Localized weighted sum method for many-objective optimization

Rui Wang, Zhongbao Zhou, Hisao Ishibuchi, *Fellow, IEEE*, Tianjun Liao, Tao Zhang

***Abstract*—Decomposition via scalarization is a basic concept for multi-objective optimization. The weighted sum method, a frequently used scalarizing method in decomposition based evolutionary multi-objective (EMO) algorithms, has good features such as computationally easy and high search efficiency, compared to other scalarizing methods. However, it is often criticized by the loss of effect on non-convex problems. This study seeks to utilize advantages of the weighted sum method, without suffering from its disadvantage, to solve many-objective problems. A novel decomposition based EMO algorithm called MOEA/D-LWS is proposed in which the weighted sum method is applied in a local manner. That is, for each search direction, the optimal solution is selected only amongst its neighboring solutions. The neighborhood is defined using a hypercone. The apex angle of a hypervcone is determined automatically in *a priori*. The effectiveness of MOEA/D-LWS is demonstrated by comparing it against three variants of MOEA/D, i.e., MOEA/D using Chebyshev method, MOEA/D with an adaptive use of weighted sum and Chebyshev method, MOEA/D with a simultaneous use of weighted sum and Chebyshev method, and four state-of-the-art many-objective EMO algorithms, i.e., PICEA-g, HypE, θ-DEA and SPEA2+SDE for the WFG benchmark problems with up to seven conflicting objectives. Experimental results show that MOEA/D-LWS outperforms the comparison algorithms for most of test problems, and is a competitive algorithm for many-objective optimization.**

***Index Terms*—Multi-objective optimization, evolutionary computation, decomposition, weighted sum, local, MOEA/D.**

I. INTRODUCTION

MULTI-OBJECTIVE problems (MOPs) arise regularly in real life, where multiple objectives have to be optimized simultaneously. Typically, an MOP can be written as follows:

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

where $\mathbf{x}$ is a decision vector in $\mathbf{\Omega}$ (which refers to a feasible search space), $\mathbb{R}^m$ refers to the objective space. $\mathbf{F}: \mathbf{\Omega} \rightarrow \mathbb{R}^m$ consists of m real-valued objective functions. Since objectives in an MOP are often conflicting with one another, the optimal solution set is not a single solution but a set of trade-off solutions, namely, Pareto optimal solutions (PS). The Pareto optimal front (PF), the image set of all the PS in the objective space, is of practical interest to a decision maker.

EMO algorithms have demonstrated their effectiveness in solving MOPs. Their population based nature enables an approximation of the PF to be obtained in a single run, and they tend to be robust to underlying cost function characteristics [1, pp. 5-7]. It has been accepted that Pareto-dominance based EMO algorithms, MOGA [2], NSGA-II [3], though perform well on MOPs with two and three objectives, often have difficulty on handling MOPs with more than three objectives (termed many-objective problems) [4]. Up to now a number of EMO algorithms have been proposed to handle many-objective problems, which can be loosely classified as follows [5]: i) modified Pareto-dominance or density estimation) based, e.g., ϵ-dominance based algorithm (ϵ-EMOA [6]) shift-density based evolutionary algorithm (SDE [7]), grid-based evolutionary algorithm (GrEA [8]); ii) Performance indicator based, e.g., the indicator based evolutionary algorithm (IBEA [9]), approximated hypervolume based evolutionary algorithm (HypE [10]); preferences-based, e.g., preference-inspired co-evolutionary algorithms (PICEA-g [11], [12], [13] and PICEA-w [14], [15]); decomposition based, e.g., cellular multi-objective genetic algorithm (CMOGA [16]), multi-objective evolutionary algorithm based on decomposition (MOEA/D [17] and its variants MOEA/D-DD [18], MOEA/D-DU [19]), NSGA-III [20], θ dominance based algorithm (θ-DEA [21]). Also, there are some other promising approaches such as the bi-goal evolutionary algorithm [22], the knee point-driven evolutionary algorithm (KnEA [23]), the improved two-archive algorithm (Two_Arch2 [24]) and [25], [26], [27], [28]. Readers are referred to [29], [30] for a detailed survey. Amongst the above mentioned algorithms, decomposition based EMO algorithms have drawn more and more attentions.

Decomposition via scalarization is a basic technique in traditional multi-objective optimization. A typical decomposition based EMO algorithm, e.g, MOEA/D, decomposes an MOP into a number of sub-problems (which are often single-objective[1]) defined by a scalarizing method using different weights, and solve these sub-problems using a population based search in a collaborative manner. The optimal solution of each single-objective problem corresponds to a Pareto optimal

This work was supported by the National Natural Science Foundation of China (Nos. 61403404, 71401167,71371067 and 71571187) and the National University of Defense Technology (No. JC14-05-01). This work was also partially supported by JSPS KAKENHI Grant Numbers 24300090 and 26540128.

Rui Wang and Tao Zhang are with the College of Information Systems and Management, National University of Defense Technology, P.R. China, 410073. E-mail: ruiwangnudt@gmail.com.

Zhongbao Zhou (corresponding author) is with Business School, Hunan University, P.R. China, 410082. E-mail: z.b.zhou@163.com.

Hisao Ishibuchi is with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Osaka 599-8531, Japan. E-mail: hisaoi@cs.osakafu-u.ac.jp.

Tianjun Liao is with State key Laboratory of Complex System Simulation, Beijing Institute of System Engineering, Beijing, P.R.China,100000. E-mail: juntianliao@163.com.

Manuscript received XX XX, 2016; revised XX XX, 2016.

[1]Note that in some variants of MOEA/D sub-problems are multi-objective [31]

solution of the MOP [32, pp.98-99]. Diversified solutions are obtained by employing different weights.

More recent studies have identified that the specification of weights [33] and/or scalarizing methods [34] has a crucial impact on the performance of MOEA/D and other decomposition based EMO algorithms.

- The specification of weights mainly impacts the distribution of the approximated PF. For example, evenly distributed weights might not lead to evenly distributed solutions for an MOP whose PF shape is non-linear (i.e., convex or concave) [35]. Effectively when the PF is known *a priori*, the studies [33], [36] investigate how to compute an optimal distribution of weights for various L_p scalarizing methods. When the PF is unknown, a few effective methods are proposed, e.g., co-evolving weights with solutions [15], [37]; and using Pareto adaptive weights [38], [39], [40], [41], [42].
- The specification of scalarizing methods mainly impacts the search efficiency. As discussed in [32, p. 79] that the weighted Chevbshev method can find solutions in both convex and non-convex PF regions whereas the weighted sum cannot; and in [43], [44], [45] that the weighted sum method (or other L_p scalarizing methods) generally leads to a better convergence performance than the weighted Chebyshev method. Moreover, it is recently reported in [36], [46] that as p increases (from 1 to ∞), the L_p scalarizing method becomes more robust on PF geometries and less effective in terms of the search efficiency. An optimal p setting exists for problems having certain PF shapes [36].

The simultaneous optimization of objectives over three (termed many-objective optimization) remains challenging in terms of obtaining a full and satisfactory approximation of the PF. These challenges include i) inefficiency of Pareto-dominance relation, ii) increased conflict between convergence and diversity, iii) difficulty of the calculation of some performance metrics, iv) inefficiency for offspring generation, v) the representation and visualization of trade-off surface.

In order to handle many-objective optimization, this study proposes a localized weighted sum method which utilizes the high search efficiency of the weighted method, meanwhile avoids its disadvantage (inefficiency on non-convex problems). Major contributions are as follows.

- A localized weighted sum (LWS) method is proposed, i.e., using the weighted sum method in a local manner. The LWS method enables a decomposition based EMO algorithm to find solutions in both convex and non-convex PF regions. The neighborhood size of a LWS, namely, its working space, is defined using a hypercone. The apex angle of the hypervcone equals to the average angle to its nearest m neighboring weights. The apex of the hypercone is automatically determined in an *a priori*.
- An instantiation of decomposition based algorithms using the LWS method, denoted as MOEA/D-LWS, is proposed. MOEA/D-LWS introduces no additional parameter, and is demonstrated to outperform MOEA/D using the Chebyshev method, MOEA/D with an adaptive use of weighted sum and Chebyshev method [45] and MOEA/D with a simultaneous use of weighted sum and Chebyshev method [47] on most of the WFG benchmark problems with up to seven conflicting objectives.
- MOEA/D-LWS is compared against three many-objective optimizers, i.e., PICEA-g and HypE and θ-DEA, and is shown as competitive.

The remainder of this paper is organized as follows. Section II provides some background knowledge. Section III elaborates the localized weighted sum method and the algorithm MOEA/D-LWS. Experimental descriptions, results and related discussions are presented in Section IV and Section V, respectively. Finally, Section VII concludes this paper and identifies some future studies.

II. Background

A. Basic definitions

Pareto-dominance: $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$, denoted by $\mathbf{x} \preceq \mathbf{y}$, if and only if $\forall i \in \{1, 2, \ldots, m\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one index $j \in \{1, 2, \ldots, m\}$.

Pareto optimal solution: A solution $\mathbf{x}^* \in \mathbf{\Omega}$ is said to be Pareto optimal if and only if $\nexists \mathbf{x} \in \mathbf{\Omega}$ such that $\mathbf{x} \preceq \mathbf{x}^*$. The set of all Pareto optimal solutions is called the Pareto optimal set (PS). The set of all Pareto optimal vectors, PF $= \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in \text{PS}\}$, is called the Pareto optimal front (PF).

***Ideal* point**: An objective vector $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$ where z_i^* is the infimum of $f_i(\mathbf{x})$ for every $i \in \{1, 2, \ldots, m\}$.

***Utopian* point**: An infeasible $\mathbf{z}^u$ whose component can be formed by $z_i^u = z_i^* - \epsilon_i,\ i = 1, 2, ..., m$, where z_i^* is the component of the *ideal* objective vector, and $\epsilon_i > 0$ is a relatively small but computationally significant scalar.

***Nadir* point**: An objective vector $\mathbf{z}^{nad} = (z_1^{nad}, \ldots, z_m^{nad})$ where z_i^{nad} is the supremum of $f_i(\mathbf{x}), \mathbf{x} \in \text{PS}$ for every $i \in \{1, 2, \ldots, m\}$.

B. Decomposition approaches: L_p scalarizing method

Though a number of scalarizing methods are available for decomposing an MOP, we focus on the family of L_p scalarizing methods due to its simpleness and popularity[2]. Mathematically, a weighted L_p scalarizing method can be written as,

$$
\begin{gathered}
g^{wd}(\mathbf{x}|\mathbf{w}, p) = \left(\sum_{i=1}^{m} \lambda_i \left(f_i(\mathbf{x}) - z_i^u \right)^p \right)^{\frac{1}{p}},\ p \geq 1 \\
\lambda_i = \left(\frac{1}{w_i} \right)
\end{gathered} \tag{2}
$$

where $\mathbf{z}^u = (z_1^u, z_2^u, \ldots, z_m^u)$ is the *utopian* point, $\mathbf{w} = (w_1, w_2, \ldots, w_m)$ is a weighting vector and $\sum_{i=1}^m w_i = 1,\ w_i > 0$.

[2]There are other scalarizing methods such as the penalty-based boundary intersection (PBI) [17], the normal boundary intersection (NBI) [48]

The two frequently used scalarizing methods, the weighted sum and Chebyshev, can be derived by setting $p = 1$ and $p = \infty$, respectively. That is,

$$\begin{aligned} g^{ws}(\mathbf{x}|\mathbf{w}) &= \sum_{i=1}^{m} (\lambda_i (f_i(\mathbf{x}) - z_i^u)) \\ g^{ch}(\mathbf{x}|\mathbf{w}) &= \max_{i=1}^{m} (\lambda_i (f_i(\mathbf{x}) - z_i^u)) \end{aligned} \tag{3}$$

By optimizing a scalarizing method with different weights, a set of Pareto optimal solutions could be obtained. If necessary, the following normalization procedure should be applied during the search:

$$\overline{f}_i = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*} \tag{4}$$

When the z_i^* and z_i^{nad} are not available, we could use the smallest and largest f_i of all non-dominated solutions found so far as approximations of z_i^* and z_i^{nad} as in many other studies (e.g., [17], [25], [49]. However, this approximation is not always very accurate especially in early generations. If necessary, more advanced methods can be applied [50].

C. Weighted sum method in multi-objective optimization

As a common concept in multi-objective optimization, the weighted sum method has been discussed prominently in the literature [32], [51], [52] since its introduction by Zadeh [53]. The method linearly aggregates all the individual objective functions in an MOP into one objective by using a weight vector.

Before EMO algorithms get popularized, the weighted sum method is mainly used in an *a priori* and *interactive* way, that is, a weight vector is pre-defined before the search, or changed during the search progressively. For instance, in [54] the weighted sum method is applied to multi-objective structure optimization. Weights are pre-defined, multiple Pareto optimal solutions are obtained by a systematic change in weights in different algorithm runs. Also, in [55] the weighted sum method is applied to topology optimization. Weights are altered to yield different Pareto optimal solutions. Within a EMO algorithm, the weighted sum method, embedded with a set of pre-defined weights, is applied to search a set of Pareto optimal solutions in a single run. For example, in [56] the weighted sum method embedded with random weights is used for selecting good solutions in a multi-objective genetic algorithm. In MSOPS [57] and MOEA/D [17], the weighted sum method (as one of the proposed methods), embedded with evenly distributed weights is employed for multi-objective optimization.

Optimizing a weighted sum could constitute either an independent method or a component of other methods. In [58], [59] a simulated annealing algorithm is proposed wherein the weighted sum method is applied as an acceptance criterion. In [60] the weighted sum method is combined with tabu search for solving bi-objective 0-1 knapsack problems. In the Two-Phase Local Search (TPLS) and Double-TPLS [61] and the MOGLS [62], [63], [64] the weighted sum method is also used to guide the local search.

Despite its wide applications, many studies demonstrate its inability on capture Pareto optimal solutions in non-convex regions [35], [65], [66]. This deficiency is often answered with alternative scalarizing methods, e.g., the Chebyshev method, the NBI method. Noticeably, Wang et al [36], [46] propose to use Pareto adaptive L_p scalarizing methods, that is, choosing a suitable L_p scalarizing method based on the estimated PF shape on line. Such alternatives can be effective and valuable, whereas they are usually independent of the weighted sum method. From another aspect, there are studies investigating the incorporation of tricks into the weighted sum method. For example, in [67], [68] Jin et al. propose that by using an archive and periodically altered weights, the weighted sum method based search is able to obtain solutions on non-convex PF. In [69], [70] Kim and de Weck propose to first use the weighted sum method to quickly obtain an approximation of the PF, then construct a mesh of Pareto front patches. The patches are further refined by imposing additional equality constraints which connect the *nadir* point and the expected Pareto optimal solutions on a piecewise planar surface in the objective space. This method is reported as effective. However, constructing the mesh of Pareto front patches becomes rather complex on many-objective problems. In [71] a bi-level weighted sum method is proposed for multi-objective optimization. However, the idea is still based on the use of Pareto front patches.

In addition to the above-mentioned approaches Ishibuchi et al. propose a mixed use of the weighted sum and Chebyshev method in attempt to harness benefits of both methods [45], [47]. Specifically, in [45] the weighted sum is used to guide the search unless a concave PF region is detected in which case the Chebyshev method is used instead. In [47] the weighted sum and Chebyshev method are simultaneously applied to evaluate candidate solutions. Experimental results show that both the methods can find solutions in non-convex PF regions. Also, their performance, especially the convergence, is much better than a single use of the Chebyshev method.

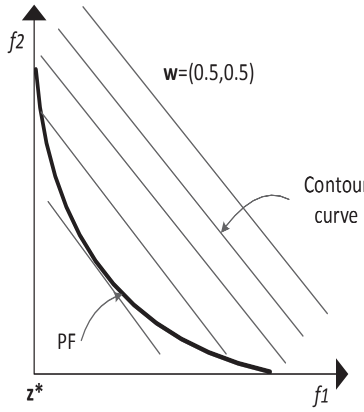
Overall the literature has repeatedly demonstrated that i) the weighted sum method cannot find solutions on non-convex PF regions, and ii) the weighted sum method has high search efficiency than the Chebyshev method. In order to harness its benefits and avoid its drawbacks, this study proposes to use it in a local manner, that is, the working space of a weighted sum method is restricted within a hypercone. Given the high search efficiency of the weighted sum method, decomposition based EMO algorithms using the localized weighted sum method are expected to have good performance on many-objective problems.

III. MOEA/D-LWS

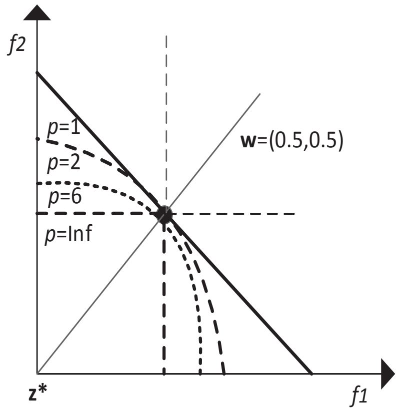
A. Motivation: high search efficiency of the weighted sum method

High search efficiency of the weighted sum method is the main motivation for promoting its application in decomposition based EMO algorithms. It is well known that Pareto dominance-based fitness evaluation mechanisms do not work well on many-objective problems [4], [72]. As explained in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2611642, IEEE Transactions on Evolutionary Computation



(a) Weighted sum



(b) L_p methods

Fig. 1. Illustration of contour curves of weighted sum method and other L_p scalarizing methods.

[73],"*Pareto-dominance methods and the Chebyshev scalarizing function are equivalent, in the sense that neither method in itself, has better probability to find superior solutions. In fact the aforementioned probabilities are the same*". Thus it is likely that the Chebyshev method does not work well on many-objective functions. Actually, [45], [73] have shown that better results are obtained by the weighted sum than the Chebyshev method for some many-objective problems. That is, it has been reported that much efficient search is realized by the weighted sum than the Chebyshev method. However, the weighted sum has its inherent disadvantage: It cannot appropriately handle non-convex PFs. In this study, we propose an idea to remedy this disadvantage in order to utilize the high search ability of the weighted sum for many-objective problem in decomposition based algorithms. Next, analyis of the search efficiency of the weighted sum and Chebyshev method, inspired by [73], is presented.

The contour curve of the weighted sum method is shown in Fig. 1(a), which is a straight line. The contour curve divides the objective space into two subspaces. Solutions in one subspace are better than those on the contour curve while solutions in the other subspace are worse. Solutions lie in the same contour curve have the identical scalar value. That is, the size of a superior region is loosely $\frac{1}{2}$, and is regardless of the number of objectives. Compared with other L_p scalarizing methods, we can observe from their contour curves, as shown in Fig. 1(b), that the size of superior region is smaller than $\frac{1}{2}$, e.g., $\frac{1}{2^m}$ for the Chebyshev method. Moreover, such value decreases significantly as m, the number of objectives, increases. This therefore indicates that i) the probability of finding a better solution (measured by the chosen scalarizing method) is $\frac{1}{2}$ for the weighted sum method and is smaller than $\frac{1}{2}$ for other L_p scalarizing methods; and ii) the probability keeps unchanged for the weighted sum method while it becomes remarkably small for other L_p scalarizing methods when m increases. As a result, it is not likely that the Chebyshev method is very efficient for many-objective problems in comparison with the weighted sum method [73].

Empirically, MOEA/D embedded with the weighted sum method and the Chebyshev method are compared for the ZDT1 benchmark [74] (ZDT is for Zitzler-Deb-Thiele). Both algorithms are run for 31 times, and each run with 250 generations. The population size is set to 100. Other settings are the same as in [75]. The PFs obtained by each algorithm, corresponding to the median hypervolume value over 31 runs, are shown in Fig. 2. From the results, it is evident that the weighted sum method offers a better performance. The obtained solutions are closer to the true PF than those obtained by the Chebyshev method.

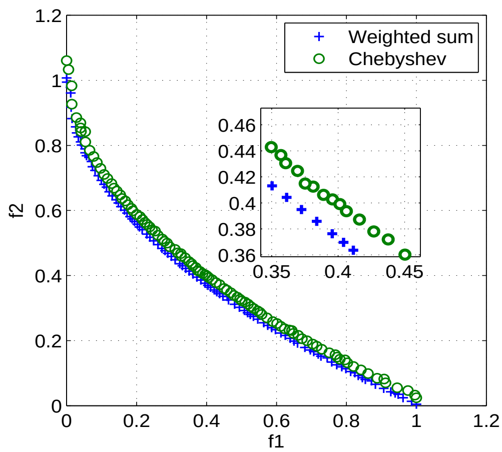


Fig. 2. PFs obtained by MOEA/D with the weighted sum and the Chebyshev method.

Therefore it is desirable to apply the weighted sum method to guide the evolutionary search. Of course an effective strategy that enables the weighted sum method to find solutions in non-convex PF regions is required.

B. Methodology: the localized weighted sum method

This section presents our idea— the localized weighted sum method, denoted as LWS. The LWS applies the weighted sum method in a local manner. That is, each weighted sum method $g^{ws}(\mathbf{x}|\mathbf{w}^i)$ is restricted to work only within a defined hypercone. The centre line of the hypercone is along the weight $\mathbf{w}^i$, and its apex angle is Θ_i, see Fig. 3 the shaded region. The main

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2611642, IEEE Transactions on Evolutionary Computation

reason for setting the working space as a hypercone (rather than a hypercylinder) is as follows. Assuming that N weight vectors: $\mathbf{w}^1$, $\mathbf{w}^2$, ... $\mathbf{w}^N$ are employed, and the PF is divided into N parts (which essentially is the way a decomposition based algorithm does). Each weighted sum method with a different weight vector tries to find a Pareto optimal solution. Given the shape of the hypercone (wide at the bottom and narrow at the top, see Fig. 3), its application, compared with the use of cylinder, implicitly enables a larger space to be explored (i.e., more solutions to be evolved) at the early stage of the search along each search direction. This is obviously beneficial in finding a global optimum. In another aspect, as the search progresses solutions gradually approach to the PF. The solution found by each weighted sum method is expected to be within the associated region of PF. Thus, the working space should narrow down. The region of the hypercone suits exactly such a process, and therefore is employed.

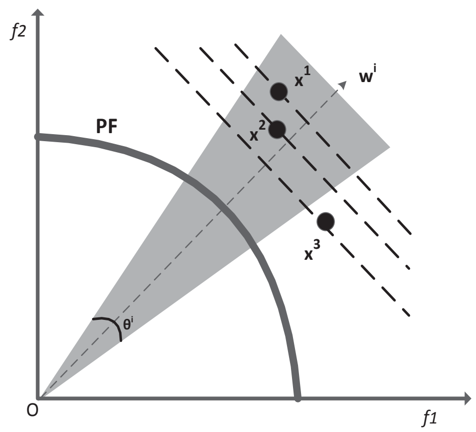


Fig. 3. Illustration of the hypercone region for weight vector $\mathbf{w}^i$.

A weighted sum method $g^{ws}(\mathbf{x}|\mathbf{w}^i)$ only measures solutions that are inside the hypercone of $\mathbf{w}^i$. For solutions that are outside the hypercone, their weighted sum values are set to ∞. For a m-objective problem, the apex angle of the hypervcone, Θ_i is defined as follows:

$$\Theta_i = \frac{\sum_{j=1}^{j=m} \theta_{ij}^{ww}}{m} \tag{5}$$

where θ_{ij}^{ww} is the angle of the jth closest weight vector to weight $\mathbf{w}^i$. Note that the dot product $\mathbf{w}^i \cdot \mathbf{w}^j$ provides the cosine of their angle. It is worth mentioning that given N evenly distributed weight vectors, Θ_i is almost identical for different hypercones. Seen from Fig. 3, by the standard weighted sum method solution $\mathbf{x}^3$ is the best, whereas by the localized weighted sum method solution $\mathbf{x}^2$ is considered as the best since $\mathbf{x}^3$ is outside of the hypercone.

Overall, by the localized strategy, the PF is divided into a number of small sub-PFs. Each localized weighted sum method corresponds to a sub-PF (which is bounded by a hypercone), and tends to find a Pareto optimal solution within this hypercone. For a convex sub-PF, the obtained solution might be along (or near) the search direction (which is sub-PF shape dependent). For a non-convex sub-PF, the obtained solution is at the boundary of the sub-PF. Since the centre line of the hypercone is along the search direction $\mathbf{w}^i$, and its apex angle is Θ, the offfset between the obtained solution and $\mathbf{w}^i$ is maximally $\frac{\Theta_i}{2}$. Given that evenly distributed weights are employed, the size of hypecones (determined by Eq. (5)) for different LWS methods is almost the same, which therefore naturally leads to a set of diversified solutions. A further discussion with respect to the uniformity of solutions obtained by LWS is given in Section VI-A.

It is worth mentioning that the very recent inspiring work MOEA/D-DU by Yuan et al [19] also proposes to update solutions that are only within the neighbourhood of the newly generated solution. Differently, the neighbourhood is defined with Euclidean distances. Moreover, an additional parameter k has to be appropriately defined before the search. In addition, effectiveness of the cone based neighbourhood has been reported in [76], [77]. Besides, in MOEA/D-DU, the Chebyshev method is used whereas our study utilizes the high convergence ability of the localized weighted sum.

C. Incorporation of the LWS method into MOEA/D

This section elaborates our proposed algorithm MOEA/D-LWS. The Pseudo-code is presented in Algorithm 1. Prior to the evolution (lines 1 to 5), the following operations are conducted. N evenly distributed weights $W \leftarrow \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N\}$ are generated (as will be described later), and the same size of solutions $S \leftarrow \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$ are initialized. Each $\mathbf{w}$ is randomly paired up with an $\mathbf{x}$. For a weight $\mathbf{w}^i$, its T neighboring weights $B(\mathbf{w}^i)$ are identified. The neighboring relation is determined in terms of θ_{ij}^{ww}. Also the associated neighboring solutions $B(\mathbf{x}^i)$ of $\mathbf{x}^i$ are identified. Lastly, the apex angle of the hypercone Θ_i for $\mathbf{w}^i$ is calculated by Eq. (5). All solutions are then evolved for $maxGen$ generations.

- Lines 6-16: new offspring solutions are reproduced (as will be described below). For a Parent solution $\mathbf{x}^i$, its offspring is generated based the neighbors of $\mathbf{x}^i$, $B(\mathbf{x}^i)$ with a probability 0.8, and based on the whole population S with a probability 0.2.
- Lines 17 to 19: all parent and offspring solutions are combined. Their objective values are computed, which are then applied to update the *ideal* and *nadir* points.
- Line 20: calculate the angle θ_{ij}^{sw} between $\mathbf{F}(\mathbf{x})^{\mathbf{i}}$ and $\mathbf{w}^j$.
- Lines 21 and 22: calculate the weighted sum value $g^{ws}(\mathbf{x}^i|\mathbf{w}^j)$, denoted as C_{ij}. Set C_{ij} as ∞ if θ_{ij}^{sw} is larger than Θ_j
- Lines 23 to 27: for each weight, find $\mathbf{x}$, the solution in the joint population with the smallest $g^{wd}(\mathbf{x}|\mathbf{w}^i)$ value.
- Line 28: update the offline archive $archiveS$ with newly obtained solutions S based on the Pareto-dominance.

Lastly, one can further obtain evenly distributed solutions by choosing the nearest solution (measured by angle) for each weight from $archiveS$.

Generation of evenly distributed weights: First evenly distributed N points on a hypersphere $f_1^2 + f_2^2 + \ldots, + f_m^2 = 1$ are obtained via minimizing a metric V defined in Eq. (6). These points are then converted into weights by $w_i = \frac{x_i}{\sum_{i=1}^{m} x_i}$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2611642, IEEE Transactions on Evolutionary Computation

Algorithm 1: MOEA/D using localized weighted sum scalarizing method, MOEA/D-LWS

Input: N evenly distributed weights, $W \leftarrow \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N\}$, N candidate solutions, $S \leftarrow \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, selection neighborhood size, T, maximum generation index, $maxGen$
Output: S, $archiveS$

1 Assign each weight, $\mathbf{w}^i$ with a randomly selected candidate solution, $\mathbf{x}^i$;
2 Find T neighboring weights $B(\mathbf{w}^i)$ of $\mathbf{w}^i$ in terms of θ_{ij}^{ww};
3 Identify the associated neighboring solutions $B(\mathbf{x}^i)$ of $\mathbf{x}^i$;
4 Calculate the apex angle Θ_i for each weight $\mathbf{w}^i$ by Eq. (5);
5 Set the archive $archiveS = \emptyset$, the mating pool $Q = \emptyset$ and the probability of mating restriction $\delta = 0.8$;
6 **for** $gen \leq maxGen$ **do**
7   **for** $i \leftarrow 1$ **to** N **do**
8   Set a temporary solution set $Sc = \emptyset$;
9   **if** $rand < \delta$ **then**
10       $Q \leftarrow B(\mathbf{x}^i)$;
11   **else**
12       $Q \leftarrow S$;
13   **end**
14   Generate a new solution $\mathbf{x}'$ by applying SBX and PM operators to solutions selected from Q;
15     $Sc \leftarrow Sc \uplus \mathbf{x}'$;
16   **end**
17   $JointS \leftarrow S \uplus Sc$;
18   Compute the objective function values $JointF$ of all candidate solutions in $JointS$;
19   Update *ideal* and *nadir* vectors, and normalize $JointF$ into [0,1];
20   Calculate the angle θ_{ij}^{sw} between $\mathbf{F}(\mathbf{x})^\mathbf{i}$ and weight $\mathbf{w}^j$, obtaining an angle matrix θ^{sw};
21   For each solution, e.g., $\mathbf{x}^i$ and its associated weight, e.g., $\mathbf{w}^j$, if $\theta_{ij}^{sw} \leq \Theta_j$ compute $g^{ws}(\mathbf{x}^i|\mathbf{w}^j)$, denoted as C_{ij};
22   Otherwise, set $C_{ij} \leftarrow \infty$;
23   $S \leftarrow \emptyset$;
24   **for** $i \leftarrow 1$ **to** N **do**
25     $\mathbf{x}^i \leftarrow \arg\min_{\mathbf{x} \in JointS} C_{ij}$;
26     $S \leftarrow S \cup \mathbf{x}^i$;
27   **end**
28   Update $archiveS$ with S using Pareto-dominance relation;
29 **end**

where x_i is the ith component of a point.

$$V = \max_{i=1}^{N} \max_{j=1, j \neq i}^{N} (\mathbf{x}^i \cdot \mathbf{x}^j) \tag{6}$$

The metric V measures the worst-case angle of two nearest neighbors. The inner maximization finds the nearest two neighbors in terms of the angle between them. The outer maximum operator finds the largest angle between two nearest neighbors. The optimal set of weights is produced when the outer maximum is minimized. More details about this method is available in [57]. Compared with the simplex-lattice design method introduced in [17], this method is able to generate any number of evenly distributed weights.

Generation of new offspring solutions: The simulated binary crossover (SBX) and polynomial mutation (PM) operators are applied to generate an offspring population Sc. As recommended in [13], [4], the SBX control parameters p_c and η_c are set to 1 and 30, respectively. The PM control parameters p_m and η are set to $1/n$ and 20 where n is the number of decision variables.

MOEA/D-LWS effectively is within a $(\mu + \lambda)$ elitism framework where $\mu = \lambda = N$. N parent solutions and their offspring solutions of the size N are pooled together. The size of the offspring population can also be different from the parent population size. Then new parents of the size N are selected from the joint population. Specifically, in MOEA/D-LWS the neighborhood structure $B(\bullet)$ is used only for choosing a pair of parents. It is not used for solution update. The generated solution $\mathbf{x}'$ for $\mathbf{w}^i$ is not necessarily compared with $\mathbf{x}^i$ for solution update. Solutions that are close to each weight vector (i.e., neighbors of a weight vector) in the merged population are compared to generate the new parent population. Also, a small number of weight vectors can share the same solution (i.e., the same solution can be selected for different weights in Line 25).

With respect to the time complexity, the calculation of the weighted sum values of all solutions on all weights runs at $\mathcal{O}(N \times N)$. Identification of the minimal weighted sum value among $2N$ values at line 25 runs at $\mathcal{O}(N)$. Therefore, the overall time complexity of the algorithm is $\mathcal{O}(N^2)$.

IV. Experiment description

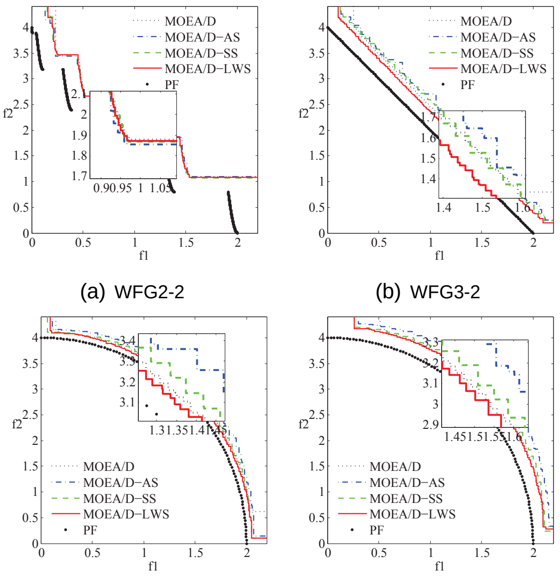
A. Test problems

Test problems 2-9 from the WFG test suite [78], invoked in two-, four-, and seven-objective instances, are applied to benchmark the considered algorithm performance[3]. In each case the WFG position parameter (k) and the distance parameter (l) are set to 6 and 94, respectively, providing a constant number of decision variables ($n = 100$) for each problem instance. The reason for setting $k = 6$ is that WFG problems require k to be divisible by $m - 1$. Choosing $n = 100$ is simply to better demonstrate the superiority of the weighted

[3]The WFG1 is not used since our preliminary experiments show that even for 1e+8 function evaluations no algorithm can approximate its PF. The reason might be the employed search operators lacks the ability for exploiting solutions with high precision. Understanding this issue may well unlock further understanding of MOEA performance. Besides, test problems 2-9 can cover most of the problem attributes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2611642, IEEE Transactions on Evolutionary Computation

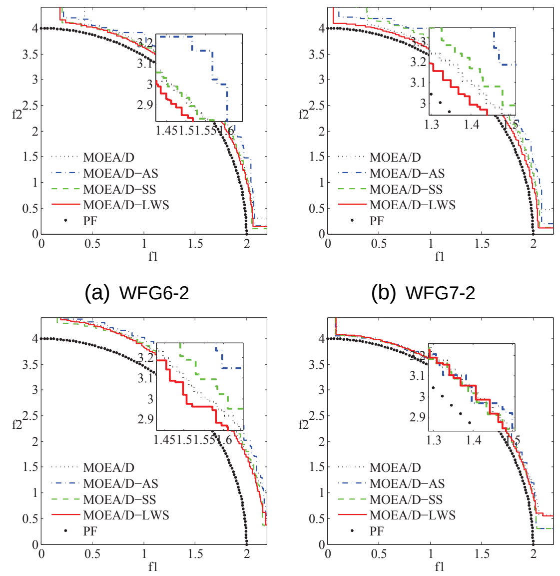
sum method. A large l creates difficulty on the convergence of an algorithm. Attributes of these problems include separability or nonseparability, unimodality or multimodality, unbiased or biased parameters, and convex or concave geometries. The *ideal* and *nadir* points for these problems are $[0, 0, \ldots, 0]$ and $[2, 4, \ldots, 2m]$, respectively. Hereafter, we use WFG*x*-m to denote the problem WFG*x* with m objectives.



(a) WFG2-2 (b) WFG3-2

(c) WFG4-2 (d) WFG5-2

Fig. 4. (Color online) Attainment surfaces for WFG2-2 to WFG5-2.



(a) WFG6-2 (b) WFG7-2

(c) WFG8-2 (d) WFG9-2

Fig. 5. (Color online) Attainment surfaces for WFG6-2 to WFG9-2.

B. Competitor algorithms and parameter settings

From the literature we select two effective methods by Ishibuchi et al. [45], [47] as the competitor algorithms.

- In [45], the weighted sum and the Chebyshev methods are adaptively chosen to guide the search. More specifically, the weighted sum is used as the scalarizing method unless a non-convex region is detected in which case the Chebyshev method is used. Note that when the weighted sum value of a solution is found to be better than all its neighbors for multiple weights, we say the solution is in a non-convex PF region. The neighborhood size is set to T, though it is a user defined value.
- In [47], the weighted sum and Chebyshev methods are simultaneously used to guide the search. The authors provide two implementations. One is to use both s-calarizing methods on each search direction. The other is to alternately assign a scalarizing method to each search direction. No additional parameter is required. The second implementation is used in this study since the first implementation doubles the population size which may result in an unfair comparison.

For clarity, MOEA/D using the *adaptive* strategy is denoted as MOEA/D-AS, and MOEA/D using the *simultaneous* strategy is denoted as MOEA/D-SS. The standard MOEA/D [75] using the Chebyshev method is used as a base line algorithm. Note that the aim of the comparison in this section is to examine the effect of using the weighted sum in MOEA/D. Thus, the proposed method is compared with other ideas of scalarization in MOEA/D. Comparison with other EMO algorithms will be shown later in Section VI.

For all test problems, each algorithm is run for 31 runs subjected to a statistical analysis, and each run for $maxGen = 250$ generations. The population size N is set to 100, 200 and 700 for two-, four- and seven-objective problems, respectively. The genetic operators and other parameters are listed in TABLE I, and are fixed across all algorithm runs.

TABLE I
SETTINGS OF GENETIC OPERATORS AND OTHER PARAMETER.

Parameters	Settings
Simulated Binary Crossover (SBX)	$p_c = 1, \eta = 30$
Polynomial Mutation (PM)	$p_m = 1/n, \eta = 20$
MOEA/D selection neighborhood size	$T =$ 10% of the N
MOEA/D mating restriction probability	δ = 0.8
MOEA/D replacement size	$nr =$ 10% of the T

C. Performance assessment

Algorithm performance is assessed by the median attainment surface (i.e., 50% attainment surface), the hypervolume (HV), the generational distance (GD) and the coverage of two sets (C metric) [79], [80]. The median attainment surface allows a visual inspection of both proximity and diversity performance. The HV metric measures the volume of the space enclosed by the Pareto approximation set and a given reference point. Both high proximity and large diversity are needed for obtaining a large HV value. The GD metric returns the average distance of the obtained solutions to their nearest neighbor on the PF. The smaller the GD the better the convergence. The C metric measures the proximity performance of one set over another. $C(A, B)$ refers to the fraction of solutions in set B that are dominated at least by one solution in set A. $C(A, B) > C(B, A)$ indicates a better convergence of set A. These performance metrics are calculated using all non-dominated solutions found during the search[4]. Prior to the calculation, all solutions are normalized by the *ideal* and *nadir* points. The reference point used for the HV calculation is set to $(1.1, 1.1, \ldots, 1.1)$.

V. Experimental Results

A. Median attainment surface results

The median attainment surfaces across the 31 runs of each algorithm are plotted in Fig. 4 and Fig. 5. The PF of each problem serves as a reference. It is observed from the results that MOEA/D-LWS appears to have comparable diversity performance as the other algorithms on all problems. However, its convergence performance is better than the others on all problems except for WFG2-2 and WFG9-2. It should be pointed out that no algorithm has completely converged to the PF. This is because all the WFG problems are configured with $n = 100$ decision variables among which 94 are distance variables ($l = 94$). As pointed out in [78], the increase of the number of WFG distance variables creates more difficulties for algorithms to converge to the PF. These results also suggest that large-scale global optimization [82] is challenging for multi-objective problems.

B. The HV and C metric results

Tables II and III show the comparison results in terms of the HV and C metrics, respectively. The non-parametric Wilcoxon-Ranksum two-sided method at the 95% confidence level is applied to test whether the results are statistically different.

From TABLE II the following results are observed.

- Comparing MOEA/D-LWS with MOEA/D, MOEA/D-LWS is better for all problems.
- Comparing MOEA/D-LWS with MOEA/D-AS, MOEA/D-LWS is better for 23 out of the 24 problems. On WFG2-2 the two algorithms perform comparably.
- Comparing MOEA/D-LWS with MOEA/D-SS, MOEA/D-LWS is also better for 22 out of the 24 problems. On WFG2-2 and WFG9-2, the two algorithms show comparable performance.

Convergence comparison results observed from TABLE III are as follows.

- Comparing MOEA/D-LWS with MOEA/D, MOEA/D-LWS is better for 22 out of the 24 problems. On WFG2-2 and WFG9-2, the two algorithms show comparable convergence performance;
- Comparing MOEA/D-LWS with MOEA/D-AS, MOEA/D-LWS is better for 20 out of the 24 problems. On WFG2-2, WFG9-2, WFG2-4 and WFG8-4 the two algorithms perform comparably.
- Comparing MOEA/D-LWS with MOEA/D-SS, MOEA/D-LWS is also better for 18 out of the 24 problems. For the three 2-objective problems (WFG2, WFG5 and WFG9) and the three 4-objective problems (WFG3, WFG8 and WFG9), the two algorithms are found to be comparable.

[4]Readers can refer to [81] for discussions about the choice of solution set for algorithm comparison.

Upon closer examination, the following results are observed.

- On WFG2-2 MOEA/D-LWS, MOEA/D-AS and MOEA/D-SS perform comparably. We find that during the search MOEA/D-AS applies the weighted Chebyshev method only for a few times at the early stage of the search. This indicates that MOEA/D-AS has successfully identified the PF shape of WFG2-2, and thus using the weighted sum method for most of time which leads to a comparable performance as MOEA/D-LWS. With respect to MOEA/D-SS, the weighted sum method is naturally embedded. Thus, its performance could be comparable with MOEA/D-LWS for WFG2-2. For the other comparisons on 2-objective problems (having non-convex PF shape), MOEA/D-LWS shows better performance than the other algorithms mainly because of the high search efficiency of LWS. In other words, on these problems MOEA/D-AS chooses to use the Chebyshev method for most of time. Regarding MOEA/D-SS, half of the weights takes no effect.
- As the number of objectives increases, the superiority of MOEA/D-LWS becomes more evident. Taking the 7-objective problems as an example, MOEA/D-LWS shows better convergence performance than any of the three algorithms for all problems. Its overall performance (measured by the HV metric) is also better than the other three algorithms for almost all problems (22 out of the 24 comparisons). The reason for the inferior performance of MOEA/D is that the algorithm applies the Chebyshev method during the whole search whose search efficiency is not as good as the weighted sum. The same reason, together with the incapability of the weighted sum to handle the non-convex region, apply to MOEA/D-AS and MOEA/D-SS. Noticeably, on WFG2-4 and WFG2-7 MOEA/D-AS is not comparable to MOEA/D-LWS. Possibly because that MOEA/D-AS cannot detect the concave region successfully with the chosen neighborhood size T.

To further compare the performance of MOEA/D-LWS and its competitor algorithms, the changes of HV and GD metrics over generations (till 1000 generations) are examined. Fig. 6 illustrates the results on WFG4-4 for instance. From the results we can observe that MOEA/D-LWS continuously exhibits better overall performance (measured by HV metric) and convergence performance (measured by GD metric) than the other three competitor MOEAs. Similar results are also obtained for most of other WFG problems.

In conclusion the localized weighted sum method is effective which makes MOEA/D-LWS perform better than the considered competitors. In particular the LWS enables MOEA/D-LWS to find solutions on both convex and non-convex PF regions. In addition the superiority of MOEA/D-LWS becomes more evident as the number of objectives increases. We can therefore make a strong claim to use MOEA/D-LWS for many-objective optimization.

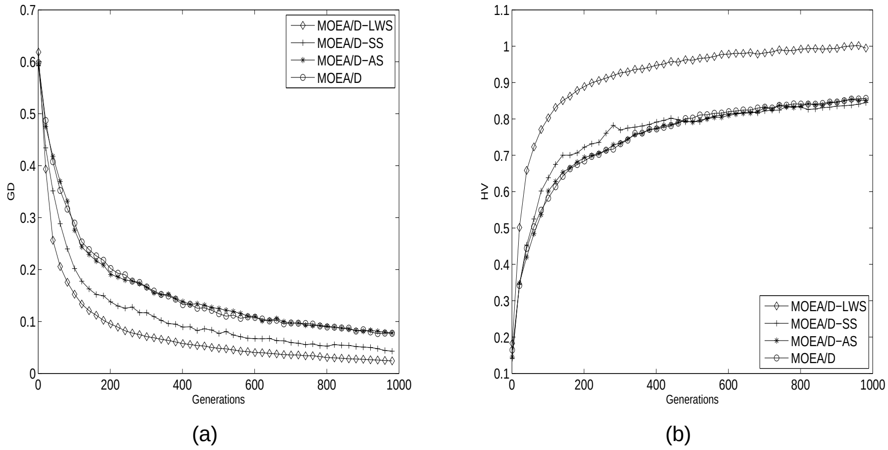


Fig. 6. The performance of algorithms over generations in terms of HV and GD metrics for WFG4-4 problems.

TABLE II
THE HV METRIC (MEAN/STD) COMPARISON RESULTS. THE SYMBOL '$-$', '$=$' OR '$+$' MEANS THE CONSIDERED ALGORITHM IS STATISTICALLY WORSE THAN, COMPARABLE TO OR BETTER THAN MOEA/D-LWS.

		MOEA/D-LWS	MOEA/D	MOEA/D-AS	MOEA/D-SS
$m=2$	WFG2	0.6098(0.0109)	0.5937(0.0218)$^-$	0.6088(0.0092)$^=$	0.6156(0.0099)$^=$
	WFG3	0.6026(0.0130)	0.5582(0.0165)$^-$	0.5642(0.0108)$^-$	0.5676(0.0121)$^-$
	WFG4	0.3726(0.0013)	0.3572(0.0079)$^-$	0.3323(0.0040)$^-$	0.3597(0.0040)$^-$
	WFG5	0.3081(0.0103)	0.3027(0.0040)$^-$	0.2550(0.0084)$^-$	0.2941(0.0033)$^-$
	WFG6	0.3600(0.0044)	0.3348(0.0056)$^-$	0.3237(0.0122)$^-$	0.3497(0.0083)$^-$
	WFG7	0.3767(0.0063)	0.3420(0.0034)$^-$	0.3002(0.0179)$^-$	0.3232(0.0090)$^-$
	WFG8	0.2822(0.0017)	0.2653(0.0069)$^-$	0.2379(0.0091)$^-$	0.2691(0.0048)$^-$
	WFG9	0.3759(0.0250)	0.3598(0.0014)$^-$	0.3669(0.0007)$^-$	0.3728(0.0071)$^=$
$m=4$	WFG2	1.2990(0.0083)	1.1748(0.0082)$^-$	1.2120(0.0107)$^-$	1.1828(0.0852)$^-$
	WFG3	1.0638(0.0166)	0.9622(0.0213)$^-$	0.9765(0.0183)$^-$	1.0130(0.0309)$^-$
	WFG4	0.9320(0.0099)	0.7791(0.0158)$^-$	0.7673(0.0288)$^-$	0.8342(0.0301)$^-$
	WFG5	0.8518(0.0070)	0.6790(0.0116)$^-$	0.6730(0.0156)$^-$	0.7722(0.0064)$^-$
	WFG6	0.9397(0.0086)	0.8411(0.0086)$^-$	0.8426(0.0131)$^-$	0.8969(0.0170)$^-$
	WFG7	1.0157(0.0040)	0.8853(0.0289)$^-$	0.8688(0.0261)$^-$	0.9415(0.0172)$^-$
	WFG8	0.7733(0.0121)	0.6667(0.0192)$^-$	0.6756(0.0068)$^-$	0.7128(0.0244)$^-$
	WFG9	0.8802(0.0294)	0.8097(0.0617)$^-$	0.8183(0.0239)$^-$	0.8449(0.0165)$^-$
$m=7$	WFG2	1.8152(0.0232)	1.7623(0.0240)$^-$	1.7455(0.0516)$^-$	1.7524(0.0233)$^-$
	WFG3	1.6772(0.0238)	1.6298(0.0315)$^-$	1.6104(0.0111)$^-$	1.5603(0.0219)$^-$
	WFG4	1.5916(0.0127)	1.2684(0.0605)$^-$	1.2761(0.0844)$^-$	1.2856(0.0751)$^-$
	WFG5	1.3588(0.0035)	1.1186(0.0441)$^-$	1.1166(0.0495)$^-$	1.1634(0.0561)$^-$
	WFG6	1.5901(0.0131)	1.2189(0.0998)$^-$	1.2012(0.1112)$^-$	1.2371(0.1159)$^-$
	WFG7	1.6898(0.0044)	1.4322(0.0696)$^-$	1.4081(0.0800)$^-$	1.4060(0.0853)$^-$
	WFG8	1.2724(0.0246)	0.9147(0.0718)$^-$	0.9538(0.1177)$^-$	0.8474(0.1014)$^-$
	WFG9	1.5244(0.0299)	1.0180(0.1637)$^-$	1.0972(0.0915)$^-$	1.1466(0.0884)$^-$
$\sharp. -/=/+$			24/0/0	23/1/0	22/1/0

VI. EXPERIMENT DISCUSSIONS

This section studies three further issues as part of a wider discussion for the effect of the localized weighted sum method: (i) uniformity performance of obtained solutions, (ii) comparison with other three evolutionary many-objective optimizers, and (iii) comparison with its variants.

A. Solutions obtained by MOEA/D-LWS with respect to uniformity

Experimental results have demonstrated the effectiveness of MOEA/D-LWS on dealing with many-objective problems as well as non-convex PFs. This section discusses the uniformity of solutions obtained by MOEA/D-LWS.

As mentioned earlier, each LWS attempts to find a Pareto optimal solution within its associated hypercone. Given a set of N hypercones, diversified solutions could be found by MOEA/D-LWS. However, it should be mentioned that the

TABLE III
THE C METRIC (MEAN/STD) COMPARISON RESULTS. THE SYMBOL ‘<’, ‘=’ OR ‘>’ MEANS THAT THE CONSIDERED ALGORITHM IS STATISTICALLY WORSE, COMPARABLE OR BETTER THAN MOEA/D-LWS. LWS REFERS TO MOEA/D-LWS, AS REFERS TO MOEA/D-AS, SS REFERS TO MOEA/D-SS.

	C(MOEA/D,LWS)		C(LWS,MOEA/D)	C(AS,LWS)		C(LWS,AS)	C(SS,LWS)		C(LWS,SS)
	2-objective problems								
WFG2	0.1564(0.3070)	<	0.7282(0.4289)	0.2485(0.3405)	=	0.6006(0.5135)	0.4165(0.3609)	=	0.3167(0.3980)
WFG3	0.2455(0.3364)	=	0.6020(0.5450)	0.0021(0.0048)	<	0.8092(0.1852)	0(0)	<	0.9493(0.0817)
WFG4	0.0537(0.0404)	<	0.6999(0.2846)	0.0022(0.0050)	<	0.9676(0.0245)	0.0768(0.0371)	<	0.7308(0.0648)
WFG5	0.2353(0.4523)	=	0.2234(0.4371)	0(0)	<	0.9841(0.0240)	0.2774(0.2515)	=	0.3117(0.4145)
WFG6	0.0266(0.0470)	<	0.8663(0.1646)	0(0)	<	0.9487(0.0477)	0.0953(0.0687)	<	0.6539(0.1817)
WFG7	0.0292(0.0653)	<	0.8840(0.2324)	0(0)	<	0.9601(0.0464)	0(0)	<	0.9468(0.0817)
WFG8	0.0360(0.0478)	<	0.7931(0.2188)	0.0148(0.0331)	<	0.8543(0.1104)	0.1179(0.1474)	<	0.7342(0.2213)
WFG9	0.2374(0.4099)	<	0.6566(0.4474)	0.3663(0.3657)	=	0.4268(0.4278)	0.4285(0.4829)	=	0.3326(0.4095)
	4-objective problems								
WFG2	0(0)	<	0.3201(0.2905)	0.0050(0.0082)	=	0.1074(0.1553)	0.0037(0.0084)	<	0.2171(0.1064)
WFG3	0.0043(0.0096)	<	0.1558(0.1074)	0(0)	<	0.1024(0.0774)	0.0100(0.0096)	=	0.0582(0.0916)
WFG4	0.0012(0.0027)	<	0.3478(0.0643)	0.0012(0.0027)	<	0.4647(0.1095)	0.0073(0.0051)	<	0.1899(0.0871)
WFG5	0(0)	<	0.3321(0.0774)	0(0)	<	0.3553(0.1023)	0.0011(0.0026)	<	0.1334(0.0303)
WFG6	0.0063(0.0109)	<	0.2745(0.1046)	0.0091(0.0057)	<	0.1826(0.0627)	0.0397(0.0195)	=	0.0194(0.0435)
WFG7	0.0013(0.0029)	<	0.2585(0.0790)	0(0)	<	0.3640(0.1674)	0.0075(0.0053)	<	0.1109(0.0499)
WFG8	0.0065(0.0066)	<	0.1079(0.0361)	0.0146(0.0088)	=	0.1029(0.0731)	0.0372(0.0116)	=	0.0338(0.0407)
WFG9	0.0137(0.0189)	<	0.3154(0.2562)	0.0197(0.0193)	<	0.0740(0.0763)	0.0164(0.0102)	=	0.0141(0.0154)
	7-objective problems								
WFG2	0(0)	<	0.0805(0.0853)	0.0007(0.0022)	<	0.0740(0.0436)	0.0009(0.0029)	<	0.1466(0.1366)
WFG3	0.0026(0.0016)	<	0.0232(0.0350)	0.0020(0.0045)	<	0.0476(0.0423)	0(0)	<	0.1984(0.0602)
WFG4	0.0043(0.0018)	<	0.1010(0.0646)	0.0043(0.0027)	<	0.1019(0.0473)	0.0057(0.0018)	<	0.0544(0.0337)
WFG5	0.0006(0.0010)	<	0.0584(0.0300)	0.0002(0.0006)	<	0.0397(0.0159)	0.0023(0.0024)	<	0.0480(0.0146)
WFG6	0.0027(0.0030)	<	0.1225(0.0273)	0.0025(0.0109)	<	0.1372(0.0556)	0.0054(0.0055)	<	0.1492(0.0928)
WFG7	0.0002(0.0007)	<	0.1155(0.0284)	0.0006(0.0029)	<	0.1206(0.0264)	0.0009(0.0012)	<	0.1723(0.0542)
WFG8	0.0150(0.0056)	<	0.0754(0.0413)	0.0071(0.0066)	<	0.0477(0.0365)	0.0157(0.0048)	<	0.0920(0.0472)
WFG9	0(0)	<	0.3214(0.1718)	0(0)	<	0.1840(0.1306)	0.0009(0.0025)	<	0.1854(0.1868)
	♯ </=/>	22/2/0		♯ </=/>	20/4/0		♯ </=/>	18/6/0	

uniformity of obtained solutions might not be as good as solutions obtained by MOEA/D with the Chebyshev method, in particular, for problems with non-convex PFs.

Fig. 8 illustrates two sets of solutions that are possibly obtained by MOEA/D-LWS using the same weights (i.e., $\mathbf{w}^1$,$\mathbf{w}^2$ and $\mathbf{w}^3$). The solutions obtained by MOEA/D with Chebyshev method severe as references. It shows that in the ideal case solutions obtained by the use of Chebyshev method (black circles) are constantly distributed along each search direction. However, for the LWS method the obtained solutions (black points) might be distributed differently, see the difference between Fig. 8(a) and Fig. 8(b). This is because that for a local non-convex PF there are more than one boundary solutions could be seen as the candidate optimal solution. The obtained solutions might not consistently sit at the same side of the weight. In other words, the uniformity of obtained solutions is not guaranteed. However, given that a number of weights are employed in MOEA/D-LWS, and each LWS is associated with a different hypercone, the diversity performance of MOEA/D-LWS would not be much deteriorated because of this limitation. As an illustration, solutions obtained by MOEA/D-LWS for 3-objective WFG problems are plotted (see the supplementary file) from which we can observe that MOEA/D-LWS achieves a relatively good diversity performance, though solutions are not uniformly distributed.

In addition, for a convex PF the LWS can maintain a good uniformity performance. As an illustration, Fig. 7 shows the performance of MOEA/D-LWS on a convex MOP, namely, ZDT1. By comparing against MOEA/D with the Chebyshev method, we can observe that MOEA/D-LWS achieves a good uniformity performance. The reason is that, as mentioned earlier, for a convex PF, the obtained solution have more chances to be along (or near) the search direction.

However, it is worth mentioning that a perfect uniformity as the Chebyshev method offers is still not guaranteed. This is because that the optimal solution for each LWS method is not always along the search direction exactly due to the contour lines of the LWS as well as the PF shape.

Overall, though the LWS method is not as good as the Chebyshev method with respect to the uniformity of obtained solutions, it is much better than the weighted sum (WS) method, being less affected by the PF shape (in particular, being able to find diversified solutions even for non-PFs). Also, the number of obtained solutions increases as the number of employed weights increases. Moreover, the LWS is clearly better than the Chebyshev method in terms of convergence property (as shown in our previous experiments).

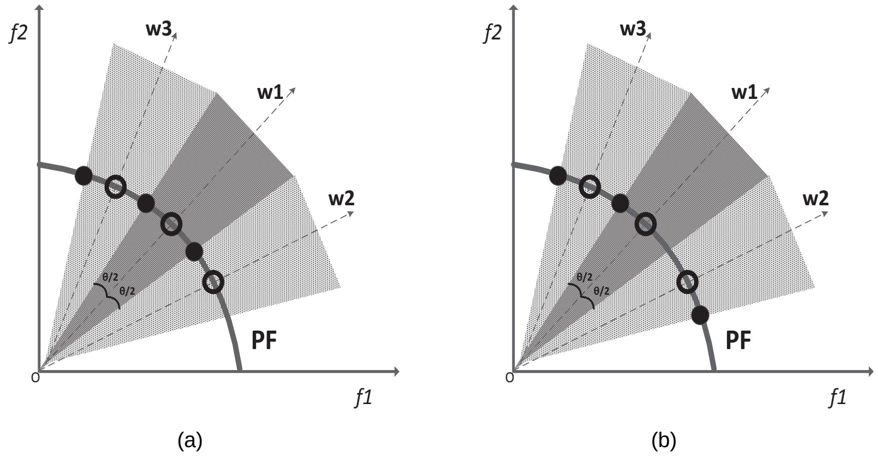


Fig. 8. Illustration of uniformity performance of MOEA/D-LWS. Solutions (black points) are obtained by the LWS method; solutions (black circles) are obtained by the Chebyshev method

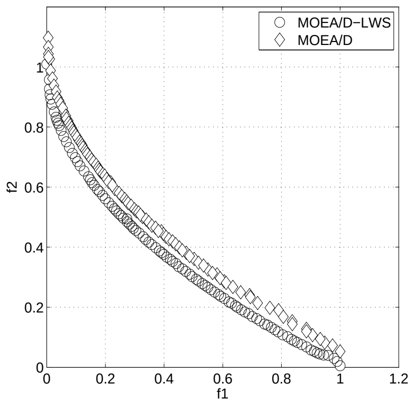


Fig. 7. Illustration of the performance of MOEA/D-LWS and MOEA/D with the Chebyshev method on ZDT1 with a convex PF.

B. Comparison with PICEA-g, HypE, θ-DEA and SPEA2+SDE

This section further investigates the performance of MOEA/D-LWS on many-objective problems by comparing it against four state-of-the-art many-objective algorithms (representing different classes), PICEA-g [11], HypE [10], θ-DEA [21] and SPEA2+SDE [7].

- PICEA-g is one realization of the preference-inspired co-evolutionary algorithms (PICEAs). The ingenuity in PICEAs lies in the simple recognition that decision-maker preferences bring additional comparability between solutions that is otherwise missing from the standard Pareto-dominance relation used in traditional EMO algorithms. What follows from this realisation is the hypothesis that, by maintaining a family of preferences during search, then there may be sufficient comparability to find the Pareto optimal set. The working principle of PICEA is, in essence, a generalization of the known decomposition approaches. PICEA generalizes the decomposition concept to any type of preference information (e.g. goals and weights) and also provides a means of adapting the set of preferences (i.e., co-evolution) to drive the search effectively in the absence of a known PF shape. In PICEA-g goal vectors are taken as preferences. Candidate solutions gain high fitness by dominating as many goals in objective space as possible. Goal vectors only gain fitness by being dominated by a candidate solution, however, the fitness is reduced if other solutions also dominate the same goal vector. In this study, the number of goal vectors N_{goal} is set as identical to the population size N.
- HypE is an effective performance indicator based evolutionary algorithm (IBEA). The IBEA transforms the optimization of an MOP into the optimization of an indicator. In HypE the HV is used as the indicator. What makes the HypE different from other HV based algorithms is that it employs a Monte Carlo method to approximate the HV value. This effectively reduces the computational effort required for the exact HV calculation [83]. In this study the exact HV is used for 2-objective problems. For 4- and 7-objective problems, the Monte Carlo method with $N_{sp} = 2N \times m$ sampling points in objective space is applied to approximate the HV value.
- θ-DEA is a very recently proposed many-objective optimizer. The algorithm is implemented within a $(\mu + \lambda)$ elitist framework, and uses a new dominance relation, namely, θ-dominance to rank solutions in the environmental selection phase. The θ-dominance, adopting the same expression of PBI scalarizing function, can effectively balance both convergence and diversity when a suitable θ is specified. In θ-DEA first the non-dominated sorting based on the Pareto-dominance is applied to sort solutions in different front levels. Then, similar to the crowding distance used in NGSA-II, the θ-dominance is further applied to select solutions. Before applying the θ-dominance, the clustering operation is conducted, i.e., each solution is assigned to its closest reference direction. The reference directions are evenly distributed. In each cluster, the θ optimal solution is selected. θ-DEA is reported to perform well on the DTLZ [84] and WFG benchmarks. Also, it is compared favourably with state-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2611642, IEEE Transactions on Evolutionary Computation

of-the-art many-objective optimizers such as NSGA-III and GrEA. In this study the same set of weights as MOEA/D-LWS is used in θ-DEA, and parameter $\theta = 5$ [21] is adopted.

- SPEA2+SDE applies a shift-based density estimation (SDE) strategy to the well-known SPEA2 [85]. The SDE strategy concerns both the distribution and convergence information of individuals. That is, it shifts the position of solutions based on their convergence information when estimating their densities. Numerical studies have demonstrated the efficiency of the SDE strategy. It can significantly improve performance of Pareto-dominance based EMO algorithms for many-objective problems. It represents an important class of many-objective optimizers, i.e., modified density based class.

In addition, in order make a fair comparison the same mating restriction strategy (Algorithm 1 lines 9-13) as MOEA/D-LWS is applied to all the three algorithms.

The HV comparison results of the three algorithms are shown in TABLE IV. To avoid repetitions HV results of MOEA/D-LWS are not shown. The same parameter settings in TABLE I are adopted. From TABLE IV it is observed that MOEA/D-LWS is not as good as HypE for 2-objective problems. However, its performance is better than HypE for all the 4- and 7-objective problems. Compared with PICEA-g, MOEA/D-LWS is better for most of the test problems. With respect to θ-DEA, MOEA/D-LWS is better on two 2-objective problems, and all the 4- and 7-objective problems. Compared with SPEA2+SDE, though MOEA/D-LWS is inferior for all 2-objective problems it exhibits better performance for five 4-objective problems and seven 7-objective problems. Such results suggest that MOEA/D-LWS is a very competitive many-objective optimizer. In addition, PICEA-g is inferior to HypE and θ-DEA for 2-objective problems. However, as the number of objectives increases, PICEA-g appears to be superior to HypE and θ-DEA.

Whereas the four competitor algorithms are found as inferior to MOEA/D-LWS for many-objective problems in this experiment, it does not mean that these algorithms are ineffective. As is known the performance of an algorithm is often impacted by its associated parameter settings. We have found that using fine-tuned parameter settings, PICEA-g and HypE could have comparable performance with MOEA/D-LWS. For example, by setting a larger N_{goal} (e.g., 4000) and a larger N_{sp} (e.g., 20,000) the three algorithms perform comparably on WFG4-4. Certainly, this is computationally more expensive. Whereas, θ-DEA is reported to perform well on WFG problems (with $n = 24$ decision variables) [21], good results are not obtained here. The reason might be that in this experiment the WFG problems are set with $n = 100$ decision variables which creates difficulty for the θ-dominance. Actually, we performed computational experiments by specifying the number of decision variables as $n = 24$, which is the same setting as in [21], In this case, comparable results are obtained from θ-DEA and MOEA/D-LWS. Besides, according to the no-free-lunch theory [86], none of algorithms can perform well on all types of problems [21], [87]. There must be some problems that θ-DEA works the best [21]. Therefore, identifying suitable algorithms for different types of problems deserves more studies.

C. MOEA/D-LWS versus its variant

MOEA/D-LWS allows multiple selections of a solution, see lines 23 to 27. This might decrease the algorithm performance, e.g., lack of solution diversity. In order to study this, the performance of its variants, denoted as MOEA/D-LWSnr, is examined. In MOEA/D-LWSnr a solution is selected to survive for maximally nr times. Specifically, lines 23 to 27 in MOEA/D-LWS are replaced by Algorithm 2.

Algorithm 2: Replacement procedure

```
S ← ∅;
J ← {1, 2, 3, ..., N};
Shuffle J randomly;
Count[i] ← 0, i = 1, 2, ..., N;
foreach j ∈ J do
    Remove j from J;
    i ← arg min_{i=1,2,...,2N} C_ij;
    Count[i] ← Count[i] + 1;
    temp ← Count[i];
    if temp ≤ nr then
        S ← S ∪ x^i;
    else
        while temp > nr do
            set C_{ij, j=1,2,...,N} to ∞;
            i' ← arg min_{j=1 to 2N} C_ij;
            Count[i'] ← Count[i'] + 1;
            temp ← Count[i'];
        end
        S ← S ∪ x^{i'};
    end
end
```

The variants of MOEA/D-LWS, i.e., MOEA/D-LWS1 ($nr = 1$) and MOEA/D-LWS2 ($nr = 2$) are examined for the same test problems as in our previous computational experiments. The HV results are shown in TABLE V. It is found that none of algorithms can beat its competitors for most of the problems. Specifically, for 2-objective problems MOEA/D-LWS1 performs better than MOEA/D-LWS for three problems, and performs worse than MOEA/D-LWS for only one problem. MOEA/D-LWS2 performs better than MOEA/D-LWS for four problems, and performs worse than MOEA/D-LWS for two problem. For 4-objective problems MOEA/D-LWS and its variants perform comparably for a majority of problems. This is demonstrated by the fact that the number of non-dominated solutions (without counting the repeated ones) in the obtained Pareto approximation set by the three algorithms is comparable for most of test problems. For example, the number of non-dominated solutions in the final generation for WFG4-4 obtained by MOEA/D-LWS, MOEA/D-LWS1 and MOEA/D-LWS2 are 168, 179 and 172 (averaged over 31 runs), respectively. For 7-objective problems, MOEA/D-LWS performs better than its variants.

TABLE IV
THE MEAN HV RESULTS OF PICEA-G, HYPE, θ-DEA AND SPEA2+SDE. THE SYMBOL '$-$', '$=$' OR '$+$' MEANS THE CONSIDERED ALGORITHM IS STATISTICALLY WORSE THAN, COMPARABLE TO OR BETTER THAN MOEA/D-LWS.

	$M=2$				$M=4$				$M=7$			
	PICEA-g	HypE	θ-DEA	SDE	PICEA-g	HypE	θ-DEA	SDE	PICEA-g	HypE	θ-DEA	SDE
WFG2	0.5964$^-$	0.6327$^+$	0.6281$^+$	0.6859$^+$	1.2796$^-$	1.2554$^-$	1.1955$^-$	1.2936$^-$	1.8345$^+$	1.8006$^-$	1.6166$^-$	1.8697$^+$
WFG3	0.5708$^-$	0.6153$^+$	0.5988$^=$	0.6376$^+$	1.0458$^-$	1.0446$^-$	0.8079$^-$	1.0890$^+$	1.6676$^=$	1.6209$^-$	0.9540$^-$	1.1236$^-$
WFG4	0.3290$^-$	0.3759$^+$	0.3650$^-$	0.3978$^+$	0.8097$^-$	0.8041$^-$	0.7307$^-$	0.8825$^-$	1.2729$^-$	1.1899$^-$	0.9080$^-$	1.3343$^-$
WFG5	0.3058$^-$	0.3354$^+$	0.3269$^+$	0.3666$^+$	0.7625$^-$	0.7456$^-$	0.6495$^-$	0.8229$^-$	1.1951$^-$	1.0766$^-$	0.7380$^-$	1.2972$^-$
WFG6	0.3223$^-$	0.3532$^-$	0.3473$^-$	0.3885$^+$	0.7904$^-$	0.7622$^-$	0.6676$^-$	0.8844$^-$	1.2519$^-$	1.0863$^-$	0.7197$^-$	1.3410$^-$
WFG7	0.3418$^-$	0.3871$^+$	0.3769$^=$	0.4104$^+$	0.9079$^-$	0.9015$^-$	0.8082$^-$	0.9775$^-$	1.5696$^-$	1.4849$^-$	1.0797$^-$	1.4867$^-$
WFG8	0.2487$^-$	0.2761$^+$	0.2808$^=$	0.3486$^+$	0.6866$^-$	0.6963$^-$	0.6575$^-$	0.7642$^-$	1.1001$^-$	1.0099$^-$	0.7940$^-$	1.2631$^-$
WFG9	0.3177$^-$	0.3631$^=$	0.3523$^-$	0.3923$^+$	0.7975$^-$	0.8047$^-$	0.7042$^-$	0.8646$^=$	1.3760$^-$	1.3153$^-$	0.8568$^-$	1.3158$^-$
	no. of –/=/+				no. of –/=/+				no. of –/=/+			
	8/0/0	1/1/6	2/3/2	0/0/8	8/0/0	8/0/0	8/0/0	5/2/1	6/1/1	8/0/0	8/0/0	7/0/1

TABLE V
THE HV RESULTS (MEAN/STD) OF MOEA/D-LWS1 ($nr = 1$) AND MOEA/D-LWS2 ($nr = 2$). THE SYMBOL '$-$', '$=$' OR '$+$' MEANS THE CONSIDERED ALGORITHM IS STATISTICALLY WORSE THAN, COMPARABLE TO OR BETTER THAN MOEA/D-LWS.

	$nr=1$	$nr=2$		$nr=1$	$nr=2$		$nr=1$	$nr=2$
WFG2-2	0.6308(0.0060)$^+$	0.6223(0.0115)$^+$	WFG2-4	1.2988(0.0089)$^=$	1.2991(0.0071)$^=$	WFG2-7	1.8624(0.0082)$^+$	1.8779(0.0040)$^+$
WFG3-2	0.6131(0.0006)$^+$	0.5920(0.0234)$^=$	WFG3-4	1.1030(0.0117)$^+$	1.0841(0.0157)$^+$	WFG3-7	1.7562(0.0132)$^+$	1.7621(0.0138)$^+$
WFG4-2	0.3720(0.0041)$^=$	0.3696(0.0031)$^-$	WFG4-4	0.9303(0.0110)$^=$	0.9353(0.0084)$^=$	WFG4-7	1.5207(0.0176)$^-$	1.5960(0.0162)$^=$
WFG5-2	0.3288(0.0046)$^+$	0.3075(0.0037)$^=$	WFG5-4	0.8538(0.0051)$^-$	0.8500(0.0088)$^-$	WFG5-7	1.3204(0.0109)$^-$	1.3451(0.0072)$^-$
WFG6-2	0.3552(0.0055)$^-$	0.3632(0.0020)$^+$	WFG6-4	0.8962(0.0124)$^=$	0.9271(0.0131)$^=$	WFG6-7	1.4533(0.0263)$^-$	1.5582(0.0150)$^-$
WFG7-2	0.3765(0.0110)$^=$	0.3664(0.0057)$^-$	WFG7-4	1.0033(0.0056)$^-$	1.0143(0.0029)$^=$	WFG7-7	1.6824(0.0050)$^-$	1.6821(0.0060)$^-$
WFG8-2	0.2880(0.0087)$^=$	0.2850(0.0046)$^+$	WFG8-4	0.7789(0.0106)$^=$	0.7837(0.0094)$^+$	WFG8-7	1.2245(0.0148)$^-$	1.2502(0.0145)$^-$
WFG9-2	0.3810(0.0133)$^=$	0.3687(0.0026)$^+$	WFG9-4	0.8729(0.0333)$^=$	0.8676(0.0295)$^=$	WFG9-7	1.4383(0.0206)$^-$	1.5154(0.0233)$^=$
♯ –/=/+	1/4/3	2/2/4		2/5/1	1/5/2		6/0/2	4/2/2

Overall restricting the number of replacements is helpful in maintaining diversity of solutions. This leads to superior performance of MOEA/D-LWS1 and MOEA/D-LWS2 for 2-objective problems. However, convergence is often considered as more challenging than diversity for many-objective optimization [4]. As the number of objectives increases the superiority of MOEA/D-LWS1 and MOEA/D-LWS2 diminishes. In another aspect, this diversity promotion strategy leads to a detrimental effect on the convergence performance. Therefore, for many-objective problems MOEA/D-LWS appears to be better than its variants.

VII. CONCLUSION

Decomposition based EMO algorithms have been repeated demonstrated as effective for addressing multi- and many-objective problems. Different from the Pareto-dominance relation, the use of weighted scalarizing methods brings additional comparability between candidate solutions, and therefore can guide candidate solutions towards the Pareto optimal front effectively. It is observed that different scalarizing methods exhibit different search efficiency. Despite the incapability of handling non-convex regions, the weighted sum method has the best search efficiency amongst all L_p scalarizing methods. This study proposes a localized weighted sum method by which the drawback of the weighted sum method is overcame while its high search efficiency is retained. Based on the localized weighted sum method, a novel decomposition based algorithm, MOEA/D-LWS is proposed. It is compared against three related methods, i.e., MOEA/D, MOEA/D with an adaptive use of the weighted sum and Chebyshev methods and MOEA/D with their simultaneous use for the well-known WFG benchmarks with upto seven objectives. Experimental results show that MOEA/D-LWS is much better than the competitor algorithms. Moreover, MOEA/D-LWS is compared against three many-objective optimizers: PICEA-g, HypE and θ-DEA, and is shown as competitive.

Regarding future studies, first we would like to assess MOEA/D-LWS on other problem types e.g. multi-objective combinatorial problems, some real-world problems, e.g., the size of renewable energy systems [88], [89], project scheduling [90], [91], [92]. Second, we would like to extend MOEA/D-LWS to a hybridized evolutionary multi-criteria decision making approach so as to assist the decision maker to find his/her preferred solutions [12], [93]. Third, as shown in the experiments, the algorithm performance often degrades as the number of decision variables increases. It is therefore worthwhile to investigate the scalability of MOEA/D-LWS for large-scale problems [94]. Fourth, a steady-state version of MOEA/D-LWS could be designed given to the merits of steady-state selection scheme. Fifthly, in MOEA/D-LWS some isolated inferior (or dominated) solutions might survive during the evolution in particular when the PF is disconnected or has a large hole. These isolated solutions might have a negative effect in terms of convergence. However, they are, to some extent good to diversity. How to effectively handle these solutions deserves further studies, e.g., using adaptive hypercone regions, adaptive weights. Lastly, decomposition-based EMO algorithms sometime struggle in maintain the uniformity of the solutions, a combination of Pareto based and decomposition based methods has shown promise for handling this issue, and thus requires further investigation [18], [95].

Source code of MOEA/D-LWS is available at http://ruiwangnudt.gotoip3.com/optimization.html.

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

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

[4] R. C. Purshouse and P. J. Fleming, "On the Evolutionary Optimization of Many Conflicting Objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 770–784, 2007.

[5] R. Wang, P. Fleming, and R. Purshouse, "General framework for localised multi-objective evolutionary algorithms," *Information Sciences*, vol. 258, no. 2, pp. 29–53, 2014.

[6] K. Deb, M. Mohan, and S. Mishra, "Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, pp. 501–25, 2005.

[7] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[8] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[9] E. Zitzler and S. Künzli, "Indicator-Based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervós, J. Bullinaria, J. Rowe, P. Tiño, A. Kabán, and H.-P. Schwefel, Eds., vol. 3242. Springer Berlin Heidelberg, 2004, pp. 832–842.

[10] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, pp. 45–76, 2011.

[11] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired Co-evolutionary Algorithms for Many-objective Optimisation," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.

[12] R. Wang, R. C. Purshouse, I. Giagkiozis, and P. J. Fleming, "The iPICEA-g: a new hybrid evolutionary multi-criteria decision making approach using the brushing technique," *European Journal of Operational Research*, vol. 243, no. 2, pp. 442–453, 2015.

[13] R. Wang, M. M. Mansor, R. C. Purshouse, and P. J. Fleming, "An analysis of parameter sensitivities of preference-inspired co-evolutionary algorithms," *International Journal of Systems Science*, vol. 46, no. 13, pp. 423–441, 2015.

[14] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-Inspired Co-Evolutionary Algorithm Using Weights for Many-objective Optimisation," in *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 101–102.

[15] ——, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.

[16] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, Eds. Springer Berlin Heidelberg, 2001, pp. 82–95.

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

[18] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[19] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 180–198, 2016.

[20] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[21] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[22] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artificial Intelligence*, vol. 228, pp. 45–65, 2015.

[23] X. Zhang, Y. Tian, and Y. Jin, "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[24] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[25] M. Asafuddoula, T. Ray, and R. Sarker, "A Decomposition Based Evolutionary Algorithm for Many Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, pp. 445–460, 2014.

[26] J. Cheng, G. G. Yen, and G. Zhang, "A many-objective evolutionary algorithm with enhanced mating and environmental selections," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 1–1, 2015.

[27] Z. He and G. Yen, "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement," vol. 20, no. 1, pp. 145–160, 2016.

[28] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," 2016.

[29] C. V. Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization & Applications*, vol. 58, no. 3, pp. 707–756, 2014.

[30] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *Acm Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[31] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[32] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999.

[33] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 428–442.

[34] H. Ishibuchi, N. Akedo, and 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.

[35] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[36] R. Wang, Q. Zhang, and T. Zhang, "Decomposition based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. To appear, 2016.

[37] B. Derbel, D. Brockhoff, and A. Liefooghe, "Force-Based Cooperative Search Directions in Evolutionary Multi-objective Optimization," in *Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 383–397.

[38] 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*. IEEE, 2011, pp. 1260–1264.

[39] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[40] F. Gu, H. Liu, and K. Tan, "A Multiobjective Evolutionary Algorithm using Dynamic Weight Design Method," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 5B, pp. 3677–3688, 2012.

[41] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with Adaptive Weight Adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2013.

[42] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[43] M. P. Hansen, "Use of substitute scalarizing functions to guide a local search based heuristic: The case of motsp," *Journal of Heuristics*, vol. 6, no. 3, pp. 419–431, 2000.

[44] P. C. Borges and M. P. Hansen, "A study of global convexity for a multiple objective travelling salesman problem," in *Essays and surveys in metaheuristics*. Springer, 2002, pp. 129–150.

[45] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and 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.

[46] R. Wang, Q. Zhang, and T. Zhang, "Pareto adaptive scalarising function for decomposition based algorithms," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 9018. Springer Berlin Heidelberg, 2015, pp. 248–262.

[47] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *GECCO 2010: Proceedings of the Genetic and Evolutionary Computation Conference*. Portland,USA: ACM, 2010, pp. 519–526.

[48] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[49] K. Li, Q. Zhang, S. Kwong, L. M., and W. R., "Stable matching based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2015.

[50] 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.

[51] R. E. Steuer, *Multiple criteria optimization: theory, computation, and applications*. Wiley, 1986.

[52] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[53] L. Zadeh, "Optimality and non-scalar-valued performance criteria," *Automatic Control, IEEE Transactions on*, vol. 8, no. 1, pp. 59–60, 1963.

[54] J. Koski and R. Silvennoinen, "Norm methods and partial weighting in multicriterion optimization of structures," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 6, pp. 1101–1121, 1987.

[55] K. Proos, G. Steven, O. Querin, and Y. Xie, "Multicriterion evolutionary structural optimization using the weighting and the global criterion methods," *AIAA journal*, vol. 39, no. 10, pp. 2006–2012, 2001.

[56] P. Hajela, E. Lee, and C. Lin, "Genetic algorithms in structural topology optimization," *Topology design of structures*, vol. 227, pp. 117–134, 1993.

[57] E. Hughes, "Multiple single objective pareto sampling," in *Evolutionary Computation (CEC), 2003 IEEE Congress on*. Piscataway, NY, USA: IEEE Press, 2003, pp. 2678–2684.

[58] P. Serafini, "Simulated annealing for multi objective optimization problems," in *10th Int. Conf. on MCDM Proc*, vol. 1, Taipei, Taiwan, 19-24 July 1992, pp. 87–96.

[59] P. Czyzżak and A. Jaszkiewicz, "Pareto simulated annealingâĂŤa metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.

[60] X. Gandibleux and A. Freville, "Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: the two objectives case," *Journal of Heuristics*, vol. 6, no. 3, pp. 361–383, 2000.

[61] L. Paquete and T. Stützle, "A two-phase local search for the biobjective traveling salesman problem," in *Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 479–493.

[62] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.

[63] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.

[64] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 204–223, 2003.

[65] J. Koski, "Defectiveness of weighting method in multicriterion optimization of structures," *Communications in applied numerical methods*, vol. 1, no. 6, pp. 333–337, 1985.

[66] A. Messac, C. Puemi-Sukam, and E. Melachrinoudis, "Aggregate objective functions and pareto frontiers: required relationships and practical implications," *Optimization and Engineering*, vol. 1, no. 2, pp. 171–188, 2000.

[67] Y. Jin, T. Okabe, and B. Sendho, "Adapting weighted aggregation for multiobjective evolution strategies," in *Evolutionary multi-criterion optimization*. Springer, 2001, pp. 96–110.

[68] Y. Jin, T. Okabe, and B. Sendhoff, "Solving three-objective optimization problems using evolutionary dynamic weighted aggregation: results and analysis," in *Genetic and Evolutionary ComputationâĂŤGECCO 2003*. Springer, 2003, pp. 636–637.

[69] I. Y. Kim and O. De Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and multidisciplinary optimization*, vol. 29, no. 2, pp. 149–158, 2005.

[70] ——, "Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 31, no. 2, pp. 105–116, 2006.

[71] K.-S. Zhang, Z.-H. Han, W.-J. Li, and W.-P. Song, "Bilevel adaptive weighted sum method for multidisciplinary multi-objective optimization," *AIAA journal*, vol. 46, no. 10, pp. 2611–2622, 2008.

[72] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Evolutionary Computation (CEC), 2008 IEEE Congress on*. Piscataway, NY, USA: IEEE Press, 2008, pp. 2419–2426.

[73] I. Giagkiozis and P. Fleming, "Methods for multi-objective optimization: An analysis," *Information Sciences*, vol. 293, no. 0, pp. 338–350, 2015.

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

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

[76] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in decomposition based multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 425–430, 2016.

[77] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2016, To appear.

[78] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 477–506, 2006.

[79] R. Wang, R. C. Purshouse, and P. J. Fleming, "On Finding Well-spread Pareto Optimal Solutions by Preference-inspired Co-evolutionary Algorithm," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 695–702.

[80] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[81] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "How to compare many-objective algorithms under different settings of population and archive sizes," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 1149–1156.

[82] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 378–224, 2014.

[83] N. Beume, C. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.

[84] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Evolutionary Computation (CEC), 2002 IEEE Congress on*. IEEE, 2002, pp. 825–830.

[85] 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.

[86] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.

[87] M. Li, S. Yang, X. Liu, and R. Shen, “A comparative study on evolutionary algorithms for many-objective optimization,” in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 7811. Springer Berlin Heidelberg, 2013, pp. 261–275.

[88] T. Zhang, R. Wang, Y. Liu, and B. Guo, “An enhanced preference-inspired co-evolutionary algorithm using orthogonal design and an ϵ-dominance archiving strategy,” *Engineering Optimization*, no. ahead-of-print, pp. 1–22, 2015.

[89] Z. Shi, R. Wang, and T. Zhang, “Multi-objective optimal design of hybrid renewable energy systems using preference-inspired coevolutionary approach,” *Solar Energy*, vol. 118, pp. 96–106, 2015.

[90] G. Wu, M. Ma, J. Zhu, and D. Qiu, “Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks,” *Systems Engineering and Electronics, Journal of*, vol. 23, no. 5, pp. 723–733, 2012.

[91] J. Xiong, L. Xing, and Y. Chen, “Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns,” *International Journal of Production Economics*, vol. 141, pp. 112–126, 2013.

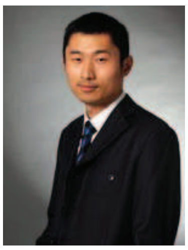
[92] J. Xiong, J. Liu, Y. Chen, and H. A. Abbass, “A knowledge-based evolutionary multi-objective approach for stochastic extended resource investment project scheduling problems,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 742–763, 2014.

[93] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, “A review of hybrid evolutionary multiple criteria decision making methods,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1147–1154.

[94] X. Li and X. Yao, “Cooperatively coevolving particle swarms for large scale optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 2, pp. 210–224, 2012.

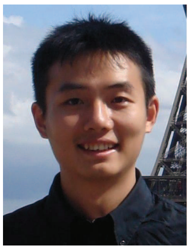
[95] M. Li, S. Yang, and X. Liu, “Pareto or Non-Pareto: Bi-Criterion Evolution in Multi-Objective Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. To appear, 2015.

Rui Wang received his B.S. degree from the National University of Defense Technology (NUDT), P.R. China, in 2008, and the Ph.D. degree from the University of Sheffield, U.K in 2013. He is currently a lecturer at the NUDT. His main research areas are evolutionary computation, multi-objective optimization, machine learning, optimization methods on energy internet network. He wins the Operational Research Society PhD prize Runners-up for the best PhD dissertation 2016.



Zhongbao Zhou received the B.S. from Department of System Engineering and Mathematics at National University of Defense Technology and the M.S. and Ph.D. from School of Information System and Management at National University of Defense Technology. His research interests include system optimization and decision, productivity and efficiency measurement, financial engineering and risk management, reliability engineering and quality management.

Hisao Ishibuchi (M’93âĂŞSM’10âĂŞF’14) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Japan, in 1992. Since 1987, he has been with Osaka Prefecture University, where he is currently a Professor with the Department of Computer Science and Intelligent Systems. His current research interests include fuzzy rule-based classifiers, evolutionary multiobjective optimization, memetic algorithms, and evolutionary games. Dr. Ishibuchi was the IEEE Computational Intelligence Society (CIS) Vice-President for Technical Activities from 2010 to 2013. He is an AdCom Member of the IEEE CIS from 2014 to 2016, an IEEE CIS Distinguished Lecturer from 2015 to 2017, and an Editor-in-Chief of the IEEE Computational Intelligence Magazine from 2014 to 2017. He is also an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE ACCESS.



Tianjun Liao received the B.S. in system engineering from the National University of Defense Technology, Changsha, China, in 2007 and the Ph.D. degree in engineering sciences from the Institute de Recherches Interdisciplinaires et de Developpements en Intelligence Artificielle (IRIDIA), Universite Libre de Bruxelles, Brussels, Belgium, in 2013. He is currently a Assistant Researcher with State Key Lab of Complex System Simulation at Beijing Institute of System Engineering. His current research interests include heuristic optimization algorithms for continuous and mixed discrete-continuous optimization problems and automated algorithm configuration.

Tao Zhang received his B.S., M.S., PhD degree from the National University of Defense Technology (NUDT), P.R. China, in 1998, 2001, and 2004, respectively. He is currently a professor at the college of information system and management, the NUDT. His main research areas are multi-criteria decision making, optimal scheduling, data mining and optimization methods on energy internet network.