BIBEA: Boosted Indicator Based Evolutionary Algorithm for Multiobjective Optimization

Dung H. Phan
Department of Computer Science
University of Massachusetts, Boston
Boston, MA 02125
phdung@cs.umb.edu

Junichi Suzuki
Department of Computer Science
University of Massachusetts, Boston
Boston, MA 02125
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. Many quality indicators have been proposed with the intention to capture different preferences in optimization. Therefore, indicator-based selection operators tend to have biased selection pressures that evolve solution candidates toward particular regions in the objective space. This paper investigates a boosting method that prioritizes and aggregates existing quality indicators to create a single indicator that outperforms those existing ones. 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 a boosted indicator can effectively operate in parent selection and environmental selection in order to solve harder problems. Experimental results show that a boosted selection operator outperforms exiting ones in optimality and diversity. It also exhibits robustness against different characteristics in different optimization problems and yields stable performance to solve them. Experimental results also demonstrate that the proposed boosted indicator based evolutionary algorithm (BIBEA) outperforms a well-known traditional EMOA and existing indicator-based evolutionary algorithms.**

***Index Terms*—Evolutionary multiobjective optimization algorithms, Quality indicators, Indicator-based selection, Boosting, Boosted selection**

I. Introduction

This paper proposes and evaluates new selection operators for evolutionary algorithms to solve multiobjective optimization problems (MOPs). In general, an MOP is formally described as follows.

$$\left.\begin{array}{r}\text{minimize } F(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \cdots, f_n(\vec{x})]^T \in \mathcal{O} \\ \text{subject to } \vec{x} = [x_1, x_2, \cdots, x_m]^T \in \mathcal{S}\end{array}\right\} \quad (1)$$

$\mathcal{S}$ denotes the decision variable space. $\vec{x} \in \mathcal{S}$ denotes a solution candidate that consists of m decision variables. It is called an *individual* in evolutionary multiobjective optimization algorithms (EMOAs). $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ consists of n real-value objective functions, which produce the objective values of $\vec{x}$ in the objective space $\mathcal{O}$. The goal of an EMOA is to find an individual(s) that minimize(s) objective values.

In MOPs, there rarely exists a single solution that is optimum with respect to all objectives because objective functions (i.e., $f_1(\vec{x}), \cdots, f_m(\vec{x})$ in Equation 1) conflict with each other. Thus, EMOAs seek the optimal trade-off individuals, or *Pareto-optimal* individuals, by considering the trade-offs among conflicting objectives. The notion of *dominance* plays an important role to seek Pareto optimality in MOPs [1]. An individual $\vec{x} \in \mathcal{S}$ is said to *dominate* another individual $\vec{y} \in \mathcal{S}$ (denoted by $\vec{x} \succ \vec{y}$) iif the both of the following conditions are hold.

- $f_i(\vec{x}) \leq f_i(\vec{y}) \ \forall \ i \ = 1, \cdots, n$
- $f_i(\vec{x}) < f_i(\vec{y}) \ \exists \ i \ = \ 1, \cdots, n$

EMOAs often rank individuals based on the dominance relationships among them and exploit their ranks in selection operators [1]. This process is called *dominance ranking*.

A research trend in the design space of EMOAs is to adopt *indicator-based selection* operators that augment or replace dominance ranking with *quality indicators* [2]. A quality indicator measures the goodness of each individual. Recent research findings (e.g., [3]) show that indicator-based EMOAs outperform traditional EMOAs that use dominance ranking.

Many quality indicators have been proposed with the intention to capture different preferences in optimization [4]–[7]. 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 [4]. An open question in this context is whether a set of existing indicators can create a single indicator that outperforms those existing ones.

In order to address this question, this paper investigates a boosting method that prioritizes and aggregates quality 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 a boosted indicator can effectively operate to solve harder problems.

Experimental results show that a boosted indicator outperforms exiting ones in terms of optimality and diversity of individuals in the population. The boosted indicator exhibits higher robustness than existing ones against different characteristics in different problems and yields more stable performance to solve a wider range of problems. Experimental results also demonstrate that the proposed boosted indicator based evolutionary algorithm (BIBEA) outperforms a well-known traditional EMOA (NSGA-II [8]) and existing indicator-based evolutionary algorithms.

II. Related Work

This paper extends the authors' prior work [9], which was the first attempt to boost quality indicators for parent selection in EMOAs. Compared with the prior work, this paper revises the proposed boosting method and evaluates it with more problems and more metrics. Moreover, this paper leverages the proposed boosting method for environmental selection as well as parent selection, while the prior work does for parent selection only.

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 [10], [11]. The Boosting Genetic Algorithm integrates boosting with a GA to discover classification rules [10]. 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 [11]. (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.

GPBoost [12] and its variants (e.g., [13]) integrate boosting with genetic programming (GP) to solve regression problems. A GP algorithm is used as a base regression solver in which each individual represents a regression solution candidate. A boosting algorithm aggregates multiple base regression solvers (i.e., GP algorithms) to build a more accurate regression solver than them.

Yalabik et al., Santana et al. and Augusto et al. propose evolutionary ensemble methods for classification problems [14]–[16]. Yalabik et al. investigate a GA that seeks the optimal permutation of base classifiers as the optimal ensemble classifier [14]. Each individual in the GA represents an ensemble classifier that aggregates a certain set of base classifiers. The fitness of an individual is computed based on the classification errors of base classifiers that the individual contains.

Santana et al. follow a similar design to encode an ensemble classifier as an individual and integrate it with a GA, an ant colony optimization algorithm and a particle swarm optimization algorithm for feature selection [15].

Augusto et al. use multiple instances of a GA that evolves individuals, each of which represents a base classifier [16]. Different GA instances evolve individuals in parallel with different training data sets, and they periodically exchange well-performing individuals. After their evolution process, they aggregate their best-performing individuals as an ensemble classifier with a boosting-like aggregation strategy.

He et al. propose a method to examine the performance of EMOAs with an ensemble of evaluation metrics (or quality indicators) such as hypervolume ratio, generational distance and spacing [17]. The method is designed with double elimination tournaments. In order to determine which EMOAs outperform which EMOAs, different sets of non-dominated individuals from different EMOAs are evaluated through those tournaments with multiple evaluation metrics.

III. Quality Indicators

This section describes 15 representative quality indicators that the proposed boosting method uses.

A. Hypervolume Indicator (I_H)

I_H measures the volume of a hypercube that an individual dominates in the objective space [18]. The hypercube is formed with the individual and the reference point representing the highest (or worst) possible objective values $\vec{r} = (r_1, r_2, .., r_n)$ 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 $\vec{x}$.

$$I_H(\vec{x}) = \prod_{i=1}^{n} |r_i - f_i(\vec{x})| \tag{2}$$

I_H is intended to favor balanced individuals in objective space rather than extreme ones [18].

B. Weighted Hypervolume Indicator (I_{W1} to I_{W9})

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 [4]. I_W of an individual $\vec{x} = (x_1, x_2, ..., x_n)$ is computed as follows.

$$\begin{aligned} I_W(\vec{x}) &= \int_{(x_1,x_2,...,x_n)}^{(r_1,r_2,...,r_n)} w(\vec{a})dz \qquad (3) \\ &where\ w(\vec{a}) = \frac{\sum_{i=1}^{n} e^{k_i(r_i - a_i)}}{\sum_{i=1}^{n} e^{k_i}} \end{aligned}$$

$w(\vec{a})$ denotes the weight of a point $\vec{a} = (a_1, a_2, ..., a_n)$ in the objective space. It is calculated by applying a weight distribution $\vec{k} = (k_1, k_2, .., k_n)$. 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 $\vec{k} = (0, 0, .., 0)$.

As shown in Table I, this paper considers nine variants of I_W (I_{W1} to I_{W9}) based on nine different combinations of k_1 and k_2 values. These value combinations are determined based on the parameter settings in [4]. Note that this paper uses a training problem whose objective space is two dimensional.

TABLE I: 9 Variants of the Weighted Hypervolume Indicator

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			

C. HypE Indicator (I_{HypE})

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 [5].

$$I_{HypE}(\vec{x}) = \sum_{i=1}^{\mu} \frac{1}{i} H_i(\vec{x}) \tag{4}$$

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

D. Binary ε Indicator ($I_{\varepsilon 1}$ and $I_{\varepsilon 2}$)

I_ε takes two individuals ($\vec{x}$ and $\vec{y}$) and measures the distance between them on a per-objective basis. It is computed as follows [6].

$$I_\epsilon(\vec{x}, \vec{y}) = max_{i \in \{1,..,n\}}(f_i(\vec{x}) - f_i(\vec{y})) \tag{5}$$

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 binary indicator values.

$$I_{\epsilon 1}(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} I_\epsilon(\vec{y}, \vec{x}) \tag{6}$$

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

$$I_{\epsilon 2}(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} -e^{-I_\epsilon(\vec{y}, \vec{x})/l} \tag{7}$$

l is a scaling coefficient. $l = 0.05$ in this paper, which is a recommended value in [6].

E. Binary Hypervolume Indicator (I_{HD1} and I_{HD2})

I_{HD} takes two individuals ($\vec{x}$ and $\vec{y}$) and measures the hypervolume dominated by $\vec{x}$ but not by $\vec{y}$ [6].

$$I_{HD}(\vec{x}, \vec{y}) = \begin{cases} H(\vec{x}) - H(\vec{y}) & \text{if } \vec{x} \text{ dominates } \vec{y} \\ H(\vec{x}) - H(\vec{x}) \cap H(\vec{y}) & \text{otherwise} \end{cases} \tag{8}$$

$H(\vec{x})$ 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 $I_{HD2}(\vec{y})$ are computed by replacing $I_{\epsilon+}(\vec{y}, \vec{x})$ with $I_{HD}(\vec{x}, \vec{y})$ in Equations 6 and 7, respectively.

IV. The Proposed Boosting Method

Algorithm 1 shows the proposed boosting method, which employs the AdaBoost algorithm [19]. It takes M indicator-based parent selection operators S and aggregates top T operators S^* ($T \leq M$). This paper uses 15 tournament selection operators that use 15 indicators described in Section III ($M = 15$). T aggregated operators have their weights: $W^* = \{\alpha_1, \alpha_2, ..., \alpha_T\}$.

The proposed boosting method is carried out through an offline training with a multiobjective optimization problem in which Pareto-optimal solutions are known. This training problem is used to generate N training populations, $\{p_1, p_2, .., p_N\}$, each of which contains μ individuals (Line 2). These individuals represent randomly-chosen points in the region that Pareto-optimal solutions dominate in the objective space. Each training population has a weight w_i ($1 \leq i \leq N$). Its initial value is $1/N$ (Line 3).

The proposed boosting method iteratively executes a loop (Line 4 to 15) T times and selects one operator into S^* in each iteration. (It selects T operators into S^* through T iterations.) 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 [20]. HVR is computed as the ratio of the hypervolume (HV) of N_p individuals (D) to the hypervolume of Pareto-optimal solutions in a training problem (P^*).

$$HVR(D) = \frac{HV(D)}{HV(P^*)} \tag{9}$$

HV measures the union of the volumes that a given set of individuals dominate in the objective space [18].

The selection of N_p individuals is said to be *successful* if $HVR(D)$ is greater than or equal to a threshold: θ ($\theta < 1$). Given this condition, the selection error of each operator is calculated as shown in Line 7. The error is adjusted with each training population's weight w_i ($1 \leq i \leq N$). Then, the proposed boosting method chooses the operator s_t^* that has the lowest selection error (Lines 8 and 9), and computes the operator's weight (Lines 10, 11 and 12). A lower selection error contributes to a higher weight.

Finally, each training population's weight is adjusted as shown in Lines 13 and 14. The weight decreases if s_t^*'s individual selection is successful; otherwise, it increases. This way, in subsequent loop iterations, the proposed boosting method places higher priorities on the training populations in which individual selection failed and favors the operators that perform successful individual selection on those populations (c.f. Line 7).

Algorithm 1 The Proposed Boosting Method

```
Input: $S$ = $\{s_1, s_2, .., s_M\}$, a set of $M$ operators
Output: $S^*$ = $\{s_1^*, s_2^*, .., s_T^*\}$, a set of $T$ aggregated operators
Output: $W^* = \{\alpha_1, \alpha_2, ..., \alpha_T\}$, Weights of $T$ aggregated operators
 1: $S^* = \phi$, $W^* = \phi$
 2: Generate $N$ training populations: $\{p_1, p_2, .., p_N\}$
 3: Initialize each training population's weight: $w_i(1) = 1/N, 1 \leq i \leq N$
 4: for $t = 1$ to $T$ do
 5:     Each operator $s_j$ performs individual selection $N_p$ times on each training population $p_i$.
 6:     Calculate the weighted selection error ($e_j$) for $s_j$
 7:         $e_j = \sum_{i=1}^{N} w_i I_{ji}$ where
            $I_{ji} = \begin{cases} 0 & \text{if } s_j\text{'s selection is successful on } p_i \\ 1 & \text{otherwise} \end{cases}$
 8:     Choose an operator $s_t^*$ such that $s_t \notin S^*$ and $s_t^*$=$\arg\min_{s_j \in S} e_j$
 9:     Add $s_t^*$ to $S^*$
10:     $e_t^*$ = the weighted selection error of $s_t^*$
11:     Calculate the weight ($\alpha_t$) of $s_t^*$ as $\alpha_t = \frac{1}{2} log \left( \frac{1-e_t^*}{e_t^*} \right)$ where $e_t^*$ denotes the weighted selection error of $s_t^*$
12:     Add $\alpha_t$ to $W^*$
13:     Adjust $w_i$ as $w_i(t+1)$ =
            $\begin{cases} w_i(t)e^{-\alpha_t} & \text{if } s_t\text{'s selection is successful} \\ w_i(t)e^{\alpha_t} & \text{otherwise} \end{cases}$
14:     Normalize $w_i(t+1)$ as $w_i(t+1) = \frac{w_i(t+1)}{\sum_{q=1}^{N} w_q(t+1)}$
15: end for
16: return $S^*$, $W^*$
```

V. Boosted Indicator Based Evolutionary Algorithm (BIBEA)

This section describes an EMOA, called Boosted Indicator Based Evolutionary Algorithm (BIBEA), which leverages a boosting method discussed in Section IV.

Algorithm 2 shows BIBEA's algorithmic structure, which is based on an existing indicator-based EMOA: IBEA [6]. In the *0*-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 a boosted parent selection operator that Algorithm 1 produces (`boostedParentSelection()`, Lines 6 and 7).

With the crossover rate P_c, two parents reproduce two offspring with the SBX (self-adaptive simulated binary crossover) operator [21] (Lines 9). Each offspring performs polynomial mutation [8] 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 $|O_g| = \mu$). The offspring ($\mathcal{O}_g$) are combined with the population $\mathcal{P}_g$ to form $\mathcal{R}_g$ ($|R_g| = 2\mu$), 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 $\mathcal{P}_{g+1}$ (`boostingDrivenEnvironmentalSelection()`, Line 20). Environmental selection performs a $(\mu + \mu)$-elitism.

Algorithm 2 The Algorithmic Structure of BIBEA

```
 1: $g = 0$
 2: $\mathcal{P}_g$ = initializePopulation($\mu$)
 3: while g < $g_{max}$ do
 4:     $\mathcal{O}_g = \emptyset$
 5:     while $|\mathcal{O}_g| < \mu$ do do
 6:         $p_1$ = boostedParentSelection($\mathcal{P}_g$)
 7:         $p_2$ = boostedParentSelection($\mathcal{P}_g$)
 8:         if random() $\leq P_c$ then
 9:             $\{o_1, o_2\}$ = crossover($p_1$, $p_2$)
10:             if random() $\leq P_m$ then
11:                 $o_1$ = mutation($o_1$)
12:             end if
13:             if random() $\leq P_m$ then
14:                 $o_2$ = 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:     $\mathcal{P}_{g+1}$ =boostingDrivenEnvironmentalSelection($\mathcal{R}_g$)
21:     $g = g + 1$
22: end while
```

A. Boosted Parent Selection

Algorithm 3 shows how the boosted parent selection operator works (c.f. `boostedParentSelection()` in Algorithm 2). It is constructed with T selection operators S^* and their weights W^*, which Algorithm 1 produces. Each of T operators first selects one individual (i.e., parent candidate) from the population P with a v-way tournament (Line 1). In a v-way tournament, a selection operator randomly draws v individuals from P and chooses the best one based on a quality indicator that the operator uses. A weight φ_i ($1 \leq i \leq T$) is assigned to each of selected T individuals with a prioritized voting by T operators (Line 2). Priorities are given to individuals based on the weights of operators ($\{\alpha_1, \alpha_2, ..., \alpha_T\}$). Finally, a boosted operator chooses one of T individuals as a parent by deriving individual selection probability δ_i from φ_i ($1 \leq i \leq T$) (Lines 3 and 4).

Algorithm 3 Boosted Parent Selection Operator

```
Input: $S^* = \{s_1^*, s_2^*, .., s_T^*\}$, $T$ aggregated operators
Input: $W^* = \{\alpha_1, \alpha_2, ..., \alpha_T\}$, Weights of $T$ aggregated operators
Input: $P$, a population of $\mu$ individuals
Output: an individual to be used as a parent for crossover
Each of $T$ operators selects one individual from the population $P$ with a $v$-way tournament. In total, $T$ individuals are selected: $\{x_1, x_2, ..., x_T\}$
Calculate the weight of each individual $x_i$ as
    $\varphi_i = \sum_{t=1}^{T} \alpha_t I_{ti}$
        where $I_{ti} = \begin{cases} 1 & \text{if } s_t \text{ selects } x_i \\ 0 & \text{otherwise} \end{cases}$
Calculate the selection probability of $x_i$ as
    $\delta_i = \frac{\varphi_i}{\sum_{i=1}^{T} \varphi_i}$
Select an individual from $\{x_1, x_2, ..., x_T\}$ based on $\delta_i$.
```

B. Boosting-driven Environmental Selection

Algorithm 4 shows how the environmental selection operator (`boostingDrivenEnvironmentalSelection()` in Algorithm 2) works. It uses S^* and W^*, which Algorithm 1 produces. It first identifies a selection operator (s^+) that has the highest weight value (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 to the next-generation population ($\mathcal{P}_{g+1}$) (Line 8).

Algorithm 4 Boosting-driven Environmental Selection Operator

```
Input: $\mathcal{R}_g, |\mathcal{R}_g| = 2\mu$
Input: $S^* = \{s_1^*, s_2^*, .., s_T^*\}$, $T$ aggregated operators
Input: $W^* = \{\alpha_1, \alpha_2, ..., \alpha_T\}$, Weights of $T$ aggregated operators
Output: $\mathcal{P}_{g+1}$, a population of individuals to be used in the next generation
1: $s^+ = \text{argmax}_{s_j \in S^*} \alpha_j$
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 produces, it does not use all T operators in S^* as parent selection does. It uses only one operator (s^+) in S^* because it intends to minimize the degree of randomization.

VI. Experimental Evaluation

This section evaluates BIBEA as well as the proposed boosting method. Experiments were configured as shown in Table II and conducted with jMetal [22] on a 3.4GHz 64bit Core i7 processor. Every experimental result is obtained with 20 independent experiments for two dimensional problems and 10 independent experiments for three dimensional problems.

TABLE II: Experimental Configurations

Parameter	Value
M (Algorithm 1)	15
T (Algorithms 1 and 3)	8
N (Algorithm 1)	10,000
μ (Algorithms 1, 2 and 4)	100
N_p (Algorithm 1)	20
θ (Algorithm 1)	0.8
g_{max} (Algorithm 2)	150, 200 or 500
v (Algorithm 3)	2, 3, 4 or 5
Crossover rate	0.9
Mutation rate	1/ (# of decision variables)

Table III shows the eight indicators that the proposed boosting method chosen from 15 indicators in order to construct a boosted selection opererator. Note that this evaluation study uses $M = 15$ and $T = 8$ in Algorithms 1 and 3.

TABLE III: Weight Values of Aggregated Indicators

Indicator	Weight (α)	Indicator	Weight (α)
$I_{\epsilon 2}$	0.876	I_{W3}	0.566
I_{HD2}	0.692	I_{W6}	0.536
I_{HypE}	0.632	I_H	0.440
I_{W0}	0.597	$I_{\epsilon 1}$	0.038

A. Training and Test Problems

This evaluation study uses ZDT1 as a training problem for the proposed boosting method. ZDT1 is the simplest problem in the ZDT family problems [23]. It has 30 decision variables[1] and a convex Pareto-optimal front in a two dimensional objective space (Figure 1).

ZDT2, ZDT3, ZDT4 and ZDT6 are used as test problems to evaluate the proposed boosting method and BIBEA. Each of the problems has a two dimensional objective space. ZDT2 and ZDT3 have 30 decision variables each[1]. ZDT4 has 10 decision variables[1]. ZDT6 has 10 decision variables[1]. ZDT2 and ZDT6 are essentially same as ZDT1 in terms of problem design and complexity; however, they have concave Pareto-optimal fronts (Figure 1). ZDT3 and ZDT4 are harder problems than ZDT1. ZDT3 has five discontiguous Pareto-optimal fronts (Figure 1). ZDT4 is a multi-modal problem that has a large number of (20^9) local optima. Its Pareto-optimal front is similar to ZDT1's.

DTLZ family problems, DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7 [24], are also used as test problems. They have 7, 12, 12, 12 and 22 decision variables, respectively[1]. All of them are harder problems than ZDT1. They have three dimensional

[1] It is the default setting in jMetal.

objective spaces (Figure 1). DTLZ1 has a continuous and planar Pareto-optimal front. DTLZ2, DTLZ3 and DTLZ4 have continuous and spherical Pareto-optimal fronts. DTLZ7 has four discontiguous Pareto-optimal fronts.

B. Evaluation Metrics

This paper uses three evaluation metrics: hypervolume ratio (HVR), inverted generational distance (IGD) 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^*) [20].

$$HVR(D) = \frac{HV(D)}{HV(P^*)} \tag{10}$$

HV measures the union of the volumes that non-dominated individuals dominate [18]. Thus, 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.

IGD is computed as follows where $d(v_i, D)$ is the minimum distance from a Pareto-optimal solution v_i to D [25].

$$IGD(D) = \sum_{i=1}^{|P^*|} \frac{d(v_i, D)}{|P^*|} \tag{11}$$

$|P^*|$ denotes the number of Pareto-otimal solutions. IGD measures the optimality and diversity (more specifically, extent) of non-dominated individuals D. A lower IGD indicates that non-dominated individuals are closer to the Pareto-optimal front and their extent is wider.

For both HVR and IGD, P^* are taken uniformly from the Pareto-optimal front. $|P^*|$ = 1,001, 1,001, 269, 1,001, 1,001, 10,000, 10,000, 4,000, 4,000 and 676 in ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7[1].

$\mathcal{C}$-metric compares two sets of non-dominated individuals [26]. 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:

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

If $\mathcal{C}(A, B) = 1$, all of B's individuals are dominated by at least one individual of A. If $\mathcal{C}(B, A) = 0$, no individuals in B are dominated by individuals in A. $\mathcal{C}(A, B) > \mathcal{C}(B, A)$ indicates that A contains better individuals than B.

C. Evaluation of the Proposed Boosting Method with HVR and IGD

This section evaluates a boosted indicator that aggregates the eight indicators listed in Table III in terms of optimality and diversity. Table IV shows the average HVR values that nine algorithms yield at the last generation in 10 different test problems. The total number of generations in each experiment (g_{max} in Table II) is 150 in ZDT problems, 500 in DTLZ3 and 200 in the other DTLZ problems. In Table IV, a number in parentheses indicates a standard deviation among different experiments. I_B represents BIBEA that uses a boosted indicator aggregating the eight indicators listed in Table III. (BIBEA uses $I_{\epsilon 2}$ for its environmental selection.) Each of the other eight algorithms represents a variant of IBEA [6] that performs parent and environmental selection with an indicator listed in Table III. For example, I_{HD2} represents a variant of IBEA that uses I_{HD2} for parent and environmental selection. v in Table IV 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 nine algorithms on a per-row basis.

In ZDT2, ZDT4, ZDT6, DTLZ1, DTLZ3, DTLZ4 and DTLZ7, I_B outperforms the other indicators when $v = 5$. In ZDT1, ZDT3 and DTLZ2, I_B and $I_\epsilon 2$ tie when $v = 5$ if HVR values are truncated to two decimal places. Table IV 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 problems. As described in Section VI-A, many of test problems are harder problems than ZDT1.

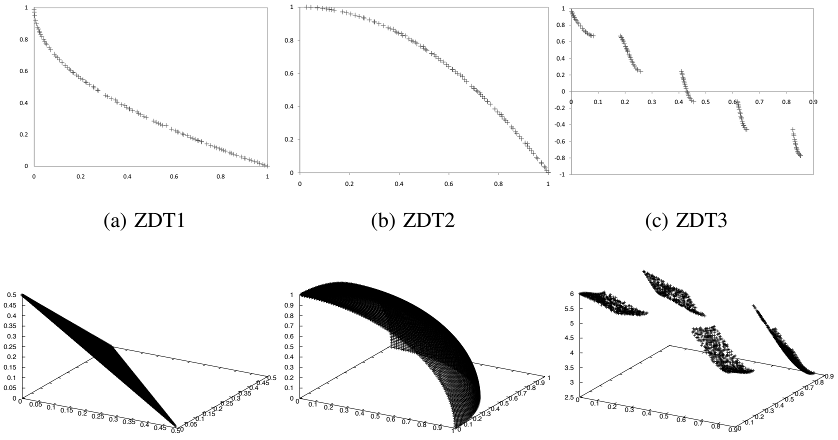
$I_{\epsilon 2}$ works well in ZDT1, ZDT3 and DTLZ7; however, its performance is marginal in DTLZ1 and DTLZ3. In DTLZ1, $I_{\epsilon 2}$ never yields 0.2 or higher HVR. The other seven (existing) indicators exhibit similar inconsistencies among different problems. For example, I_{HD2} performs well in ZDT1 and ZDT3 but performs poorly in ZDT4, DTLZ3 and DTLZ4. (It never yields 0.62 or higher HVR in ZDT4.)

In contrast, I_B's HVR performance is much more consistent among different problems. Its worst HVR is 0.84 (DTLZ7) while $I_{\epsilon 2}$'s worst is 0.09 (DTLZ1), I_{HD2}'s is 0.44 (ZDT4) and I_{HypE}'s is 0.67 (DTLZ3). The worst HVR of I_{W0}, I_{W3}, I_{W6}, I_H and $I_{\epsilon 1}$ is 0. This shows that I_B allows different indicators to complement with each other well.

In summary, Table IV 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.

Table V shows the average IGD values that nine algorithms yield at the last generation in 10 different test problems. In ZDT4, DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7, I_B outperforms the other indicators when $v = 5$. In ZDT 1 and ZDT3, I_B and $I_\epsilon 2$ tie when $v = 5$ if HVR values are truncated to four decimal places. In ZDT6, I_B and I_{HD2} tie when $v = 5$ if HVR values are truncated to four decimal places. Similar to the observation in Table IV, Table V 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 problems.

In Table V, I_B's IGD performance is more consistent than the other indicators among different test problems. Its worst IGD is 0.01 (DTLZ7) while $I_{\epsilon 2}$'s worst is 0.01 (DTLZ3), I_{HD2}'s is 0.01 (ZDT4 and DTLZ7), I_{HypE}'s is 0.01 (DTLZ7), I_{W0}'s is 0.84 (DTLZ1), I_{W3}'s 0.65 is (DTLZ1), I_{W6}'s is 0.88 (DTLZ1), I_H's is 0.04 (ZDT3), and $I_{\epsilon 1}$'s is 0.01 (ZDT2, ZDT3



(a) ZDT1

(b) ZDT2

(c) ZDT3

(d) DTLZ1

(e) DTLZ3

(f) DTLZ7

Fig. 1: The Pareto Front Shapes of ZDT1, ZDT2, ZDT3, DTLZ1, DTLZ3 and DTLZ7

and DTLZ7). This shows that I_B allows different indicators to complement with each other well.

In summary, similar to the observation in Table IV, Table IV demonstrates that I_B performs better than, or equally to, existing indicators in IGD (i.e., in optimality and diversity/extent) in all test problems and I_B is more robust and stable than existing indicators under different characteristics in different problems.

D. Evaluation of BIBEA with HVR and IGD

Tables VI and VII show the average HVR and IGD values, respectively, which BIBEA and four other EMOAs (IBEA-$\epsilon 2$, IBEA-HD2, NSGAII and SMS-EMOA) yield at the last generation in 10 different test problems. IBEA-$\epsilon 2$ is a variant of IBEA that performs parent and environmental selection with $I_{\epsilon 2}$, and IBEA-HD2 is a variant of IBEA that performs parent and environmental selection with I_{HD2} [6]. NSGAII is a traditional EMOA that uses dominance ranking in its parent and environmental selection [8]. SMS-EMOA is a steady state EMOA which uses the hypervolume indicator in its environmental selection [27]. It is one of the most successfully and widely used indicator-based EMOAs [3], [28]

All algorithms perform 5-way tournament in parent selection. In Tables VI and VII, a number in parentheses indicates a standard deviation among different experiments. A bold number indicates the best result among five algorithms on a per-row basis. A double star (**) or a single star (*) is placed for an average HVR/IGD result when the result is significantly different from BIBEA's result based on a single-tail t-test. A double star is placed with the confidence level of 99% while a single star is placed with the confidence level of 95%.

As shown in Table VI, BIBEA yields the best average HVR among five algorithms in all problems except for ZDT1, ZDT3, DTLZ2. BIBEA outperforms IBEA-$\epsilon 2$ in ZDT6, DTLZ1 and DTLZ3 with the confidence level of 99%. It significantly outperforms IBEA-HD2 in six problems. It significantly outperforms NSGA-II in eight problems. It significantly outperforms SMS-EMOA in ZDT1, ZDT2, ZDT4, ZDT6 and DTLZ7. BIBEA significantly outperforms all the other five algorithms in ZDT6. In ZDT1, ZDT3 and DTLZ2, BIBEA's average HVR is not the best among four algorithms; however, it is not outperformed by the other algorithms with the significance level of 95%.

BIBEA's IGD performance is not as good as its HVR performance. In ZDT6, BIBEA significantly outperforms the other five algorithms. NSGA-II yields the best IGD in seven problems.

It is noticeable that several or all algorithms yield high standard deviation in HVR and IGD (Tables VI and VII). For example, in Table VI, all five algorithms yield high HVR standard deviation in DTLZ4. In Table VII, BIBEA, IBEA-$\epsilon 2$ and NSGA-II yield high IGD standard deviation in ZDT4. Therefore, this paper uses boxplots to analyze the distributions of HVR and IGD values in the problems in which standard deviations are high (Figure 2).

A box in each boxplot contains the middle 50% (or the interquartile range; IQR) of individuals. The upper edge of the box indicates the 75th percentile (the upper quartile) of individuals, and the lower edge indicates the 25th percentile (the lower quantile). The middle horizontal line in the box indicates the 50th percentile (the median). The ends of a vertical line, or a *whisker*, indicate the maximum and minimum individuals unless outlier individuals exist. An individual is said to be an outliner when it is not between 1.5 IQR of the lower quartile and 1.5 IQR of the upper quartile. Outlier individuals are shown as points outside the ends of a whisker. Each boxplot is drawn with the individuals that an algorithm produces at the last generation in each of experiments.

Figures 2a to 2h show the distributions of HVR and IGD values that four algorithms yield in ZDT4, DTLZ1, DTLZ4 and DTLZ7. BIBEA yields the best or second best HVR/IGD median values in all of these problems. It also maintains

TABLE IV: Comparison of I_B and Other Eight Indicators with HVR

	v	I_B	$I_{\varepsilon 2}$	I_{HD2}	I_{HypE}	I_{W0}	I_{W3}	I_{W6}	I_H	$I_{\varepsilon 1}$
ZDT1	2	0.99076 (0.00059)	**0.99108** (0.00051)	0.99053(0.00065)	0.99019 (0.00081)	0.0112 (0.01381)	0.00882 (0.00762)	0.02979 (0.05572)	0.02286 (0.03741)	0.97119 (0.00232)
	3	0.99111 (0.00057)	**0.99124** (0.0005)	0.99108 (0.00052)	0.9872 (0.00133)	0.014 (0.01154)	0.02446 (0.04202)	0.00934 (0.00827)	0.00914 (0.00604)	0.97048 (0.00311)
	4	0.99125 (0.00048)	**0.9914** (0.00044)	0.99131 (0.00061)	0.98373 (0.003)	0.03333 (0.06476)	0.01908 (0.04225)	0.02241 (0.05115)	0.02021 (0.02265)	0.96996 (0.00293)
	5	0.99124 (0.00073)	**0.9914** (0.00053)	0.99124 (0.00057)	0.98007 (0.00306)	0.01118 (0.01533)	0.01331 (0.02165)	0.02871 (0.07231)	0.01737 (0.02354)	0.96703 (0.00524)
ZDT2	2	0.96048 (0.05558)	**0.96861** (0.01234)	0.67788 (0.26256)	0.97276 (0.00368)	0(0)	0 (0)	0 (0)	0 (0)	0.00191 (0.00124)
	3	**0.97732** (0.00241)	0.97635 (0.00258)	0.80336 (0.22041)	0.96998 (0.00451)	0(0)	0 (0)	0 (0)	0 (0)	0.00119 (0.00124)
	4	**0.97852** (0.00167)	0.97757 (0.00224)	0.88847 (0.17961)	0.95958 (0.01163)	0(0)	0 (0)	0 (0)	0 (0)	0.00104 (0.00107)
	5	**0.97855** (0.00201)	0.97816 (0.00141)	0.85718 (0.22804)	0.94279 (0.02943)	0(0)	0 (0)	0 (0)	0 (0)	0.00249 (0.00369)
ZDT3	2	0.97979 (0.02658)	**0.98468** (0.01757)	0.97633 (0.02599)	0.98584 (0.01706)	0.00762 (0.01091)	0.01899 (0.03018)	0.03113 (0.07114)	0.01086 (0.01417)	0.58206 (0.17689)
	3	0.98922 (0.00231)	**0.98923** (0.00069)	0.97772 (0.02378)	0.98843 (0.01018)	0.02199 (0.03572)	0.03807 (0.07873)	0.00869 (0.01302)	0.007 (0.00644)	0.58693 (0.15734)
	4	0.98245 (0.02488)	**0.98906** (0.00091)	0.98537 (0.00082)	0.98908 (0.00689)	0.01329 (0.01952)	0.01115 (0.02243)	0.0203 (0.04805)	0.01601 (0.04528)	0.59774 (0.14276)
	5	0.98618 (0.01717)	**0.9892** (0.00125)	0.98569 (0.00098)	0.98277 (0.02218)	0.01554 (0.03695)	0.01937 (0.04578)	0.0077 (0.00841)	0.00698 (0.0082)	0.49046 (0.1915)
ZDT4	2	0.89967 (0.03925)	0.82759 (0.09139)	0.4487 (0.27688)	**0.95183** (0.03788)	0 (0)	0 (0)	0 (0)	0 (0)	0.90873 (0.06488)
	3	**0.94945** (0.02699)	0.89975 (0.06417)	0.61921 (0.19496)	0.94363 (0.06164)	0 (0)	0 (0)	0 (0)	0 (0)	0.91156 (0.07297)
	4	**0.97071** (0.01868)	0.92762 (0.03617)	0.52386 (0.25191)	0.94837 (0.03878)	0 (0)	0 (0)	0 (0)	0 (0)	0.94117 (0.01884)
	5	**0.96062** (0.03139)	0.93968 (0.03962)	0.59782 (0.23131)	0.93607 (0.07498)	0 (0)	0 (0)	0 (0)	0 (0)	0.93024 (0.0595)
ZDT6	2	0.89221 (0.02391)	**0.89694** (0.01971)	0.86634 (0.02646)	0.90851 (0.01496)	0.01934 (0.0593)	0 (0)	0.00079 (0.00311)	0.01035 (0.04207)	0.00646 (0.00409)
	3	**0.93746** (0.01184)	0.93181 (0.00818)	0.93458 (0.01476)	0.94867 (0.00984)	0.00026 (0.00115)	0.02046 (0.07058)	0 (0)	0.01969 (0.08804)	0.00225 (0.00221)
	4	**0.96077** (0.00623)	0.94592 (0.01112)	0.95305 (0.01261)	0.96787 (0.00669)	0 (0)	0.01711 (0.07653)	0.01997 (0.08929)	0.0006 (0.00183)	0.00076 (0.00093)
	5	**0.97168** (0.00502)	0.95166 (0.00771)	0.96303 (0.00709)	0.97581 (0.00445)	0.00469 (0.02098)	0.01111 (0.04691)	0.00012 (0.00038)	0.00004 (0.0002)	0.00041 (0.00112)
DTLZ1	2	0.93498 (0.00348)	0.16563 (0.16349)	0.87292 (0.1565)	**0.93612** (0.00366)	0 (0)	0 (0)	0 (0)	0 (0)	0.00067 (0.00213)
	3	0.87669 (0.19738)	0.18356 (0.12278)	0.86789 (0.16297)	**0.93523** (0.0037)	0 (0)	0 (0)	0 (0)	0 (0)	0.01453 (0.02837)
	4	**0.93419** (0.0039)	0.13599 (0.16671)	0.82699 (0.15773)	0.93057 (0.00657)	0 (0)	0 (0)	0 (0)	0 (0)	0.01371 (0.03418)
	5	**0.93482** (0.00364)	0.09898 (0.08299)	0.85456 (0.14999)	0.90112 (0.09922)	0 (0)	0 (0)	0 (0)	0 (0)	0.00233 (0.00498)
DTLZ2	2	**0.89693** (0.00082)	0.89671 (0.00165)	0.8817 (0.0013)	0.89665 (0.00171)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	3	0.89708 (0.0014)	**0.89722** (0.00055)	0.8815 (0.00075)	0.89561 (0.00174)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	4	**0.89699** (0.00201)	0.89624 (0.00185)	0.88098 (0.00107)	0.86899 (0.01116)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	5	0.89625 (0.00135)	**0.8967** (0.00087)	0.88071 (0.00115)	0.75846 (0.0847)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
DTLZ3	2	**0.87846** (0.01391)	0.15156 (0.11169)	0.67875 (0.21244)	0.88077 (0.01817)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	3	**0.88482** (0.01005)	0.12678 (0.09993)	0.79458 (0.13355)	0.8793 (0.00867)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	4	**0.8753** (0.02285)	0.18072 (0.13898)	0.81021 (0.12525)	0.77406 (0.06568)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	5	**0.87064** (0.03315)	0.13781 (0.05099)	0.81664 (0.06968)	0.67406 (0.10538)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
DTLZ4	2	0.88189 (0.20625)	**0.88301** (0.20706)	0.55533 (0.3879)	0.88276 (0.20657)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	3	**0.88289** (0.20683)	0.8827 (0.2069)	0.56097 (0.38541)	0.78462 (0.25305)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	4	0.88238 (0.20645)	**0.88293** (0.20701)	0.61755 (0.39914)	0.83329 (0.23659)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	5	**0.88277** (0.20642)	0.7853 (0.25371)	0.61014 (0.40272)	0.73481 (0.2574)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
DTLZ7	2	0.84901 (0.08944)	**0.90849** (0.04448)	0.74587 (0.17607)	0.80901 (0.09467)	0 (0)	0 (0)	0 (0)	0 (0)	0.07872 (0.0021)
	3	0.84607 (0.08521)	**0.90558** (0.04829)	0.71182 (0.1039)	0.80016 (0.11071)	0 (0)	0 (0)	0 (0)	0 (0)	0.07913 (0.00244)
	4	0.84646 (0.08547)	**0.92182** (0.00907)	0.70199 (0.16297)	0.80508 (0.10569)	0 (0)	0 (0)	0 (0)	0 (0)	0.08059 (0.00152)
	5	**0.9176** (0.00504)	0.89312 (0.06185)	0.68389 (0.14252)	0.78243 (0.11719)	0 (0)	0 (0)	0 (0)	0 (0)	0.08049 (0.00148)

the smallest box (IQR) in all those problems except ZDT4. In DTLZ4 and DTLZ7, BIBEA yields lower median values and smaller IQR values in IGD than NSGA-II (Figures 2f and 2h) although NSGA-II yields lower IDG average values than BIBEA (Table VII). Figures 2a to 2h demonstrate that BIBEA achieves very low dispersion and skewness in HVR and IGD. BIBEA yields more stable performance than the other three algorithms.

E. Evaluation of BIBEA with $\mathcal{C}$-metric

Table VIII compares BIBEA with the other four algorithms with $\mathcal{C}$-metric. The first two rows compare BIBEA and IBEA-ε2. A bold font face is used to indicate a higher $\mathcal{C}$-metric value between $\mathcal{C}$(BIBEA, IBEA-ε2) and $\mathcal{C}$(IBEA-ε2, BIBEA). $\mathcal{C}$(BIBEA, IBEA-ε2) $>$ $\mathcal{C}$(IBEA-ε2, BIBEA) in six problems. This means that BIBEA outperforms IBEA-ε2 in those six problems. In ZDT3 and DTLZ1, IBEA-ε2 outperforms BIBEA; however, they yield very similar $\mathcal{C}$-metric values. The differences are 1.2% in ZDT3 and 0.3% in DTLZ1.

BIBEA outperforms IBEA-HD2 in five problems. IBEA-HD2 outperforms BIBEA in DTLZ1 and DTLZ7; however, they yield very similar $\mathcal{C}$-metric values. The differences are 1.8% in DTLZ1 and 1% in DTLZ7.

BIBEA outperforms NSGA-II in all ten problems. In ZDT6 and DTLZ2, no individuals of NSGA-II cannot dominate BIBEA individuals. A very limited number of NSGA-II individuals can dominate BIBEA individuals in other problems; for example, 0.1% in ZDT1 and DTLZ7, 0.4% in ZDT2, 0.6% in ZDT3, 1% in DTLZ1, and 1.7% in DTLZ3.

BIBEA outperforms SMS-EMOA in all problems except DTLZ4. In ZDT6 no individuals of SMS-EMOA cannot dominate BIBEA individuals.

F. Evaluation of BIBEA in Convergence Velocity

This section evaluates the convergence velocity of four different algorithms with HVR. Figure 3 how four algorithms improve their HVR values as the number of function evaluations grows. BIBEA improves its HVR faster than the other three algorithms in ZDT1, ZDT2, ZDT4, ZDT6, DTLZ4, DTLZ7. Particularly, in DTLZ4, BIBEA converge significantly faster than other algorithms.

In DTLZ1 and DTLZ3, BIBEA's convergence velocity is not the best but fairly acceptable. In fact, it is only slower than IBEA$_{HD2}$ at the beginning of experiments. However, it can converge to a higher HVR value toward the last function evaluation in both problems.

TABLE V: Comparison of I_B and Other Eight Indicators with IGD

	v	I_B	$I_{\varepsilon 2}$	I_{HD2}	I_{HypE}	I_{W0}	I_{W3}	I_{W6}	I_H	$I_{\varepsilon 1}$
ZDT1	2	0.00018 (0.00001)	**0.00017** (0.00001)	0.00018 (0.00001)	0.0002 (0.00002)	0.02869 (0.00022)	0.02872 (0.00012)	0.02837 (0.00096)	0.02849 (0.00062)	0.00071 (0.00006)
	3	0.00018 (0.00001)	**0.00017** (0.00001)	0.00018 (0.00001)	0.00031 (0.00004)	0.02864 (0.00018)	0.02847 (0.00069)	0.02871 (0.00013)	0.02871 (0.00009)	0.00074 (0.00011)
	4	0.00018 (0.00001)	**0.00017** (0.00001)	0.00018 (0.00001)	0.00041 (0.00011)	0.02831 (0.00109)	0.02855 (0.0007)	0.02849 (0.00087)	0.02854 (0.00036)	0.00076 (0.0001)
	5	0.00018 (0.00001)	**0.00017** (0.00001)	0.00018 (0.00001)	0.00049 (0.00009)	0.02868 (0.00024)	0.02865 (0.00034)	0.02838 (0.00126)	0.02858 (0.00037)	0.00084 (0.00018)
ZDT2	2	0.00077 (0.00137)	**0.00055** (0.0004)	0.00666 (0.0053)	0.00043 (0.00004)	0.02316 (0.00003)	0.02316 (0.00003)	0.02316 (0.00003)	0.02318 (0.00005)	0.01299 (0.00001)
	3	0.00045 (0.00007)	**0.00041** (0.00006)	0.00411 (0.00458)	0.00046 (0.00005)	0.02311 (0.00001)	0.02311 (0.00001)	0.02311 (0.00001)	0.02311 (0.00001)	0.013 (0.00001)
	4	**0.00041** (0.00004)	**0.00041** (0.00006)	0.00222 (0.0038)	0.00053 (0.0001)	0.0231 (0)	0.0231 (0.00001)	0.0231 (0.00001)	0.0231 (0.00001)	0.01301 (0.00001)
	5	0.00043 (0.00003)	**0.00042** (0.00006)	0.00307 (0.00489)	0.00069 (0.00034)	0.0231 (0)	0.0231 (0)	0.0231 (0)	0.0231 (0)	0.01299 (0.00003)
ZDT3	2	0.00313 (0.00277)	**0.00211** (0.001)	0.00375 (0.00245)	0.00215 (0.00151)	0.04307 (0.00022)	0.04283 (0.00066)	0.0425 (0.00174)	0.04301 (0.0003)	0.01304 (0.00366)
	3	0.00242 (0.00166)	**0.00199** (0.00076)	0.00323 (0.00138)	0.00147 (0.00017)	0.04276 (0.0008)	0.04234 (0.00189)	0.04305 (0.00027)	0.04308 (0.00013)	0.01279 (0.00336)
	4	0.00202 (0.00132)	**0.00201** (0.00075)	0.00327 (0.0014)	0.00152 (0.00018)	0.04295 (0.00041)	0.04299 (0.00049)	0.04277 (0.00112)	0.04286 (0.00107)	0.0127 (0.003)
	5	0.00223 (0.00148)	**0.00215** (0.00105)	0.00294 (0.00123)	0.00165 (0.00081)	0.04289 (0.00086)	0.0428 (0.00107)	0.04307 (0.00017)	0.04308 (0.00016)	0.01496 (0.00385)
ZDT4	2	**0.0052** (0.002)	0.00739 (0.00312)	0.01845 (0.00797)	0.00189 (0.00218)	0.0346 (0.00474)	0.03386 (0.0052)	0.0347 (0.00606)	0.03377 (0.00508)	0.00167 (0.00087)
	3	**0.00211** (0.00182)	0.00466 (0.00352)	0.01467 (0.00518)	0.00197 (0.00334)	0.03106 (0.00266)	0.03118 (0.00363)	0.0309 (0.00268)	0.03017 (0.00196)	0.00175 (0.00129)
	4	**0.00103** (0.00131)	0.00309 (0.00254)	0.01678 (0.00697)	0.00186 (0.00245)	0.0297 (0.00085)	0.0298 (0.00156)	0.03031 (0.00207)	0.03089 (0.00251)	0.00126 (0.00048)
	5	**0.00143** (0.0021)	0.0021 (0.00238)	0.01564 (0.00633)	0.00219 (0.00397)	0.02994 (0.00201)	0.03069 (0.0027)	0.03006 (0.00198)	0.03048 (0.00231)	0.00131 (0.00082)
ZDT6	2	0.00111 (0.00027)	0.00115 (0.00025)	0.00151 (0.00035)	**0.00089** (0.00016)	0.03373 (0.01507)	0.04223 (0.00083)	0.03852 (0.01065)	0.03609 (0.01297)	0.00641 (0.00007)
	3	0.00058 (0.00013)	0.00069 (0.00008)	0.00066 (0.00016)	**0.00043** (0.00009)	0.03588 (0.01271)	0.03604 (0.01229)	0.0411 (0.00034)	0.03984 (0.00493)	0.00662 (0.00004)
	4	0.00035 (0.00006)	0.00055 (0.00013)	0.00047 (0.00013)	**0.00027** (0.00004)	0.04089 (0.00024)	0.03925 (0.00691)	0.03944 (0.00612)	0.03566 (0.01254)	0.00668 (0.00002)
	5	0.00025 (0.00003)	0.0005 (0.0001)	0.00037 (0.00007)	**0.00022** (0.00002)	0.03894 (0.00773)	0.03384 (0.01395)	0.03725 (0.01048)	0.03899 (0.00763)	0.00671 (0.00002)
DTLZ1	2	**0.00057** (0.00002)	0.00697 (0.00172)	0.00096 (0.00106)	**0.00057** (0.00001)	0.84562 (0.77686)	0.65843 (0.70531)	0.88428 (0.48513)	0.0076 (0.00011)	0.00487 (0.00002)
	3	0.00092 (0.00125)	0.0066 (0.00075)	0.00123 (0.00153)	**0.00057** (0.00002)	0.7055 (0.43464)	0.63019 (0.52838)	0.80142 (0.41844)	0.00752 (0.00002)	0.0048 (0.0001)
	4	**0.00058** (0.00002)	0.00778 (0.00126)	0.00159 (0.00154)	0.00062 (0.00006)	0.56043 (0.29091)	0.49692 (0.46022)	0.40978 (0.27154)	0.00754 (0.00012)	0.00482 (0.00011)
	5	**0.00058** (0.00002)	0.00773 (0.00157)	0.00124 (0.00126)	0.0008 (0.00056)	0.64759 (0.67722)	0.35375 (0.29561)	0.71856 (0.5923)	0.00751 (0.00002)	0.00485 (0.00003)
DTLZ2	2	**0.001** (0.00003)	0.00101 (0.00003)	0.00138 (0.00002)	**0.001** (0.00005)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00535 (0)
	3	**0.001** (0.00002)	0.001 (0.00003)	0.00139 (0.00002)	0.00102 (0.00004)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00535 (0)
	4	**0.00101** (0.00003)	0.001 (0.00002)	0.00138 (0.00003)	0.00114 (0.00008)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00535 (0)
	5	**0.00101** (0.00002)	0.00102 (0.00002)	0.0014 (0.00003)	0.00179 (0.0002)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00859 (0)	0.00535 (0)
DTLZ3	2	0.00162 (0.00162)	0.01054 (0.0029)	0.00356 (0.00218)	**0.00158** (0.00005)	0.01439 (0.00005)	0.01438 (0.00003)	0.0144 (0.00002)	0.01437 (0.00002)	0.00891 (0.00003)
	3	**0.00158** (0.00005)	0.01065 (0.00294)	0.00261 (0.00068)	0.00161 (0.00005)	0.01437 (0.00001)	0.01439 (0.00004)	0.01438 (0.00004)	0.01437 (0.00002)	0.00891 (0.00002)
	4	**0.00161** (0.00006)	0.00939 (0.00339)	0.00279 (0.00132)	0.00252 (0.00046)	0.01437 (0.00001)	0.01438 (0.00002)	0.01437 (0.00002)	0.01437 (0.00002)	0.00891 (0.00001)
	5	**0.00169** (0.00018)	0.01075 (0.00173)	0.00257 (0.00059)	0.00325 (0.00046)	0.01439 (0.00004)	0.01439 (0.00005)	0.01438 (0.00003)	0.01437 (0.00002)	0.0089 (0.00001)
DTLZ4	2	**0.00211** (0.00236)	0.00232 (0.00277)	0.00516 (0.00362)	**0.00211** (0.00236)	0.01447 (0.00536)	0.01752 (0.00429)	0.01651 (0.00492)	0.00939 (0)	0.00625 (0.00084)
	3	**0.00213** (0.00235)	0.00232 (0.00277)	0.00516 (0.00361)	0.00319 (0.00292)	0.0165 (0.00491)	0.01854 (0.00321)	0.01854 (0.00321)	0.00939 (0)	0.00637 (0.00103)
	4	0.00214 (0.00234)	**0.00211** (0.00236)	0.00467 (0.00386)	0.00277 (0.00288)	0.0165 (0.00491)	0.0165 (0.00491)	0.01854 (0.00321)	0.00939 (0)	0.00601 (0.00042)
	5	**0.0021** (0.00236)	0.00343 (0.00317)	0.00452 (0.0037)	0.00387 (0.00309)	0.01752 (0.00429)	0.01549 (0.00525)	0.01854 (0.00321)	0.00939 (0)	0.00625 (0.00084)
DTLZ7	2	**0.01053** (0.00831)	0.00441 (0.00449)	0.01402 (0.01093)	0.01407 (0.00823)	0.16264 (0.01842)	0.14682 (0.01723)	0.16529 (0.02265)	0.0377 (0)	0.01163 (0.00001)
	3	**0.01049** (0.00832)	0.00458 (0.0045)	0.01675 (0.00589)	0.01371 (0.00934)	0.16265 (0.01492)	0.1677 (0.02078)	0.16242 (0.01552)	0.0377 (0)	0.01163 (0.00001)
	4	**0.01063** (0.00822)	0.00304 (0.00019)	0.01616 (0.00925)	0.01325 (0.0088)	0.16551 (0.02734)	0.15856 (0.0208)	0.16778 (0.01719)	0.0377 (0)	0.01163 (0.00001)
	5	**0.00315** (0.00024)	0.00581 (0.00602)	0.01873 (0.00952)	0.01412 (0.00942)	0.17422 (0.03043)	0.16881 (0.03114)	0.16702 (0.01116)	0.0377 (0)	0.01163 (0)

TABLE VI: Comparison of BIBEA with Other Four EMOAs with HVR

	BIBEA	IBEA-ε2	IBEA-HD2	NSGAII	SMS-EMOA
ZDT1	0.99124 (0.00073)	**0.99140** (0.00053)	0.99124 (0.00057)	0.98490 (0.00137)**	0.99048(6.5E-4)*
ZDT2	**0.97855** (0.00201)	0.97816 (0.00141)	0.85718 (0.22804)*	0.96948 (0.00330)**	0.91286(0.09076)*
ZDT3	0.98618 (0.01717)	0.98920 (0.00125)	0.98569 (0.00098)	**0.99284** (0.00202)	0.98882(0.01754)
ZDT4	**0.96062** (0.03139)	0.93968 (0.03962)	0.59782 (0.23131)**	0.95103 (0.04873)*	0.71216(0.23092)**
ZDT6	**0.97168** (0.00502)	0.95166 (0.00771)**	0.96303 (0.00709)**	0.93456 (0.01069)**	0.85502(0.02427)**
DTLZ1	**0.93482** (0.00364)	0.09898 (0.08299)**	0.85456 (0.14999)	0.46676 (0.46002)**	0.93192(0.06896)
DTLZ2	0.89625 (0.00135)	0.89670 (0.00087)	0.88071 (0.00115)**	0.80160 (0.01253)**	**0.91111**(5.6E-4)**
DTLZ3	**0.87064** (0.03315)	0.13781 (0.05099)**	0.81664 (0.06968)*	0.76162 (0.03454)**	**0.88058**(0.05345)
DTLZ4	**0.88277** (0.20642)	0.78530 (0.25371)	0.61014 (0.40272)	0.80650 (0.28387)	0.72147(0.37901)
DTLZ7	**0.91760** (0.00504)	0.89312 (0.06185)	0.68389 (0.14252)**	0.84875 (0.01250)**	0.76956(0.01022)**

Moreover, IBEA-$\epsilon 2$ encounters premature convergence in DTLZ1, DTLZ3 and DTLZ4. IBEA-HD2 does in ZDT4, DTLZ4 and DTLZ7. NSGAII has the same problem in DTLZ2, DTLZ4 and DTLZ7. In contrast, BIBEA never encounters premature convergence. This means BIBEA's convergence velocity is more consistent and robust than the other's among different problems.

G. Evaluation of BIBEA in Experimental Execution Time

Table IX shows the average execution time of each algorithm. An execution time indicates how long it takes for an algorithm to run a single experiment for a test problem. BIBEA is the slowest in six of ten test problems. In ZDT problems (i.e., two-objective problems), BIBEA and IBEA-HD2 are the two slowest algorithms while NSGA-II is the fastest. In DTLZ problems (i.e., three-objective problems), BIBEA is the slowest algorithm while NSGA-II is still the fastest. BIBEA is slow because it uses the HypE indicator (Table III), which is generally expensive and significantly expensive in higher-dimensional problems.

The average execution time of the proposed boosting method (Algorithm 1) is 18 minutes. This is fairly acceptable for an offline training phase.

TABLE VII: Comparison of BIBEA with Other Four EMOAs with IGD

	BIBEA	IBEA-ε2	IBEA-HD2	NSGAII	SMS-EMOA
ZDT1	0.00018 (0.00001)	0.00017 (0.00001)**	0.00018 (0.00001)	0.00022 (0.00001)**	**0.00015**(0.0)**
ZDT2	0.00043 (0.00003)	0.00042 (0.00006)	0.00307 (0.00489)*	**0.00022** (0.00002)**	0.00206(0.00239)*
ZDT3	0.00223 (0.00148)	0.00215 (0.00105)	0.00294 (0.00123)	**0.00074** (0.00149)**	0.00141(0.00214)
ZDT4	0.00143 (0.0021)	0.00210 (0.00238)	0.01564 (0.00633)**	**0.00109** (0.00146)	0.00946(0.00561)**
ZDT6	**0.00025** (0.00003)	0.00050 (0.00010)**	0.00037 (0.00007)**	0.00068 (0.00014)**	0.00181(0.00033)**
DTLZ1	0.00058 (0.00002)	0.00773 (0.00157)**	0.00124 (0.00126)	0.00464 (0.00490)*	**0.00054**(0.00035)
DTLZ2	0.00101 (0.00002)	0.00102 (0.00002)*	0.00140 (0.00003)**	**0.00080** (0.00003)**	0.00084(1.0E-5)**
DTLZ3	0.00169 (0.00018)	0.01075 (0.00173)**	0.00257 (0.00059)**	**0.00135** (0.00007)**	0.00137(0.00025)**
DTLZ4	0.00210 (0.00236)	0.00343 (0.00317)	0.00452 (0.00370)	**0.00206** (0.00259)	0.00375(0.00364)
DTLZ7	0.00315 (0.00024)	0.00581 (0.00602)	0.01873 (0.00952)**	**0.00239** (0.00023)**	0.00701(0.00487)**

TABLE VIII: Comparison of BIBEA with Other Four EMOAs with $\mathcal{C}$-metric

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ7
$\mathcal{C}$(BIBEA, IBEA-ε2)	**0.396**	**0.602**	0.384	0.528	**0.992**	0.042	**0.081**	0.005	**0.059**	**0.032**
$\mathcal{C}$(IBEA-ε2, BIBEA)	0.185	0.123	**0.396**	**0.738**	0.089	**0.045**	0.030	**0.150**	0.020	0.011
$\mathcal{C}$(BIBEA, IBEA-HD2)	**0.287**	**0.853**	0.342	**0.858**	**0.914**	0.038	0.038	0.163	**0.055**	0.014
$\mathcal{C}$(IBEA-HD2, BIBEA)	0.227	0.085	**0.454**	0.326	0.265	**0.056**	**0.089**	**0.311**	0.048	**0.024**
$\mathcal{C}$(BIBEA, NSGAII)	**0.973**	**0.956**	**0.63**	**0.917**	**0.989**	**0.55**	**0.255**	**0.524**	**0.231**	**0.512**
$\mathcal{C}$(NSGAII, BIBEA)	0.001	0.004	0.006	0.591	0.0	0.01	0	0.017	0	0.001
$\mathcal{C}$(BIBEA, SMS-EMOA)	**0.821**	**0.815**	**0.614**	**0.94**	**0.991**	**0.049**	**0.021**	**0.323**	0.016	**0.023**
$\mathcal{C}$(SMS-EMOA, BIBEA)	0.041	0.048	0.011	0.375	0.0	0.027	0.018	0.281	**0.023**	0.008

TABLE IX: Comparison of BIBEA with Other Four EMOAs in Experimental Execution time (seconds)

	BIBEA	IBEA-ε2	IBEA-HD2	NSGAII	SMS-EMOA
ZDT1	49.34	39.16	50.65	**6.04**	32.39
ZDT2	49.44	38.82	48.62	**6.30**	23.37
ZDT3	47.67	37.02	48.97	**4.25**	29.18
ZDT4	49.15	36.27	50.29	**6.21**	10.27
ZDT6	50.05	38.95	51.75	**6.87**	12.69
DTLZ1	812.01	127.41	170.36	**89.87**	295.97
DTLZ2	747.85	115.85	148.82	**71.30**	656.21
DTLZ3	1685.88	154.98	271.33	**67.20**	332.54
DTLZ4	944.10	76.96	115.56	**29.86**	410.57
DTLZ7	902.77	54.17	94.18	**9.40**	231.34

VII. Conclusions

This paper proposes and evaluates a novel method that leverages a boosting algorithm to obtain an aggregated selection operator from various existing indicator-based selection operators. Experimental results show that a boosted selection operator outperforms exiting ones in optimality, diversity and convergence velocity. The proposed boosting process can work with a simple training problem, and the boosted operator can effectively solve harder problems. The boosted operator also exhibits robustness against different characteristics in different problems and yields stable performance to solve them.

Several future extensions are planned for the proposed boosting method. First, it will be enhanced so that it can work with a training problem in which Pareto-optimal solutions are unknown. Second, the notion of boosted indicator-based selection will be evaluated in other problems than ZDT and DTLZ problems.

References

[1] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evol. Computat.*, vol. 2, no. 3, 1995.

[2] C. C. Coello, "Evolutionary multi-objective optimization: Some current research trends and topics that remain to be explored," *Front. Computat. Sci. China*, vol. 3, no. 1, 2009.

[3] 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.

[4] 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.

[5] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Computat.*, vol. 19, no. 1, 2011.

[6] E. Zitzler and S. Kuenzli., "Indicator-based selection in multiobjective search," in *Proc. Int'l Conference on Parallel Problem Solving from Nature*, 2004.

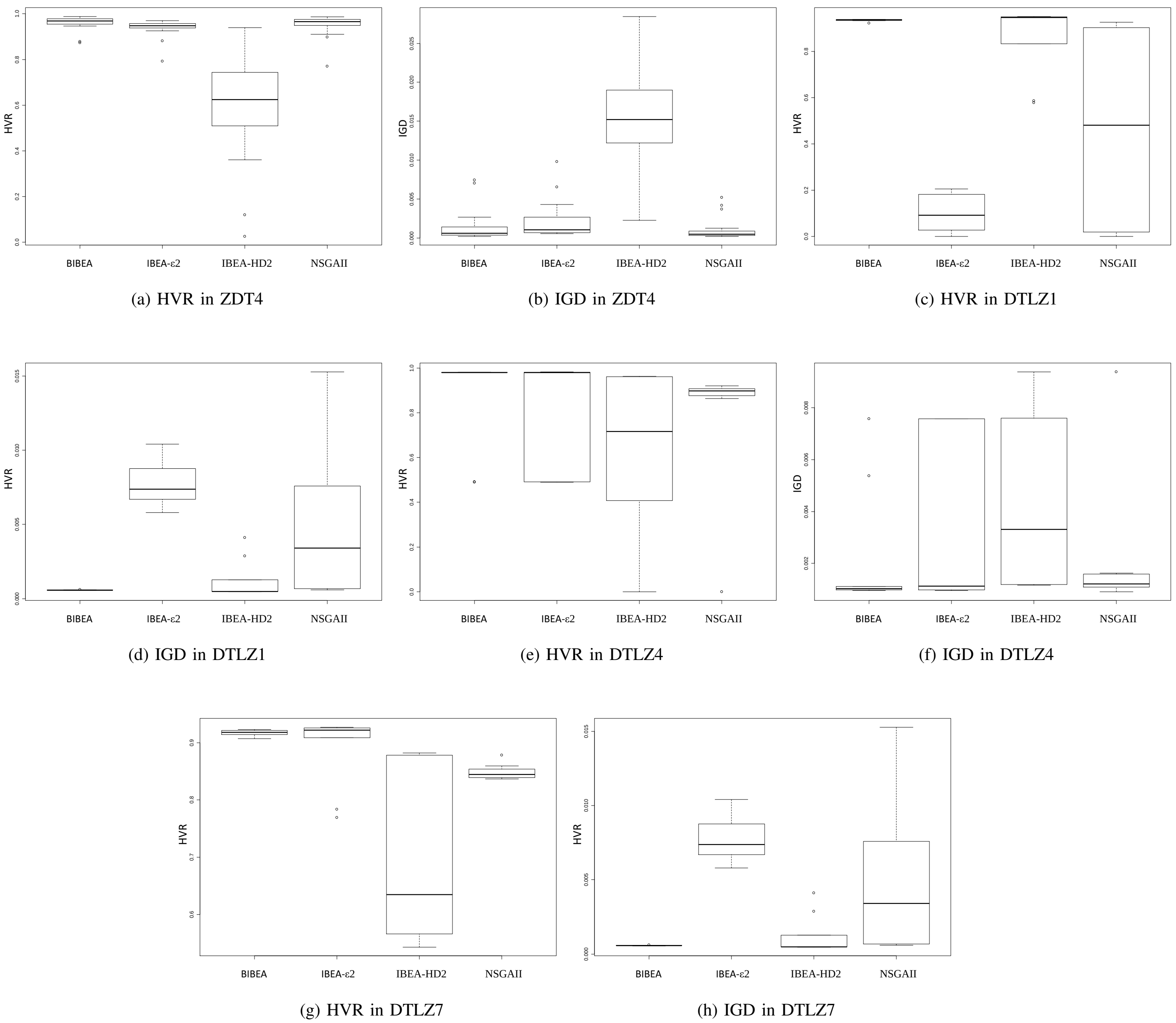
[7] P. Boonma and J. Suzuki, "Prospect indicator based evolutionary multiobjective optimization algorithm," in *Proc. IEEE Congress on Evolutionary Computation*, 2011.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans Evol. Computat.*, vol. 6, no. 2, 2002.

[9] 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.

[10] B. Liu, M. B., and A. H.A., "Improving genetic classifiers with a boosting algorithm," in *Proc. IEEE Congress on Evol. Computat.*, 2003.

[11] 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.



(a) HVR in ZDT4

(b) IGD in ZDT4

(c) HVR in DTLZ1

(d) IGD in DTLZ1

(e) HVR in DTLZ4

(f) IGD in DTLZ4

(g) HVR in DTLZ7

(h) IGD in DTLZ7

Fig. 2: Boxplots for ZDT4, DTLZ1, DTLZ4 and DTLZ7

[12] G. Paris, D. Robilliard, and C. Fonlupt, “Applying boosting techniques to genetic programming,” in *Proc. European Conference on Artificial Evolution*, 2002.

[13] L. V. Souza, A. R. T. Pozo, J. C. M. Da Rosa, and C. Neto, “Technique using correlation coefficient to improve time series forecasting accuracy,” in *Proc. IEEE Congress on Evolutionary Computation*, 2007.

[14] I. Yalabik and T.-V. Fatos, “A pattern classification approach for boosting with genetic algorithms,” in *Proc. Int’l Symposium on Computer and Information Sciences*, 2007.

[15] L. E. Santana, A. M. P. Canuto, and L. Silva, “Bio-inspired metaheuristic as feature selector in ensemble systems: A comparative analysis,” in *Proc. Int’l Joint Conference on Neural Networks*, 2011.

[16] D. A. Augusto, H. J. Barbosa, and N. F. Ebecken, “Coevolutionary multipopulation genetic programming for data classification,” in *Proc. ACM Intl Genetic and Evolutionary Computation Conference*, 2010.

[17] Z. He and G. G. Yen, “An ensemble method for performance metrics in multiobjective evolutionary algorithms,” in *Proc. IEEE Congress on Evolutionary Computation*, 2010.

[18] 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.

[19] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, 1997.

[20] D. A. V. Veldhuizen and G. B. Lamont, “Multiobjective evolutionary algorithm test suites,” in *Proc. ACM Symposium on Applied Computing*, 1999.

[21] 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.

[22] 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.

[23] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Computat.*, vol. 8, no. 2, 2000.

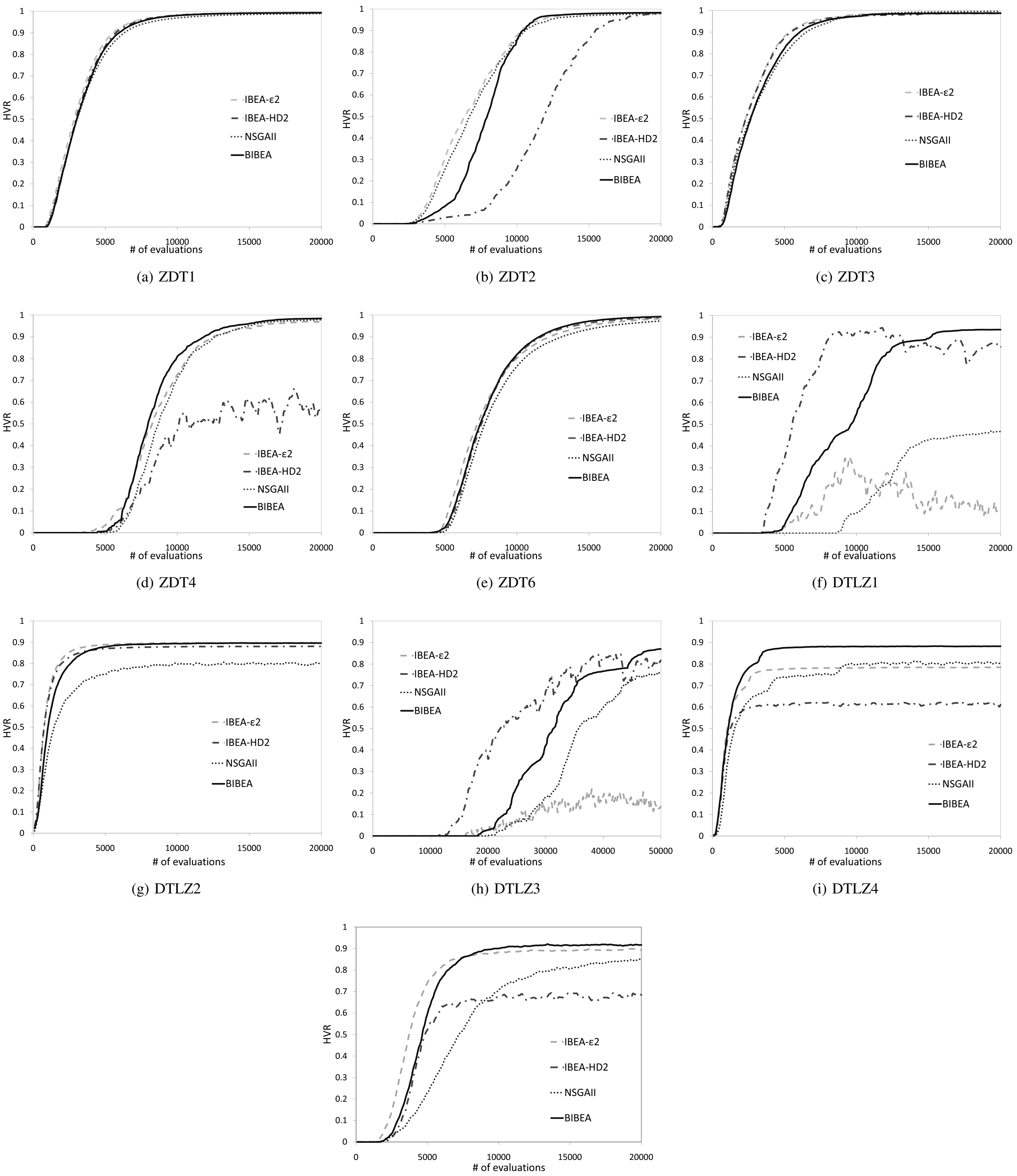
[24] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multi-objective Optimization*, A. Abraham, R. Jain, and R. Goldberg, Eds. Springer, 2005.

[25] D. A. V. Veldhuizen and G. B. Lamont, “Multiobjective evolutionary algorithm research: A history and analysis,” in *Technical Report TR-98-03, Air Force Institute of Technology*, 1998.

[26] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE Trans.Evol. Comput.*, vol. 3, no. 4, 1999.

[27] N. Beume, B. Naujoks, and M. Emmerich, “SMS-EMOA: Multiobjective selection based on dominated hypervolume,” *Eur. J. Oper. Res.*, vol. 181, no. 3, 2007.

[28] D. H. Phan, J. Suzuki, and P. Boonma, “SMSP-EMOA: Augmenting sms-emoa with the prospect indicator for multiobjective optimization,” in *Proc. IEEE Int’l Conf. on Tools with Artificial Intelligence*, 2011.



(a) ZDT1

(b) ZDT2

(c) ZDT3

(d) ZDT4

(e) ZDT6

(f) DTLZ1

(g) DTLZ2

(h) DTLZ3

(i) DTLZ4

(j) DTLZ7

Fig. 3: Convergence Velocity