# On the effect of the cooperation of indicator-based multiobjective evolutionary algorithms

Falcón-Cardona, J.G.; Ishibuchi, H.; Coello Coello, C.A.; Emmerich, M.T.M.

# On the Effect of the Cooperation of Indicator-Based Multiobjective Evolutionary Algorithms

Jesús Guillermo Falcón-Cardona, *Member, IEEE*, Hisao Ishibuchi, *Fellow, IEEE*, Carlos A. Coello Coello, *Fellow, IEEE*, and Michael Emmerich

***Abstract*—For almost 20 years, quality indicators (QIs) have promoted the design of new selection mechanisms of multiobjective evolutionary algorithms (MOEAs). Each indicator-based MOEA (IB-MOEA) has specific search preferences related to its baseline QI, producing Pareto front approximations with different properties. In consequence, an IB-MOEA based on a single QI has a limited scope of multiobjective optimization problems (MOPs) in which it is expected to have a good performance. This issue is emphasized when the associated Pareto front geometries are highly irregular. In order to overcome these issues, we propose here an island-based multiindicator algorithm (IMIA) that takes advantage of the search biases of multiple IB-MOEAs through a cooperative scheme. Our experimental results show that the cooperation of multiple IB-MOEAs allows IMIA to perform more robustly (considering several QIs) than the panmictic versions of its baseline IB-MOEAs as well as several state-of-the-art MOEAs. Additionally, IMIA shows a Pareto-front-shape invariance property, which makes it a remarkable optimizer when tackling MOPs with complex Pareto front geometries.**

***Index Terms*—Island model, multiobjective optimization, quality indicators (QIs), selection mechanism.**

## I. Introduction

QUALITY indicators (QIs) have been especially noteworthy in the evolutionary multiobjective optimization field [1]. QIs evaluate the quality of Pareto front approximations[1] generated by multiobjective evolutionary algorithms (MOEAs) [2], focusing on three specific aspects: 1) convergence toward the Pareto front; 2) diversity of solutions; and 3) the coverage of the Pareto front. Mathematically, a QI is a set function that assigns a real value to one or more approximation sets simultaneously. This implies that QIs impose a total order in the set $\Psi$ of all approximation sets, depending on their particular preferences [3]. From the plethora of currently available QIs [1], the unary indicators that measure convergence and coverage at the same time by a single value, without requiring any knowledge of the true Pareto front, have a preponderant position among them and have been extensively used to compare MOEAs' performance. Remarkable examples of unary convergence indicators are the hypervolume indicator (HV) [4], $R2$ indicator [5], inverted generational distance (IGD) [6], IGD plus ($\text{IGD}^+$) [7], additive $\epsilon$ indicator ($\epsilon^+$) [3], and averaged Hausdorff distance ($\Delta_p$) [8].

In addition to the assessment of approximation sets, QIs have also promoted the design of selection mechanisms of MOEAs, giving rise to the so-called indicator-based MOEAs (IB-MOEAs) [9]. IB-MOEAs include the incorporation of QIs into environmental selection, density estimation, and archive update rules. The underlying idea of these indicator-based mechanisms is to select from a set of $N$ solutions a subset of size $k < N$ such that the indicator value is optimized. In the context of MOEA design, IB-MOEAs have remarkable properties. The indicator-based mechanisms allow increasing the selection pressure, which impacts in solving multiobjective optimization problems (MOPs) with more than three objective functions, i.e., the so-called many-objective optimization problems (MaOPs). This is an important property since Pareto-based MOEAs (i.e., MOEAs whose environmental selection is based on the Pareto dominance relation)[2] have poor performance when solving MaOPs due to the dilution of the selection pressure related to the exponential increase of mutually nondominated solutions in high-dimensional objective spaces. Additionally, due to the preferences of each indicator,

Manuscript received August 17, 2020; revised December 22, 2020; accepted February 15, 2021. Date of publication February 23, 2021; date of current version July 30, 2021. The work of Carlos A. Coello Coello was supported in part by CONACyT under Project 1920; in part by 2018 SEP-Cinvestav Grant (application no. 4); and in part by the Basque Government through the Program by the Spanish Ministry of Science under Grant BERC 2018-2021. *(Corresponding author: Hisao Ishibuchi.)*

Jesús Guillermo Falcón-Cardona is with the Computer Science Department, CINVESTAV-IPN, Mexico City 07360, Mexico (e-mail: jfalcon@computacion.cs.cinvestav.mx).

Hisao Ishibuchi is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hisao@sustech.edu.cn).

Carlos A. Coello Coello is with the Computer Science Department, CINVESTAV-IPN, Mexico City 07360, Mexico, also with the Department of Combinatorial Optimization, Basque Center for Applied Mathematics, 48009 Bilbao, Spain, and also with the Ikerbasque, 48009 Bilbao, Spain (e-mail: ccoello@cs.cinvestav.mx).

Michael Emmerich is with the Multiobjective Optimization and Decision Analysis Group, Leiden Institute of Advanced Computer Science, 2333 CA Leiden, The Netherlands, and also with the Multiobjective Optimization Group, Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland (e-mail: m.t.m.emmerich@liacs.leidenuniv.nl).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2021.3061545.

Digital Object Identifier 10.1109/TEVC.2021.3061545

[1]A Pareto front approximation, or approximation set, denoted as $\mathcal{A}$, is a set of solutions in objective space (especially mutually nondominated solutions) that aims to approximate a Pareto front.

[2]Given $\vec{x}, \vec{y} \in \mathbb{R}^n$, $\vec{x}$ is said to Pareto dominate $\vec{y}$ (denoted as $\vec{F}(\vec{x}) \prec \vec{F}(\vec{y})$) if and only if $f_i(\vec{x}) \leq f_i(\vec{y})$ for all $i = 1, \ldots, m$ and there is at least an index $j \in \{1, \ldots, m\}$ such that $f_j(\vec{x}) < f_j(\vec{y})$. In case that $f_i(\vec{x}) \leq f_i(\vec{y})$ for all $i = 1, \ldots, m$, then $\vec{x}$ is said to weakly Pareto dominate $\vec{y}$ (denoted as $\vec{F}(\vec{x}) \preceq \vec{F}(\vec{y})$).

IB-MOEAs generate approximation sets with different distributions for a specific Pareto front geometry [10], [11]. In the specialized literature, there is a wide range of IB-MOEAs, each one having particular advantages and drawbacks [9].

QIs have particular preferences that result in a different order of the approximation sets in $\Psi$ [12]. For instance, HV prefers solutions on the boundaries of concave Pareto fronts (when the reference point is not close to the Nadir point). In contrast, $R2$ tends to prefer solutions uniformly distributed (more evidently for concave and linear Pareto front shapes) because of the regular use of a set of convex weight vectors[3] for its calculation. Hence, if a QI is employed to guide the selection process of an MOEA, the generated Pareto front approximations will inherit characteristics related to it [10]. In consequence, an IB-MOEA using a single baseline QI will have a good performance on some MOPs and bad performance on some others. This situation resembles the No-Free Lunch theorem [13]. To overcome this issue, a possible solution is to compensate for the weaknesses of an indicator with the strengths of others, which motivates the creation of multiindicator-based MOEAs (MIB-MOEAs). Additionally, the combination of QIs' preferences in MIB-MOEAs could bring collateral improvements in the design of MOEAs whose performance does not depend on the Pareto front shapes as stated in [14].

Currently, there are a few MIB-MOEAs in the specialized literature [14]–[22]. According to the available proposals, MIB-MOEAs have been mainly designed in two ways: 1) using a single selection mechanism that incorporates the search biases of multiple QIs [15]–[17], [21], [22] and 2) utilizing simultaneously multiple selection mechanisms, each one based on a single QI [14], [18]–[20]. From these approaches, a remarkable one is the cooperative MIB-MOEA (cMIB-MOEA) [20]. The core idea of cMIB-MOEA is the cooperation of five steady-state IB-MOEAs[4] based on HV, $R2$, $\text{IGD}^+$, $\epsilon^+$, and $\Delta_p$ under the master–slave paradigm to exploit the search properties of the selected algorithms. All the IB-MOEAs are connected bidirectionally to a master node that maintains a global Pareto front approximation in an archive. During $f_{\text{mig}}$ iterations, the IB-MOEAs are independently executed to evolve in isolation their populations. After this period, each IB-MOEA sends its whole population to the master node where the subpopulations are combined with the current contents of the global archive to obtain the set of nondominated solutions. If the maximum archive size is exceeded, a pruning process is performed, aiming to get the set of solutions that minimize the Riesz $s$-energy [25]. Finally, a migration procedure is performed to improve the diversity of IB-MOEAs. In each island, the individual contributions to the associated indicator of all the solutions are calculated, and $n_{\text{mig}}$ ones with the worst contribution values are replaced by the same number of solutions randomly selected from the global archive. The only restriction is that the replacing solutions must had been generated by other IB-MOEAs. cMIB-MOEA exhibited better performance than its baseline IB-MOEAs and a Pareto front shape invariant performance emerged from the cooperative scheme.

In this article, we present an extension of cMIB-MOEA, denoted as island-based multiindicator algorithm (IMIA), to study the effect of the cooperation of multiple IB-MOEAs. Unlike cMIB-MOEA that uses a master–slave model and it is not parallelized, IMIA is a parallel MOEA that strictly follows the island model. Five steady-state IB-MOEAs, each one using a density estimator based on HV, $R2$, $\text{IGD}^+$, $\epsilon^+$, and $\Delta_p$, evolve micropopulations in isolation and, after a predefined number of iterations, they synchronously communicate through an improved migration process, adopting a user-defined communication topology. Additionally, instead of adopting a master node with a global archive to manage a centralized Pareto front approximation, IMIA uses a distributed approach where each island maintains its own approximation set. As a result, this allows IMIA to increase the selection pressure when tackling MaOPs and increase the diversity of solutions in comparison with cMIB-MOEA that has some problems in MaOPs. Finally, IMIA improves the migration method of cMIB-MOEA by iteratively replacing the worst contributing solutions to the given indicator by solutions coming from IB-MOEAs indicated in the communication topology. Overall, these improvements make IMIA a better optimizer than cMIB-MOEA. The experimental results show that IMIA takes advantage of the properties of each IB-MOEA to increase its exploration ability, producing Pareto front approximations with a high-quality degree of convergence, diversity, and coverage, regardless of the Pareto front geometry of the MOP being solved. The main contributions of this article are the following.

1) We propose the first island-based IB-MOEA, called IMIA, whose core idea is the cooperation of multiple IB-MOEAs. Moreover, we implement IMIA using the multicore parallel scheme to reduce its computational cost.
2) We show that due to the cooperation of multiple IB-MOEAs, IMIA can perform more robustly (under seven QIs: HV, $R2$, $\text{IGD}^+$, $\epsilon^+$, $\Delta_p$, Riesz $s$-energy, and the Solow–Polasky Diversity [26]) than the panmictic versions of its baseline IB-MOEAs. In this regard, we define a robust performance as the capacity of an MOEA to consistently obtain the best results under several QIs (measuring convergence and diversity) for MOPs with different Pareto front shapes and scaling the dimensionality of the objective space.
3) IMIA generates Pareto front approximations with high diversity (especially for MOPs with highly irregular Pareto front geometries) in comparison with its baseline panmictic IB-MOEAs and several state-of-the-art MOEAs specifically designed to tackle different Pareto front shapes.

The remainder of this article is organized as follows. Section II provides the mathematical definitions of QIs

[3]A vector $\vec{w} \in \mathbb{R}^m$ is a convex weight vector if $\forall i = 1, \ldots, m, w_i \geq 0$ and $\sum_{i=1}^{m} w_i = 1$.

[4]The IB-MOEAs are based on the framework of the $\mathcal{S}$-metric selection evolutionary multiobjective algorithm (SMS-EMOA) [23]. Hence, they use a steady-state selection where the nondominated sorting algorithm [24] and an indicator-based density estimator are the primary and secondary selection criteria, respectively.

employed in the proposal. An overview of the current multiindicator-based MOEAs is provided in Section III. Our proposed approach is outlined in Section IV. Section V presents the experimental results. Finally, Section VI concludes this article and sketches the future research directions.

## II. Background

In this section, we formally define an MOP and the QIs: HV, $R2$, $\mathrm{IGD}^+$, $\epsilon^+$, $\Delta_p$, Riesz $s$-energy ($E_s$), and the Solow Polasky diversity (SPD). In all cases, let $\mathcal{A}$ be an approximation set and $\mathcal{Z}$ be a reference set. $m$ is the dimensionality of the objective space.

Throughout this article, we focus, without loss of generality, on unconstrained multiobjective optimization problems for minimization [2], which are defined as follows:

$$\min_{\vec{x}\in\mathcal{X}} \vec{F}(\vec{x}) := \left[f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})\right]^T \tag{1}$$

where $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ is the $n$-dimensional vector of decision variables and $\mathcal{X} \subseteq \mathbb{R}^n$; $f_j : \mathcal{X} \to \mathbb{R}, j = 1, \ldots, m$ are the objective functions.

*Definition 1 (Unary Quality Indicator):* A unary quality indicator $I$ is a function $I : \Psi \to \mathbb{R}$, which assigns a real value to a Pareto front approximation. $\Psi$ is the set of all approximation sets.

*Definition 2 (Hypervolume Indicator [4]):* Given an anti-optimal reference point $\vec{r} \in \mathbb{R}^m$, HV is defined as follows:

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

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

*Definition 3 (Unary $R2$ Indicator [5]):* The unary $R2$ indicator is defined as follows:

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

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

*Definition 4 ($IGD^+$ Indicator [7]):* The $\mathrm{IGD}^+$ for minimization is defined as follows:

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

where $d^+(\vec{a}, \vec{z}) = \sqrt{\sum_{i=1}^m (\max\{a_i - z_i, 0\})^2}$.

*Definition 5 (Unary $\epsilon^+$ Indicator [3]):* The unary $\epsilon^+$ 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:

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

To define the averaged Hausdorff distance ($\Delta_p$), it is first necessary to introduce a variant of the indicators generational distance (GD) [27] and IGD [6], denoted as $\mathrm{GD}_p$ and $\mathrm{IGD}_p$, respectively.

*Definition 6 ($GD_p$ Indicator [8]):*

$$\mathrm{GD}_p(\mathcal{A}, \mathcal{Z}) = \left(\frac{1}{|\mathcal{A}|} \sum_{\vec{a}\in\mathcal{A}} d(\vec{a}, \mathcal{Z})^p\right)^{1/p} \tag{6}$$

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

*Definition 7 ($IGD_p$ Indicator [8]):*

$$\mathrm{IGD}_p(\mathcal{A}, \mathcal{Z}) = \mathrm{GD}_p(\mathcal{Z}, \mathcal{A}) = \left(\frac{1}{|\mathcal{Z}|} \sum_{\vec{z}\in\mathcal{Z}} d(\vec{z}, \mathcal{A})^p\right)^{1/p}. \tag{7}$$

*Definition 8 (Averaged Hausdorff Distance Indicator ($\Delta_p$) [8]):* For a given $p > 0$, $\Delta_p$ is defined as follows:

$$\Delta_p(\mathcal{A}, Z) = \max\left\{\mathrm{GD}_p(\mathcal{A}, Z), \mathrm{IGD}_p(\mathcal{A}, Z)\right\}. \tag{8}$$

As with IGD, the $\Delta_p$ indicator requires an aspiration set. $\Delta_p$ was proposed to eliminate some shortcomings of IGD such as its sensitivity to the cardinality of sets [8].

*Definition 9 (Indicator Contribution):* Let $\mathcal{I}$ be any indicator in the set $\{\mathrm{HV}, R2, \mathrm{IGD}^+, \epsilon^+, \Delta_p\}$. The individual contribution $C$ of a solution $\vec{a} \in \mathcal{A}$ to the indicator value is given as follows:

$$C_{\mathcal{I}}(\vec{a}, \mathcal{A}) = |\mathcal{I}(\mathcal{A}) - \mathcal{I}(\mathcal{A} \setminus \{\vec{a}\})|. \tag{9}$$

*Definition 10 (Riesz $s$-Energy [25]):* For a given $s > 0$, the Riesz $s$-energy indicator is defined as follows:

$$E_s(\mathcal{A}) = \sum_{\vec{a}\in\mathcal{A}} \sum_{\vec{b}\in\mathcal{A}\setminus\{\vec{a}\}} \left\|\vec{a} - \vec{b}\right\|^{-s} \tag{10}$$

where $\|\cdot\|$ represents the Euclidean distance. As $s \to \infty$, $E_s$ prefers more uniform solutions. This indicator measures the even distribution of a set of points in $d$-dimensional manifolds.

*Definition 11 (Riesz $s$-Energy Individual Contribution):* The individual contribution $C$ of a solution $\vec{a} \in \mathcal{A}$ to the Riesz $s$-energy indicator is as follows:

$$C_{E_s}(\vec{a}, \mathcal{A}) = \frac{1}{2}\left[E_s(\mathcal{A}) - E_s(\mathcal{A} \setminus \{\vec{a}\})\right]. \tag{11}$$

Unlike (9), $C_{E_s}$ involves the term 1/2 since $\left\|\vec{a} - \vec{b}\right\| = \left\|\vec{b} - \vec{a}\right\|$ for all $\vec{a}, \vec{b} \in \mathcal{A}$.

*Definition 12 (Solow Polasky Diversity [26]):* Let $\mathbf{C} \in \mathbb{R}^{N\times N}$ be a full-rank matrix. The $(i, j)$-element of $\mathbf{C}$ is defined by $c_{ij} = e_2^{-\theta\cdot\|\vec{a}_i - \vec{a}_j\|}$, $i, j = 1, \ldots, N$ where $\vec{a}_i, \vec{a}_j \in \mathcal{A}$ and $\theta > 0$ is a user-defined parameter. Each $c_{ij}$ denotes the correlation between $\vec{a}_i$ and $\vec{a}_j$. If two points are of the same species, the correlation is one. Let $\mathbf{M} = \mathbf{C}^{-1}$. Hence, the SPD is given by

$$\mathrm{SPD}(\mathcal{A}) = \sum_{i=1}^N \sum_{j=1}^N m_{ij}. \tag{12}$$

According to Basto-Fernandes *et al.* [26], $\mathrm{SPD}(\mathcal{A})$ tends to $N$ if the distance between all species tends to be very large. In contrast, $\mathrm{SPD}(\mathcal{A})$ tends to one if species are very similar with respect to each other. The parameter $\theta$ indicates how fast the population tends to $N$ when the distances increase.

## III. Previous Related Work

In this section, we briefly review some MIB-MOEAs and MOEAs that were designed to tackle MOPs with different Pareto front shapes.

### A. Multiindicator-Based MOEAs

To the best of our knowledge, the first MIB-MOEA was proposed by Phan and Suzuki [15] in which multiple indicator-based mating selection mechanisms were combined, using the AdaBoost algorithm. The proposal, denoted as boosting indicator-based evolutionary algorithm (BIBEA), aimed to select potential parents for crossover, avoiding the bias of a single selection mechanism, increasing the convergence speed of the algorithm and producing well-diversified Pareto fronts. In further work, Phan *et al.* [16] proposed BIBEA-P, which improves the previous multiindicator mating selection scheme by using PDI-Boosting instead of AdaBoost. The authors also proposed a multiindicator environmental selection mechanism, ensembling HV and $\epsilon^+$, among other QIs.

In 2016, Li *et al.* [17] introduced the stochastic ranking-based multiindicator algorithm (SRA) that aims to balance the search biases of the indicator $\epsilon^+$ and the shift-based density estimator (SDE) [28]. SRA uses the stochastic ranking algorithm as its environmental selection mechanism as well as to balance the search biases of $\epsilon^+$ and SDE. SRA exhibited promising performance on different benchmark problems. However, the authors stated that further studies are necessary.

In 2017, Hernández Gómez and Coello Coello [18] proposed to use an environmental selection mechanism based on $R2$ in conjunction with the Riesz $s$-energy as the backbone of a density estimator to break the ties of the former mechanism by promoting good diversity. The simultaneous utilization of both schemes allows to improve the diversity of solutions due to the properties of the Riesz $s$-energy, while keeping a high selection pressure due to the $R2$ indicator.

Focusing on taking advantage of the search properties of $R2$, $\text{IGD}^+$, $\epsilon^+$, and $\Delta_p$, Falcón-Cardona and Coello Coello [19] proposed a hyperheuristic method that, according to the current state of the evolutionary process, selects the best-suited indicator-based density estimator (IB-DE), using a Markov chain. The so-called multiindicator hyperheuristic (MIHPS) gave insights about the competition of IB-DEs and the way in which each one is preferred according to the state of the search. In 2019, Falcón-Cardona *et al.* [14] proposed a density estimator that, depending on a statistical analysis of convergence, switches between an $\text{IGD}^+$-based density estimator to increase diversity and a Riesz $s$-energy-based density estimator to promote diversity in the case that the convergence behavior is stagnated. Due to the use of this mechanism, a Pareto front shape invariance property emerges in an MOEA. In 2020, Falcón-Cardona *et al.* [21] proposed a density estimator that ensembles five IB-DEs, using the AdaBoost algorithm as in the case of BIBEA. Unlike BIBEA that requires an offline learning process, the ensemble IB-DE performs an online learning to adjust the weights of the linear combination of IB-DEs. This proposal showed robust performance under several QIs, i.e., due to its use, an MOEA is able to obtain the best results with respect to different indicators.

### B. MOEAs Focused on Irregular Pareto Front Shapes

AR-MOEA [29] uses a density estimator based on the enhanced IGD (IGD-NS) which, unlike the original IGD, promotes in a better way the convergence and uniformity of solutions. However, the main contribution of AR-MOEA is the utilization of an adaptative technique that, at each generation, creates a new reference set, based on the solutions stored in an external archive, aiming to approximate the current Pareto front shape. This reference set is employed to calculate the IGD-NS contributions of all the solutions. This adaptive reference set, in conjunction with IGD-NS, allows AR-MOEA to effectively tackle MOPs with different Pareto front shapes.

Balancing convergence and diversity is the underlying idea of GrEA [30]. To this aim, GrEA exploits a grid-based scheme to increase the selection pressure toward the Pareto front while maintaining a good diversity of solutions. Compared to other grid-based MOEAs, GrEA focuses on each individual instead of the whole grid and it employs three grid-based selection criteria. Moreover, an adaptive mechanism is applied to shape the neighborhood of individuals, promoting a better diversity. Finally, both the neighborhood structure and the three grid-based selection criteria are used to adjust the fitness values of the solutions.

SPEA2+SDE [28] combines SPEA2 [31] with the SDE [28]. SDE is a general method that can be embedded into any distance-based density estimator to increase the selection pressure and to preserve a good diversity of solutions which is desirable when solving MaOPs. For this sake, SDE adjusts the position of solutions according to their relative proximity to the Pareto front, by using the $d^+$ distance of $\text{IGD}^+$ instead of the Euclidean distance. This slight modification allows a distance-based density estimator to take into account convergence and diversity information in order to increase the selection pressure.

Two_Arch2 [32] is a hybrid MOEA that uses two subpopulations: 1) one dedicated to maintain convergence and 2) the other to preserve diversity. Two_Arch2 was especially designed to tackle MaOPs. The convergence subpopulation is updated based on the $\epsilon^+$ indicator. The other subpopultion aims to maintain diversity by using an update rule based on an $L_{1/m}$ norm, where $m$ is the number of objective functions. Both subpopulations interact to produce a Pareto front approximation with both convergence and diversity properties regardless of the Pareto front geometry.

### C. State-of-the-Art MOEAs

Currently, big data optimization problems are one important research field. To deal with these problems Yi *et al.* [33] proposed an adaptive mutation operator, embedded into the NSGA-III [34] since the variation operators importantly influence the performance of MOEAs on MOPs with many decision variables. Following this attempt of solution where the genetic operators have a decisive role, Yi *et al.* [35] benchmarked the performance of three crossover operators,

using NSGA-III, on a human electroencephalogram signal processing problem, which is a large-scale MOP. The use of different crossover operators allowed NSGA-III to present a better performance when solving large-scale MOPs. More recently, Gu and Wang [36] and Zhang *et al.* [37] introduced the information feedback models (which is the use of historical information of individuals to update the process of the current generation) to deal with large-scale MOPs. This methodology was used to improve the performance of both NSGA-III and MOEA/D [38], generating significant improvements. In addition to large-scale MOPs, other important and demanding problems are the interval MOPs (IMOPs) since to obtain a reasonable good solution, a large number of function evaluations are required. Sun *et al.* [39] incorporated several local searches to increase the performance of an MOEA by improving its exploration skills. These local search mechanisms are activated only when the underlying MOEA reaches a specific hypervolume value. In consequence, a significant performance improvement is achieved. Another important technique to solve large-scale MOPs is the use of multiple populations since they allow to explore different regions of the search space. Tian *et al.* [40] proposed a multipopulation MOEA to deal with large-scale multimodal MOPs. The proposed approach guides the multiple populations using adaptive search directions to provide efficient convergence in the huge search space, differentiating the plethora of regions. Last but not least, dynamic MOPs are also challenging problems that deserve the attention from the evolutionary multiobjective optimization community. Under this direction, the use of multiobjective particle swarm optimizers (MOPSOs) is a viable alternative. Kouka *et al.* [41] designed an MOPSO to tackle dynamic MOPs where the key contribution is the use of multiple populations and cooperative agents that share knowledge to deal with the changing search environment. Their experimental results showed the effectiveness of this approach.

## IV. Island-Based Multiindicator Algorithm

The proposed IMIA algorithm is an island-based MOEA where in each island an steady-state IB-MOEA (following the framework of SMS-EMOA) evolves a micropopulation[5] in isolation during a given number of iterations ($f_{\text{mig}}$ iterations). After that, a synchronous migration process is performed where each island sends $n_{\text{mig}}$ solutions to each island indicated by a connection topology. Additionally, due to the use of micropopulations, each island maintains an external archive where the best-found solutions are stored to preserve high diversity. In the following, we broadly describe the general framework of IMIA, the generic steady-state IB-MOEA employed in each island, the migration process, and the management of the external archives.

[5]The decision of using micropopulations is because Hernández-Gómez *et al.* [42] found that the computational cost of SMS-EMOA when solving MaOPs does not considerably grow if populations of no more than 15 individuals are used.

**Algorithm 1** IMIA General Framework

**Require:** Set of indicators $\mathcal{I} = \{I_1, \ldots, I_k\}$; Population size $\mu$; migration frequency $f_{\text{mig}}$; number of solutions to migrate $n_{\text{mig}}$; topology matrix $M$.
**Ensure:** Pareto front approximation $\mathcal{A}$.
1: Randomly initialize $P_j, j = 1, \ldots, k$ with $|P_j| = \mu/k$
2: $A_j = \text{Non-dominated}\left(\bigcup_{i=1}^{k} P_i\right), j = 1, \ldots, k$
3: **while** stopping criterion is not fulfilled **do**
4: **parallel for** $j = 1$ to $k$ **do**
5: $\{P_j, A_j\} = \text{IB-MOEA}(P_j, I_j, A_j, \mu, f_{\text{mig}}, n_{\text{mig}}, M)$
6: **end parallel for**
7: **end while**
8: $\mathcal{A} = \bigcup_{j=1}^{k} A_j \cup P_j$
9: $\mathcal{A} = \text{Non-dominated}(\mathcal{A})$
10: **if** $|\mathcal{A}| > \mu$ **then**
11: Obtain $\vec{z}^*$ and $\vec{z}^{\text{nad}}$ from $\mathcal{A}$ and normalize it
12: **end if**
13: **while** $|\mathcal{A}| > \mu$ **do**
14: $\vec{a}_{\text{worst}} = \arg\max_{\vec{a} \in \mathcal{A}} C_{E_s}(\vec{a}, \mathcal{A})$
15: $\mathcal{A} = \mathcal{A} \setminus \{\vec{a}_{\text{worst}}\}$
16: **end while**
17: **return** $\mathcal{A}$

### A. General Framework

The general framework of IMIA is presented in Algorithm 1. To execute IMIA, the user needs to provide the indicator $I_j$ that each island $j = 1, \ldots, k$ will use; the size $\mu$ of the approximation set to be generated; and the migration parameters, namely, $f_{\text{mig}}$, $n_{\text{mig}}$, and the topology matrix $M$ (or adjacency matrix). In line 1, all the subpopulations $P_j$ are randomly initialized and then each archive $A_j$ is set to have the globally nondominated solutions. The main loop of IMIA (lines 4–8) consists of the parallel execution of the IB-MOEAs (described in Algorithm 2) where they communicate synchronously to migrate solutions after being executed $f_{\text{mig}}$ iterations. This process continues until the stopping criterion is met. In line 9, all the subpopulations and archives are merged in a single set $\mathcal{A}$ from which we obtain the nondominated solutions. If the cardinality of $\mathcal{A}$ is greater than $\mu$, the solutions in $\mathcal{A}$ are normalized so that we iteratively reduce its cardinality by deleting at each iteration the worst contributing solution to the Riesz $s$-energy until $\mathcal{A}$ has $\mu$ individuals [43]. Finally, $\mathcal{A}$ is returned as the approximation set.

### B. Generic Steady-State IB-MOEA

Algorithm 2 introduces the generic steady-state IB-MOEA (following the SMS-EMOA framework [23]) that is executed on every island. Each IB-MOEA requires seven inputs for its execution: the population to be evolved, the baseline indicator $I$ employed in the density estimator, and the local archive with the given maximum size that is set to $\mu$, $f_{\text{mig}}$, $n_{\text{mig}}$, and $M$. This generic IB-MOEA does not have initialization instructions since $P$ and $\mathcal{A}$ were initialized in lines 1 and 2 from Algorithm 1, respectively. Hence, this allows IB-MOEAs to be executed iteratively in the main loop of IMIA. The main loop of the IB-MOEA is executed in lines 2–17 for $f_{\text{mig}}$ iterations. First, a new offspring solution is generated from $P$, using roulette-wheel parent selection, simulated binary crossover

**Algorithm 2** Generic Steady-State IB-MOEA

**Require:** Population $P$; indicator $I$; local archive $\mathcal{A}$; maximum archive size $\mu$; migration frequency $f_{\text{mig}}$; number of solutions to migrate $n_{\text{mig}}$; topology matrix $M$.
**Ensure:** Updated population $P$ and local archive $\mathcal{A}$

$g = 0$
**while** $g < f_{\text{mig}}$ **do**
  Generate offspring $\vec{q}$ from population $P$
  $Q = P \cup \{\vec{q}\}$
  Obtain $\vec{z}^*$ and $\vec{z}^{\text{nad}}$ from $Q$ and normalize it
  $\{R_1, \ldots, R_t\} = \text{Non-dominated-sorting}(Q)$
  **if** $|R_t| > 1$ **then**
    $\vec{r}_{\text{worst}} = \arg\min_{\vec{r} \in R_t} C_I(\vec{r}, R_t)$
  **else**
    $\vec{r}_{\text{worst}}$ is the single solution in $R_t$
  **end if**
  **if** $\vec{q} \neq \vec{r}_{\text{worst}}$ **then**
    $\mathcal{A} = \text{Insert}(\mathcal{A}, \vec{q}, \mu)$
  **end if**
  $P = Q \setminus \{\vec{r}_{\text{worst}}\}$
  $g = g + 1$
**end while**
$\{P, \mathcal{A}\} = \text{Migration}(P, I, \mathcal{A}, \mu, f_{\text{mig}}, n_{\text{mig}}, M)$
**return** $\{P, \mathcal{A}\}$

(SBX), and polynomial-based mutation (PBM) [24]. This solution is added to $P$ to generate the set $Q$ that is then normalized and processed by the nondominated sorting algorithm [24] to create a set of dominance layers $\{R_1, \ldots, R_t\}$. If the last layer $R_t$ contains more than one solution, then the worst contributing solution $\vec{r}_{\text{worst}} \in R_t$ to $I$ is identified. Otherwise, $\vec{r}_{\text{worst}}$ is set to be the sole solution in $R_t$. In case that $\vec{r}_{\text{worst}}$ is not the newly created solution, the latter is inserted in the local archive $\mathcal{A}$ using Algorithm 3. Finally, $\vec{r}_{\text{worst}}$ is removed from $Q$ to shape the next population $P$ and the iteration counter $g$ is increased by one. Once the main loop is broken, the migration process of Algorithm 4 is performed. Since the migration is a blocking process due to its synchronous design, IB-MOEA will wait until all the immigrant solutions have arrived. Finally, the updated $P$ and $\mathcal{A}$ are returned.

### C. Archive Management

The external archive (Algorithm 3) is managed by two selection criteria: 1) Pareto dominance and 2) Riesz $s$-energy minimization [43]. A solution $\vec{r}$ to be inserted is first tested using the Pareto dominance relation against all the individuals in $\mathcal{A}$. Every time that $\vec{r}$ dominates a solution $\vec{a} \in \mathcal{A}$, the latter is removed from $\mathcal{A}$. However, if at least one solution in the archive weakly dominates $\vec{r}$, then the process is stopped and $\mathcal{A}$ is returned without adding $\vec{r}$. Assuming that $\vec{r}$ is not weakly dominated by any of the elements in $\mathcal{A}$, the former is added to the archive. If the cardinality of the archive is greater than its maximum possible size $\mu$, it is necessary to prune it following an iterative process. At each iteration, the solution with the worst contribution to the Riesz $s$-energy indicator is removed until the desired size $\mu$ is reached. Finally, $\mathcal{A}$ is returned.

### D. Migration Process

The exchange of individuals via migration is crucial for the overall performance of IMIA. It increases the diversity of both the main population and the local archive in each island. In Algorithm 4, the migration is performed in lines 1–6 where the invoked island sends $n_{\text{mig}}$ solutions (randomly selected from the main population $P$) to each of its neighboring islands that are determined by the topology matrix $M$. After sending the solutions, the island is ready for receiving immigrant solutions. Hence, it is first necessary to determine which solutions from $P$ will be replaced. In our case, we iteratively delete from $P$ the $L \cdot n_{\text{mig}}$ worst contributing solutions to the given indicator $I$ in lines 9–13, where $L$ is the number of source islands of the current island. The blocking reception process is described in lines 15–21, where the algorithms wait until receiving the total $L \cdot n_{\text{mig}}$ of immigrant solutions. Each time a solution is received, it is inserted in the archive (using Algorithm 3) and it is directly added to $P$ in one of the available places. Once all the immigrant solutions were received, the updated $P$ and $\mathcal{A}$ are returned.

**Algorithm 3** Insert

**Require:** Archive $\mathcal{A}$; solution $\vec{r}$ to be inserted; maximum archive size $\mu$.
**Ensure:** Updated archive $\mathcal{A}$

**for all** $\vec{a} \in \mathcal{A}$ **do**
  **if** $\vec{r} \prec \vec{a}$ **then**
    $\mathcal{A} = \mathcal{A} \setminus \{\vec{a}\}$
  **else if** $\vec{a} \preceq \vec{r}$ **then**
    **return** $\mathcal{A}$
  **end if**
**end for**
$\mathcal{A} = \mathcal{A} \cup \{\vec{r}\}$
**while** $|\mathcal{A}| > \mu$ **do**
  $\vec{a}_{\text{worst}} = \arg\max_{\vec{a} \in \mathcal{A}} C_{E_s}(\vec{a}, \mathcal{A})$
  $\mathcal{A} = \mathcal{A} \setminus \{\vec{a}_{\text{worst}}\}$
**end while**
**return** $\mathcal{A}$

**Algorithm 4** Migration

**Require:** Population $P$; indicator $I$; archive $\mathcal{A}$; maximum archive size $\mu$; migration frequency $f_{\text{mig}}$; number of solutions to migrate $n_{\text{mig}}$; topology matrix $M$.
**Ensure:** Updated population $P$ and archive $\mathcal{A}$

1: **for all** destination islands in $M$ of the current island **do**
2:  **for** $j = 1$ to $n_{\text{mig}}$ **do**
3:  Randomly select a solution $\vec{r} \in P$ to migrate
4:  Send solution $\vec{r}$
5:  **end for**
6: **end for**
7: Let $L$ denote the number of source islands of this island
8: $t = 1$
9: **while** $t \leq L \cdot n_{\text{mig}}$ **do**
10:  $\vec{r}_{\text{worst}} = \arg\min_{\vec{r} \in P} C_I(\vec{r}, P)$
11:  $P = P \setminus \{\vec{r}\}$
12:  $t = t + 1$
13: **end while**
14: $t = 1$
15: **while** $t \leq L \cdot n_{\text{mig}}$ **do**
16:  **if** a solution $\vec{r}$ is received **then**
17:  $\mathcal{A} = \text{Insert}(\mathcal{A}, \vec{r}, \mu)$
18:  $P = P \cup \{\vec{r}\}$
19:  $t = t + 1$
20:  **end if**
21: **end while**
22: **return** $\{P, \mathcal{A}\}$

### *E. Runtime Complexity*

In this section, we provide the runtime complexity of a single iteration of the parallel *for* in lines 4–6 in Algorithm 1. Hence, it is necessary to first determine the runtime complexity of the `Insert` and `Migration` operations, shown in Algorithms 3 and 4, respectively. It is worth noting that the cardinality of each $A_j$ is at most $\mu$ and $|P_j| = \mu/k$ for all $j = 1, \ldots, k$. The complexity of the `Insert` operation is dominated by the *while* loop in lines 9–12 since the *for* loop takes $\mathcal{O}(m|A_j|) = \mathcal{O}(m\mu)$. The *while* loop requires the selection of the worst contributing solution to the Riesz $s$-energy, which is performed using the algorithm proposed in [43] that takes $\Theta(m\mu^2)$. Hence, the overall complexity of the `Insert` algorithm is $\Theta(m\mu^2)$.

The `Migration` algorithm (see Algorithm 4) is mainly composed of three loops. The *for* loop in lines 1–6 takes $\mathcal{O}((k-1) \cdot n_{\text{mig}})$, where an island can send solutions to at most $k-1$ islands. The *while* loop in lines 15–21 invokes the `Insert` algorithm a total number of $L \cdot n_{\text{mig}}$ times, where $L$ is at most $k-1$. Hence, this *while* loop is performed in $\mathcal{O}((k-1) \cdot n_{\text{mig}} \cdot m\mu^2)$. The remaining *while* loop dominates the complexity of the `Migration` because it involves the selection of the worst contributing solution to the $I_j$ indicator. Let $\tau_j(\mu)$ denote the complexity of determining the worst contributing solution to $I_j$. In consequence, line 10 of Algorithm 4 takes $\mathcal{O}(\tau_j(\mu/k))$ and this operation is repeated $L \cdot n_{\text{mig}}$ times. Thus, the overall complexity of the `Migration` algorithm is $\mathcal{O}((k-1) \cdot n_{\text{mig}} \cdot \tau_j(\mu/k))$. Regarding $\tau_j(\mu)$, in the case of using HV via the walking-fish-group (WFG) algorithm [44], $\tau(\mu) = \mathcal{O}(2^\mu)$. On the other hand, in the case of using IGD$^+$, $R2$, $\epsilon^+$, and $\Delta_p$, computing the worst contributing solution takes $\tau(\mu) = \Theta(m\mu^2)$, according to [45].

The runtime complexity of Algorithm 2 is dominated by line 8 where it is determined the worst contributing solution to the given indicator in $\mathcal{O}(\tau(\mu/k))$. The remaining operations are at most $\mathcal{O}(\mu^2)$. For instance, the nondominated sorting is performed in $\mathcal{O}(m(\mu/k)^2)$ and the `Insert` operation in line 13 takes $\mathcal{O}(m\mu^2)$. Since the *while* loop is repeated $f_{\text{mig}}$ times and in line 18, the `Migration` is executed, the overall complexity of Algorithm 2 is $\mathcal{O}(f_{\text{mig}} \cdot (\tau(\mu/k) + m\mu^2) + (k-1) \cdot n_{\text{mig}} \cdot \tau(\mu/k))$. Since $f_{\text{mig}}$, $n_{\text{mig}}$, and $k$ are constants, the complexity can be written as $\mathcal{O}(\tau(\mu/k) + m\mu^2)$, which also corresponds with the complexity of the parallel *for* in Algorithm 1.

## V. Experimental Results

This section is devoted to analyzing the performance of IMIA, employing islands with SMS-EMOA [23], $R2$-EMOA [5], IGD$^+$-MaOEA [45], $\epsilon^+$-MaOEA, and $\Delta_p$-MaOEA (the last two algorithms are similar to IGD$^+$-MaOEA). We decided to utilize these five IB-MOEAs because according to Falcón-Cardona and Coello Coello [10], they exhibit different convergence and diversity properties that can be combined to compensate for the weaknesses of a given IB-MOEA with the strengths of the others. The islands are linked through a fully connected graph topology and each IB-MOEA uses a micropopulation of $\mu/5$ individuals. IMIA is compared with panmictic versions[6] of its five baseline IB-MOEAs and five state-of-the-art MOEAs[7]: 1) AR-MOEA [29]; 2) GrEA [30]; 3) SPEA2+SDE [28]; 4) Two_Arch2 [32]; and 5) SRA [28] (which is a MIB-MOEA). These five state-of-the-art MOEAs share one property: they have been designed to tackle MOPs with different Pareto front shapes.

TABLE I
Summary of Pareto Front Shapes Related to the Selected MOPs

| MOP name | Pareto front shape | Simplex-like |
|---|---|---|
| DTLZ1 | Linear | Yes |
| DTLZ1$^{-1}$ | Inverted linear | No |
| DTLZ2 - DTLZ4 | Concave | Yes |
| DTLZ2$^{-1}$ - DTLZ4$^{-1}$ | Convex | No |
| DTLZ5 & DTLZ6 | Concave ($m = 2$)<br>Degenerate ($m = 3$)<br>Unknown ($m > 3$) | No |
| DTLZ5$^{-1}$ & DTLZ6$^{-1}$ | Convex | No |
| DTLZ7 & DTLZ7$^{-1}$ | Disconnected | No |
| WFG1 | Mixed | Yes |
| WFG1$^{-1}$ | Mixed | No |
| WFG2 | Disconnected | Yes |
| WFG2 | Convex | No |
| WFG3 | Linear ($m = 2$)<br>Degenerate ($m \geq 3$) | Yes<br>No |
| WFG3$^{-1}$ | Inverted linear | No |
| WFG4 - WFG9 | Concave | Yes |
| WFG4$^{-1}$ - WFG9$^{-1}$ | Convex | No |

For comparison purposes, we adopted the test suites Deb-Thiele-Laumanns-Zitzler (DTLZ) [47], WFG [48], and their inverted versions DTLZ$^{-1}$ and WFG$^{-1}$ [49], using 2, 3, 4, 5, 6, and 7 objective functions. Table I presents an overview of the Pareto front shapes related to the considered test problems, where it is emphasized if the Pareto front geometry is correlated with the shape of a simplex formed by a set of convex weight vectors. Regarding the DTLZ and DTLZ$^{-1}$ test problems, the number of variables was set to $n = m + K - 1$, where $m$ is the number of objective functions, and $K = 5$ for DTLZ1, $K = 10$ for DTLZ2-DTLZ6, and $K = 20$ for DTLZ7. Their inverted counterparts share the same value of $K$. Concerning the WFG and WFG$^{-1}$ problems, Table II shows the number of variables and position-related parameters together with the number of objective functions. For each test instance, we performed 30 independent executions and, to have statistical confidence, we employed the one-tailed Wilcoxon rank-sum test, using a confidence level of $\alpha = 0.05$.

IMIA[8] was implemented using the C programming language (compiler GCC 4.7.2 20121109) and we adopted the OpenMP library to deal with the parallel execution of the islands (line 4 of Algorithm 1). The running environment is the following: Intel Core i7-3930K CPU @ 3.20 GHz (6 cores), having 8-GB RAM and Red Hat 4.7.2-8 as the operating system. It is worth noting that all the algorithms were executed under the same running environment.

[6]They were implemented following Algorithm 2, adding an initialization phase for the main population and not using the external archive nor the migration process.

[7]We employed the algorithms implemented in the PlatEMO platform [46].

[8]The source code of IMIA is available at http://computacion.cs.cinvestav.mx:/jfalcon/IMIA/.

TABLE II
COMMON PARAMETER SETTINGS APPLICABLE TO ALL THE $\mu$ VALUES

| Num. objectives ($m$) | | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Max. function evaluations ($\times 10^3$) | | 40 | 50 | 60 | 70 | 80 | 90 |
| WFG | Variables ($n$) | 24 | 26 | 28 | 30 | 32 | 34 |
| | Position-related parameters | 2 | 4 | 6 | 8 | 10 | 12 |
| GrEA's divisions of objective space | | 45 | 15 | 10 | 9 | 9 | 8 |

TABLE III
REFERENCE POINTS FOR THE HV CALCULATION

| MOP name | Reference point |
|---|---|
| DTLZ1 | $(1, 1, \ldots, 1)$ |
| DTLZ2-DTLZ6 | $(2, 2, \ldots, 2)$ |
| DTLZ7 | $(1, \ldots, 1, 21)$ |
| DTLZ1$^{-1}$ - DTLZ6$^{-1}$ | $(1, 1, \ldots, 1)$ |
| DTLZ7$^{-1}$ | $(0.1, \ldots, 0.1, -10)$ |
| WFG1-WFG9 | $\{2j+1\}_{j=1,\ldots,m}$ |
| WFG1$^{-1}$-WFG9$^{-1}$ | $(1, 1, \ldots, 1)$ |

### A. Parameters Settings

For a fair comparison, IMIA and all the selected MOEAs use the same population size $\mu$. On the one hand, when comparing IMIA with the panmictic IB-MOEAs, we used different values of $\mu = 50, 75, 100, 120, 140$. On the other hand, we set $\mu = 140$ for the comparison of IMIA with AR-MOEA, GrEA, SPEA2+SDE, Two_Arch2, and SRA. IMIA and all the selected MOEAs utilize simulated binary crossover and PBM as their genetic operators [24]. For two- and three-objective MOPs, we set the crossover probability to 0.9, and the crossover distribution index to 20, while for MaOPs these values are set to 1.0 and 30, respectively. For all test instances, the mutation probability is set to $1/n$ (where $n$ is the number of decision variables), and the mutation distribution index is set to 20. We employed a maximum number of function evaluations as the stopping criterion (see Table II). Regarding IMIA, $f_{\text{mig}} = \mu/5$, $n_{\text{mig}} = 1$, and a fully connected graph topology is employed as $M$ in all cases. The global and local archives of IMIA have a cardinality that is equal to $\mu$. Due to the use of the island model, IMIA was implemented adopting a multicore parallel approach via OpenMP. To generate the weight vectors that $R2$-EMOA requires, Uniform Design, using the Hammersley method (UDH) [50] is employed. It is worth noting that UDH can produce sets of weight vectors of any cardinality, unlike the simplex-lattice design method. Additionally, we used the achievement scalarizing function as the utility function for $R2$-EMOA. $\{\text{IGD}^+, \epsilon^+, \Delta_p\}$-MaOEA uses the current set of nondominated solutions as reference set. Regarding the PlatEMO implementations, AR-MOEA, SPEA2+SDE, and SRA do not need special parameters settings while GrEA and Two_Arch2 do. The number of divisions of the objective space employed by GrEA is shown in Table II. The size of the convergence archive of Two_Arch2 is equal to the population size and the fractional distance is set to $1/m$ for all the test instances. These parameter values are suggested by the authors of GrEA and Two_Arch2 in [30] and [32], respectively.

To assess the Pareto front approximations, we decided to utilize seven QIs: HV, $R2$, IGD$^+$, $\epsilon^+$, $\Delta_p$, Riesz $s$-energy, and the SPD. The reason of this decision is that we aimed to determine if the MOEAs' performance is robust under several quality measures, i.e., we wanted to know if the MOEA's performance was consistently good or bad in the light of the selected convergence and diversity indicators. Table III shows the reference points that HV employs per each test problem. $R2$-EMOA uses UDH-based weight vectors and the vector-angle distance scaling function. For the calculation of IGD$^+$, $\epsilon^+$, and $\Delta_p$, a reference set is required. The reference sets are constructed by merging the Pareto front approximations from the MOEAs, getting the nondominated solutions and, then, applying a Riesz $s$-energy-based subset selection, with $s = m - 1$ [43]. According to Hardin and Saff [51], a uniform point set is favored if $s$ is greater or equal to the dimension of the manifold covered. The cardinalities of the reference sets are equal to $100 \cdot m$. Finally, the parameter $\theta$ of SPD is set to 10 for all cases.

### B. Comparing IMIA With Panmictic IB-MOEAs

In this section, we discuss the performance of IMIA in comparison with the panmictic IB-MOEAs. Due to the high computational cost of executing a panmictic SMS-EMOA on MaOPs, we decided to compare IMIA with the IB-MOEAs in MOPs with 2, 3, and 4 objective functions. However, to allow an exhaustive experimentation, we used different population sizes, i.e., $\mu = 50, 75, 100, 120, 140$. Due to space limtiations, Table IV summarizes all the numerical results, showing the statistical ranks obtained by each MOEA per quality indicator and population size. The complete numerical results are available in the supplementary material.

For a population size of 50 individuals, Table IV shows that SMS-EMOA is the best algorithm, getting the first place for all the convergence indicators and the second place for diversity indicators where IMIA has the best performance. In contrast, IMIA is consistently the best-ranked algorithm for $R2$, $\Delta_p$, $E_s$, and SPD and it obtains the second and third places regarding $\epsilon^+$ and HV, respectively, for $\mu = 75, 100, 120$, and 140 individuals. For these $\mu$ values, SMS-EMOA is the best-ranked algorithm for HV, IGD$^+$, and $\epsilon^+$. This is an expected result since SMS-EMOA optimizes HV and the preferences of this QI are highly correlated with those of IGD$^+$ and $\epsilon^+$ [12], [52]. A reason that explains why IMIA does not get the first places for HV, IGD$^+$, and $\epsilon^+$ is that the Pareto front approximations of our proposed approach have a high degree of diversity as it is shown in Fig. 1. For example, for the three-objective DTLZ2 problem (see Fig. 1), SMS-EMOA, IGD$^+$-MaOEA, and $\epsilon^+$-MaOEA are the three best-ranked algorithms according to Table 34 in the supplementary material and, from the figure, it is clear that their Pareto front approximations are similar but lacking diversity. However, IMIA, $\Delta_p$-MaOEA, and $R2$-EMOA, whose approximation sets are remarkably more diversified, are ranked fourth, fifth, and sixth, respectively. Hence, having well-diversified Pareto fronts does not necessarily imply a large hypervolume value. Under this direction, it is possible to observe that an important advantage of IMIA

TABLE IV
MEAN RELATED TO THE STATISTICAL RANKS OF THE COMPARISON BETWEEN IMIA AND THE PANMICTIC IB-MOEAS. A SYMBOL # IS PLACED WHEN IMIA'S RANK IS SIGNIFICANTLY BETTER THAN THE OTHER IB-MOEAS BASED ON A ONE-TAILED WILCOXON TEST, USING A SIGNIFICANCE LEVEL OF $\alpha = 0.05$. THE TWO BEST VALUES ARE SHOWN IN GRAY SCALE, WHERE THE DARKER TONE CORRESPONDS TO THE BEST VALUE. THE SUBINDEX IS THE RANK OF EACH MOEA

| Pop. size | QI | IMIA | SMS-EMOA | $R2$-EMOA | IGD$^+$-MaOEA | $\epsilon^+$-MaOEA | $\Delta_p$-MaOEA |
|---|---|---|---|---|---|---|---|
| 50 | HV | $3.833_4$ | $1.406_1$ | $4.354_5$# | $3.396_2$ | $3.448_3$ | $4.562_6$# |
| | $R2$ | $2.802_2$ | $2.531_1$ | $3.354_4$# | $4.688_5$# | $4.698_6$# | $2.927_3$ |
| | IGD$^+$ | $4.104_5$ | $1.531_1$ | $4.031_4$ | $3.27_3$ | $3.250_2$ | $4.812_6$# |
| | $\epsilon^+$ | $3.729_3$ | $1.344_1$ | $4.000_5$ | $3.865_4$ | $3.625_2$ | $4.438_6$# |
| | $\Delta_p$ | $2.719_2$ | $2.615_1$ | $3.396_4$# | $4.406_5$# | $4.583_6$# | $3.281_3$# |
| | $E_s$ | $1.833_1$ | $2.271_2$# | $5.792_6$# | $3.896_4$# | $4.333_5$# | $2.875_3$# |
| | SPD | $1.625_1$ | $2.635_2$# | $4.885_6$# | $4.385_4$# | $4.677_5$# | $2.792_3$# |
| 75 | HV | $3.531_3$ | $1.344_1$ | $4.500_5$# | $3.208_2$ | $3.750_4$ | $4.667_6$# |
| | $R2$ | $2.448_1$ | $2.604_2$ | $3.542_4$# | $4.479_5$# | $4.885_6$# | $3.042_3$# |
| | IGD$^+$ | $3.646_3$ | $1.458_1$ | $4.177_5$# | $3.073_2$ | $3.729_4$ | $4.917_6$# |
| | $\epsilon^+$ | $3.188_2$ | $1.323_1$ | $4.05_42$# | $3.677_3$# | $4.125_5$# | $4.635_6$# |
| | $\Delta_p$ | $2.198_1$ | $2.854_2$# | $3.542_4$# | $4.354_5$# | $4.812_6$# | $3.240_3$# |
| | $E_s$ | $1.562_1$ | $2.573_2$# | $5.865_6$# | $3.615_4$# | $4.479_5$# | $2.906_3$# |
| | SPD | $1.292_1$ | $2.958_3$# | $4.948_6$# | $4.271_4$# | $4.875_5$# | $2.656_2$# |
| 100 | HV | $3.542_3$ | $1.292_1$ | $4.406_5$# | $3.375_2$ | $3.750_4$ | $4.635_6$# |
| | $R2$ | $2.531_1$ | $2.677_2$ | $3.365_4$# | $4.594_5$# | $4.885_6$# | $2.948_3$# |
| | IGD$^+$ | $3.708_4$ | $1.438_1$ | $4.135_5$# | $3.104_2$ | $3.635_3$ | $4.979_6$# |
| | $\epsilon^+$ | $3.271_2$ | $1.344_1$ | $3.854_4$# | $3.729_3$# | $4.135_5$# | $4.667_6$# |
| | $\Delta_p$ | $2.125_1$ | $2.979_2$# | $3.469_4$# | $4.469_5$# | $4.708_6$# | $3.250_3$# |
| | $E_s$ | $1.604_1$ | $2.698_2$# | $5.875_6$# | $3.750_4$# | $4.229_5$# | $2.844_3$# |
| | SPD | $1.521_1$ | $3.135_3$# | $4.792_6$# | $4.344_4$# | $4.792_5$# | $2.417_2$# |
| 120 | HV | $3.552_3$ | $1.385_1$ | $4.438_5$# | $3.271_2$ | $3.688_4$ | $4.667_6$# |
| | $R2$ | $2.635_1$ | $2.802_2$ | $3.542_4$# | $4.500_5$# | $4.667_6$# | $2.854_3$# |
| | IGD$^+$ | $3.740_4$ | $1.458_1$ | $4.271_5$# | $3.073_2$ | $3.500_3$ | $4.958_6$# |
| | $\epsilon^+$ | $3.229_2$ | $1.354_1$ | $3.948_4$# | $3.698_3$# | $4.073_5$# | $4.698_6$# |
| | $\Delta_p$ | $2.073_1$ | $3.062_2$# | $3.740_4$# | $4.406_5$# | $4.469_6$# | $3.250_3$# |
| | $E_s$ | $1.635_1$ | $2.906_3$# | $5.802_6$# | $3.823_4$# | $4.260_5$# | $2.573_2$# |
| | SPD | $1.646_1$ | $3.229_3$# | $4.771_6$# | $4.365_4$# | $4.667_5$# | $2.323_2$# |
| 140 | HV | $3.531_2$ | $1.333_1$ | $4.260_5$ | $3.531_3$# | $3.729_4$# | $4.615_6$# |
| | $R2$ | $2.510_1$ | $2.771_2$ | $3.427_4$# | $4.635_5$# | $4.719_6$# | $2.938_3$# |
| | IGD$^+$ | $3.760_4$ | $1.396_1$ | $4.146_5$ | $3.188_2$ | $3.500_3$ | $5.010_6$# |
| | $\epsilon^+$ | $3.073_2$ | $1.417_1$ | $3.875_3$# | $3.917_4$# | $3.917_5$# | $4.802_6$# |
| | $\Delta_p$ | $2.021_1$ | $3.031_2$# | $3.677_4$# | $4.375_5$# | $4.594_6$# | $3.302_3$# |
| | $E_s$ | $1.562_1$ | $2.948_3$# | $5.865_6$# | $3.792_4$# | $4.271_5$# | $2.562_2$# |
| | SPD | $1.531_1$ | $3.312_3$# | $4.771_5$# | $4.344_4$# | $4.781_6$# | $2.260_2$# |

over its baseline panmictic IB-MOEAs is its ability to generate Pareto front approximations with high diversity. This skill is due to a better exploration of the search space as a result of the cooperation of the islands, where different solutions are found, following the inner preferences of the baseline QIs. In fact, IMIA is the best algorithm for both Riesz $s$-energy and SPD for all the $\mu$ values. This is supported by Fig. 1 where regardless of the Pareto front shape, IMIA is able to produce approximation sets covering the whole Pareto front with well-diversified solutions. Overall, IMIA is the most robust algorithm according to Table IV, since for all the seven QIs and different population sizes, the performance of IMIA is consistently good. This is a strong insight that supports the fact that the cooperation of multiple IB-MOEAs through IMIA is responsible for obtaining better performance than using their panmictic implementations. In contrast, the panmictic IB-MOEAs are restricted to their own search abilities, i.e., it is not possible for them to keep solutions out of the scope of their QI preferences. In consequence, when assessing an IB-MOEA with multiple QIs, including its baseline QI, it is expected that it has a good performance on its baseline QI (because by design, the IB-MOEA aims to optimize it). However, regarding the other indicators, it is not very likely for the IB-MOEA to present a good performance because it does not fulfill the solutions rewarded by them. Hence, overall the experimental results show that IMIA looks for a balance between the preferences of its baseline IB-MOEA, gathering solutions from different promising regions of the search space, which results in a more robust performance that its panmictic baseline IB-MOEAs.

One may argue that SMS-EMOA is also a robust algorithm. This is partially true due to the following reasons.

1) HV, IGD$^+$, and $\epsilon^+$ systematically reward SMS-EMOA due to their high correlation of preferences.
2) SMS-EMOA is not able to produce well-diversified Pareto front approximations regardless of the associated manifold geometry as shown in Fig. 1.
3) It requires a high computational effort even for MOPs with 2, 3, and 4 objective functions (we refer the reader to Section 3 of the supplementary material).

Regarding the last point, Tables 49–53 of the supplementary material show the speedups that IMIA obtains. Even though SMS-EMOA is a remarkable well-performing algorithm, its computational cost is too high in comparison with IMIA and the remaining IB-MOEAs. For instance, IMIA gets speedups of up to 189.136× for the DTLZ2 problem with a population

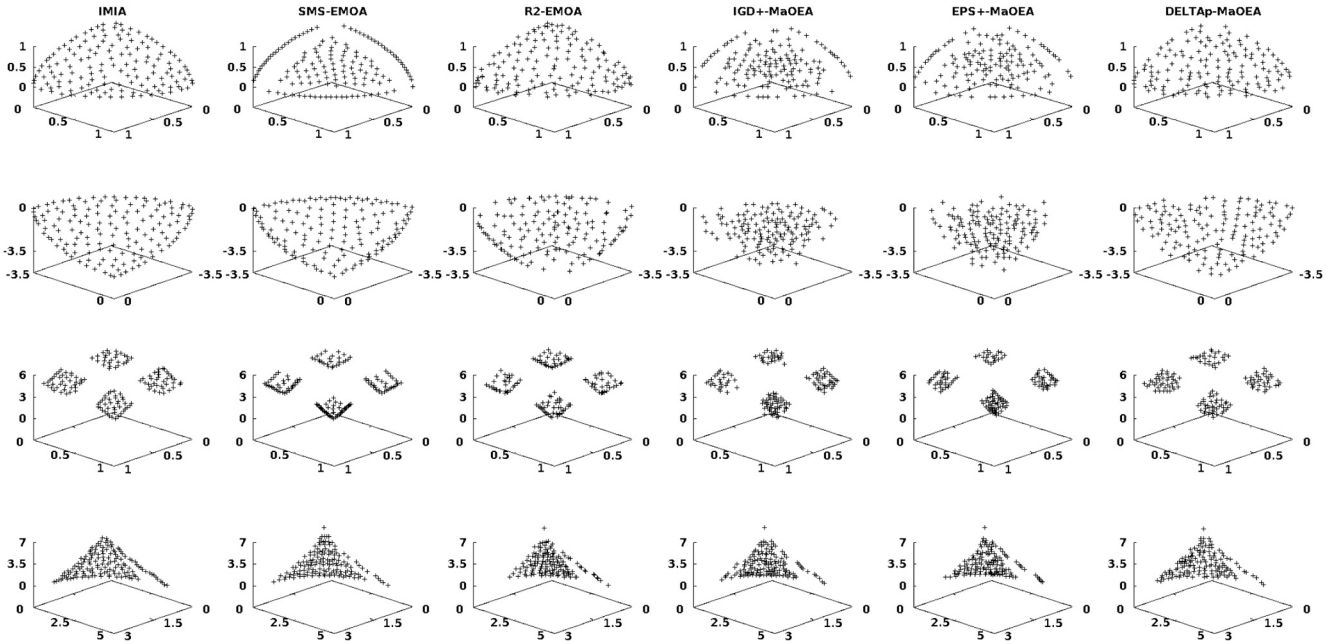


Fig. 1. Pareto front approximations of the three-objective DTLZ2, DTLZ2$^{-1}$, DTLZ7, and WFG1 (from top to bottom). The approximation sets correspond to the median of the hypervolume.

size of 140 individuals (see Table 53 of the supplementary material) as we increase the number of objective functions with respect to SMS-EMOA. The advantage of IMIA with respect to the panmictic SMS-EMOA is the use of a micropopulation as stated by Hernández-Gómez *et al.* [42]. Hence, this observation strengthens our claim that IMIA is better than panmictic IB-MOEAs since their high-quality results are obtained in lower computational time.

In the following, we describe the MOPs on which IMIA systematically attains a good performance. These claims are based on the heat maps presented in Section 1 of the supplementary material. For all the $\mu$ values, IMIA always present a robust performance (being ranked first, second, or third on all the QIs) on problems DTLZ2$^{-1}$, DTLZ5$^{-1}$, DTLZ6, DTLZ6$^{-1}$, WFG3$^{-1}$, WFG4$^{-1}$, WFG5$^{-1}$, WFG6$^{-1}$, WFG7$^{-1}$, WFG8$^{-1}$, and WFG9$^{-1}$. For $\mu \geq 75$, IMIA is very good for problems DTLZ1$^{-1}$, DTLZ2, DTLZ4$^{-1}$, and DTLZ7. Under this light, it is possible to see the superiority of IMIA on the inverted DTLZ and WFG test instances, due to the combination of the strengths of its baseline IB-MOEAs. In addition, IMIA has always a competitive performance for problems WFG4, WFG6, WFG7, WFG8, and WFG9. For WFG1, WFG2, and WFG3, IMIA has a good performance as long as $\mu$ increases. Hence, it is possible to see that IMIA is a good optimizer under MOPs with different search difficulties and Pareto front shapes. Finally, the heat maps show that IMIA is not a good option for problems similar to DTLZ3 (which has a highly multifrontal MOP) and DTLZ7$^{-1}$. A possible reason for the bad performance on DTLZ3 is that due to the use of micropopulations, the islands get stuck on local optima since there is not enough genetic material to exploit. In comparison, the panmictic IB-MOEAs have better performance on this problem. Concerning DTLZ7$^{-1}$, we observed that IMIA has some difficulties to generate all the disconnected regions of the Pareto front, especially for two objective functions. However, for DTLZ7 and WFG2, which also have a disconnected Pareto front, IMIA presents a competitive performance.

### C. Pareto-Front-Shape Invariance

A few years ago, Ishibuchi *et al.* [49] pointed out that some MOEAs are overspecialized on benchmark problems whose Pareto front shapes are correlated to the form of a simplex. In other words, the performance of these MOEAs depends on the Pareto front geometries. To effectively overcome this issue, IMIA is designed to take advantage of the search skills of different IB-MOEAs throughout their cooperation. To show that IMIA is a Pareto-front-shape invariant optimizer, we first need to analyze how its Pareto front approximations are generated. The final approximation set generated by IMIA is constructed by merging all the subpopulations and archives of all the islands (see lines 8–16 of Algorithm 1). It is worth analyzing how each island contributes to the final Pareto front approximations. The underlying intuition is that the percentage of contribution of each island will change depending on the Pareto front shapes due to the specific preferences of its baseline QI. Fig. 2 presents the percentage of solutions contributed by each island for the problems DTLZ2, WFG8, DTLZ2$^{-1}$, and WFG8$^{-1}$ with $\mu = 140$. We should mention that both DTLZ2 and WFG8 have a concave Pareto front geometry correlated with the shape of a simplex. In contrast, both DTLZ2$^{-1}$ and WFG8$^{-1}$ have a nonsimplex-like convex Pareto front geometry. From the figure, it is interesting to see some patterns. For both DTLZ2 and WFG8, it is clear that as the number of objective functions increases from 2 to 7, the contribution of solutions of the $\Delta_p$ island increases as

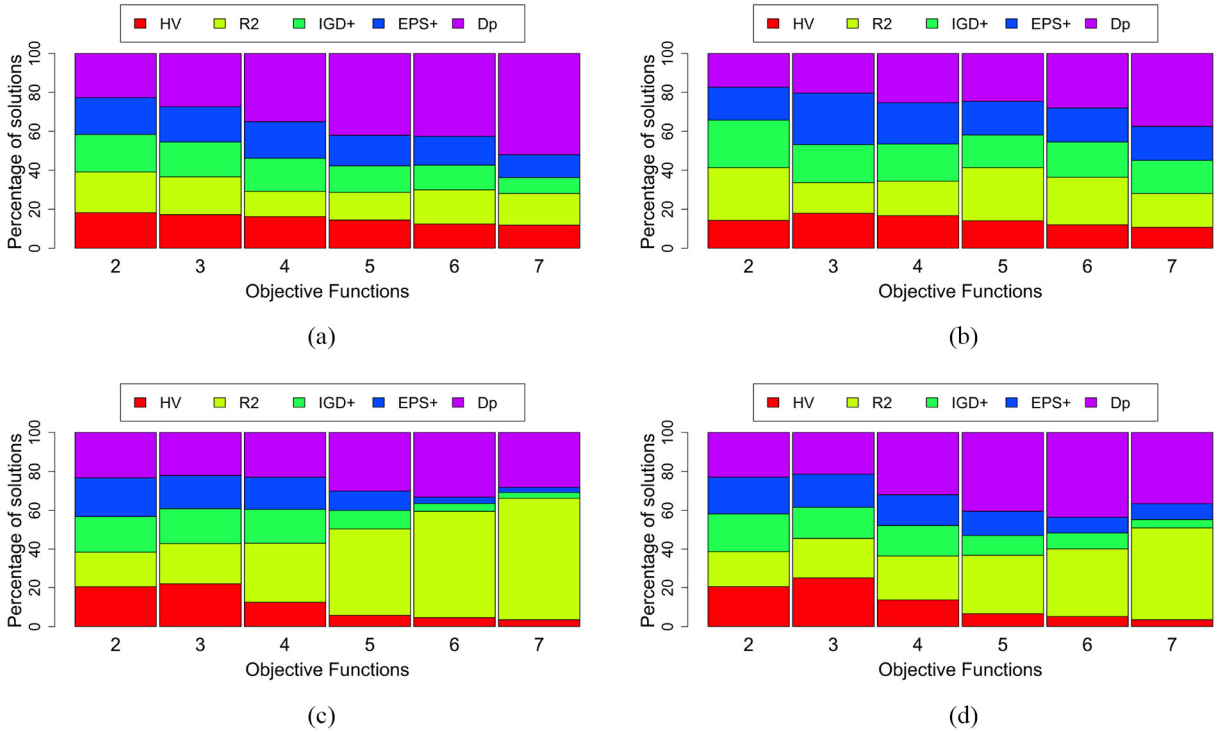


Fig. 2. Percentage of solutions contributed by each island to the final approximation sets with $\mu = 140$. Different composition patterns are formed according to the Pareto front shape. (a) DTLZ2. (b) WFG8. (c) DTLZ2$^{-1}$. (d) WFG8$^{-1}$.

well while the contribution of the HV, IGD$^+$, and $\epsilon^+$ islands decreases and the contribution of the $R2$ island varies a little without being significant. In contrast, for both DTLZ2$^{-1}$ and WFG8$^{-1}$, the contribution of the $R2$ island is the dominant as the dimension of the objective space increases while the impact of the $\Delta_p$ island is relatively constant and the decrease of the contribution of the HV, IGD$^+$, and $\epsilon^+$ islands is more significant. These contribution patterns are also exhibited for different $\mu$ values. In general, we found that for groups of test problems, sharing similar Pareto front shapes, a specific contribution pattern appeared (the complete results are available at http://computacion.cs.cinvestav.mx/~jfalcon/IMIA/). The existence of a contribution pattern for similar problems and the difference of contributions when the Pareto front shape changes is a clear insight into the efficacy of each island on specific problems and how IMIA can compensate for the weaknesses of an IB-MOEA with the strengths of others. Additionally, this is the main reason for the Pareto-front-shape invariance of IMIA.

To support the above-mentioned results, it is necessary to analyze the indicator results of IMIA when it is compared with panmictic IB-MOEAs and state-of-the-art MOEAs. First, Table IV shows that for all the population sizes, IMIA is the best-ranked algorithm for $R2$ and $\Delta_p$ (which are convergence-diversity indicators) and, more importantly, it is the best algorithm for Riesz $s$-energy and SPD, which are diversity QIs. These indicator values exhibit the superiority of IMIA to produce Pareto front approximations with high diversity. Fig. 1 compares the approximation sets generated by IMIA and the panmictic IB-MOEAs, where IMIA generates the best ones regardless of the geometry of the manifold.

Table V presents the statistical ranks obtained by IMIA and the selected state-of-the-art MOEAs. These comparisons are based on test problems with 2–7 objective functions. The complete comparison is available in Section 2 of the supplementary material. The statistical ranks show that IMIA obtains once again the first place in the comparison regarding $\Delta_p$, $E_s$, and SPD, and the second place with respect to $R2$. This is a direct consequence of its generation of well-diversified Pareto fronts. On the other hand, Two_Arch2 is the best for $R2$ and $\epsilon^+$ and it achieves the second place for HV, IGD$^+$, $\Delta_p$, and SPD. The obtention of the first place on $\epsilon^+$ is expected since Two_Arch2 optimizes this QI and its good performance on the other mentioned QIs is due to the interaction between its subpopulations. SPEA2+SDE is the third-best algorithm. This MOEA is the best ranked for HV and IGD$^+$ and the second best for $\epsilon^+$, which is explained by the use of the SDE method that shifts the position of solutions using the $d^+$ distance of IGD$^+$. IMIA outperformed the remaining MOEAs, i.e., AR-MOEA, GrEA, and SRA. The wide variety of problems employed in the comparison showed that the adaptation method for the weight vectors of AR-MOEA cannot perfectly match the shape of the underlying Pareto front, which causes the loss of some portions of it (see Fig. 3). For both GrEA and SRA, their poor performance is due to the loss of some parts of the Pareto fronts. Fig. 3 compares some five- and seven-objective approximation sets produced by IMIA and the state-of-the-art MOEAs for problems DTLZ2, DTLZ2$^{-1}$, WFG9, and WFG$^{-1}$. It is clear that both IMIA and Two_Arch2 are the only algorithms that completely cover all the Pareto fronts, having a high diversity degree. For DTLZ2$^{-1}$, AR-MOEA, SPEA2+SDE, and SRA are not able to cover all the Pareto front. Hence, this supports

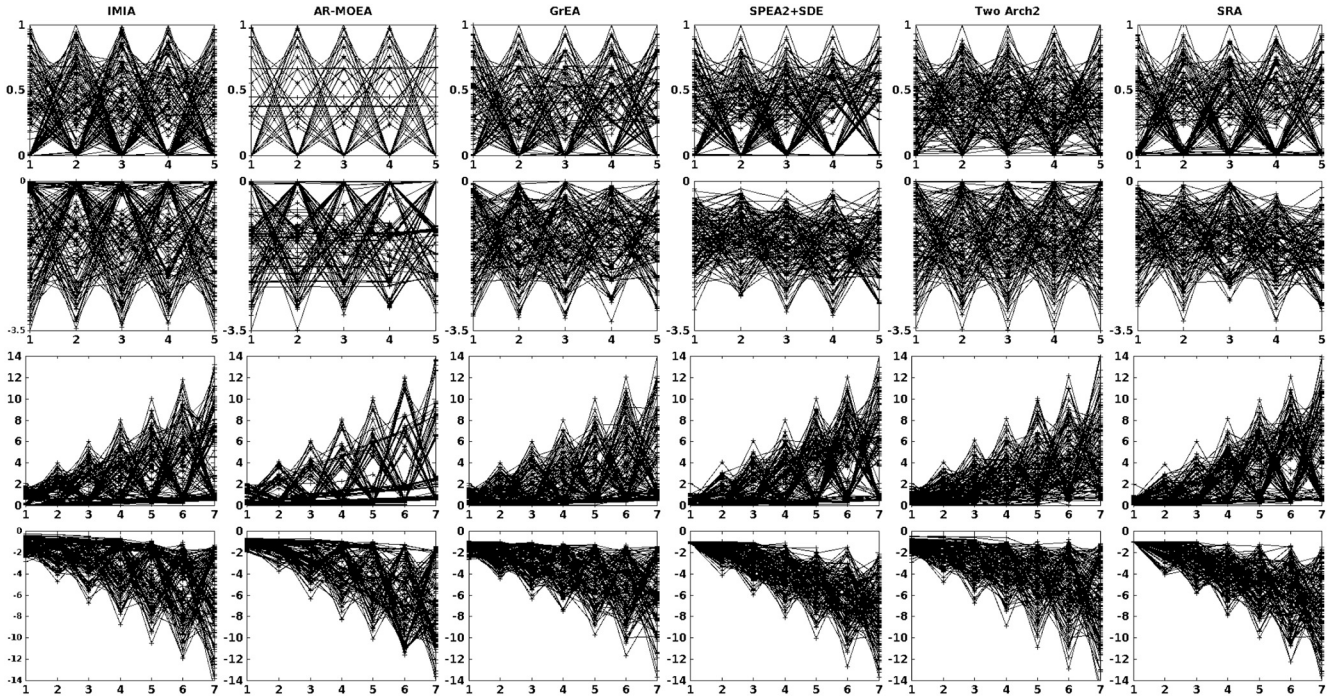


Fig. 3. Pareto front approximations of the five-objective DTLZ2 and DTLZ2$^{-1}$ and the seven-objective WFG9 and WFG9$^{-1}$ (from top to bottom). The approximation sets correspond to the hypervolume median.

TABLE V
MEAN RELATED TO THE STATISTICAL RANKS OF THE COMPARISON BETWEEN IMIA AND THE STATE-OF-THE-ART MOEAS. A SYMBOL # IS PLACED WHEN IMIA'S RANK IS SIGNIFICANTLY BETTER THAN THE OTHER MOEAS BASED ON A ONE-TAILED WILCOXON TEST, USING A SIGNIFICANCE LEVEL OF $\alpha = 0.05$. THE TWO BEST VALUES ARE SHOWN IN GRAY SCALE, WHERE THE DARKER TONE CORRESPONDS TO THE BEST VALUE. THE SUBINDEX IS THE RANK OF EACH MOEA

| QI | IMIA | AR-MOEA | GrEA | SPEA2+SDE | Two_Arch2 | SRA |
|---|---|---|---|---|---|---|
| HV | $3.568_4$ | $3.901_5$# | $3.401_3$ | $2.844_1$ | $3.156_2$ | $4.130_6$# |
| $R2$ | $2.630_2$ | $3.026_3$# | $3.958_4$# | $4.724_6$# | $2.443_1$ | $4.219_5$# |
| IGD$^+$ | $3.396_3$ | $3.844_5$# | $3.417_4$ | $3.021_1$ | $3.083_2$ | $4.240_6$# |
| $\epsilon^+$ | $3.552_4$ | $3.500_3$ | $3.552_5$ | $3.495_2$ | $3.089_1$ | $3.812_6$ |
| $\Delta_p$ | $2.422_1$ | $3.000_3$# | $3.833_4$# | $4.875_6$# | $2.526_2$# | $4.344_5$# |
| $E_s$ | $1.552_1$ | $5.250_6$# | $4.990_5$# | $2.708_2$# | $3.432_4$# | $3.068_3$# |
| SPD | $1.573_1$ | $4.573_5$# | $5.062_6$# | $3.531_4$# | $3.042_2$# | $3.219_3$# |
| **Average** | $2.670_1$ | $3.870_5$ | $4.030_6$ | $3.599_3$ | $2.967_2$ | $3.861_4$ |

our claim that one of the clear advantages of IMIA is its invariance with respect to the Pareto front shape and the generation of approximation sets with high coverage and diversity.

### D. Parallel Performance

In this section, we analyze in depth the overall computational time of IMIA and the computational time during which each island is executed. Due to the high amount of data generated from this study, the complete results are available at http://computacion.cs.cinvestav.mx/∼jfalcon/IMIA/.

IMIA is a parallel MOEA where islands (IB-MOEAs) are executed simultaneously, and every $f_{\text{mig}}$ iterations they communicate with each other following a synchronous scheme. A critical factor related to the execution time of IMIA is the size of the subpopulations on each island. Fig. 4 compares the execution time of IMIA when solving the DTLZ2 problem with 2–7 objective functions and total population sizes of 50 and 140 individuals, which implies subpopulations of 10 and 28 individuals on each island, respectively. In Fig. 4(a), related to 50 individuals, the execution time follows a linear behavior while in Fig. 4(c) a nonlinear behavior is shown for 140 individuals. This is a consequence of the subpopulation size and the cost of the HV island. The boxplots in Fig. 4(b) and (d) show for how much time is each island executed. For 140 individuals and as we increase the dimensionality of objective space, the HV island controls the execution time of IMIA, while the execution time of the remaining IB-MOEAs is very low. In contrast, for 50 individuals, the execution times of all the islands are similar regardless of the number of objective functions. However, from the results in Section V-B, we know that the performance of IMIA increases as the population size does. Consequently, there is a tradeoff between execution time and performance.

An important point to emphasize is the idle times on each island. Since the HV island controls the overall execution time

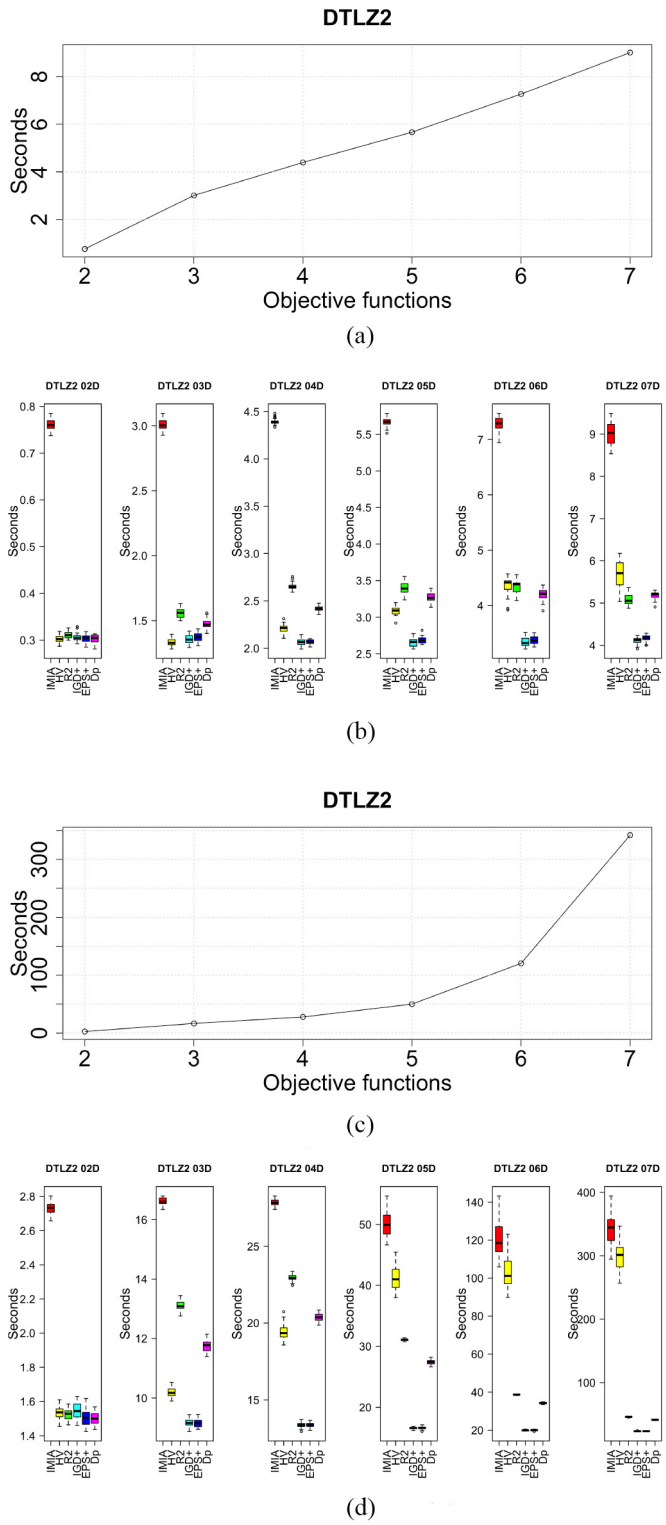


Fig. 4. Comparison of execution time when varying the dimensionality of the objective space and the population size for DTLZ2. (a) Execution time with a population size of 50 individuals. (b) Overall execution time of IMIA and each island with a population size of 50 individuals. (c) Execution time with a population size of 140 individuals. (d) Overall execution time of IMIA and each island with a population size of 140 individuals.

of IMIA, as we increase the population size and the number of objective functions, the remaining islands have too much idle time due to the synchronous migration [see Fig. 4(b) and (d)].

TABLE VI
MEAN RELATED TO THE STATISTICAL RANKS OF THE COMPARISON BETWEEN IMIA AND CMIB-MOEA. A SYMBOL # IS PLACED WHEN IMIA'S RANK IS SIGNIFICANTLY BETTER THAN CMIB-MOEA BASED ON A ONE-TAILED WILCOXON TEST, USING A SIGNIFICANCE LEVEL OF $\alpha = 0.05$. THE BEST VALUE IS SHOWN IN GRAY SCALE

| QI | IMIA | cMIB-MOEA |
|---|---|---|
| HV | 1.649# | 1.350 |
| $R2$ | 1.402 | 1.597# |
| $\text{IGD}^+$ | 1.655# | 1.344 |
| $\epsilon^+$ | 1.534# | 1.465 |
| $\Delta_p$ | 1.454 | 1.545# |
| $E_s$ | 1.385 | 1.614# |
| SPD | 1.287 | 1.712# |

In a future improvement of IMIA, an asynchronous migration scheme could be considered to tackle this issue. However, from the parallel MOEAs, we know that there is also a tradeoff between performance quality and the communication scheme adopted.

### *E. Comparing IMIA With cMIB-MOEA*

This section is devoted to briefly show that IMIA is better than cMIB-MOEA. In a similar fashion to IMIA, cMIB-MOEA is set to use five islands based on HV, $R2$, $\text{IGD}^+$, $\epsilon^+$, and $\Delta_p$, adopting micropopulations of size $\mu/5$, where $\mu = 140$ and it uses the same $n_{\text{mig}}$ and $f_{\text{mig}}$ values. We compared both algorithms using the DTLZ, $\text{DTLZ}^{-1}$, WFG, and $\text{WFG}^{-1}$ test suites with 2–7 objective functions. We employed HV, $R2$, $\text{IGD}^{-1}$, $\epsilon^+$, $\Delta_p$, $E_s$, and SPD to compare the performances. The parameters settings stated in Table II are utilized for the experimentation. Due to the large amount of data, Table VI shows the statistical ranks obtained by both algorithms. Similar to the comparisons between IMIA and the panmictic IB-MOEAs and the state-of-the-art MOEAs, IMIA is the best algorithm regarding $R2$, $\Delta_p$, $E_s$, and SPD. As stated before, this implies that IMIA generates Pareto front approximations with higher diversity in comparison with cMIB-MOEA.

## VI. CONCLUSION AND FUTURE WORK

In this article, we analyzed the cooperation of multiple IB-MOEAs as the key idea to generate an optimizer with a robust performance. Our proposed approach, called IMIA, is an island-based MOEA in which multiple IB-MOEAs, using micropopulations, cooperate to combine their search preferences for producing high-quality Pareto front approximations. Our experimental results based on a plethora of MOPs with different search difficulties and Pareto front geometries showed that IMIA has a more robust performance than the panmictic versions of its baseline IB-MOEAs. Furthermore, due to the Pareto front shape invariance of IMIA, our proposal is able to generate approximation sets with higher diversity in comparison with several state-of-the-art MOEAs specifically designed to tackle MOPs with irregular Pareto front geometries. As part of our future work, we aim to study the impact of the migration parameters and the connection topology of the

islands. Finally, we aim to design an asynchronous migration mechanism that allows IMIA to reduce the idle times in its islands.

## ACKNOWLEDGMENT

J. G. Falcón-Cardona acknowledges support from CONACyT and CINVESTAV-IPN to pursue graduate studies in computer science.

## REFERENCES

[1] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Surveys*, vol. 52, no. 2, p. 26, Mar. 2019.

[2] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY, USA: Springer, Sep. 2007.

[3] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[4] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Ph.D. dissertation, Institut Technische Informatik Kommunikationsnetze Comput. Eng. Netw. Lab., Swiss Federal Inst. Technol. (ETH), Zurich, Switzerland, Nov. 1999.

[5] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the $R2$ indicator," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Philadelphia, PA, USA, Jul. 2012, pp. 465–472.

[6] C. A. Coello Coello and N. C. Cortés, "An approach to solve multiobjective optimization problems based on an artificial immune system," in *Proc. 1st Int. Conf. Artif. Immune Syst. (ICARIS)*, Sep. 2002, pp. 212–221.

[7] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Proc. 8th Int. Conf. Evol. Multi-Criterion Optim.*, Mar./Apr. 2015, pp. 110–125.

[8] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 504–522, Aug. 2012.

[9] J. G. Falcón-Cardona and C. A. Coello Coello, "Indicator-based multi-objective evolutionary algorithms: A comprehensive survey," *ACM Comput. Surveys*, vol. 53, no. 2, pp. 1–35, Mar. 2020.

[10] J. G. Falcón-Cardona and C. A. Coello Coello, "Convergence and diversity analysis of indicator-based multi-objective evolutionary algorithms," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 524–531.

[11] R. Tanabe and H. Ishibuchi. "An analysis of quality indicators using approximated optimal distributions in a three-dimensional objective space," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 853–867, Oct. 2020.

[12] A. Liefooghe and B. Derbel, "A correlation analysis of set quality indicator values in multiobjective optimization," in *Proc. Genet. Evol. Comput. Conf.*, Jul. 2016, pp. 581–588.

[13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[14] J. G. Falcón-Cardona, C. A. Coello Coello, and M. Emmerich, "CRI-EMOA: A Pareto-front shape invariant evolutionary multi-objective algorithm," in *Proc. 10th Int. Conf. Evol. Multi-Criterion Optim.*, Mar. 2019, pp. 307–318.

[15] D. H. Phan and J. Suzuki, "Boosting indicator-based selection operators for evolutionary multiobjective optimization algorithms," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 276–281.

[16] D. H. Phan, J. Suzuki, and I. Hayashi, "Leveraging indicator-based ensemble selection in evolutionary multiobjective optimization algorithms," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Jul. 2012, pp. 497–504.

[17] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924–938, Dec. 2016.

[18] R. H. Gómez and C. A. Coello Coello, "A hyper-heuristic of scalarizing functions," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Berlin, Germany, Jul. 2017, pp. 577–584.

[19] J. G. Falcón-Cardona and C. A. Coello Coello, "A multi-objective evolutionary hyper-heuristic based on multiple indicator-based density estimators," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Kyoto, Japan, Jul. 2018, pp. 633–640.

[20] J. G. Falcón-Cardona, M. T. M. Emmerich, and C. A. Coello Coello, "On the cooperation of multiple indicator-based multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 2050–2057.

[21] J. G. Falcón-Cardona, A. Liefooghe, and C. A. Coello Coello, "An ensemble indicator-based density estimator for evolutionary multi-objective optimization," in *Parallel Problem Solving From Nature—PPSN XVI*, T. Bäck *et al.*, Eds. Leiden, The Netherlands: Springer, 2020, pp. 201–214.

[22] J. G. Falcón-Cardona, M. T. M. Emmerich, and C. A. Coello Coello, "On the construction of Pareto-compliant combined indicators," CINVESTAV-IPN, Mexico City, CDMX, Mexico, Rep. EVOCINV-01-2020, Jan. 2020.

[23] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA–II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[25] D. P. Hardin and E. B. Saff, "Discretizing manifolds via minimum energy points," *Notices AMS*, vol. 51, no. 10, pp. 1186–1194, 2004.

[26] V. Basto-Fernandes, I. Yevseyeva, A. Deutz, and M. Emmerich, "A survey of diversity oriented optimization: Problems, indicators, and algorithms," in *EVOLVE: A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation VII*, M. Emmerich, A. Deutz, O. Schütze, P. Legrand, E. Tantar, and A.-A. Tantar, Eds. Cham, Switzerland: Springer, 2017, pp. 3–23.

[27] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Dept. Electr. Comput. Eng., Graduate Sch. Eng., Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, May 1999.

[28] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[29] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.

[30] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[31] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Problems (EUROGEN)*, Athens, Greece, 2001, pp. 95–100.

[32] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.

[33] J.-H. Yi, S. Deb, J. Dong, A. H. Alavi, and G.-G. Wang, "An improved NSGA-III algorithm with adaptive mutation operator for big data optimization problems," *Future Gener. Comput. Syst.*, vol. 88, pp. 571–585, Nov. 2018.

[34] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[35] J.-H. Yi *et al.*, "Behavior of crossover operators in NSGA-III for large-scale optimization problems," *Inf. Sci.*, vol. 509, pp. 470–487, Jan. 2020.

[36] Z.-M. Gu and G.-G. Wang, "Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization," *Future Gener. Comput. Syst.*, vol. 107, pp. 49–69, Jun. 2020.

[37] Y. Zhang, G.-G. Wang, K. Li, W.-C. Yeh, M. Jian, and J. Dong, "Enhancing MOEA/D with information feedback models for large-scale many-objective optimization," *Inf. Sci.*, vol. 522, pp. 1–16, Jun. 2020.

[38] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[39] J. Sun, Z. Miao, D. Gong, X.-J. Zeng, J. Li, and G. Wang, "Interval multiobjective optimization with memetic algorithms," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3444–3457, Aug. 2020.

[40] Y. Tian, R. Liu, X. Zhang, H. Ma, K. C. Tan, and Y. Jin, "A multi-population evolutionary algorithm for solving large-scale multi-modal multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, early access, Dec. 15, 2020, doi: 10.1109/TEVC.2020.3044711.

[41] N. Kouka, R. Fdhila, A. Hussain, and A. M. Alimi, "Dynamic multi objective particle swarm optimization with cooperative agents," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2020, pp. 1–8.

[42] R. Hernández-Gómez, C. A. Coello Coello, and E. Alba, "A parallel version of SMS-EMOA for many-objective optimization problems," in *Proc. 14th Int. Conf. Parallel Problem Solving Nat. (PPSN XIV)*, Sep. 2016, pp. 568–577.

[43] J. G. Falcón-Cardona, H. Ishibuchi, and C. A. Coello Coello, "Riesz s-energy-based reference sets for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2020, pp. 1–8.

[44] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, Feb. 2012.

[45] J. G. Falcón-Cardona and C. A. Coello Coello, "Towards a more general many-objective evolutionary optimizer," in *Proc. 15th Int. Conf. Parallel Problem Solving Nat. (PPSN XV)*, Sep. 2018, pp. 335–346.

[46] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[47] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[48] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[49] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.

[50] S. Z. Martínez, V. A. S. Hernández, H. Aguirre, K. Tanaka, and C. A. Coello Coello, "Using a family of curves to approximate the pareto front of a multi-objective optimization problem," in *Proc. 13th Int. Conf. Parallel Problem Solving From Nature (PPSN XIII)*, Sep. 2014, pp. 682–691.

[51] D. P. Hardin and E. B. Saff, "Minimal Riesz energy point configurations for rectifiable $d$-dimensional manifolds," *Adv. Math.*, vol. 193, no. 1, pp. 174–204, 2005.

[52] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Comparison of hypervolume, IGD and IGD+ from the viewpoint of optimal distributions of solutions," in *Evolutionary Multi-Criterion Optimization*, K. Deb *et al.*, Eds. East Lansing, MI, USA: Springer, 2019, pp. 332–345.