

Leveraging Indicator-based Ensemble Selection in Evolutionary Multiobjective Optimization Algorithms

Dung H. Phan
Dept. of Computer Science
University of Massachusetts,
Boston
Boston, MA, USA 02125-3393
phdung@cs.umb.edu

Junichi Suzuki
Dept. of Computer Science
University of Massachusetts,
Boston
Boston, MA, USA 02125-3393
jxs@cs.umb.edu

Isao Hayashi
Faculty of Informatics
Kansai University
Osaka, Japan 569-1095
ihaya@res.kutc.kansai-
u.ac.jp

ABSTRACT

Various evolutionary multiobjective optimization algorithms (EMOAs) have replaced or augmented the notion of dominance with quality indicators and leveraged them in selection operators. Recent studies show that indicator-based EMOAs outperform traditional dominance-based EMOAs. This paper proposes and evaluates an ensemble learning method that constructs an ensemble of existing indicators with a novel boosting algorithm called Pdi-Boosting. The proposed method is carried out with a training problem in which Pareto-optimal solutions are known. It can work with a simple training problem, and an ensemble of indicators can effectively aid parent selection and environmental selection in order to solve harder problems. Experimental results show that the proposed method is efficient thanks to its dynamic adjustment of training data. An ensemble of indicators outperforms existing individual indicators in optimality, diversity and robustness. The proposed ensemble-based evolutionary algorithm outperforms a well-known dominance-based EMOA and existing indicator-based EMOAs.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

Evolutionary multiobjective optimization algorithms, Quality indicators, Indicator-based ensemble selection, Boosting

1. INTRODUCTION

A recent research trend in the design space of evolutionary multiobjective optimization algorithms (EMOAs) is to

adopt *indicator-based selection* operators that augment or replace traditional dominance ranking with *quality indicators* [4]. A quality indicator measures the goodness of each individual. Recent research findings (e.g., [2, 18, 21]) show that indicator-based EMOAs outperform traditional EMOAs that use dominance ranking in selection.

Many quality indicators have been proposed with the intention to capture different preferences in optimization [1, 3, 19, 21]. Therefore, indicator-based selection operators tend to have biased selection pressures that evolve individuals toward particular regions in the objective space. For example, the hypervolume indicator favors *balanced* individuals that equally balance the trade-offs among all objectives, while the weighted hypervolume indicator favors *extreme* individuals that yield superior performance only in a limited number of objectives [19]. A research question in this context is whether a set of existing indicators can create a single indicator that outperforms those existing ones.

The authors of the paper have addressed this question with the notion of *indicator-based ensemble selection*, which means selection with an ensemble of indicators, and verified that an ensemble of indicators outperforms each of those indicators [15, 16]. In [15, 16], a traditional boosting algorithm called AdaBoost [9] was used to construct ensembles.

This paper enhances the prior work by replacing AdaBoost with a novel boosting algorithm called Pdi-Boosting (Probabilistic data interpolation-Boosting). Pdi-Boosting interpolates data around misclassified data using a given probability density function, while AdaBoost labels misclassified data with associated weights.

Designed with Pdi-Boosting, the proposed boosting method constructs an ensemble of indicators for two types of selection operators in EMOAs: (1) *parent selection*, which chooses parent individuals from the population to reproduce offspring and (2) *environmental selection*, which chooses a set of individuals used in the next generation from the union of the current population and its offspring. The proposed boosting method is carried out with a training problem in which Pareto-optimal solutions are known. It can work with a simple training problem, and an ensemble of indicators can effectively aid to solve harder problems.

This paper also proposes a new EMOA, called BIBEA-P, which uses an ensemble selection operator constructed with the proposed boosting method. It is designed as a variant of an existing indicator-based EMOA: IBEA [21].

Experimental results show that the proposed boosting method outperforms its previous AdaBoost-based method in opti-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.
Copyright 2012 ACM 978-1-4503-1177-9/12/07 ...\$10.00.

mality, diversity and convergence metrics. It is 44% faster than the AdaBoost-based method thanks to its dynamic adjustment of training data. Experimental results also demonstrate that an ensemble of indicators outperforms existing individual indicators in optimality and diversity. The ensemble exhibits higher robustness than existing indicators against different characteristics in different problems and yields more stable performance to solve a wider range of problems. Moreover, BIBEA-P outperforms a well-known traditional EMOA (NSGA-II [5]) and existing indicator-based EMOAs (IBEA [21] and SMS-EMOA [2]).

2. RELATED WORK

This paper extends the authors' prior work [15,16], which investigated indicator-based ensemble selection with AdaBoost. As discussed in the previous section, this paper leverages a new boosting algorithm, Pdi-Boosting, in order to construct more efficient and effective ensembles of indicators.

Several existing work have integrated ensemble methods, including boosting algorithms, with evolutionary algorithms (EAs) although they have never used ensemble methods for selection operators in EAs. For example, boosting algorithms have been integrated with genetic algorithms (GAs) to solve classification problems [12,13]. The Boosting Genetic Algorithm integrates boosting with a GA to discover classification rules [12]. The GA is used as a base classifier in which each individual represents a classification rule. A boosting algorithm aggregates multiple base classifiers (i.e., GAs) to build a more accurate classifier than them.

Liu et al. integrate boosting with a GA for feature selection [13]. (Feature selection aims to identify the features that strongly contribute to classification accuracy.) The GA evolves a set of individuals, each of which encodes a feature selection candidate, and seeks the optimal feature selection that minimizes classification error. A feature selection candidate represents a set of boosted classifiers, each of which considers a single feature to perform classification. Boosted classifiers are constructed on a feature by feature basis.

Naujoks et al. propose an online statistical technique to determine the termination condition (i.e., the maximum number of objective function evaluations) for an indicator-based EMOA: SMS-EMOA [14]. This paper does not focus on dynamic adjustment of the termination condition, but uses statically-fixed termination conditions.

3. QUALITY INDICATORS

This section describes 19 representative quality indicators that the proposed boosting method uses.

3.1 Hypervolume Indicator

I_H measures the volume of a hypercube that an individual dominates in the objective space [22]. The hypercube is formed with the individual and the reference point representing the highest (or worst) possible objective values $\vec{r} = (r_1, r_2, \dots, r_n)$ where n denotes the number of objectives. I_H of an individual \vec{x} is calculated as follows where $f_i(\vec{x})$ denotes the i -th objective function value of \vec{x} .

$$I_H(\vec{x}) = \prod_{i=1}^n |r_i - f_i(\vec{x})| \quad (1)$$

I_H is intended to favor balanced individuals in the objective space rather than extreme ones [22].

3.2 Weighted Hypervolume Indicators

I_W is an extension to I_H in that I_W places different weights on different regions in the objective space while I_H places the uniform weight on all regions [19]. I_W of an individual $\vec{x} = (x_1, x_2, \dots, x_n)$ is computed as follows.

$$I_W(\vec{x}) = \int_{(x_1, x_2, \dots, x_n)}^{(r_1, r_2, \dots, r_n)} w(\vec{a}) dz \quad (2)$$

$$\text{where } w(\vec{a}) = \frac{\sum_{i=1}^n e^{k_i(r_i - a_i)}}{\sum_{i=1}^n e^{k_i}}$$

$w(\vec{a})$ denotes the weight of a point $\vec{a} = (a_1, a_2, \dots, a_n)$ in the objective space. It is calculated by applying a weight distribution $\vec{k} = (k_1, k_2, \dots, k_n)$. k_i is the weight assigned to the i -th objective. Given a greater k_i value, I_W favors extreme individuals that are closer to the f_i axis in the objective space. Note that I_W is equal to I_H when $\vec{k} = (0, 0, \dots, 0)$.

This paper uses nine variants of I_W (I_{W1} to I_{W9}) based on nine different combinations of k_1 and k_2 values (Table 1). These value combinations follow the parameter settings in [19]. Note that this paper considers two weight values (k_1 and k_2) because it uses a training problem whose objective space is two dimensional.

Table 1: Nine Variants of I_W

I_W variants	k_1	k_2	I_W variants	k_1	k_2
I_{W1}	10	10	I_{W6}	0	20
I_{W2}	10	0	I_{W7}	30	30
I_{W3}	0	10	I_{W8}	30	0
I_{W4}	20	20	I_{W9}	0	30
I_{W5}	20	0			

3.3 HypE Indicator

I_{HypE} is also an extension to I_H . This indicator places different weights on different portions in the hypervolume that an individual dominates. The hypervolume is divided into multiple portions based on how many other individuals dominate it as well. I_{HypE} of \vec{x} is computed as follows [1].

$$I_{HypE}(\vec{x}) = \sum_{i=1}^{\mu} \frac{1}{i} H_i(\vec{x}) \quad (3)$$

μ denotes the population size (the number of individuals in the population). $H_i(\vec{x})$ denotes the hypervolume that is dominated by \vec{x} and other $(i - 1)$ individuals in the population. H_1 is the hypervolume that \vec{x} dominates exclusively. The highest weight (1) is given to H_1 . H_2 is the hypervolume that \vec{x} and another individual dominate. The second highest weight ($\frac{1}{2}$) is given to H_2 . The lowest weight ($\frac{1}{\mu}$) is given to H_{μ} , which all individuals in the population dominate.

3.4 Binary ε Indicators

I_{ε} takes two individuals (\vec{x} and \vec{y}) and measures the distance between them on a per-objective basis [21].

$$I_{\varepsilon}(\vec{x}, \vec{y}) = \max_{i \in \{1, \dots, n\}} (f_i(\vec{x}) - f_i(\vec{y})) \quad (4)$$

This paper considers two methods to evaluate the quality of an individual (\vec{x}) against the other individuals in the population P . The first method is to sum up I_{ε} values.

$$I_{\epsilon 1}(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} I_{\epsilon}(\vec{y}, \vec{x}) \quad (5)$$

The second method amplifies the influence of dominating individuals over dominated ones.

$$I_{\epsilon 2}(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} -e^{-I_{\epsilon}(\vec{y}, \vec{x})/l} \quad (6)$$

l is a scaling coefficient. $l = 0.05$ in this paper, which is a recommended value in [21].

3.5 Binary Hypervolume Indicators

I_{HD} takes two individuals (\vec{x} and \vec{y}) and measures the hypervolume dominated by \vec{x} but not by \vec{y} [21].

$$I_{HD}(\vec{x}, \vec{y}) = \begin{cases} H(\vec{x}) - H(\vec{y}) & \text{if } \vec{x} \text{ dominates } \vec{y} \\ H(\vec{x}) - H(\vec{x}) \cap H(\vec{y}) & \text{otherwise} \end{cases} \quad (7)$$

$H(\vec{x})$ denotes the hypervolume that \vec{x} dominates.

Similar to $I_{\epsilon 1}$ and $I_{\epsilon 2}$, this paper considers two variants, I_{HD1} and I_{HD2} , to evaluate the quality of an individual (\vec{x}) against the other individuals in the population. $I_{HD1}(\vec{x})$ and $I_{HD2}(\vec{x})$ are computed by replacing $I_{\epsilon}(\vec{y}, \vec{x})$ with $I_{HD}(\vec{x}, \vec{y})$ in Equations 5 and 6, respectively.

3.6 Fuzzy Indicators

I_F is computed based the relationship of the fuzzy Pareto dominance between two individuals (\vec{x} and \vec{y}) [10].

$$I_F(\vec{x}, \vec{y}) = \frac{\prod_i \min(f_i(\vec{x}), f_i(\vec{y}))}{\prod_i f_i(\vec{x})} \quad (8)$$

Similar to $I_{\epsilon 1}$ and $I_{\epsilon 2}$, this paper considers two variants, I_{F1} and I_{F2} , to evaluate the quality of an individual (\vec{x}) against the other individuals in the population. I_{F1} and I_{F2} are computed with Equations 5 and 6, respectively. $l = 0.01$ for I_F in this paper.

3.7 Volume Indicators

I_V is computed as follows [11].

$$I_V(\vec{x}, \vec{y}) = \frac{H(\vec{x}) - H(\vec{y})}{H(\vec{x}) \cap H(\vec{y})} \quad (9)$$

I_V also has two variants, I_{V1} and I_{V2} , which are computed as I_{F1} and I_{F2} are computed.

4. THE PROPOSED BOOSTING METHOD

Algorithm 1 shows the proposed boosting method, which employs Pdi-Boosting. It takes M indicator-based selection operators (S) and aggregates some of them as an ensemble (S^*). This paper uses 19 tournament selection operators that use 19 indicators described in Section 3 ($M = 19$).

The proposed boosting method is carried out through of-line training with a multiobjective optimization problem in which Pareto-optimal solutions are known. This training problem is used to generate N_0 initial training populations, $\{p_1, p_2, \dots, p_{N_0}\}$, each of which contains μ individuals (Line 1). These individuals represent randomly-chosen points in the region that the Pareto-optimal solutions dominate in the objective space.

Algorithm 1 The Proposed Boosting Method

Input: $S = \{s_1, s_2, \dots, s_M\}$, M indicator-based selection operators

Output: S^* , an ensemble of operators

- 1: Generate the initial N_0 training populations: $\mathcal{P} = \{p_1, p_2, \dots, p_{N_0}\}$
 - 2: $t = 0, N = N_0, S^* = \phi$
 - 3: **repeat**
 - 4: $t = t + 1$
 - 5: Each operator s_j performs individual selection N_p times on each training population p_i .
 - 6: Calculate the selection error rate (e_j) for s_j

$$e_j = \frac{\sum_{i=1}^N I_{ji}}{N}$$
 where
$$I_{ji} = \begin{cases} 0 & \text{if } s_j \text{'s selection is successful on } p_i \\ 1 & \text{otherwise} \end{cases}$$
 - 7: Choose an operator s_t^* such that $s_t^* = \arg \min_{s_j \in S} e_j$ and $s_t^* \notin S^*$
 - 8: $S^* = S^* \cup \{s_t^*\}$
 - 9: $d = N \times (1 - 2e_j)$
 - 10: Let \mathcal{P}' be a set of training populations on which s_t^* 's individual selection is unsuccessful.
 - 11: Let \mathcal{W} be a set of HVR values that s_t^* yields with training populations in \mathcal{P}'
 - 12: $D = \text{interpolateExtraPopulations}(\mathcal{P}', \mathcal{W}, d)$
 - 13: $\mathcal{P} = \mathcal{P} \cup D$
 - 14: $N = N + d$
 - 15: **until** $e_t < 0.5$
 - 16: **return** S^*
-

The proposed boosting method iteratively executes a loop (Line 3 to 15) and selects one operator into S^* in each iteration. In each iteration, each of M operators selects an individual N_p times (i.e., N_p individuals in total) from each training population (Line 5). The quality of those N_p individuals is evaluated with the hypervolume ratio (HVR) metric [17]. HVR is computed as the ratio of the hypervolume (HV) of N_p individuals (P^S) to the hypervolume of μ individuals in a training population p_i .

$$HVR(P^S) = \frac{HV(P^S)}{HV(p_i)} \quad (10)$$

HV measures the union of the volumes that a given set of individuals dominate in the objective space [22].

The selection of N_p individuals is said to be *successful* if $HVR(P^S)$ is greater than or equal to a threshold: $\theta < 1$. Given this condition, the selection error of each operator (e_j) is calculated as shown in Lines 6 and 7. Then, the proposed boosting method chooses an operator (s_t^*) that has the lowest selection error (Line 7) and includes the operator in the output ensemble S^* (Line 8).

In Line 9, d is computed as the difference between the number of training populations on which individual selection is successful ($N \times (1 - e_j)$) and the number of training populations on which individual selection is unsuccessful ($N \times e_j$). The proposed boosting method generates d extra training populations that are similar to P' , which is a set of training populations on which s_t^* 's individual selection is unsuccessful (Line 12, $|\mathcal{P}'| = N \times e_j$), and adds the extra populations to P (Line 13).

This way, the proposed boosting method dynamically adjusts the number of training populations by interpolating d

populations around \mathcal{P}' so that $|\mathcal{P}'| + d$ becomes equal to the number of populations on which s_i^* 's individual selection is successful. As a result, in subsequent loop iterations, the proposed method favors the operators that perform successful individual selection on interpolated populations.

Algorithm 2 Interpolate Extra Training Populations

Input: $\mathcal{P}' = \{p_1, p_2, \dots, p_m\}$, m training populations
Input: $\mathcal{W} = \{w_1, w_2, \dots, w_m\}$, m HVR values associated with the populations in \mathcal{P}'
Input: d , the number of populations to be interpolated
Output: $\mathcal{P}^* = \{p_1^*, p_2^*, \dots, p_d^*\}$, the populations to be interpolated

- 1: Sort the populations in \mathcal{P}' in ascending order of their HVR values
- 2: $i = 1$
- 3: **while** $i \leq d$ **do**
- 4: Create new empty population p_i^*
- 5: **for each** individual x in p_i **do**
- 6: Create new individual x^*
- 7: **for** $j = 1$ to the total number of objectives **do**
- 8: $f_j(x^*) = f_j(x) + \mathcal{N}(0, \sigma^2)$
- 9: **end for**
- 10: Add x^* to p_i^* .
- 11: **end for**
- 12: Add p_i^* to \mathcal{P}^*
- 13: $i = i + 1$
- 14: **end while**
- 15: **return** \mathcal{P}^*

Algorithm 2 shows how to generate d training populations and interpolate them around $\mathcal{P}' = \{p_1, p_2, \dots, p_m\}$ (`interpolateExtraPopulations()` in Algorithm 1). The m populations in \mathcal{P}' are sorted in ascending order of HVR values (Line 1). This ensures that the populations with smaller HVR values are used more often for interpolation than the populations with higher HVR values when $d < m$. A new population (p_i^*) is generated by modifying the objective values of individuals in $p_i \in \mathcal{P}'$ with a normal distribution $\mathcal{N}(0, \sigma^2)$.

5. THE PROPOSED ENSEMBLE-BASED EMOA: BIBEA-P

This section describes an EMOA, called Boosted Indicator Based Evolutionary Algorithm (BIBEA-P), which leverages the boosting method discussed in Section 4.

Algorithm 3 shows BIBEA-P's algorithmic structure, which is based on an existing indicator-based EMOA: IBEA [21]. In the g -th generation, μ individuals are randomly generated as the initial population (Line 2). In each generation (g), a pair of individuals, called parents (p_1 and p_2), are chosen from the current population with an ensemble of indicator-based selection operators that Algorithm 1 produces (`ensembleBasedParentSelection()`, Lines 6 and 7).

With the crossover rate P_c , two parents reproduce two offspring with the SBX (self-adaptive simulated binary crossover) operator [6] (Lines 9). Each offspring performs polynomial mutation [5] with the probability P_m (Lines 10 to 15). The boosted parent selection, crossover and mutation operators are repeatedly executed on \mathcal{P}_g until μ offspring are reproduced (i.e., until $|\mathcal{O}_g| = \mu$). The offspring (\mathcal{O}_g) are combined with the population \mathcal{P}_g to form \mathcal{R}_g ($|\mathcal{R}_g| = 2\mu$),

which is a pool of candidates for the next-generation individuals (Line 19).

Environmental selection follows offspring reproduction. μ individuals are selected from 2μ individuals in \mathcal{R}_g as the next-generation population \mathcal{P}_{g+1} (`ensembleDrivenEnvSelection()`, Line 20). Environmental selection performs a $(\mu + \mu)$ -elitism.

Algorithm 3 The Algorithmic Structure of BIBEA-P

- 1: $g = 0$
- 2: $\mathcal{P}_g = \text{initializePopulation}(\mu)$
- 3: **while** $g < g_{max}$ **do**
- 4: $\mathcal{O}_g = \emptyset$
- 5: **while** $|\mathcal{O}_g| < \mu$ **do do**
- 6: $p_1 = \text{ensembleBasedParentSelection}(\mathcal{S}^*, \mathcal{P}_g)$
- 7: $p_2 = \text{ensembleBasedParentSelection}(\mathcal{S}^*, \mathcal{P}_g)$
- 8: **if** $\text{random}() \leq P_c$ **then**
- 9: $\{o_1, o_2\} = \text{crossover}(p_1, p_2)$
- 10: **if** $\text{random}() \leq P_m$ **then**
- 11: $o_1 = \text{mutation}(o_1)$
- 12: **end if**
- 13: **if** $\text{random}() \leq P_m$ **then**
- 14: $o_2 = \text{mutation}(o_2)$
- 15: **end if**
- 16: $\mathcal{O}_g = \{o_1, o_2\} \cup \mathcal{O}_g$
- 17: **end if**
- 18: **end while**
- 19: $\mathcal{R}_g = \mathcal{P}_g \cup \mathcal{O}_g$
- 20: $\mathcal{P}_{g+1} = \text{ensembleDrivenEnvSelection}(\mathcal{R}_g, \mathcal{S}^*)$
- 21: $g = g + 1$
- 22: **end while**

Algorithm 4 Ensemble-based Parent Selection Operator

Input: $\mathcal{S}^* = \{s_1^*, s_2^*, \dots, s_T^*\}$, an ensemble of T indicator-based selection operators
Input: \mathcal{P} , a population of μ individuals
Output: an individual to be used as a parent for crossover

- 1: Each of T operators selects one individual from the population \mathcal{P} with a v -way tournament. In total, T individuals are selected: $\{x_1, x_2, \dots, x_T\}$
- 2: Calculate the number of votes for each individual x_i as $\varphi_i = \sum_{t=1}^T I_{ti}$
 where $I_{ti} = \begin{cases} 1 & \text{if } s_t^* \text{ selects } x_i \\ 0 & \text{otherwise} \end{cases}$
- 3: **return** $x^* = \text{argmax}_{x_i} \varphi_i$.

5.1 Ensemble-based Parent Selection

Algorithm 4 shows how the proposed ensemble-based parent selection works (c.f. `ensembleBasedParentSelection()` in Algorithm 3). It takes an ensemble of T indicator-based selection operators \mathcal{S}^* , which Algorithm 1 produces.

Each of T operators first selects one individual (i.e., parent candidate) from the population \mathcal{P} with a v -way tournament (Line 1). In a v -way tournament, a selection operator randomly draws v individuals from \mathcal{P} and chooses the best one based on a quality indicator that the operator uses. The count of votes φ_i ($1 \leq i \leq T$) is assigned to each of selected T individuals (Line 2). Finally, the proposed ensemble-based parent selection operator chooses an individual with the highest vote count as a parent. If more

than one individuals have the same highest vote number, one of them is randomly chosen.

5.2 Ensemble-driven Environmental Selection

Algorithm 5 shows how the proposed ensemble-driven environmental selection (`ensembleDrivenEnvSelection()` in Algorithm 3) works. It takes an ensemble of T indicator-based selection operators S^* , which Algorithm 1 produces. It first identified an operator that Algorithm 3 selected in its first loop iteration (s^+) (Line 1). Then, with the indicator (I^+) that s^+ uses, μ individuals are removed from 2μ individuals in \mathcal{R}_g (Lines 2 to 6). Finally, the remaining μ individuals are selected as the next-generation population (\mathcal{P}_{g+1}) (Line 8).

Algorithm 5 Boosting-driven Environmental Selection Operator

Input: \mathcal{R}_g , a population of 2μ individuals

Input: $S^* = \{s_1^*, s_2^*, \dots, s_T^*\}$, an ensemble of T indicator-based selection operators

Output: \mathcal{P}_{g+1} , a population of individuals to be used in the next generation

```

1:  $s^+ = s_1^*$ 
2: while  $|\mathcal{R}_g| > \mu$  do
3:   Rank all the individuals in  $\mathcal{R}_g$  with  $I^+$ , which is the
   indicator that  $s^+$  uses.
4:   Select an individual  $d$  that has the worst  $I^+$  value in
    $\mathcal{R}_g$ 
5:    $\mathcal{R}_g = \mathcal{R}_g \setminus \{d\}$ 
6: end while
7:  $\mathcal{P}_{g+1} = \mathcal{R}_g$ 
8: return  $\mathcal{P}_{g+1}$ 

```

Although environmental selection depends on what the proposed boosting method (Algorithm 3) produces, it does not use all T operators in S^* as parent selection does. It uses only one operator (s_1^*) in S^* because it intends to minimize the degree of randomization in environmental selection.

6. EXPERIMENTAL EVALUATION

This section evaluates BIBEA-P as well as the proposed boosting method. Experiments were configured as shown in Table 6 and conducted with jMetal [8] on a 3.4GHz 64bit Core i7 processor.

Table 2: Experimental Configurations

Parameter	Value
M (Algorithm 1)	19
N_0 (Algorithm 1)	300
μ (Algorithms 1, 3 and 5)	100
N_p (Algorithm 1)	20
θ (Algorithm 1)	0.87
σ (Algorithm 2)	0.05
g_{max} (Algorithm 3)	150, 200 or 500
v (Algorithm 4)	2, 3, 4 or 5
Crossover rate	0.9
Mutation rate	$1/(\# \text{ of decision variables})$

This evaluation study uses ZDT1 as a training problem for the proposed boosting method. ZDT1 is the simplest problem in the ZDT family problems [20]. It has a convex Pareto-optimal front in a two dimensional objective space.

ZDT2, ZDT3, ZDT4 and ZDT6 are used as test problems. Each of them has a two dimensional objective space. ZDT2 and ZDT6 are essentially same as ZDT1 in terms of problem design and complexity; however, they have concave Pareto-optimal fronts. ZDT3 and ZDT4 are harder problems than ZDT1. ZDT3 has five discontinuous Pareto-optimal fronts. ZDT4 possesses a large number of (20^9) local optima.

DTLZ family problems, DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7 [7], are also used as test problems. All of them are harder problems than ZDT1 (i.e., the training problem in this evaluation study). They have three dimensional objective spaces. DTLZ1 has a continuous and planar Pareto-optimal front. DTLZ2, DTLZ3 and DTLZ4 have continuous and spherical Pareto-optimal fronts. DTLZ7 has four discontinuous Pareto-optimal fronts.

All ZDT and DTLZ problems are configured with the default settings of jMetal. Every experimental result is obtained with 20 independent experiments for ZDT problems and 10 independent experiments for DTLZ problems.

6.1 Evaluation Metrics

This paper uses two evaluation metrics: hypervolume ratio (HVR) and coverage metric (\mathcal{C} -metric). HVR is calculated as the ratio of the hypervolume (HV) of non-dominated individuals (D) to the hypervolume of Pareto-optimal solutions (P^*) [17].

$$HVR(D) = \frac{HV(D)}{HV(P^*)} \quad (11)$$

HVR quantifies the optimality and diversity of non-dominated individuals D . A higher HVR indicates that non-dominated individuals are closer to the Pareto-optimal front and more diverse in the objective space.

\mathcal{C} -metric compares two sets of non-dominated individuals [23]. Given non-dominated individual sets A and B , $\mathcal{C}(A, B)$ measures the fraction of individuals in B that at least one individual in A dominates:

$$\mathcal{C}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \succ b\}|}{|B|} \quad (12)$$

6.2 Boosting Overhead

The left half of Table 3 shows a set of indicators that the proposed boosting method chose from 19 indicators with Pdi-Boosting. It also depicts how many training populations were used to choose individual indicators. Note that the number of training populations is initially 300 ($N_0 = 300$ in Algorithm 1. See Table 6.) and it gradually grows. Therefore, the proposed boosting method uses different numbers of training populations for selecting different indicators. In total, it used 19,515 training populations.

The right half of Table 3 shows a set of indicators that an AdaBoost-based method chosen [15, 16]. In this method, the number of training populations is fixed. In total, it used 80,000 training populations. (10,000 per indicator was the minimum number of training populations for the AdaBoost method to obtain the best experimental results.)

The number of training populations impacts the overhead of a boosting algorithm (i.e., how long it takes for the algorithm to complete its boosting process.) The boosting overhead was 10 minutes in the proposed boosting method with Pdi-Boosting while it was 18 minutes with AdaBoost.

Thanks to Pdi-Boosting’s dynamic adjustment of the number of training populations, the proposed boosting method is 44% faster than an AdaBoost-based method.

Table 3: The Number of Training Populations used by Pdi-Boosting and AdaBoost

Indicator	# of training populations in Pdi-Boosting	Indicator	# of training populations in AdaBoost
$I_{\epsilon 2}$	300	$I_{\epsilon 2}$	10,000
I_{HD2}	450	I_{HD2}	10,000
I_{F2}	686	I_{HypE}	10,000
I_{HypE}	889	I_{W0}	10,000
I_{W0}	1,193	I_{W3}	10,000
I_{W3}	1,619	I_{W6}	10,000
I_{W6}	2,171	I_H	10,000
I_H	2,895	$I_{\epsilon 1}$	10,000
I_{HD1}	3,930		
I_{V1}	5,382		
Total	19,515	Total	80,000
Overhead	10 (mins)	Overhead	18 (mins)

6.3 Evaluation of Ensemble Selection

This section evaluates a boosted indicator that aggregates the ten indicators listed in Table 3 in terms of optimality and diversity. Table 4 shows the average HVR values that five algorithms yield at the last generation in 10 different test problems. The total number of generations in each experiment (g_{max} in Table 6) is 150 in ZDT problems, 500 in DTLZ3 and 200 in the other DTLZ problems. In Table 4, a number in parentheses indicates a standard deviation among different experiments. I_B represents BIBEA-P that uses an ensemble indicator aggregating the ten indicators listed in Table 3. (BIBEA-P uses $I_{\epsilon 2}$ for its environmental selection.) Each of the other four algorithms represents a variant of IBEA [21] that performs parent and environmental selection with an indicator listed in Table 3. For example, I_{HD2} represents a variant of IBEA that uses I_{HD2} for parent and environmental selection. v in Table 4 indicates the size of a tournament in parent selection. In each test problem, 2-way to 5-way tournament selections are examined. A bold number indicates the best result among five algorithms on a per-row basis. The results of other six selected indicators (i.e., I_{W0} , $I_{W0,}$, I_{W2} , I_{W6} , I_H , I_{HD1} and I_{VD1}) are not shown due to the space limitation Their results are significantly worse than the top four selected indicators.

In five of ten test problems (i.e., ZDT2, ZDT4, ZDT6, DTLZ1 and DTLZ3), I_B outperforms the other indicators when $v = 5$. In three other problems (i.e., ZDT1, ZDT3 and DTLZ2), I_B and $I_{\epsilon 2}$ tie when $v = 5$ if HVR values are truncated to two decimal places. Table 4 demonstrates that the proposed boosting method can work with a simple training problem (i.e., ZDT1) and I_B can effectively operate to solve harder test problems.

$I_{\epsilon 2}$ works well in ZDT1, ZDT3, DTLZ2, DTLZ4 and DTLZ7; however, its performance is marginal in DTLZ1 and DTLZ3. In both problems, $I_{\epsilon 2}$ never yields 0.19 or higher HVR. The other three (existing) indicators exhibit similar performance inconsistencies among different problems. For example, I_{F2} performs well in ZDT1 and ZDT2, while it performs poorly in DTLZ4. (It never yields 0.05 or higher HVR in DTLZ4.)

In contrast, I_B ’s HVR performance is much more consistent among different problems. Its worst HVR is 0.68 (DTLZ4) while $I_{\epsilon 2}$ ’s worst is 0.09 (DTLZ1), I_{HD2} ’s is 0.44 (ZDT4) and I_{F2} ’s is 0.01 (DTLZ4).

In summary, Table 4 demonstrates that I_B performs better than, or equally to, existing indicators in HVR (i.e., in optimality and diversity) in all test problems and I_B is more robust and stable than existing indicators under different characteristics in different problems.

6.4 Comparative Evaluation of BIBEA-P

Tables 5 shows the average HVR values that BIBEA-P and five other EMOAs (BIBEA, IBEA- $\epsilon 2$, IBEA-HD2, NSGAI and SMS-EMOA) yield at the last generation in 10 different test problems. BIBEA (Boosted Indicator Based Evolutionary Algorithm) is a variant of IBEA that uses an ensemble selection operator constructed with AdaBoost [16]. IBEA- $\epsilon 2$ and IBEA-HD2 are variants of IBEA that perform parent and environmental selection with $I_{\epsilon 2}$ and I_{HD2} , respectively [21]. NSGAI is a traditional EMOA that uses dominance ranking in its parent and environmental selection [5]. SMS-EMOA is a steady state EMOA that uses the hypervolume indicator in its environmental selection [2]. All algorithms use a 5-way tournament in parent selection.

In Tables 5, a number in parentheses indicates a standard deviation among different experiments. A bold number indicates the best result among six algorithms. A double star (**) or a single star (*) is placed when a HVR result is significantly different from BIBEA’s result based on a single-tail t -test. A double and a single stars are placed with the confidence level of 99% and 95%, respectively.

As shown in Table 5, BIBEA-P yields the best HVR in four of ten test problems (i.e., ZDT2, ZDT4, ZDT6, DTLZ3). BIBEA yields the best HVR in three problems (i.e., DTLZ1, DTLZ4 and DTLZ7). BIBEA-P outperforms BIBEA in four problems. Of one of the four problems (i.e., ZDT6), BIBEA-P outperforms BIBEA with the confidence level of 99%.

BIBEA-P outperforms IBEA- $\epsilon 2$ significantly in ZDT4, ZDT6, DTLZ1 and DTLZ3 with the 95% or higher confidence level. It significantly outperforms IBEA-HD2, NSGA-II and SMS-EMOA in seven, six and five problems, respectively. BIBEA-P significantly outperforms all the other five EMOAs in ZDT6. In ZDT3, DTLZ1 and DTLZ4, BIBEA-P’s average HVR is not the best among six EMOAs; however, it is not outperformed significantly by the others.

Table 6 compares BIBEA-P with the other five algorithms with \mathcal{C} -metric. The first two rows compare BIBEA-P and BIBEA. A bold font face is used to indicate a higher \mathcal{C} -metric value between $\mathcal{C}(\text{BIBEA-P}, \text{BIBEA})$ and $\mathcal{C}(\text{BIBEA}, \text{BIBEA-P})$. $\mathcal{C}(\text{BIBEA-P}, \text{BIBEA}) > \mathcal{C}(\text{BIBEA}, \text{BIBEA-P})$ in eight of 10 problems. This means that BIBEA-P outperforms BIBEA in those eight problems. In DTLZ1 and DTLZ3, BIBEA outperforms BIBEA-P; however, they yield very similar \mathcal{C} -metric values. The differences are 5.4% in DTLZ1 and 0.1% in DTLZ3.

Note that Table 6 shows BIBEA-P outperforms BIBEA in DTLZ4 and DTLZ7 while BIBEA outperforms BIBEA-P in the average HVR (Table 5). This conflict occurs due to high standard deviations in HVR (Tables 4 and 5).

BIBEA-P outperforms IBEA- $\epsilon 2$, IBEA-HD2 and SMS-EMOA in eight, nine and eight problems, respectively. Even when BIBEA-P is outperformed, the difference in \mathcal{C} -metric values is not significant.

Table 4: Comparison of I_B and Other Top Four Selected Indicators with HVR

	v	I_B	I_{e2}	I_{HD2}	I_{HypE}	I_{F2}
ZDT1	2	0.99038(0.00078)	0.99108 (0.00051)	0.99053(0.00065)	0.99019 (0.00081)	0.98815(0.00069)
	3	0.99114(0.00059)	0.99124 (0.0005)	0.99108 (0.00052)	0.9872 (0.00133)	0.9863(0.00069)
	4	0.99078(0.00054)	0.9914 (0.00044)	0.99131 (0.00061)	0.98373 (0.003)	0.98976 (0.0006)
	5	0.99085(0.00055)	0.9914 (0.00053)	0.99124 (0.00057)	0.98007 (0.00306)	0.99013(0.0007)
ZDT2	2	0.9683(0.02064)	0.96861 (0.01234)	0.67788 (0.26256)	0.97276 (0.00368)	0.94773(0.04736)
	3	0.97863 (0.00248)	0.97635 (0.00258)	0.80336 (0.22041)	0.96998 (0.00451)	0.96932(0.01206)
	4	0.97873 (0.00219)	0.97757 (0.00224)	0.88847 (0.17961)	0.95958 (0.01163)	0.97486(0.00272)
	5	0.97864 (0.00211)	0.97816 (0.00141)	0.85718 (0.22804)	0.94279 (0.02943)	0.97541(0.00569)
ZDT3	2	0.97186(0.03306)	0.98468 (0.01757)	0.97633 (0.02599)	0.98584 (0.01706)	0.82823(0.00073)
	3	0.98678(0.01748)	0.98923 (0.00069)	0.97772 (0.02378)	0.98843 (0.01018)	0.82899(0.0008)
	4	0.98481(0.01768)	0.98906 (0.00091)	0.98537 (0.00082)	0.98908 (0.00689)	0.82913(0.00052)
	5	0.98279(0.0251)	0.9892 (0.00125)	0.98569 (0.00098)	0.98277 (0.02218)	0.82927(0.0006)
ZDT4	2	0.89002(0.06255)	0.82759 (0.09139)	0.4487 (0.27688)	0.95183 (0.03788)	0.82026(0.10946)
	3	0.91844(0.0632)	0.89975 (0.06417)	0.61921 (0.19496)	0.94363 (0.06164)	0.87938(0.09757)
	4	0.96153 (0.01557)	0.92762 (0.03617)	0.52386 (0.25191)	0.94837 (0.03878)	0.93339(0.05858)
	5	0.96446 (0.01789)	0.93968 (0.03962)	0.59782 (0.23131)	0.93607 (0.07498)	0.89258(0.10842)
ZDT6	2	0.91442 (0.0142)	0.89694(0.01971)	0.86634 (0.02646)	0.90851 (0.01496)	0.90671(0.00992)
	3	0.95499 (0.00717)	0.93181 (0.00818)	0.93458 (0.01476)	0.94867 (0.00984)	0.9459(0.00824)
	4	0.97157 (0.0051)	0.94592 (0.01112)	0.95305 (0.01261)	0.96787 (0.00669)	0.96024(0.00577)
	5	0.979 (0.00362)	0.95166 (0.00771)	0.96303 (0.00709)	0.97581 (0.00445)	0.96884(0.00482)
DTLZ1	2	0.92785(0.01461)	0.16563 (0.16349)	0.87292 (0.1565)	0.93612 (0.00366)	0.72269(0.31834)
	3	0.9337(0.00307)	0.18356 (0.12278)	0.86789 (0.16297)	0.93523 (0.0037)	0.91004(0.03724)
	4	0.84968(0.2651)	0.13599 (0.16671)	0.82699 (0.15773)	0.93057 (0.00657)	0.91907(0.02753)
	5	0.93431 (0.00477)	0.09898 (0.08299)	0.85456 (0.14999)	0.90112 (0.09922)	0.86101(0.18988)
DTLZ2	2	0.8967(0.00127)	0.89671 (0.00165)	0.8817 (0.0013)	0.89665 (0.00171)	0.8713(0.003)
	3	0.89635(0.00135)	0.89722 (0.00055)	0.8815 (0.00075)	0.89561 (0.00174)	0.87548(0.0045)
	4	0.8961(0.00183)	0.89624 (0.00185)	0.88098 (0.00107)	0.86899 (0.01116)	0.8785(0.00315)
	5	0.89533(0.0014)	0.8967 (0.00087)	0.88071 (0.00115)	0.75846 (0.0847)	0.87945(0.00234)
DTLZ3	2	0.8751 (0.01467)	0.15156 (0.11169)	0.67875 (0.21244)	0.88077 (0.01817)	0.8496(0.02864)
	3	0.87677 (0.03506)	0.12678 (0.09993)	0.79458 (0.13355)	0.8793 (0.00867)	0.84667(0.03259)
	4	0.88044 (0.01828)	0.18072 (0.13898)	0.81021 (0.12525)	0.77406 (0.06568)	0.86444(0.01944)
	5	0.88113 (0.0124)	0.13781 (0.05099)	0.81664 (0.06968)	0.67406 (0.10538)	0.8524(0.02157)
DTLZ4	2	0.73641(0.34721)	0.88301 (0.20706)	0.55533 (0.3879)	0.88276 (0.20657)	0.04929(0.06082)
	3	0.73536(0.25797)	0.8827 (0.2069)	0.56097 (0.38541)	0.78462 (0.25305)	0.01464(0.03364)
	4	0.7354(0.41662)	0.88293 (0.20701)	0.61755 (0.39914)	0.83329 (0.23659)	0.00656(0.02075)
	5	0.68672(0.34287)	0.7853 (0.25371)	0.61014 (0.40272)	0.73481 (0.2574)	0.00211(0.00669)
DTLZ7	2	0.80116(0.08808)	0.90849 (0.04448)	0.74587 (0.17607)	0.80901 (0.09467)	0.84283(0.05248)
	3	0.80705(0.11318)	0.90558 (0.04829)	0.71182 (0.1039)	0.80016 (0.11071)	0.85851(0.05549)
	4	0.83449(0.08327)	0.92182 (0.00907)	0.70199 (0.16297)	0.80508 (0.10569)	0.84186(0.04963)
	5	0.82417(0.07712)	0.89312 (0.06185)	0.68389 (0.14252)	0.78243 (0.11719)	0.82051(0.07021)

BIBEA-P outperforms NSGA-II in all ten problems. In ZDT6, DTLZ2 and DTLZ7, no individuals of NSGA-II can not dominate BIBEA individuals. A very limited number of NSGA-II individuals can dominate BIBEA individuals; for example, 0.5% in ZDT1, ZDT2 and ZDT3, 0.4% in DTLZ3, and 0.3% in DTLZ4.

In summary, Tables 4 and 5 demonstrate that Pdi-Boosting is superior than AdaBoost in that it allows BIBEA-P to outperform BIBEA in optimality and diversity. They also illustrate that ensemble indicator based EMOAs outperform existing indicator-based EMOAs and dominance ranking based EMOAs.

7. CONCLUSIONS

This paper proposes and evaluates an ensemble learning method that constructs an ensemble of quality indicators with a novel boosting algorithm called Pdi-Boosting. The proposed method is efficient thanks to its dynamic adjustment of training data. It can work with a simple training problem, and an ensemble of indicators can effectively aid parent selection and environmental selection in order to solve harder problems. An ensemble of indicators outperforms existing individual indicators in optimality, diversity and robustness. The proposed ensemble-based evolutionary algorithm outperforms a well-known dominance-based EMOA and existing indicator-based EMOAs.

8. REFERENCES

- [1] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Computat.*, 19(1), 2011.
- [2] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.*, 181(3), 2007.
- [3] P. Boonma and J. Suzuki. Prospect indicator based evolutionary multiobjective optimization algorithm. In *Proc. IEEE Congress on Evol. Computat.*, 2011.
- [4] C. C. Coello. Evolutionary multi-objective optimization: Some current research trends and topics that remain to be explored. *Front. Computat. Sci. China*, 3(1), 2009.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol. Computat.*, 6(2), 2002.
- [6] K. Deb, K. Sindhya, and T. Okabe. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proc. ACM Int'l Genetic and Evolutionary Computation Conference*, 2007.
- [7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, R. Jain, and

Table 5: Comparison of BIBEA-P and Other Five EMOAs with HVR

	BIBEA-P	BIBEA	IBEA- ϵ 2	IBEA-HD2	NSGAI1	SMS-EMOA
ZDT1	0.99085(0.00055)	0.99124 (0.00073)	0.99140 (0.00053)**	0.99124 (0.00057)*	0.98490 (0.00137)**	0.99048(6.5E-4)*
ZDT2	0.97864 (0.00211)	0.97855 (0.00201)	0.97816 (0.00141)	0.85718 (0.22804)*	0.96948 (0.00330)**	0.91286(0.09076)*
ZDT3	0.98279(0.0251)	0.98618 (0.01717)	0.98920 (0.00125)	0.98569 (0.00098)	0.99284 (0.00202)	0.98882(0.01754)
ZDT4	0.96446 (0.01789)	0.96062(0.03139)	0.93968 (0.03962)*	0.59782 (0.23131)**	0.95103 (0.04873)	0.71216(0.23092)**
ZDT6	0.979 (0.00362)	0.97168 (0.00502)**	0.95166 (0.00771)**	0.96303 (0.00709)**	0.93456 (0.01069)**	0.85502(0.02427)**
DTLZ1	0.89533(0.0014)	0.93482 (0.00364)	0.09898 (0.08299)**	0.85456 (0.14999)	0.46676 (0.46002)**	0.93192(0.06896)
DTLZ2	0.89533(0.0014)	0.89625 (0.00135)	0.89670 (0.00087)	0.88071 (0.00115)**	0.80160 (0.01253)**	0.91111 (5.6E-4)**
DTLZ3	0.88113 (0.0124)	0.87064 (0.03315)	0.13781 (0.05099)**	0.81664 (0.06968)**	0.76162 (0.03454)**	0.88058(0.05345)
DTLZ4	0.68672(0.34287)	0.88277 (0.20642)	0.78530 (0.25371)	0.61014 (0.40272)	0.80650 (0.28387)	0.72147(0.37901)
DTLZ7	0.82417(0.07712)	0.91760 (0.00504)**	0.89312 (0.06185)*	0.68389 (0.14252)*	0.84875 (0.01250)	0.76956(0.01022)*

Table 6: Comparison of BIBEA-P and Other Five EMOAs with C-metric

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ7
C (BIBEA-P, BIBEA)	0.267	0.441	0.558	0.758	0.937	0.062	0.079	0.185	0.067	0.027
C (BIBEA, BIBEA-P)	0.199	0.114	0.165	0.729	0.137	0.120	0.018	0.186	0.030	0.011
C (BIBEA-P, IBEA- ϵ 2)	0.454	0.693	0.565	0.623	0.971	0.062	0.084	0.299	0.098	0.032
C (IBEA- ϵ 2, BIBEA-P)	0.170	0.060	0.154	0.860	0.0720	0.109	0.014	0.117	0.008	0.005
C (BIBEA-P, IBEA-HD2)	0.475	0.647	0.547	0.690	0.939	0.098	0.141	1.000	0.155	1.000
C (IBEA-HD2, BIBEA-P)	0.139	0.069	0.252	0.793	0.103	0.000	0.059	0.000	0.021	0.000
C (BIBEA-P, SMS-EMOA)	0.851	0.900	0.644	0.888	0.991	0.080	0.033	0.243	0.020	0.048
C (SMS-EMOA, BIBEA-P)	0.057	0.027	0.007	0.594	0.000	0.124	0.016	0.148	0.044	0.002
C (BIBEA-P, NSGAI1)	0.973	0.965	0.673	0.869	0.988	0.526	0.283	0.610	0.195	0.497
C (NSGAI1, BIBEA-P)	0.005	0.005	0.005	0.615	0.000	0.058	0.000	0.004	0.034	0.000

R. Goldberg, editors, *Evolutionary Multiobjective Optimization*. Springer, 2005.

- [8] J. Durillo, A. Nebro, and E. Alba. The jMetal framework for multi-objective optimization: Design and architecture. In *Proc. IEEE Congress on Evolutionary Computation*, 2010.
- [9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computat. Syst. Sci.*, 55, 1997.
- [10] M. Koppen, R. Vicente-Garcia, and B. Nickolay. Fuzzy-pareto dominance and its application in evolutionary multi-objective optimization. In *Proc. Int'l Conf. on Evol. Multi-criterion Optimization*, 2005.
- [11] K. Le and D. Landa-Silva. Obtaining better non-dominated sets using volume dominance. In *Proc. IEEE Congress on Evolutionary Computation*, 2007.
- [12] B. Liu, M. B., and A. H.A. Improving genetic classifiers with a boosting algorithm. In *Proc. IEEE Congress on Evol. Computat.*, 2003.
- [13] B. Liu, B. McKay, and H. A. Abbass. Feature selection combining genetic algorithm and adaboost classifiers. In *Proc. Int'l Conference on Pattern Recognition*, 2008.
- [14] B. Naujoks and H. Trautmann. Online convergence detection for multiobjective aerodynamic applications. In *Proc. IEEE Congress on Evol. Computat.*, 2009.
- [15] D. H. Phan and J. Suzuki. Boosting indicator-based selection operators for evolutionary multiobjective optimization algorithms. In *Proc. IEEE Int'l Conference on Tools with Artificial Intelligence*, 2011.
- [16] D. H. Phan, J. Suzuki, and I. Hayashi. BIBEA: Boosted indicator based evolutionary algorithm for multiobjective optimization. In *Proc. Asia Pacific Symposium of Intelligent and Evol. Systems*, 2011.
- [17] D. A. V. Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithm test suites. In *Proc. ACM Symposium on Applied Computing*, 1999.
- [18] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Proc. Int'l Conf. Evol. Multi-criterion Optimization*, 2007.
- [19] E. Zitzler, D. Brockho, and L. Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In *Proc. Int'l Conference on Evolutionary Multi-Criterion Optimization*, 2007.
- [20] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Computat.*, 8(2), 2000.
- [21] E. Zitzler and S. Kuenzli. Indicator-based selection in multiobjective search. In *Proc. Int'l Conference on Parallel Problem Solving from Nature*, 2004.
- [22] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: A comparative study. In *Proc. Int'l Conference on Parallel Problem Solving from Nature*, 1998.
- [23] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.*, 3(4), 1999.