An Ensemble Indicator-based Density Estimator for Evolutionary Multi-objective Optimization*

Jesús Guillermo Falcón-Cardona[1], Arnaud Liefooghe[2], and Carlos A. Coello Coello[0000−0002−8435−680X]1

[1] CINVESTAV-IPN, Computer Science Department, Mexico City 07300, Mexico
jfalcon@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx
[2] JFLI – CNRS IRL 3527, University of Tokyo, 113-0033 Tokyo, Japan
arnaud.liefooghe@univ-lille.fr

Abstract. Ensemble learning is one of the most employed methods in machine learning. Its main ground is the construction of stronger mechanisms based on the combination of elementary ones. In this paper, we employ AdaBoost, which is one of the most well-known ensemble methods, to generate an ensemble indicator-based density estimator for multi-objective optimization. It combines the search properties of five density estimators, based on the hypervolume, R2, IGD$^+$, ϵ^+, and Δ_p quality indicators. Through the multi-objective evolutionary search process, the proposed ensemble mechanism adapts itself using a learning process that takes the preferences of the underlying quality indicators into account. The proposed method gives rise to the ensemble indicator-based multi-objective evolutionary algorithm (EIB-MOEA) that shows a robust performance on different multi-objective optimization problems when compared with respect to several existing indicator-based multi-objective evolutionary algorithms.

Keywords: Multi-objective optimization · quality indicators · ensemble learning · AdaBoost.

1 Introduction

In many scientific and industrial fields arise the so-called multi-objective optimization problems (MOPs), that involve the simultaneous optimization of two or more conflicting objective functions. Mathematically, an MOP is defined as follows:

$$\min_{\boldsymbol{x}\in\Omega} \{\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))\}, \tag{1}$$

where $\boldsymbol{x}$ is the vector of decision variables, $\Omega \subseteq \mathbb{R}^n$ is the decision space and $\boldsymbol{F}(\boldsymbol{x})$ is the vector of $m \geq 2$ objective functions such that $f_i : \mathbb{R}^n \to \mathbb{R}$

* The first author acknowledges support from CONACyT and CINVESTAV-IPN to pursue graduate studies in Computer Science. The third author gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a SEP-Cinvestav grant (application no. 4).

for $i \in \{1, 2, \ldots, m\}$. Unlike single-objective optimization problems which have a single global optimal solution, the solution of an MOP is a set of solutions that represents the best possible trade-offs among the objective functions. Given $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ and $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m$, we say that $\boldsymbol{x}$ dominates $\boldsymbol{y}$, denoted as $\boldsymbol{F}(\boldsymbol{x}) \prec \boldsymbol{F}(\boldsymbol{y})$, if and only if $\forall i \in \{1, \ldots, m\}, f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ and there exists at least an index $j \in \{1, \ldots, m\}$ such that $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$. The particular set that yields the optimum values, according to the Pareto dominance relation, is the Pareto set; its image is known as the Pareto front.

Multi-objective evolutionary algorithms (MOEAs) constitute a popular choice to tackle complex MOPs [1]. MOEAs are stochastic black-box optimizers based on the principles of Darwin's natural selection. MOEAs are population-based metaheuristics that can generate a Pareto front approximation (or approximation set) in a single execution. Ideally, an MOEA should produce solutions as close as possible to the Pareto front, covering it all and with good diversity. There exist four main design methodologies for MOEAs [1]: (1) MOEAs using the Pareto dominance relation or any of its relaxed forms, (2) decomposition-based MOEAs, (3) reference set-based MOEAs, and (4) indicator-based MOEAs (IB-MOEAs). In the last fifteen years, IB-MOEAs have attracted considerable attention due to their ability to solve MOPs having more than three objective functions (i.e., the so-called many-objective optimization problems) [2]. The underlying idea of IB-MOEAs is the use of a quality indicator (QI) [3], which is a set function that evaluates the quality of an approximation set based on specific preferences, in order to guide the evolutionary search process by focusing on the selection mechanisms. Currently, there exist several QIs, such as the hypervolume indicator (HV) [4], R2 [5], the inverted generational distance plus (IGD^+) [6], the additive epsilon indicator (ϵ^+) [7], and the averaged Hausdorff distance (Δ_p) [8], being these ones the most popular within the currently available IB-MOEAs [2].

An IB-MOEA produces a Pareto front approximation exhibiting the preferences of its underlying QI [9]. As such, different IB-MOEAs yield different results in terms of the distribution of solutions in the approximation set, due to the underlying properties of the QI they employ. Moreover, there are MOPs where a specific IB-MOEA performs well, but there are others on which it does not. As a consequence, it is not clear which QI to consider beforehand, and an open question is whether a set of existing indicator-based selection mechanisms can create a single operator that reaches a consensus that outperforms the existing ones. In 2011, Phan and Suzuki were the first to investigate this question by boosting a set of indicator-based mating selection operators [10]. The boosted indicator-based mating selection operator uses 15 quality indicators from which it ensembles the best suited ones using the AdaBoost algorithm [11] with an offline training. The proposed mechanism was embedded into the non-dominated sorting genetic algorithm II (NSGA-II) [12], giving rise to the boosted indicator-based evolutionary algorithm (BIBEA). According to the reported results, BIBEA was able to outperform NSGA-II, exhibiting robustness in MOPs with different characteristics. Later on, Phan *et al.* [13] proposed BIBEA-P which allows BIBEA to

use an additional ensemble indicator-based mechanism for environmental selection. Moreover, BIBEA-P uses Pdi-Boosting instead of AdaBoost. Similarly to BIBEA, the ensemble operators of BIBEA-P needs to be trained using a given MOP in an offline fashion. The experimental results showed that BIBEA-P is better than BIBEA, NSGA-II, and SMS-EMOA (which is a HV-based MOEA) [14] when using MOPs with different Pareto front shapes.

In this paper, we propose an ensemble indicator-based density estimator using the HV, R2, IGD$^+$, ϵ^+, and Δ_p indicators. Unlike BIBEA and BIBEA-P, our mechanism adapts the combination of the indicator-based density estimators (IB-DEs) in an online fashion. The underlying reason to use an online ensemble mechanism is to allow our approach to produce high-quality results for problems with different characteristics, and to reach a robust performance with respect to the multiple indicators being considered. This approach allows our proposed ensemble indicator-based MOEA (EIB-MOEA) to tackle problems with different Pareto front geometries from the test suites DTLZ, DTLZ^{-1}, WFG, and WFG^{-1}. Furthermore, the proposed approach consistently obtains competitive results with respect to other IB-MOEAs.

The remainder of this paper is organized as follows. Section 2 provides the mathematical definitions of the QIs that we consider in our analysis. Section 3 describes the algorithmic design of EIB-MOEA. Section 4 shows our experimental results and Section 5 provides our final conclusions and some possible paths for future work.

2 Background

In this section, we describe a selection of five QIs, corresponding to those which are most frequently used in the specialized literature. They will be considered as constituent QIs in our proposed ensemble indicator-based density estimator. In the following, $\mathcal{A}$ denotes an approximation set.

Definition 1 (Hypervolume indicator [4]) *Given an anti-optimal reference point $\boldsymbol{r} \in \mathbb{R}^m$, the hypervolume is defined as follows:*

$$HV(\mathcal{A}, \boldsymbol{r}) = \mathcal{L}\left(\bigcup_{\boldsymbol{a} \in \mathcal{A}} \{\boldsymbol{b} \mid \boldsymbol{a} \prec \boldsymbol{b} \prec \boldsymbol{r}\}\right), \tag{2}$$

where $\mathcal{L}(\cdot)$ denotes the Lebesgue measure in $\mathbb{R}^m$.

Definition 2 (Unary $R2$ indicator [5]) *The unary R2 indicator is defined as follows:*

$$R2(\mathcal{A}, W) = -\frac{1}{|W|} \sum_{\boldsymbol{w} \in W} \max_{\boldsymbol{a} \in \mathcal{A}} \{u_{\boldsymbol{w}}(\boldsymbol{a})\}, \tag{3}$$

where W is a set of weight vectors and $u_{\boldsymbol{w}} : \mathbb{R}^m \to \mathbb{R}$ is a scalarizing function defined by $\boldsymbol{w} \in W$ that assigns a real value to each m-dimensional vector.

Definition 3 (IGD$^+$ indicator [6]) *The IGD$^+$, for minimization, is defined as follows:*

$$IGD^+(\mathcal{A}, Z) = \frac{1}{|Z|} \sum_{\boldsymbol{z} \in Z} \min_{\boldsymbol{a} \in \mathcal{A}} d^+(\boldsymbol{a}, \boldsymbol{z}), \tag{4}$$

where $d^+(\boldsymbol{a}, \boldsymbol{z}) = \sqrt{\sum_{k=1}^{m} \left(\max\{a_k - z_k, 0\}\right)^2}$.

Definition 4 (Unary ϵ^+ indicator [7]) *The unary ϵ^+-indicator gives the minimum distance by which a Pareto front approximation needs to or can be translated in each dimension in the objective space such that a reference set is weakly dominated. Mathematically, it is defined as follows:*

$$\epsilon^+(\mathcal{A}, \mathcal{Z}) = \max_{\boldsymbol{z} \in \mathcal{Z}} \min_{\boldsymbol{a} \in \mathcal{A}} \max_{1 \leq i \leq m} \{a_i - z_i\}. \tag{5}$$

Definition 5 (Δ_p indicator [8]) *For a given $p > 0$, the Δ_p is defined as follows:*

$$\Delta_p(\mathcal{A}, Z) = \max\{GD_p(\mathcal{A}, Z), IGD_p(\mathcal{A}, Z)\}. \tag{6}$$

Δ_p is defined on the basis of two indicators: GD_p and IGD_p which are slight modifications of the indicators Generational Distance (GD) and Inverted Generational Distance (IGD) [3], respectively. These are defined in the following.

Definition 6 (GD_p indicator [8])

$$GD_p(\mathcal{A}, \mathcal{Z}) = \left(\frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{a} \in \mathcal{A}} d(\boldsymbol{a}, \mathcal{Z})^p \right)^{1/p}, \tag{7}$$

where $d(\boldsymbol{a}, \mathcal{Z}) = \min_{\boldsymbol{z} \in \mathcal{Z}} \sqrt{\sum_{i=1}^{m} (a_i - z_i)^2}$.

Definition 7 (IGD_p indicator [8]) *It is defined as follows:* $IGD_p(\mathcal{A}, \mathcal{Z}) = GD_p(\mathcal{Z}, \mathcal{A})$.

Definition 8 (Indicator contribution) *Let $\mathcal{I}$ be any indicator in the set $\{HV$, $R2$, IGD^+, ϵ^+, $\Delta_p\}$. The individual contribution C of a solution $\boldsymbol{a} \in \mathcal{A}$ to the indicator value is given as follows:*

$$C_{\mathcal{I}}(\boldsymbol{a}, \mathcal{A}) = |\mathcal{I}(\mathcal{A}) - \mathcal{I}(\mathcal{A} \setminus \{\boldsymbol{a}\})|. \tag{8}$$

Interestingly, the QIs presented above have different properties, and express different preferences in terms of set approximation quality [7]. Moreover, they do not always agree with each other [15], so that good-quality approximation sets for a given QI typically contain different solutions than for other QIs. This motivates the ensemble indicator-based approach introduced below.

3 The Proposed EIB-MOEA Approach

In this section, we first give the general description of EIB-MOEA, then we detail the learning model and the adaptive strategy considered to update the relative importance given to each QI at different iterations.

Algorithm 1 EIB-MOEA's general framework

Require: Set of indicators $\{I_1, \ldots, I_k\}$; time window size T_w
Ensure: Pareto front approximation
1: Randomly initialize population $\mathcal{A}$
2: $w_i = 1/k, i \in \{1, \ldots, k\}$
3: Initialize performance matrix $P \in \mathbb{R}^{k \times T_w}$
4: Initialize learning matrix $\Psi \in \{0, 1\}^{k \times T_w}$
5: $g = 0$
6: **while** stopping criterion is not fulfilled **do**
7: Create an offspring solution $\boldsymbol{q}$ based on $\mathcal{A}$
8: $Q = \mathcal{A} \cup \{q\}$
9: $\{R_1, \ldots, R_\ell\} = NondominatedSorting(Q)$
10: **if** $|R_\ell| > 1$ **then**
11: $z_i^{\min} = \min_{\boldsymbol{a} \in \mathcal{A}} f_i(\boldsymbol{a}), i \in \{1, \ldots, m\}$
12: $z_i^{\max} = \max_{\boldsymbol{a} \in \mathcal{A}} f_i(\boldsymbol{a}), i \in \{1, \ldots, m\}$
13: Normalize $\{R_1, \ldots, R_\ell\}$ using $\boldsymbol{z}^{\min}$ and $\boldsymbol{z}^{\max}$
14: **for** $j = 1$ to k **do**
15: $C_{I_j}(\boldsymbol{r}, R_\ell) = |I_j(R_\ell) - I_j(R_\ell \setminus \{\boldsymbol{F}(\boldsymbol{r})\})|, \forall \boldsymbol{r} \in R_\ell$
16: Sort C_{I_j} in ascending order
17: $\forall \boldsymbol{z} \in R_\ell$, compute $\text{rank}_{I_j}(\boldsymbol{F}(\boldsymbol{r}))$, using the sorted C_{I_j}
18: **end for**
19: $\boldsymbol{a}_{\text{worst}} = \arg\min_{\boldsymbol{r} \in R_\ell} \left\{ H\left(\boldsymbol{z} = \boldsymbol{F}(\boldsymbol{r})\right) = \sum_{j=1}^{k} w_j \text{rank}_{I_j}(\boldsymbol{z}) \right\}$
20: Learning$(Q, R_\ell, \{I_1, \ldots, I_k\}, g, \boldsymbol{a}_{\text{worst}}, P, \Psi)$
21: $g = g + 1$
22: **else**
23: Let $\boldsymbol{a}_{\text{worst}}$ be the sole solution in R_ℓ
24: **end if**
25: $\mathcal{A} = Q \setminus \{\boldsymbol{a}_{\text{worst}}\}$
26: **if** $g = T_w$ **then**
27: UpdateWeights$(\boldsymbol{w}, P, \Psi, T_w, k)$
28: $g = 0$
29: **end if**
30: **end while**
31: **return** $\mathcal{A}$

3.1 General Description

The proposed EIB-MOEA is a steady-state MOEA based on SMS-EMOA [14]. Its general framework is outlined in Algorithm 1. EIB-MOEA requires a set of k indicators $\{I_1, \ldots I_k\}$ and a time window frame T_w as input parameters. In Line 2, all the components of the weight vector $\boldsymbol{w}$ are set to $1/k$. This weight vector is employed in the ensemble indicator-based density estimator (EIB-DE), and contains the relative importance given to each indicator at the current iteration. Lines 6 to 30 describe the main loop of EIB-MOEA. At each iteration, a single offspring solution $\boldsymbol{q}$ is created using variation operators. This newly created solution is added to the population $\mathcal{A}$ to create the temporary population Q. The non-dominated sorting algorithm [12] processes Q to create a set of layers $\{R_1, \ldots, R_\ell\}$. If R_ℓ contains more than one solution, EIB-DE is executed. First, the population is normalized in Line 13. Then, for each indicator $I_j, j \in \{1, \ldots, k\}$, the individual indicator contributions of all solutions in R_ℓ are computed and stored in the vector C_{I_j}. By sorting this vector in ascending order, for each $\boldsymbol{r} \in R_\ell$ we obtain $\text{rank}_{I_j}(\boldsymbol{F}(\boldsymbol{r})) \in \{1, 2, \ldots, |R_l|\}$ that returns the ranking of the solution in the sorted C_{I_j}, where rank 1 corresponds to the worst-contributing solution to I_j. In Line 19, the worst-contributing solution,

Algorithm 2 Learning

Require: Population $\mathcal{A}$; worst set R; set of indicators $\{I_1, \ldots, I_k\}$; index t; selected solution $\boldsymbol{a}_{\text{worst}}$; performance matrix P; learning matrix Ψ
Ensure: Updated Ψ
1: **for** $j = 1$ to k **do**
2: $\quad \boldsymbol{a}^j_{\text{worst}} = \arg\min_{\boldsymbol{r} \in R} |I_j(R) - I_j(R \setminus \{\boldsymbol{F}(\boldsymbol{r})\})|$
3: $\quad \mathcal{A}^j = \mathcal{A} \setminus \{\boldsymbol{a}^j_{\text{worst}}\}$
4: $\quad P_{jt} = I_j(\mathcal{A}^j)$
5: $\quad$ **if** $P_{jt} > P_{j,t-1 \bmod T_w} \wedge \boldsymbol{a}^j_{\text{worst}} = \boldsymbol{a}_{\text{worst}}$ **then**
6: $\quad\quad \Psi_{jt} = 0$
7: $\quad$ **else**
8: $\quad\quad \Psi_{jt} = 1$
9: $\quad$ **end if**
10: **end for**
11: **return** Ψ

using EIB-DE, is obtained. The learning process (see Algorithm 2), which is a fundamental part to update the weight vector $\boldsymbol{w}$, is performed in Line 20, and then, the counter g is incremented by one. In Line 25, $\boldsymbol{a}_{\text{worst}}$ is eliminated from Q to shape the population for the next generation. In case g is equal to T_w, $\boldsymbol{w}$ is updated following Algorithm 3 and g is set to zero. Finally, once the stopping condition is satisfied, $\mathcal{A}$ is returned as the Pareto front approximation.

3.2 Learning Process

The learning process, described in Algorithm 2, is based on analyzing the behavior of the population using all indicators. For each indicator $I_j, j \in \{1, \ldots, k\}$, we obtain its worst-contributing solution $\boldsymbol{a}^j_{\text{worst}}$, where R represents the last layer of solutions with respect to non-dominated sorting. In Line 3, we simulate the elimination of $\boldsymbol{a}^j_{\text{worst}}$ from the population $\mathcal{A}$ to generate the set $\mathcal{A}^j$ that is assessed by I_j. This indicator value is stored in the performance matrix at position (j, t), i.e., $P_{jt} = I_j(\mathcal{A}^j)$. It is worth noting that each row of P, represented as P_j, works as a circular array of size T_w. If P_{jt} is greater than the previous sample in P_j (which implies an increase in quality) and $\boldsymbol{a}^j_{\text{worst}}$ is the same as the worst-contributing solution to EIB-DE, the selection is marked as successful and a zero value is stored in the learning matrix Ψ in the same position (j, t). Otherwise, we set $\Psi_{jt} = 1$.

3.3 Updating the Relative Importance of QIs

After executing EIB-DE and the learning algorithm a total of T_w times, the weight vector has to be updated. Algorithm 3 sketches the update process which is based on the AdaBoost algorithm [11], whose aim is to minimize the exponential loss. For each indicator $I_j, j \in \{1, \ldots, k\}$, the selection error e_j is calculated using the jth row of the learning matrix Ψ, taking into account that e_j should be in the open interval $(0, 1)$ to avoid numerical problems in the calculation of the factor α_j. Using the indicator values in P_j, a linear model is constructed to obtain its angle θ. In Line 7, we set the weight $w_j = w_j\, e^{-\alpha_j}$ if θ is strictly

Algorithm 3 UpdateWeights

Require: Weight vector $\boldsymbol{w}$; performance matrix P; learning matrix Ψ; time window size T_w; number of indicators k
Ensure: Updated $\boldsymbol{w}$
1: **for** $j = 1$ to k **do**
2: $e_j = \frac{w_j}{T_w} \sum_{i=1}^{T_w} \Psi_{ji}$
3: Validate that $e_j \in (0, 1)$
4: $\alpha_j = \frac{1}{2} \ln \left(\frac{1-e_j}{e_j} \right)$
5: Build linear performance model based on P_j
6: Get the angle θ_j of the linear model
7: $w_j = \begin{cases} w_j e^{-\alpha_j}, & \theta > 0 \\ w_j e^{\alpha_j}, & \text{otherwise} \end{cases}$
8: Validate that $w_j > 0$
9: **end for**
10: $w_j = \frac{w_j}{\sum_{i=1}^{k} w_i}, j \in \{1, \ldots, k\}$
11: **return** $\boldsymbol{w}$

positive, which implies an increasing quality of the population due to the use of the density estimator based on I_j. Otherwise, we set $w_j = w_j\, e^{\alpha_j}$. To avoid having EIB-DE composed of a single indicator, we do not allow the existence of zero weights. At last, all weights are normalized in Line 10 and the updated weight vector is returned.

4 Experimental Analysis

In this section, we analyze the performance of the proposed approach[3]. First, we compare EIB-MOEA with its average ranking version, i.e, an EIB-MOEA where the weights for the ensemble are the same for all indicators (denoted as avgEIB-MOEA) to show that the adaptive mechanism produces better quality results. Then, we perform an exhaustive analysis where we compare EIB-MOEA with SMS-EMOA, R2-EMOA, IGD$^+$-MaOEA, ϵ^+-MaOEA, and Δ_p-MaOEA, which are all steady-state MOEAs using density estimators based on the HV, R2, IGD$^+$, ϵ^+, and Δ_p indicators, respectively. In all test instances, each MOEA is independently executed 30 times.

4.1 Parameters Settings

We employ the benchmark functions DTLZ1, DTLZ2, DTLZ5, DTLZ7, WFG1, WFG2, WFG3, and WFG4, together with their corresponding minus versions proposed in [16] for two and three objective functions. We adopted these problems because they all have different search difficulties and Pareto front shapes. The number n of decision variables was set as follows. For DTLZ instances and their minus versions, $n = m + K - 1$, where m is the number of objective functions and $K = 5$ for DTLZ1, $K = 10$ for both DTLZ2 and DTLZ5, and $K = 20$

[3] The source code of EIB-MOEA is available at `http://computacion.cs.cinvestav.mx/~jfalcon/Ensemble/EIB-MOEA.html`.

for DTLZ7. Regarding the WFG and WFG^{-1} test problems, n was set to 24 and 26, for two- and three-objective instances and in both cases the number of position-related parameters was set to 2. For a fair comparison, all the MOEAs employ the same population size $\mu = 120$, and the same variation operators: simulated binary crossover (SBX) and polynomial-based mutation (PBM) [12] for all test instances. The crossover probability is set to 0.9, the mutation probability is $1/n$ (where n is the number of decision variables), and both the crossover and mutation distribution indexes are set to 20. We considered 50,000 function evaluations as the stopping criterion for all MOPs. We employ the achievement scalarizing function for the R2-based density estimator. In every generation, we employ the currently population's non-dominated solutions as the reference set required by IGD$^+$, ϵ^+, and Δ_p. Regarding EIB-MOEA and avgEIB-MOEA, we set $T_w = \mu$.

4.2 Experimental Results

For the performance assessment of EIB-MOEA, avgEIB-MOEA and the other IB-MOEAs, we used eight quality indicators: HV, HV relative deviation (HVRD), R2, IGD$^+$, ϵ^+, Δ_p, and, for diversity, we employed Riesz s-energy [17] and the Solow-Polasky Diversity indicator [18]. The indicator values for two- and three-objective instances of the DTLZ and DTLZ^{-1} test problems are shown with boxplots in Figures 1 and 2, respectively. The boxplots for the WFG and WFG^{-1} instances with two and three objective functions correspond to Figures 3 and 4, respectively. Figure 5 shows the statistical ranks obtained by each algorithm over all benchmark functions with respect to each considered indicator. For a given benchmark function, the rank corresponds to the number of algorithms that significantly outperform the algorithm under consideration with respect to a Mann-Whitney non-parametric statistical test with a p-value of 0.05 and a Bonferroni correction (a lower value is better). The complete numerical results related to Figure 5 are available at `http://computacion.cs.cinvestav.mx/~jfalcon/Ensemble/EIB-MOEA.html` due to space limitations.

Regarding the comparison of EIB-MOEA with avgEIB-MOEA, Figure 5 shows that the former gets better statistical ranks for all the considered indicators except for Δ_p. From these QIs, the increase in quality is more evident for the hypervolume indicator. This means that the online ensemble allows EIB-MOEA to produce approximation sets closer to the Pareto front. This is supported by the other convergence indicators. However, producing better convergent approximation sets is not strictly related to producing higher diversity, as shown by the Riesz s-energy and SPD values, where EIB-MOEA is hardly better than avgEIB-MOEA. Overall, these results support that EIB-MOEA performs better than avgEIB-MOEA. On the other hand, for the comparison of EIB-MOEA against the steady-state IB-MOEAs, Figure 5 shows that our proposed approach maintains a robust performance over all the considered QIs. Figures 1 to 4 illustrate that EIB-MOEA and SMS-EMOA obtained the best HV values. Overall, SMS-EMOA performs better on the original benchmark problems, but the quality of its approximate Pareto fronts is just slightly better than those produced by

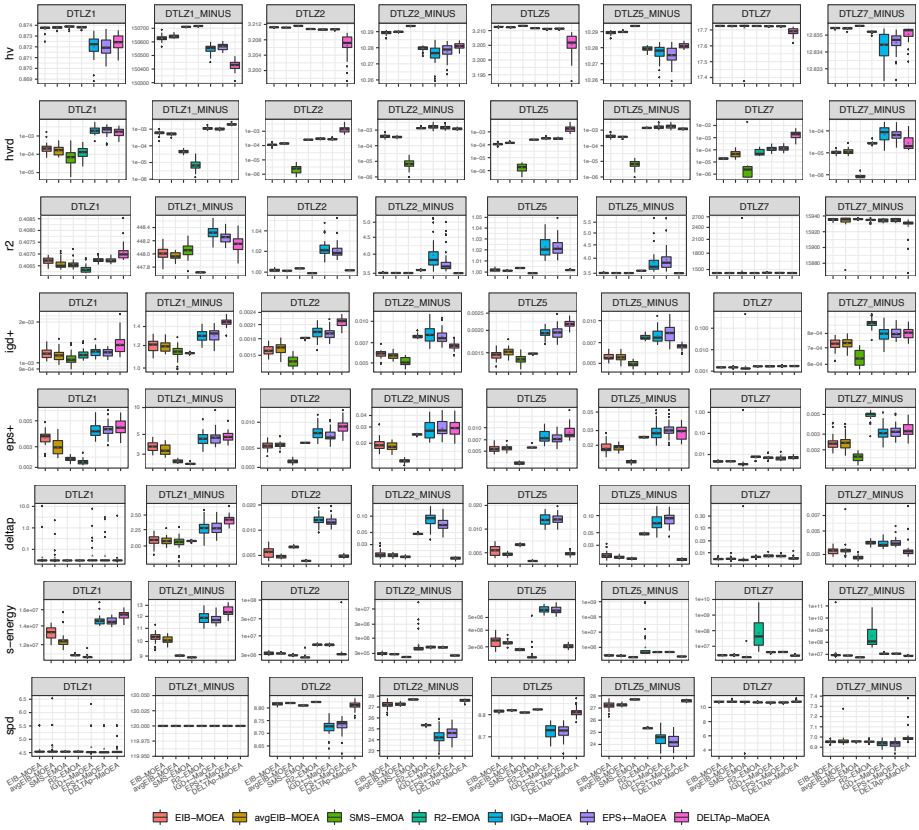

Fig. 1: Indicator values for two-objective DTLZ benchmark functions.

EIB-MOEA. In contrast, for the DTLZ^{-1} and WFG^{-1} test suites, EIB-MOEA significantly outperforms SMS-EMOA. This is because EIB-MOEA is able to produce Pareto front approximations with better distribution and coverage, due to the influence of all the indicators, for these modified problems, which increases the HV value. Moreover, for the original problems, EIB-MOEA still produces good approximations but with other type of distribution that does not maximize the hypervolume value as SMS-EMOA does. In consequence, EIB-EMOA performs more robustly than SMS-EMOA. Additionally, for IGD$^+$ and ϵ^+ which are QIs whose preferences are highly correlated to those of HV, Figure 5 shows a similar behavior as in the case of HV. This is also supported by the detailed boxplots reported for the different test problems. Regarding the R2 indicator, R2-EMOA presents the best results for MOPs whose Pareto front maps to the simplex shape; e.g., DTLZ1, DTLZ2, and WFG4. This behavior is expected since R2-EMOA uses a set of convex weight vectors [16]. However, for the DTLZ^{-1} and WFG^{-1} test suites, R2-EMOA does not perform well and EIB-MOEA presents

Fig. 2: Indicator values for three-objective DTLZ benchmark functions.

the best overall results. This indicates that the ensemble mechanism of EIB-MOEA allows to circumvent the weaknesses of the individual indicator-based density estimators, in this case the one based on R2. Finally, in terms of diversity, Figures 1–4 show that EIB-MOEA generates well-diversified approximation sets when dealing with MOPs whose Pareto front is irregular; i.e., different from the simplex shape. This is the case, for example, of WFG1, WFG1^{-1}, DTLZ1^{-1}, and DTLZ^{-1}. Nevertheless, EIB-MOEA is able to produce competitive results with respect to Riesz s-energy and SPD, while SMS-EMOA is the best-ranked algorithm for the former indicator and Δ_p-MaOEA is the best for the latter. As such, although EIB-MOEA is able to obtain very good HV values, there is still room for improvement in terms of diversity, e.g. by adding diversity-related indicators into the ensemble controlled by EIB-MOEA.

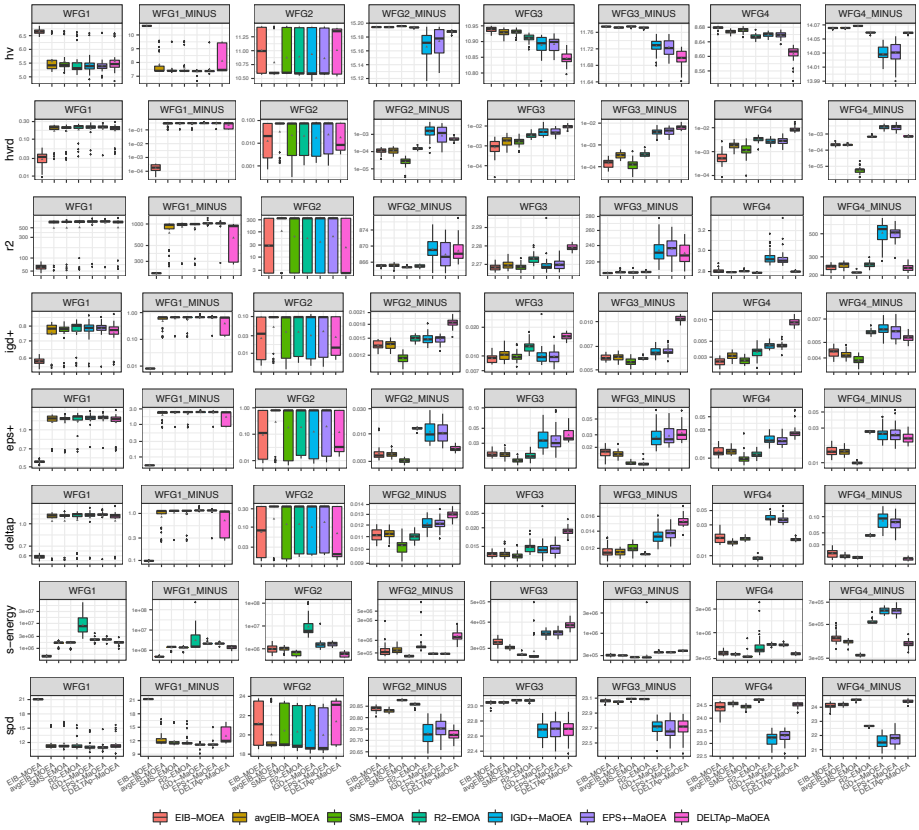

Fig. 3: Indicator values for two-objective WFG benchmark functions.

5 Conclusions and Future Work

In this paper, we explored the effectiveness of an ensemble indicator-based density estimator, using the AdaBoost algorithm. The proposed mechanism adapts the ensemble in an online fashion depending on the performance of the underlying density estimators based on the indicators HV, R2, IGD$^+$, ϵ^+, and Δ_p. The adaptive ensemble mechanism was embedded into a steady-state MOEA, giving rise to the EIB-MOEA. First, we showed that EIB-MOEA outperforms an average ranking EIB-MOEA that sets all the weights to the same value for the ensemble. Then, we compared EIB-MOEA with respect to SMS-EMOA, R2-EMOA, IGD$^+$-MaOEA, ϵ^+-MaOEA, and Δ_p-MaOEA. The experimental results showed that EIB-MOEA is able to maintain a robust performance with respect to multiple quality indicators. As part of our future work, we aim at studying the performance of a generational EIB-MOEA and at improving the learning mechanism for the ensemble. We would also like to assess the performance of our proposed EIB-MOEA in many-objective optimization problems.

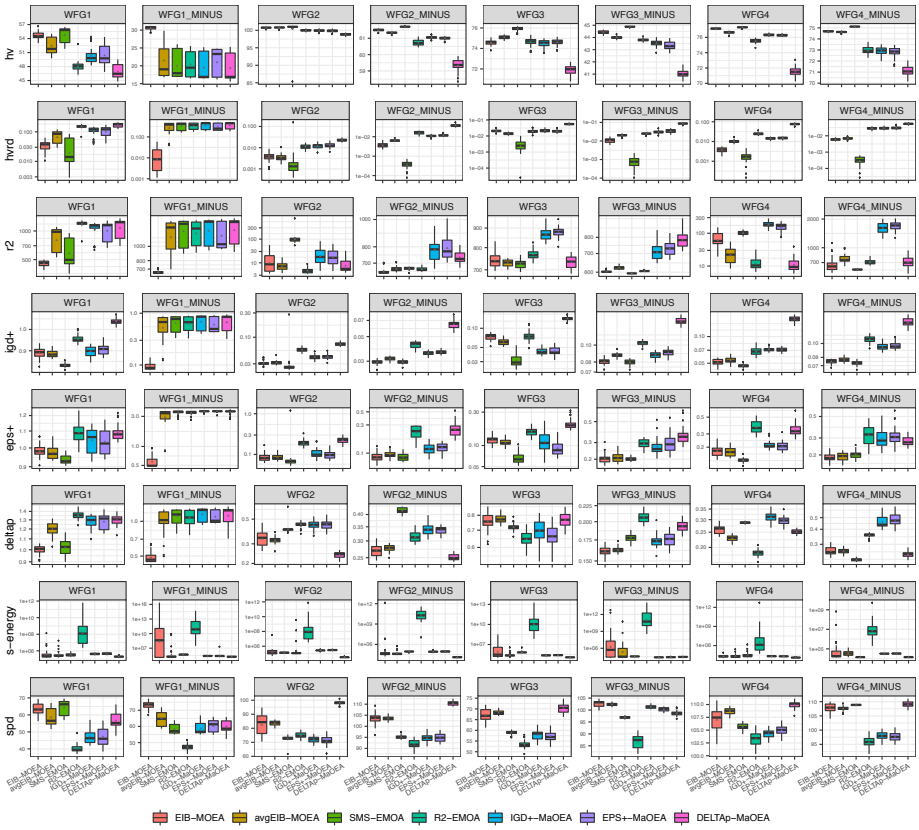

Fig. 4: Indicator values for three-objective WFG benchmark functions.

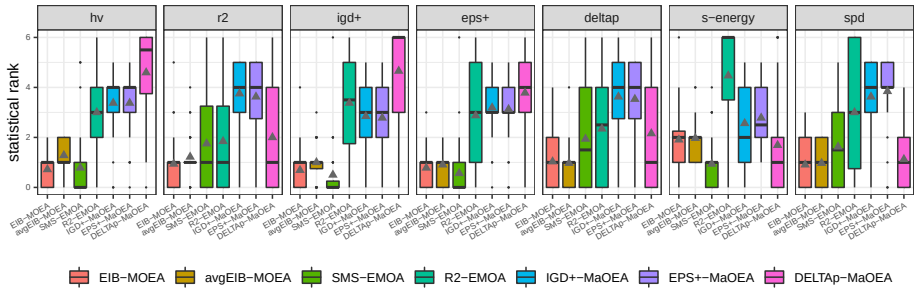

Fig. 5: Statistical ranks obtained by each algorithm over all benchmark functions with respect to each considered indicator.

References

1. Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. Many-Objective Evolutionary Algorithms: A Survey. *ACM Computing Surveys*, 48(1), September 2015.
2. Jesús Guillermo Falcón-Cardona and Carlos A. Coello Coello. Indicator-based Multi-Objective Evolutionary Algorithms: A Comprehensive Survey. *ACM Computing Surveys*, 53(2), March 2020.
3. Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys*, 52(2):26:1–26:38, March 2019.
4. Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.
5. Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.
6. Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.
7. Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.
8. Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, August 2012.
9. Jesús Guillermo Falcón-Cardona and Carlos A. Coello Coello. Convergence and Diversity Analysis of Indicator-Based Multi-Objective Evolutionary Algorithms. In *2019 Genetic and Evolutionary Computation Conference (GECCO'2019)*, pages 524–531, Prague, Czech Republic, 2019. ACM Press.
10. Dugh H. Phan and Junichi Suzuki. Boosting indicator-based selection operators for evolutionary multiobjective optimization algorithms. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 276–281. IEEE press, Nov 2011.
11. Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.
12. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
13. Dũng H. Phan, Junichi Suzuki, and Isao Hayashi. Leveraging Indicator-Based Ensemble Selection in Evolutionary Multiobjective Optimization Algorithms. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 497–504, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.
14. Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

15. Arnaud Liefooghe and Bilel Derbel. A Correlation Analysis of Set Quality Indicator Values in Multiobjective Optimization. In *2016 Genetic and Evolutionary Computation Conference (GECCO'2016)*, pages 581–588, Denver, Colorado, USA, 20-24 July 2016. ACM Press. ISBN 978-1-4503-4206-3.
16. Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190, April 2017.
17. D. P. Hardin and E. B. Saff. Discretizing Manifolds via Minimum Energy Points. *Notices of the AMS*, 51(10):1186–1194, November 2004.
18. Michael T.M. Emmerich, André H. Deutz, and Johannes W. Kruisselbrink. On Quality Indicators for Black-Box Level Set Approximation. In Emilia Tantar, Alexandru-Adrian Tantar, Pascal Bouvry, Pierre Del Moral, Pierrick Legrand, Carlos A. Coello Coello, and Oliver Schütze, editors, *EVOLVE - A bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, chapter 4, pages 157–185. Springer-Verlag. Studies in Computational Intelligence Vol. 447, Heidelberg, Germany, 2013. 978-3-642-32725-4.