On the effect of localized PBI method in MOEA/D for multi-objective optimization

Rui Wang*, Hisao Ishibuchi†, Yan Zhang*, Xiaokun Zheng* and Tao Zhang*
* College of Information Systems and Management, National University of Defense Technology, P.R. China
† Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Japan
E-mail: ruiwangnudt@gmail.com, hisaoi@cs.osakafu-u.ac.jp

***Abstract*—The idea of localization, e.g., mating restriction, local search, is often employed in the design of evolutionary multi-objective algorithms, and has been demonstrated effective in many studies. This paper proposes a localized penalty boundary intersection (PBI) method which is then used in the seminal decomposition based algorithm, i.e., MOEA/D. The localized PBI (LPBI) method works in a pre-defined hypercone, that is, solutions compete with its neighbors in the same hypercone for survive. The size of the hypercone is determined automatically prior to the search. Experimental results show that i) the LPBI method improves the performance of MOEA/D-PBI for a wide range of penalty parameter values, in particular, for small penalty values; ii) for most of test problems, MOEA/D-LPBI with a small penalty value offers the best performance; and iii) The LPBI tends to be less sensitive than the PBI method on penalty parameter values.**

I. Introduction

MANY real world optimization problems are multi-objective in nature where more than one objectives are required to be optimized, simultaneously. Since these objectives are often conflicting with each other, the optimal solution of a multi-objective problem (MOP) is a set of trade-off solutions (so called the Pareto optimal set). Evolutionary multi-objective (EMO) algorithms are well-suited for addressing MOPs as they can approximate the trade-off surface (so called the Pareto front, PF) in a single run [1].

In the last three decades, a variety of EMO algorithms have been proposed. Pareto-dominance based algorithms, e.g., NSGA-II [2] are one of the earliest approaches for evolutionary multi-objective optimization, which have been demonstrated as effective for MOPs with two and three objectives. However, their performance degrades significantly as the number of objectives increases [3]. In addition to Pareto-dominance based algorithms, there are performance indicator based algorithms, e.g., HypE [4]; preference-inspired algorithms, e.g., PICEA-g [5], [6], PICEA-w [7], [8], decomposition based algorithms, e.g., C-MOGA [9], MOEA/D [10]. Performance of these algorithms is not much deteriorated for MOPs with more than three objectives.

Amongst these algorithms, decomposition based algorithms become increasingly popular in recent years. Typically, a decomposition based algorithm decomposes a MOP into a set of sub-problems, and solves these sub-problems in a collaborative manner. The sub-problem could either be single-objective [10] or multi-objective [11]. In the seminal decomposition based algorithm, i.e., MOEA/D, the sub-problem is single-objective, and is realized by means of scalarizing methods. MOEA/D has been intensively studied. The algorithm is shown to have high search ability for combinatorial optimization, high compatibility with local search and good performance on problems with complex Pareto sets [12], [13].

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

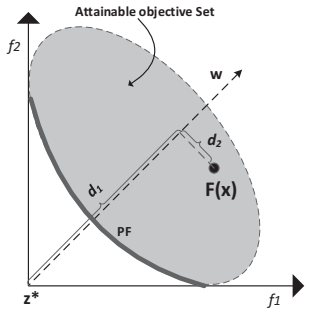

Fig. 1. The d_1 and d_2 distances in the PBI method.

The penalty boundary intersection (PBI) method is a frequently used scalarizing method in decomposition based algorithms. It returns a penalized distance value of the form $d_1+\theta d_2$ for a solution $\mathbf{x}$ with respect to a weight $\mathbf{w}$ (also called reference direction or search direction) wherein d_1 denotes the distance along the reference line, and d_2 denotes the perpendicular distance to $\mathbf{w}$, see Fig. 1 and Eq. (1). θ is a user-defined penalty parameter.

$$\begin{aligned} d_1 &= \frac{\|(\mathbf{F}(\mathbf{x})-\mathbf{z}^*)^T\mathbf{w}\|}{\|\mathbf{w}\|} \\ d_2 &= \|\mathbf{F}(\mathbf{x})-(\mathbf{z}^*+d_1\frac{\mathbf{w}}{\|\mathbf{w}\|})\| \end{aligned} \tag{1}$$

MOEA/D with PBI method (MOEA/D-PBI) performs well when a suitable θ is chosen. Usually, the smaller the θ value, the faster the algorithm converges [14]. However, a small θ might not work effectively on non-convex PFs [12].

This study is going to examine the performance of a localized PBI method (LPBI) in MOEA/D. The LPBI works within a pre-defined hypercone. The size of the hypercone is

determined automatically prior to the search. MOEA/D with LPBI (MOEA/D-LPBI) is compared against MOEA/D-PBI on a set of benchmarks. Experimental results show that the localized version is much more efficient than the standard version of PBI method, especially when a small penalty value is used. The LPBI method can nicely work on non-convex PFs. LPBI with a small penalty value tends to perform better than PBI with a large penalty value. Additionally, though LPBI is also impacted by the choice of a penalty value, its impact is not as much as PBI exhibits.

The rest of the paper is structured as follows: Section II provides some background knowledge. Section III introduces the localized PBI method. This is followed, in Section IV, by a description of MOEA/D-LPBI. Section V and Section VI present the experiment descriptions , results and discussions. Section VII concludes this paper and identifies some future studies.

II. Background

Without loss of generality, a minimization MOP can be written as follows:

$$\text{Minimize} \quad \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})\} \tag{2}$$

where m is the number of objectives, $\mathbf{F}(\mathbf{x})$ is the m-dimensional objective vector, $f_i(\mathbf{x})$ is the i-th objective to be minimized, and $\mathbf{x} \in \mathbf{\Omega}$ is the decision vector.

A. Basics of multi-objective optimization

Definition 1: $\mathbf{x}$ 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 $j \in \{1, 2, \ldots, m\}$.

Definition 2: 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 $\mathbf{x} \preceq \mathbf{x}^*$.

Definition 3: 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).

Definition 4: The *ideal* objective vector $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$, where $\mathbf{z}_i^*$ is the infimum of $f_i(\mathbf{x})$ for every $i \in \{1, 2, \ldots, m\}$.

Definition 5: An *utopian* objective vector $\mathbf{z}^u$ is an infeasible objective vector whose component can be formed by $\mathbf{z}_i^u = z_i^* - \epsilon_i,\ i = 1, 2, ..., m$, where z_i^* is the component of the *ideal* objective vector, and $\epsilon_i > 0$ is a relatively small but computationally significant scalar.

Definition 6: The *nadir* objective vector $\mathbf{z}^{nad} = (z_1^{nad}, \ldots, z_m^{nad})$, where z_i^{nad} is the supremum of $f_i(\mathbf{x}), \mathbf{x} \in$ PS for every $i \in \{1, 2, \ldots, m\}$.

B. Principle of decomposition based EMO algorithm

Decomposition based EMO algorithms[1] decompose a MOP into a number of sub-problems. These sub-problems are then solved in a collaborative manner. In most of the existing studies, sub-problems are single-objective, and are defined by means of scalarizing methods. Take the PBI method as an example, the following single-objective problem is defined.

[1]Hereafter we use decomposition based algorithms for short

$$\begin{gathered} g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^u) = d_1 + \theta d_2 \\ \mathbf{x} \in \mathbf{\Omega} \end{gathered} \tag{3}$$

where d_1 and d_2 are defined in Eq. (1). The goal is to push $g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^u)$ as low as possible so that it can reach the boundary of the attainable objective set, see Fig. 1.

The optimal solution of each single objective problem corresponds to a Pareto optimal solution of the MOP [15, pp.98-99]. By minimizing the scalarizing function with different weight vectors, a set of diversified Pareto optimal solutions could be obtained.

Since objectives might have totally different scales, the following normalization procedure is often applied:

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

When 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 to estimate z_i^* and z_i^{nad}, respectively. However, this might not be always accurate especially in the early stage of the evolution.

C. Related studies on the PBI method

The PBI method is built on the normal-boundary intersection method [16] whose equality constraint is handled by a penalty function [10], as shown in Eq. (3). Due to its simplicity and promising performance on MOPs reported in [12], [14], [17], the PBI method has been applied in many recently proposed decomposition based approaches such as [18], [19], [20], [21].

Fig. 2 illustrates contour lines of PBI methods in 2-D cases with $\theta = 0$, $\theta = 1$, $\theta = 10$. Drawn from Fig. 2 and literature, the following remarks with respect to the PBI method are obtained:

- Different penalty values, i.e., θ, exhibit different contour lines, resulting in different search behaviours;
- For PBI with a small θ value, a move of a solution away from the reference line $\mathbf{w}$ (an increase of d_2 distance) is acceptable if it decreases the distance from the reference point, i.e., d_1 distance. That is, the optimal solution could be far away from its corresponding reference line. Meanwhile, this also implies that a small penalty value is more prone to convergence performance.
- For PBI with a large θ value, a small increase in d_2 is not compensable for a large decrease in d_1. That is, a large θ value makes the PBI robust on various PF shapes.
- As θ decreases the search behaviour of PBI tends to be similar to the weighted sum method. That is, PBI with a small θ might encounter the same difficulty as the weighted sum method on non-convex PFs. Effectively, the impact of θ in the PBI method is similar to the p value in L_p scalarizing methods [22], [23].

Although many studies have demonstrated that PBI with different θ values show different search behaviours, surprisingly, scholars tend to employ $\theta = 5$ in their studies, e.g., [18], [20]. None of studies (to the best of the authors' knowledge) have

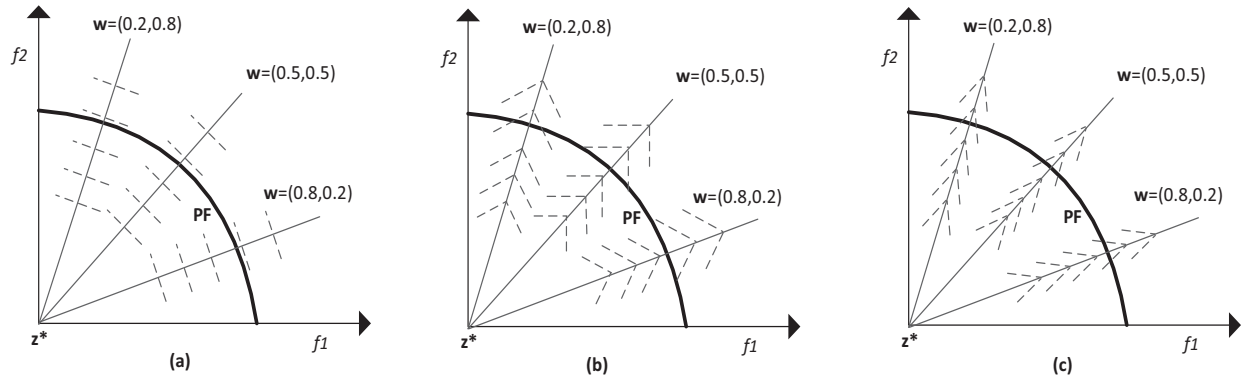

Fig. 2. Illustration of contour lines of PBI with different penalty values. $\theta = 0$, 1, 10 from the left to right.

investigated issues such as how to appropriately select the θ value. Specifically, how to utilize the advantages of PBI with a small θ value, i.e., its fast convergence. This is study is going to answer this question.

III. Localized PBI method

Motivations for the localized PBI methods are that i) PBI is sensitive to the choice of a penalty value, ii) PBI with a small penalty value has a fast convergence, but encounters difficulty on non-convex PF shapes. It is expected that a localized PBI method can handle non-convex PF shapes and maintain a fast convergence. Also, it can be less sensitive to the choice of a penalty value.

As the name says, the localized PBI method, denoted as LPBI, implements the PBI method in a local manner. Similar discussions on this issue are [24], [25]. Specifically, the newly generated solution is not compared with the entire population but only with solutions that are inside the same hypercone. Each weight $\mathbf{w}$ is associated with a hypercone, see Fig. 3. The center line of the hypercone is identical to the weight. The apex angle (Φ_i) of the hypercone for a weight $\mathbf{w}_i$ is defined as the average angle between $\mathbf{w}$ and its m closest weights (where m is the number of objectives), see Eq. (5):

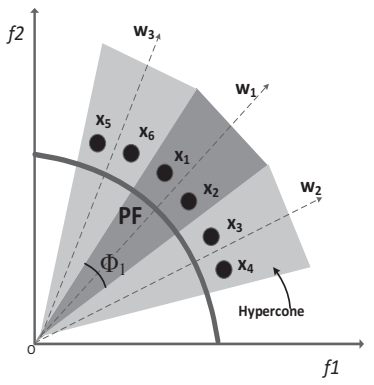

Fig. 3. Illustration of the localized PBI method.

$$\Phi_i = \frac{\sum_{j=1}^{j=m} \phi_i^j}{m} \tag{5}$$

where ϕ_i^j is the angle of the j-th closest weight vector to weight $\mathbf{w}_i$. Given two weights, e.g. $\mathbf{w}_1$ and $\mathbf{w}_2$, the cosine value of the acute angel ϕ between the two vectors can be calculated as follows:

$$\phi = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\|\mathbf{w}_1\|\|\mathbf{w}_2\|} \tag{6}$$

When weight vectors are evenly distributed, Φ_i is almost identical for different hypercones. That is, the size of hypercone for each weight is almost identical.

In the LPBI method, the solution corresponding to each $\mathbf{w}$ is selected from solutions that are only inside the hypercone of this weight. For example, in Fig. 3, the solution for $\mathbf{w}_1$ is selected from $\mathbf{x}_1$ and $\mathbf{x}_2$ only. Therefore, the obtained solution would not be very far away from its corresponding reference line. Even in the worst case, the angle between the obtained solution and its associated reference line is $\frac{\Phi}{2}$. This enables the LPBI to handle non-convex PFs. Specifically, implementation of the LPBI is as follows: PBI values of solutions that are outside the hypercone are set as ∞; while PBI values of solutions that are inside the hypercone are kept unchanged.

Overall, by the LPBI method, the PF is divided into N sub-regions (assuming that the number of weight vectors is N), each LPBI is responsible for searching for an optimal solution within its associated sub-region, i.e., hypercone. Ideally, LPBI methods with different weight vectors would find different optimal solutions, and thus, providing a set of diversified solutions.

IV. MOEA/D-LPBI

Pseudo-code of MOEA/D-LPBI is shown in Algorithm 1. Prior to the evolution (lines 1 to 6), the following operations are conducted: generate N evenly distributed weight vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$ where the canonical simplex lattice design method [10] is employed; initialize the same size of solutions $S \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$; assign a randomly selected candidate solution to each weight vector; find the neighboring weight

vectors $B(\mathbf{w})$ for each weight vector; and identify the associated neighboring solutions $B(\mathbf{x})$ of $\mathbf{x}$; and compute apex angle of the hypercone for each weight. Then, all solutions are evolved for $maxGen$ generations (lines 7 to 30).

Algorithm 1: MOEA/D-LPBI

Input: N candidate solutions, $S \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, penalty value, θ, selection neighborhood size, T, maximum generation index, $maxGen$
Output: S, $archiveS$

1 Generate N evenly distributed weight vectors, $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$;
2 Randomly paired up $\mathbf{w}$ with a candidate solution $\mathbf{x}$;
3 Find T neighboring weight vectors $B(\mathbf{w})$ of $\mathbf{w}$ in terms of ϕ;
4 Identify the associated neighboring solutions $B(\mathbf{x})$ of $\mathbf{x}$;
5 Compute Φ of the hypercone for each weight by Eq. (5);
6 Set the archive $archiveS = \emptyset$, the mating pool $Q = \emptyset$ and the probability of mating restriction $\delta = 0.8$;
7 **for** $gen \leq maxGen$ **do**
8 **for** $i \leftarrow 1$ **to** N **do**
9 Set a temporary solution set $Sc = \emptyset$;
10 **if** $rand < \delta$ **then**
11 $Q \leftarrow B(\mathbf{x}^i)$;
12 **else**
13 $Q \leftarrow S$;
14 **end**
15 Generate a new solution $\mathbf{x}'$ by applying SBX and PM operators to solutions selected from Q;
16 $Sc \leftarrow Sc \uplus \mathbf{x}'$;
17 **end**
18 $JointS \leftarrow S \uplus Sc$;
19 Compute the objective function values $JointF$ of all candidate solutions in $JointS$;
20 Update *ideal* and *nadir* vectors, and normalize $JointF$ into [0,1];
21 Calculate the angle ϕ_{ij}^{sw} between $\mathbf{F}(\mathbf{x})^i$ and $\mathbf{w}_j$, obtaining an angle matrix ϕ^{sw};
22 For each solution, e.g., $\mathbf{x}_i$ and its associated weight vector, e.g., $\mathbf{w}_j$, if $\phi_{ij}^{sw} \leq \Phi_j$ compute $g^{PBI}(\mathbf{x}_i|\mathbf{w}_j)$, denoted as C_{ij};
23 Otherwise, set $C_{ij} \leftarrow \infty$;
24 $S \leftarrow \emptyset$;
25 **for** $i \leftarrow 1$ **to** N **do**
26 $\mathbf{x}_i \leftarrow \arg\min_{\mathbf{x} \in JointS} C_{ij}$;
27 $S \leftarrow S \cup \mathbf{x}_i$;
28 **end**
29 Update $archiveS$ with S using Pareto-dominance relation;
30 **end**

- Lines 8-17: offspring solutions are reproduced using the SBX and PM operators. Parents are selected from $B(\mathbf{x})$ with a probability 0.8, otherwise from the whole population S.
- Lines 18-20: parent solutions and the newly generated offspring solutions are merged. The *ideal* and *nadir* points are updated based on the joint population
- Line 21: calculate the angle ϕ_{ij}^{sw} between $\mathbf{F}(\mathbf{x})^i$ and $\mathbf{w}_j$.
- Lines 22 and 23: calculate the PBI value $g^{PBI}(\mathbf{x}_i|\mathbf{w}_j, \theta)$, denoted as C_{ij}. Set C_{ij} as ∞ if $\phi_{ij}^{sw} > \Phi_j$
- Lines 24 to 28: for each weight, find the solution $\mathbf{x}$ whose PBI value is the smallest amongst the joint solutions.
- Line 29: update the offline archive $archiveS$ with newly obtained solutions S based on Pareto-dominance.

Lastly, we would like to make the following remarks about MOEA/D-LPBI:

- The algorithm is implemented within a $(\mu + \mu)$ elitism framework as NSGA-II does [2];
- The neighborhood structure $B(\bullet)$ is used only in the solution reproduction while in the solution update procedure another neighborhood structure, i.e. hypercone is applied;
- Multiple replacements are allowed, i.e., the same solution can be selected for different weights;
- One can obtain N solutions with a good distribution by choosing the nearest solution (measured by angle) for each weight from $archiveS$.
- Calculating PBI values for all solutions on all weight vectors runs at $\mathcal{O}(N \times N)$, obtaining the minimal PBI value from $2N$ values runs at $\mathcal{O}(N)$. Thus, the overall time complexity is $\mathcal{O}(N^2)$.

V. Experiments description

A. Test problems

Test problems are generated using the WFG (Walking Fish Group) test suite . Specifically, problem 2 to 9 are invoked in two and four-objective instances [2]. These problems cover most of problem attributes such as separability/non-separability, unimodality/multimodality, unbiased/biased. The number of decision variables n for each problem is set to 24 amongst which the position parameter k is 6 and the distance parameter l is 18. Hereafter, we use WFGx-m refers to problem WFGx with m objectives.

B. Parameter settings

Some general parameters such as population size, the genetic operators, i.e., simulated binary crossover (SBX) and polynomial mutation (PM) are set as follows:

- the number of generations is set to 250; the population size is set to $N = 100$ and $N = 200$ for 2- and 4-objective problems, respectively.
- as suggested in [3], [26] the SBX probability p_c and distribution index η_c are set to 1 and 30, respectively;

[2]The WFG1 benchmark is not employed since our preliminary results show that even for 1e+8 function evaluations our algorithm still cannot approximate its PF. The reason might be that WFG1 is highly biased. In order to approximate the PF, the employed search operators must have the ability for exploiting solutions with high precision. However, search operators used in this study lacks such ability.

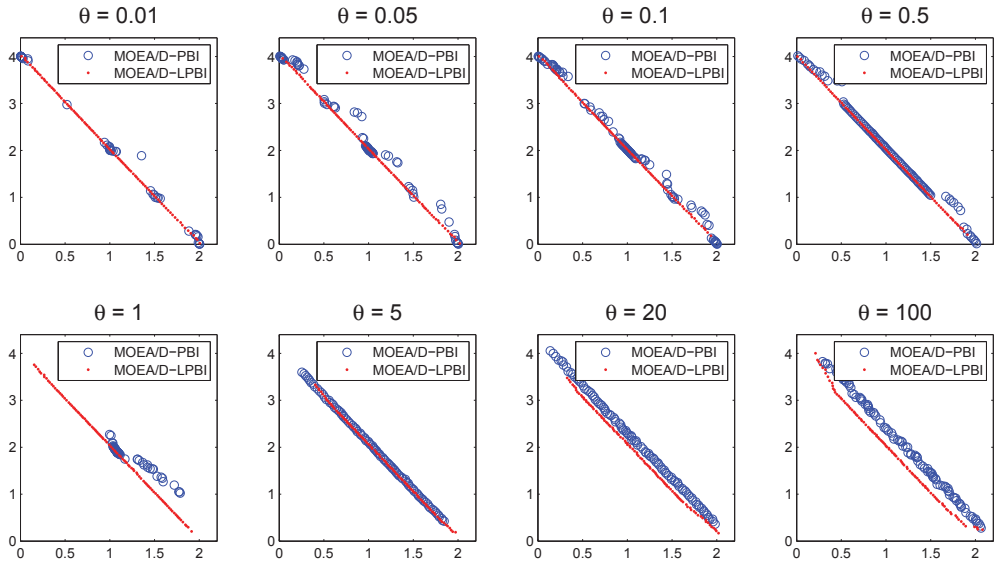

Fig. 4. PFs obtained by MOEA/D-PBI (○) and MOEA/D-LPBI (●) under different penalty values for WFG3-2.

and the PM probability p_m and distribution index η_m are set to $1/n$ and 20, respectively;

In addition, the considered PBI penalty values are $\theta = \{0.01, 0.05, 0.1, 0.5, 1, 5, 20, 100\}$. Moreover, in order to demonstrate the efficiency of LPBI method, MOEA/D with the standard PBI is employed as a reference algorithm. In MOEA/D-PBI, the selection neighborhood size T is set to 10, and the replacement neighborhood size nr is set to 2.

C. Performance assessment

For each test problem, 31 runs of each algorithm test are performed to facilitate a robust statistical analysis. The hypervolume (HV) indicator is selected as the performance metric. Prior to calculating the HV metric, solutions are normalized by the *nadir* and *ideal* points which are $(2, 4, ..., 2m)$ and $(0, 0, ..., 0)$, respectively. The reference point for computing HV is set to $(1.2, 1.2, ..., 1.2)$. Lastly, for each run the HV is computed using the offline archive solution set, however, the set is pruned to the maximum size N (i.e., population size) using the SPEA2 truncation procedure [27].

VI. Results and Discussions

A. Visual results

This section visually examines the PF obtained by MOEA/D-LPBI and MOEA/D-PBI with different penalty values for 2-objective WFG problems. For instance, results for WFG3-2 (whose PF is non-convex, i.e., a straight line) are shown in Fig. 4. The presented PF corresponds to the median HV value across 31 algorithm runs. From the results we can observe that:

- When penalty values are small, MOEA/D-PBI exhibits poor performance, that is, solutions on some PF regions are not obtained. However, MOEA/D-LPBI exhibits good performance.
- As the penalty value increases (from 0.01 to 100), the performance of MOEA/D-PBI varies significantly, i.e., first improves then degrades. $\theta = 5$ offers the best performance.
- MOEA/D-LPBI tends to be robust when θ is small, i.e., $\theta = (0.01, 0.05, 0.1, 0.5)$. Moreover, MOEA/D-LPBI with small penalty values performs better than MOEA/D-LPBI with large penalty values.
- When penalty values are large, MOEA/D-LPBI shows a better convergence than MOEA/D-PBI. One possible reason is that PBI with a large penalty value is not Pareto dominance compatible while the LPBI reduces the probability of Pareto dominance incompatible move because of the use of hypercone.

Due to the space limitation, we have only showed results for WFG3-2. However, similar results are observed for the other 2-objective test problems.

B. Statistical results

MOEA/D-LPBI is quantitatively compared with MOEA/D-PBI in terms of the HV metric. Due to the page limitation, TABLE I only presents the mean HV values for WFG8-2, WFG9-2, WFG8-4 and WFG9-4. In addition, plots of the mean HV values obtained by MOEA/D-PBI and MOEA/D-LPBI under different penalty values for all test problems are shown in Fig. 5 and Fig. 6, which offer a more straightforward comparison.

TABLE I

COMPARISON RESULTS IN TERMS OF THE MEAN HV VALUES. THE BEST HV VALUE FOR EACH PROBLEM IS MARKED AS **BOLD**. THE SYMBOL '$-$', '$=$' OR '$+$' MEANS MOEA/D-LPBI IS STATISTICALLY WORSE THAN, COMPARABLE TO OR BETTER THAN MOEA/D-PBI UNDER THE SAME θ SETTING.

Problem	Alg.	$\theta = 0.01$	$\theta = 0.05$	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$	$\theta = 5$	$\theta = 20$	$\theta = 100$
WFG8-2	PBI	0.1742	0.1744	0.1938	0.2749	0.2996	0.3031	0.2732	0.2327
	LPBI	0.3237^{+}	0.3202^{+}	0.3230^{+}	$\mathbf{0.3270}^{+}$	0.3254^{+}	0.3160^{+}	0.3127^{+}	0.3111^{+}
WFG9-2	PBI	0.2179	0.2076	0.2287	0.3412	0.3866	0.3517	0.3222	0.3113
	LPBI	0.3364^{+}	$\mathbf{0.3906}^{+}$	0.3700^{+}	0.3707^{+}	0.3703^{-}	0.3714^{+}	0.3445^{+}	0.3509^{+}
WFG8-4	PBI	0.5992	0.5902	0.5797	0.5154	0.4545	0.7573	0.7289	0.7027
	LPBI	0.8848^{+}	0.8834^{+}	0.8843^{+}	0.8902^{+}	$\mathbf{0.8953}^{+}$	0.8308^{+}	0.6871^{-}	0.6654^{-}
WFG9-4	PBI	0.6862	0.6708	0.6746	0.7091	0.5493	0.7499	0.7038	0.7117
	LPBI	0.8575^{+}	0.8703^{+}	0.8573^{+}	0.9412^{+}	$\mathbf{0.9445}^{+}$	0.9341^{+}	0.8614^{+}	0.8358^{+}

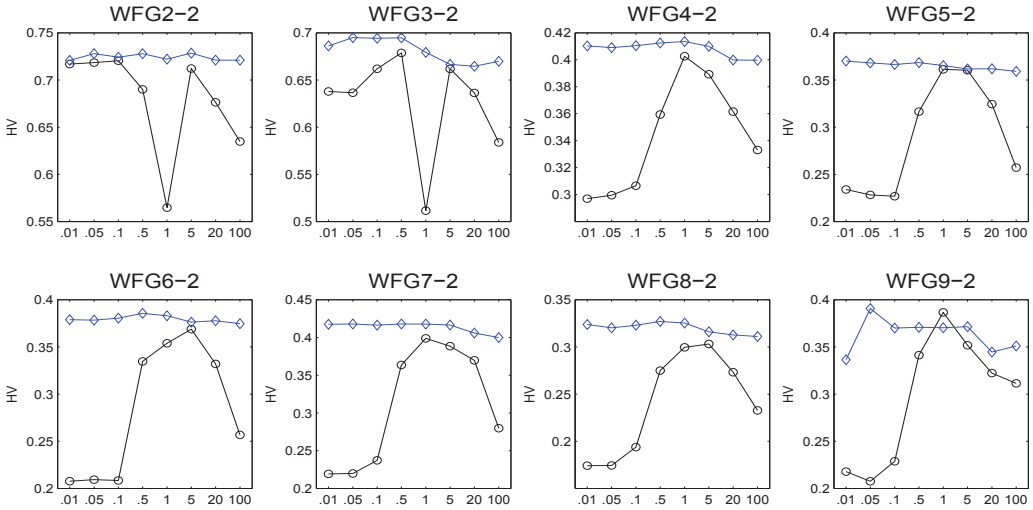

Fig. 5. Plots of mean HV values obtained by MOEA/D-PBI (-○-) and MOEA/D-LPBI (– ◇ –) under different penalty values for all the 2-objective problems.

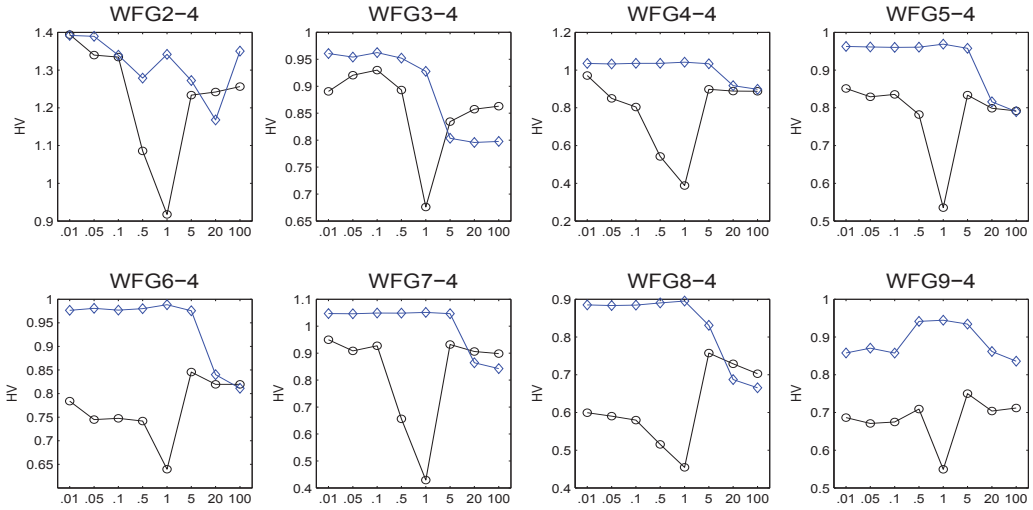

Fig. 6. Plots of mean HV values obtained by MOEA/D-PBI (-○-) and MOEA/D-LPBI (-◇-) under different penalty values for all the 4-objective problems.

From TABLE I, first we observe that for all the four problems the best results are obtained by MOEA/D-LPBI. Second, under the same θ setting, we observe that i) for WFG8-2 and WFG9-4, MOEA/D-LPBI is always better than MOEA/D-PBI; ii) for WFG9-2, MOEA/D-LPBI is better than MOEA/D-PBI for all cases except for $\theta = 1$; and iii) for WFG8-4, MOEA/D-LPBI is better than MOEA/D-PBI for six out of the eight cases. MOEA/D-LPBI is inferior when $\theta = 20$ and $\theta = 100$;

More generally, from Fig. 5 it is observed that

- MOEA/D-LPBI offers better performance than MOEA/D-PBI for most of the test problems under a wide range of θ settings;
- LPBI with a small penalty value is much more effective than PBI with a small penalty value;
- The performance of MOEA/D-PBI varies significantly as θ changes while the performance of MOEA/D-LPBI does not vary as much as MOEA/D-PBI does.
- A clear performance deterioration is observed for MOEA/D-PBI with $\theta = 1$ for WFG2-2 and WFG3-2.

From Fig. 6 it is observed that

- The best results are obtained by MOEA/D-LPBI for all problems;
- MOEA/D-LPBI tends to be robust on the choice of small penalty values for some problems, i.e., from WFG4-2 to WFG8-2.
- MOEA/D-LPBI with small penalty values is clearly superior to MOEA/D-LPBI with large penalty values and MOEA/D-PBI;
- For some large penalty values, e.g., $\theta = 100$, MOEA/D-LPBI is inferior to MOEA/D-PBI;
- Again, a clear performance deterioration is observed for MOEA/D-PBI with $\theta = 1$ for all problems.

Overall from the above comparison we can tentatively draw the following conclusions:

- The effect of PBI and LPBI methods are both dependent on the choice of a penalty value. However, the influence induced by different penalty values (especially when penalty values are small) on LPBI is smaller than that on PBI, that is, MOEA/D-PBI much less sensitive to the specification of small penalty values.
- PBI with a small penalty value performs poorly on non-convex PFs. However, LPBI does not suffer from the issue of small penalty values. For the PBI method, good performance is usually achieved when a moderate penalty value, e.g, 5 is selected. For the LPBI method, good performance is achieved when a small penalty value is selected.
- The localization of PBI improves the performance of MOEA/D-PBI for a wide range of penalty parameter values except for very large values. For all the examined test problems, better results are obtained by LPBI with a small penalty value than PBI with any settings of the penalty parameter values. Thus, the LPBI with a small parameter value (instead of PBI) is advisable. The superior performance of MOEA/D-LPBI with small penalty values is mainly because that the LPBI retains the high search efficiency of PBI with small penalty values, and also mitigates the limitation of PBI on non-convex PFs (i.e., improving the uniformity of obtained solutions).

Interestingly, MOEA/D-LPBI is found to be inferior to MOEA/D-PBI when the penalty value is very large, e.g., $\theta = 100$, for 4-objective problems. The reason might be that the diversity improvement (mainly uniformity) made by LPBI with large penalty values is not as significant as that for LPBI with small penalty values. Effectively, PBI with a large penalty value is already good to maintain good uniformity performance. However, it should be mentioned that the localized PBI is originally proposed for PBI with small penalty values, aiming to utilize the high search efficiency of PBI with small penalty values. Besides, it has been demonstrated that PBI with a large penalty value is less efficient than PBI with a small penalty value in pushing solutions towards the PF. Moreover, our experimental results have shown that in most cases neither PBI nor LPBI with a very large penalty value offers the best performance. Thus, the use of large penalty values is not advisable.

Additionally, in MOEA/D-PBI a clear performance deterioration is observed when $\theta = 1$, especially for 4-objective problems. As explained in [28], this is mainly due to the loss of diversity. Most of solutions obtained by PBI with $\theta = 1$ are clustered around the center of the PF. This can also be observed from Fig. 4. Besides, it is not precise to say that PBI with $\theta = 1$ is similar to the Chebyshev method whereas their contour lines are the same for the weight vector (0.5, 0.5), see Fig. 2.

C. A further study

Previous experiments have demonstrated the efficiency of MOEA/D-LPBI on problems having non-convex PFs. Next we examine the algorithm's performance on problems having convex PF shape. The modified WFG4 test problem denoted as WFG42 [8] is employed. The WFG42 is built on WFG4 where the concave shape function in WFG4 is replaced with the convex shape provided in the WFG test suite. All settings are the same as that adopted in Section VI. TABLE II presents the comparison results.

TABLE II
THE HV COMPARISON RESULTS OF MOEA/D-LPBI AND MOEA/D-PBI. THE BEST RESULTS FOR EACH PROBLEMS IS MARKED AS **BOLD**

Problem	Alg.	$\theta = 0.01$	$\theta = 100$
WFG42-2	PBI	**0.9889**	0.9165
	LPBI	0.9831	0.9743
WFG42-4	PBI	1.4554	1.4212
	LPBI	**1.4557**	1.4177

It is again observed that better results are obtained by using small penalty values. More specifically, MOEA/D-LPBI is inferior to MOEA/D-PBI for WFG42-2. This is because in 2-objective case the localized technique degrades the convergence of the PBI method. About 25% solutions

obtained by MOEA/D-LPBI are dominated by those obtained by MOEA/D-PBI for WFG42-2. Effectively, according to the contour lines of PBI with $\theta = 0.01$ we find that the region for obtaining better solutions is greatly reduced because of the restriction of hypercone. When $m = 4$ the side effect induced by the hypercone becomes very small, and thus, MOEA/D-LPBI and MOEA/D-PBI performs comparably.

VII. Conclusion

The effect of the PBI method varies as the penalty value changes. This study proposes a localized PBI method. It is found that the localized PBI improves the performance of MOEA/D-PBI for a wide range of penalty parameter values except for very large values. This improvement is mainly achieved by significantly improving the uniformity of obtained solutions at the cost of very small deterioration of the convergence ability (when the penalty parameter value is not large). This improvement makes the performance of MOEA/D-PBI much less sensitive to the specification of the penalty parameter value.

As for future studies, first it is observed that when a large penalty value is chosen, MOEA/D-LPBI is inferior to MOEA/D-PBI. Unlocking this issue may further understand the search behaviour of PBI method, and thus is worth studying. Second, we would like to extend the localized technique to other scalarizing methods. Third, the localized idea may retain some isolated (and inferior) solutions especially when the PF is disconnected or has a large hole. Also, some weight vectors may have to keep waiting for solutions generated inside their cones. How to effectively handle these issues, i.e., matching mechanisms between solutions and weight vectors, deserves further studies. Lastly, it is worth studying how an appropriate penalty value can be selected for different problems.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] ——, "Preference-inspired co-evolutionary algorithm using adaptively generated goal vectors," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. Piscataway, NY, USA: IEEE Press, June 2013, pp. 916–923.

[7] ——, "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.

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

[9] 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.

[10] 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.

[11] 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.

[12] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 2, pp. 264–283, 2015.

[13] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Evolutionary Computation (CEC), 2009 IEEE Congress on*. IEEE, 2009, pp. 203–208.

[14] H. Ishibuchi, 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[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] 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.

[23] ——, "Decomposition based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. To appear, 2016.

[24] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization." *Cybernetics IEEE Transactions on*, vol. 45, no. 10, pp. 2076–2088, 2015.

[25] 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.

[26] 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.

[27] 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.

[28] H. Ishibuchi, K. Doi, and Y. Nojima, "Characteristics of Many-Objective Test Problems and Penalty Parameter Specification in MOEA/D," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*, Vancouver, Canada, July 2016, p. To appear.