10-objective problems.

Here, we show the obtained p values for the search direction $\mathbf{w} = (0.5, 0.5)$ over generations, as an evidence of the superior performance of MOEA/D-par over MOEA/D. Due to the limited space, Fig. 8(a) only illustrates the obtained p values for WFG41-4. The Pareto optimal front of WFG41-4 is a hyper-sphere. This indicates that the threshold p value is 2, and thus, the obtained p value should be 3 provided on the considered candidate p values, i.e., $p \in P = \{\frac{1}{2}, \frac{2}{3}, 1, 2, 3, 4, 5, 10, 1000\}$. As is expected, it is observed from Fig. 8(a) that the p values gradually converge to 3. As previously analysed, the $L_{p=3}$ scalarising function is able to find all Pareto optimal solutions for a sphere type Pareto front, i.e., WFG41, and simultaneously, $L_{p=3}$ has a better search ability than the Chebyshev scalarising function.

Analysis of the range of the initial candidate *p* values In principle p can be any value within the interval $(0, \infty]$. However, regarding the computational efficiency, we expect to shrink the range of p as much as possible. Of course, such a shrink should not lead to a severe deterioration of the algorithm performance. In this section, we conduct a simple experimental analysis on the effect of different p values so as to set an upper bound of the candidate p value.

Let us consider a set of $1000 \times m$ points that are uniformly sampled from the hypercube $(0, 2]^m$, where m is the dimension of objective space. Also, consider the contour lines of the L_p scalarising functions along the direction of $\mathbf{w} = \{1/m, \cdots, 1/m\}$. Such contour lines intersect the point $\mathbf{x} = (1, \cdots, 1)$. Then we count the number of points that satisfy the condition $\sum_{i=1}^{m} (\mathbf{x}_i)^p < m$, indicating that $\mathbf{x}$ is better than solutions on the contour line of the $g^{wd}(\mathbf{x}|\mathbf{w}, p) = m$. The experiments are repeated for 100 times. The mean proportion of better points over p values varying from 1 to 10 are plotted in Fig. 8(b) for $m = 2, 4$ and 10-dimension spaces, respectively. From the figure, we find that the search ability of the L_p scalarising function decreases dramatically from $p = 1$ to $p = 5$ whereas slightly when $p > 5$. Moreover, the larger the problem dimension, the faster the decrease of the L_p search ability. The search ability of $L_{p=10}$ appears no significant advantage over the weighted Chebyshev function, in particular, in the 10-dimension space. Therefore, we tentatively recommend that despite the Chebyshev function, $p = 10$ might be considered as an upper bound for the candidate p values.

5 Conclusion

Decomposition based algorithms comprise a popular class of multi-objective evolutionary algorithms, and have been demonstrated to perform well when a suitable set of weighted scalarising functions are provided. The effect of weights, including methods for determining suitable weights, have been intensively studied. However, the effect of scalarising functions is far from being well understood. In this paper we study the properties of the family of L_p scalarising functions, and identify that the p value corresponds to a trade-off between the scalarising function's search ability and its robustness on Pareto front geometry. Moreover, we propose to use different Pareto adaptive scalarising functions along different search directions. A naive method is employed to perform an on line Pareto front geometry estimation, and thus, identifying a suitable L_p function. Experimental results show that MOEA/D using Pareto adaptive scalarising functions outperforms the standard MOEA/D for problems having different Pareto front geometries.

It should be pointed out that there are a number of ways in which the central contributions of this study are limited. First, we are aware of some other methods handling the choice of scalarising functions, for example, an adaptive use (a simultaneous use) of the Chebyshev and weighted sum approaches by Ichibuchi et al. [29, 30]. In future, a comprehensive analysis regarding the advantages and disadvantages of these methods will be conducted. Second, though the employed Pareto front geometry estimation strategy appears to work well on the considered test problems, it is rather limited, more effective methods are required. As a start, it is non-trivial to investigate how a suitable set of neigobouring solutions should be chosen as this plays an important role for discontinuous Pareto front geometry estimation. Third, adaptation of scalarising functions accounts effectively varying the subproblems. As discussed in [34], the adaptation can lead to reduced convergence rates, and thus, the effect of adaptation of scalarising functions should be investigated further. Lastly, findings of this study are based on three basic continuous MOPs. It is also important to assess the performance of MOEA/D-par on problems having other complex geometries, other problem types, e.g. multi-objective combinatorial problems, and also, crucially, real-world problems.

Acknowledgment

This research was conducted in the Department of computer science, City University of Hong Kong, and the first author is grateful for the facilities and support provided by the University. The first author would also like to thank the National Natural Science Foundation of China (No. 61403404,61473241) and the National University of Defense Technology (No. JC 14-05-01) for their financial support.

References

1. K. Deb, Multi-objective optimization using evolutionary algorithms, Wiley, 2001.
2. C. Fonseca, 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. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
4. R. Wang, R. Purshouse, P. Fleming, Preference-inspired Co-evolutionary Algorithms for Many-objective Optimisation, IEEE Transactions on Evolutionary Computation 17 (4) (2013) 474–494.
5. E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature – PPSN VIII, Springer, 2004, pp. 832–842.
6. J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (2011) 45–76.
7. T. Murata, H. Ishibuchi, M. Gen, Specification of genetic search directions in cellular multi-objective genetic algorithms, in: Evolutionary Multi-Criterion Optimization, Springer, 2001, pp. 82–95.
8. E. Hughes, Multiple single objective pareto sampling, in: Evolutionary Computation (CEC), 2003 IEEE Congress on, IEEE, 2003, pp. 2678–2684.
9. Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
10. R. Wang, P. Fleming, R. Purshouse, General framework for localised multi-objective evolutionary algorithms, Information Sciences 258 (2) (2014) 29–53.
11. H.-L. Liu, F. Gu, Q. Zhang, Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 450–455.
12. H. Li, Q. Zhang, Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 284–302.
13. Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: Evolutionary Computation (CEC), 2009 IEEE Congress on, IEEE, 2009, pp. 203–208.
14. H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, in: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1, IEEE, San Antonio, TX, USA, 2009, pp. 1820–1825.
15. Q. Zhang, Research articles and applications related to MOEA/D, `http://dces.essex.ac.uk/staff/zhang/webofmoead.html`.
16. I. Giagkiozis, R. C. Purshouse, P. J. Fleming, Generalized decomposition, in: Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 428–442.

17. H. Ishibuchi, N. Akedo, Y. Nojima, A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems, in: Learning and Intelligent Optimization, Springer, 2013, pp. 231–246.
18. I. Giagkiozis, R. Purshouse, P. Fleming, Generalized decomposition and cross entropy methods for many-objective optimization, Information Sciences In Press (2014) 1–25.
19. R. Wang, R. Purshouse, P. Fleming, Preference-inspired co-evolutionary algorithms using weight vectors, European Journal of Operational Research, In press-doi:10.1016/j.ejor.2014.05.019.
20. R. Wang, R. Purshouse, P. Fleming, Preference-Inspired Co-Evolutionary Algorithm Using Weights for Many-objective Optimisation, in: GECCO 2013: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, Amsterdam,Netherlands, 2013, pp. 101–102.
21. S. Jiang, Z. Cai, J. Zhang, Y. Ong, Multiobjective optimization by decomposition with Pareto-adaptive weight vectors, in: Natural Computation (ICNC), 2011 Seventh International Conference on, IEEE, 2011, pp. 1260–1264.
22. H. Li, D. Landa-Silva, An adaptive evolutionary multi-objective approach based on simulated annealing, Evolutionary Computation 19 (4) (2011) 561–595.
23. Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with Adaptive Weight Adjustment, Evolutionary Computation 22 (2) (2013) 231–264.
24. B. Derbel, D. Brockhoff, A. Liefooghe, Force-Based Cooperative Search Directions in Evolutionary Multi-objective Optimization, in: Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 383–397.
25. Y. Jin, T. Okabe, Adapting weighted aggregation for multiobjective evolution strategies, in: Evolutionary Multi-Criterion Optimization, 2001, pp. 96–110.
26. K. Miettinen, Nonlinear multiobjective optimization, Springer, 1999.
27. B. Derbel, D. Brockhoff, A. Liefooghe, S. Verel, On the impact of multiobjective scalarizing functions, in: Parallel Problem Solving from Nature–PPSN XIII, Springer, 2014, pp. 548–558.
28. I. Giagkiozis, P. Fleming, Methods for multi-objective optimization: An analysis, Information Sciences 293 (0) (2015) 338 C 350.
29. H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm, in: Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 438–452.
30. H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Simultaneous use of different scalarizing functions in MOEA/D, in: GECCO 2010: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, Portland,USA, 2010, pp. 519–526.
31. R. Wang, Towards understanding of selection strategies in many-objective optimisation., Research Report No. 1096, College of Information Systems and Management, National University of Defense Technology (November 2014).
32. S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (2006) 477–506.
33. A. Hernández-Díaz, L. Santana-Quintero, C. Coello Coello, J. Molina, Pareto-adaptive ε-dominance, Evolutionary computation 15 (4) (2007) 493–517.
34. I. Giagkiozis, R. C. Purshouse, P. J. Fleming, Towards Understanding the Cost of Adaptation in Decomposition-Based Optimization Algorithms, in: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, IEEE, 2013, pp. 615–620.