




# DeepEMO: A Multi-indicator Convolutional Neural Network-Based Evolutionary Multi-objective Algorithm

Emilio Bernal-Zubieta<sup>(✉)</sup>, Jesús Guillermo Falcón-Cardona<sup>(ID)</sup>,  
and Jorge M. Cruz-Duarte<sup>(ID)</sup>

Tecnologico de Monterrey, School of Engineering and Sciences, Ave. Eugenio Garza  
Sada 2501, 64849 Monterrey, NL, México  
{a01570751, jfalcon, jorge.cruz}@tec.mx

**Abstract.** Quality Indicators (QIs) have been used in numerous Evolutionary Multi-objective Optimization Algorithms (EMOAs) as selection mechanisms within the evolutionary process. Because each QI prefers specific point-distribution properties, an Indicator-based EMOA (IB-EMOA) that uses a single QI has an intrinsically limited scope of problems it can solve accurately. To overcome the issues that IB-EMOAs have, we present the first results of a new general multi-indicator-based multi-objective evolutionary algorithm, denoted as DeepEMO. It uses a Convolutional Neural Network (CNN) as a hyper-heuristic to choose, depending on the Pareto-front geometry, the appropriate indicator-based selection mechanism at each generation of the evolutionary process. We employ state-of-the-art benchmark problems with different Pareto front geometries to test our approach. Our experimental results show that DeepEMO obtains competitive performance across multiple QIs. This is because the CNN is employed to classify the geometry of the point cloud that approximates the Pareto front. Hence, DeepEMO compensates for the weaknesses of a single QI with the strengths of others, showing that its performance is invariant to the Pareto front geometry.

**Keywords:** Quality Indicators · Multi-Objective Optimization · Convolutional Neural Networks · Hyper-heuristics · Higher Education · Educational Innovation

## 1 Introduction

In many scientific, industrial, and engineering fields, some problems involve simultaneously optimizing  $m$  conflicting objective functions. These problems are known as Multi-objective Optimization Problems (MOPs) [8]. Unlike single-objective optimization problems, the solution to a MOP is a set denoted as the *Pareto set* of optimal solutions, and its corresponding image in objective space is the so-called *Pareto front* that shows the trade-off between the conflicting objectives. (In multi-objective optimization, it is expected to use the Pareto

dominance relation to induce a strict partial order and, thus, define an optimality criterion.) It is worth noting that the Pareto front is a manifold of dimension at most  $m - 1$ .

In the specialized literature, different techniques exist to solve MOPs, ranging from mathematical programming to bio-inspired metaheuristics [8]. Despite mathematical programming methods ensuring optimal solutions, they require the objectives to be differentiable once (or even twice), which is only possible if the objectives have an analytical definition. Another critical issue is that these techniques often generate a single solution per execution. In consequence, bio-inspired metaheuristics, such as Evolutionary Multi-objective Optimization Algorithms (EMOAs) [12, 24, 28, 32, 34], have emerged as promising methods to tackle MOPs. EMOAs are stochastic, population-based, and derivative-free methods that approximate the MOP's solution. Although they cannot ensure the optimality of solutions, EMOAs have been successfully applied to different complex real-world problems where mathematical programming techniques have difficulties.

In this regard, the output of an EMOA stands for a finite set of approximately optimal solutions whose image composes a Pareto front approximation. Such an approximation is a finite representation of the manifold associated with the Pareto front, *i.e.*, an  $N$ -point cloud. Ideally, the Pareto front approximation should be as close to the true Pareto front as possible. Hence, these points should also cover the whole Pareto front, showing a good distribution regardless of the Pareto front shape [21]. Nevertheless, in recent years, Ishibuchi *et al.* emphasized that the performance of some EMOAs depends on the Pareto front shape [15]. Consequently, different approaches have been proposed to tackle this critical issue [12, 24, 28, 32, 34].

On the one hand, an effective approach to designing EMOAs with performance invariant to the Pareto front geometry is the use of multiple indicator-based selection mechanisms, giving rise to the Multi-Indicator-based EMOAs (MIB-EMOAs) [12, 32]. Quality indicators (QIs) are the core of every MIB-EMOA [21]. A unary QI is a set function that evaluates a Pareto front approximation's quality (convergence, spread, or distribution) based on specific preferences. In other words, a QI assigns a real number to a Pareto front approximation. Hence, it is possible to search for the Pareto front approximation that optimizes a QI. That is, we can define an Indicator-based Subset Selection Problem (IBSSP) that, in terms of EMOAs, involves the selection of the fittest solutions according to the QI value. Thus, those objective vectors that approximate the solution of an IBSSP exhibit the preferences of the baseline QI; *i.e.*, they approach the optimal  $\mu$ -distribution of the QI. Considering the previous concepts, an MIB-EMOA exploits the strengths of a set of QIs to compensate for the weaknesses of a particular one. For instance, Wang *et al.* proposed the Two\_Arch2 algorithm that uses two archives, each based on a specific QI, to improve the convergence and diversity properties of a Pareto front approximation [32]. Notwithstanding, another design strategy is conceptualized by the Island-based Multi-Indicator Algorithm (IMIA), where the cooperation between multiple Indicator-based EMOAs (IB-

EMOAs) is exploited [12]. In this strategy, each island of IMIA evolves a micro-population using an IB-EMOA with a different QI. After some generations, some individuals migrate between islands, aiming to improve the diversity of the other islands.

On the other hand, data processing by learning models is at the heart of today’s artificial intelligence revolution. Point clouds, like those produced by the EMOA approximation sets, are an essential data type that these models can process. Some applications of point clouds worth mentioning include robotics, indoor navigation, and self-driving vehicles. Plus, their analysis, namely point cloud classification and segmentation, has become relevant in recent years. Though traditional Deep Neural Networks (DNNs) require input data with a regular structure, point clouds have an irregular structure. Thus, it is clear that permutation invariance within the DNN is crucial due to point clouds’ lack of topological information. Consequently, designing a DNN that can extract topological features from them is relevant. One can corroborate this claim from several point cloud classifiers proposed to tackle these issues. For instance, PointNet [6] uses the *max-pooling* symmetric function to deal with the unordered input set of points. Later, PointNet++ [26] builds upon PointNet’s design and adds a local feature extractor by grouping points into neighborhoods similar to CNNs. Finally, Dynamic Graph CNN (DGCNN) [33] further exploits the CNNs implementation in point clouds by analyzing dynamically computed graphs in each network layer.

Despite EMOAs generating a point cloud in the objective space at each iteration, learning mechanisms do not exploit this information. In addition, we find no research done into using the type of geometry associated with the Pareto front approximation as a mechanism that selects from a pool the best-fitted indicator-based mechanism. So, exploiting geometric information from the point cloud can eliminate the need for sophisticated methods by leveraging the geometric biases inherent to QIs. In this regard, CNNs have yet to be used to classify Pareto front geometries and guide the selection process of an MIB-EMOA. Hence, our proposal is a pioneer work in this area. Geometric classification as a guide for MIB-EMOAs allows for exploiting the properties of individual indicator-based selection mechanisms as a hyper-heuristic. The main contributions of our work are the following.

- We propose the first CNN-based MIB-EMOA, called DeepEMO, that uses DGCNN to classify the geometry associated with the current Pareto front approximation at each generation. Then, DeepEMO chooses the best-fitted one from a pool of indicator-based selection mechanisms to guide the selection process. This is based on predefined rules that consider the effectiveness of indicator-based selection mechanisms on different geometries. For this proof-of-concept, we employed the Hypervolume Indicator (HV) [2], the discrete  $R2$  indicator [4], and the Riesz  $s$ -energy ( $E_s$ ) [3].
- We constructed a particular dataset to train DGCNN based on the Pareto fronts from several state-of-the-art benchmark problems. We also selected problems with different Pareto front geometries.

- We present a comprehensive study of the performance of DeepEMO, considering two- and three-objective problems with different Pareto front shapes. Moreover, we validate the performance of DeepEMO by comparing it to IB-EMOAs that use the baseline QIs, *i.e.*, HV,  $R2$ , and  $E_s$ . Based on different QIs, we realize that DeepEMO is a promising direction to combine EMOAs and Deep Learning.

The remainder of this paper is structured as follows. Section 2 provides the concepts that make this paper self-contained. Section 3 details DeepEMO, and Sect. 4 presents and analyzes the experimental results. Finally, Sect. 5 outlines the conclusions and possible improvements for future work.

## 2 Background

This section introduces some mathematical concepts that sustain our proposed approach. Thus, we start defining a MOP, then the notion of QI, HV,  $R2$ , and  $E_s$ , the generic IB-EMOA, and DGCNN.

### 2.1 Multi-objective Optimization Problem (MOP)

Throughout this paper, we focus on tackling, without loss of generality, unconstrained MOPs for minimization, which are defined as follows:

$$\min_{\vec{x} \in \Omega} \{f(x) := (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}))^\top\} \quad (1)$$

where  $x = (x_1, \dots, x_n)^\top$  is an  $n$ -dimensional decision vector and  $\Omega \subseteq \mathbb{R}^n$  is the decision space.  $f : \Omega \mapsto \mathbb{R}^m$  is the objective vector of  $m \geq 2$  conflicting objective functions  $f_i : \Omega \mapsto \mathbb{R}$ ,  $\forall i = 1, 2, \dots, m$ .

The most common definition of optimality in multi-objective optimization is based on the Pareto dominance relation that induces a strict partial order among the decision vectors. Then, given two solutions  $\vec{x}, \vec{y} \in \Omega$ ,  $\vec{x}$  is said to Pareto dominate  $\vec{y}$  (denoted as  $\vec{x} \prec \vec{y}$ ) if  $f_i(\vec{x}) \leq f_i(\vec{y})$ ,  $\forall i = 1, 2, \dots, m$ , and there exists at least an index  $j \in \{1, 2, \dots, m\}$  such that  $f_j(\vec{x}) < f_j(\vec{y})$ . One can claim that  $\vec{x}^* \in \Omega$  is a Pareto optimal solution if there is no other  $\vec{x} \in \Omega$  such that  $\vec{x} \prec \vec{x}^*$ . Due to the conflict among the objectives, there is not a single Pareto optimal solution but a set of Pareto optimal solutions denoted as the Pareto set, whose image is the so-called Pareto front. Since the Pareto set cardinality could be infinite, some algorithms that tackle MOPs produce a finite approximation set  $\mathcal{A} = \{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_N\}$ , where  $\vec{a}_i \in \Omega$ . Ideally,  $\vec{a}_i \not\prec \vec{a}_j$  and  $\vec{a}_j \not\prec \vec{a}_i$  for every  $i \neq j$ , *i.e.*,  $\mathcal{A}$  has mutually non-dominated solutions. The Pareto front approximation is the image  $f(\mathcal{A})$ .

### 2.2 Quality Indicator (QI)

A QI ( $\mathcal{I}$ ) is a set function that assigns a real value to a given number  $k$  of Pareto front approximations [21]. That is, a  $k$ -ary indicator is defined as  $\mathcal{I} : \Psi^k \mapsto \mathbb{R}$ ,

where  $\Psi$  is the set of all possible finite Pareto front approximations. When  $k = 1$ , the QI is known as a unary indicator. Currently, many QIs measure the three main properties of a Pareto front approximation, *i.e.*, convergence, uniformity, and spread [21]. In the following lines, we briefly describe three well-known indicators considered in this work.

The Hypervolume Indicator (HV) is the most popular QI due to its mathematical properties [2]. HV measures the region weakly dominated by  $\mathcal{A}$  and bounded by an anti-optimal reference point  $\vec{r}$ . It simultaneously measures convergence and spread and is the only Pareto-compliant QI. Therefore, given an approximation set  $\mathcal{A}$  and a reference point  $\vec{r} \in \mathbb{R}^m$  dominated by all points in  $\mathcal{A}$ , HV is defined as:

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

where  $\mathcal{L}$  is the Lebesgue measure in  $\mathbb{R}^m$ . It is worth mentioning that we abuse notation since  $\vec{r}$  is in the objective space. However, the Pareto dominance relation (defined above) induces a strict partial order in  $\Omega$  by checking the objective vectors of the solutions. Thus, we can compare  $f(\vec{a})$ ,  $f(\vec{b})$ , and  $\vec{r}$ .

Another well-known QI is the discrete  $R2$  indicator [4].  $R2$  is a convergence-uniformity indicator that uses a set of weight vectors ( $W$ ) in  $\mathbb{R}^m$  to measure the average minimum utility value generated by a Pareto front approximation. Unlike HV, whose computational cost is high, the cost of  $R2$  is  $\mathcal{O}(m|\mathcal{A}||W|)$ , but it is weakly Pareto-compliant. So, for a given set of  $m$ -dimensional weight vectors  $W$  and a utility function  $u_{\vec{w}} : \mathbb{R}^m \mapsto \mathbb{R}$ , the  $R2$  indicator is defined as follows:

$$R2(\mathcal{A}, W) = \frac{1}{|W|} \sum_{\vec{w} \in W} \min_{\vec{a} \in \mathcal{A}} u_{\vec{w}}(f(\vec{a})). \quad (3)$$

Lastly and more recently, the Riesz  $s$ -energy ( $E_s$ ) has been employed in evolutionary multi-objective optimization to generate well-diversified solution sets [11].  $E_s$  is a pair-potential energy function taken from physics that measures the interaction between pairs of particles in an  $N$ -point set. Despite  $E_s$  being used mainly for subset selection in EMO, it can also be used as a diversity indicator. Hence, given a Pareto front approximation  $\mathcal{A}$  and  $s > 0$ ,  $E_s$  is determined by:

$$E_s(\mathcal{A}) = \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \frac{1}{\|f(\vec{a}_i) - f(\vec{a}_j)\|^s}. \quad (4)$$

### 2.3 Indicator-Based EMOA (IB-EMOA)

This section introduces a generic steady-state IB-EMOA, which is based on the framework of  $\mathcal{S}$ -Metric Selection EMOA (SMS-EMOA), that employs HV [2]. Regardless of the QI, the backbone of this generic IB-EMOA is the contribution

( $C$ ) of a single solution ( $\vec{x} \in \mathcal{A}$ ) to the overall indicator value. This contribution value is calculated as:

$$C_{\mathcal{I}}(\vec{x}, \mathcal{A}) = |\mathcal{I}(\mathcal{A}) - \mathcal{I}(\mathcal{A} \setminus \{\vec{x}\})|. \quad (5)$$

Considering the contribution value, it is possible to define a heuristic method to approximate the solution of an indicator-based subset selection problem. In other words, given a Pareto front approximation of size  $\mu + \lambda$ , we aim to find  $\mathcal{A}'$  such that  $|\mathcal{A}'| = \mu$  and  $\mathcal{I}(\mathcal{A}')$  is maximum. (Without loss of generality, we assume that maximizing  $\mathcal{I}$  implies better quality.)

Algorithm 1 outlines the generic steady-state IB-EMOA whose main loop comprises lines 3 to 14. First, a new solution  $\vec{y}$  is generated via variation operators and joined with the current population  $P_t$  to define a temporary population  $Q$  of size  $N + 1$ . Then, in line 6,  $Q$  is sorted using the non-dominated sorting algorithm [9] to define a set of layers  $\{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_p\}$ . It is worth noting that layer  $\mathcal{L}_p$  contains a subset of solutions of  $Q$ , which are the worst regarding the Pareto dominance relation. If the cardinality of  $\mathcal{L}_p$  is greater than 1, then we calculate which is the worst-contributing  $\vec{x}_{\text{worst}}$  solution to  $\mathcal{I}$  according to (5). Otherwise,  $\vec{x}_{\text{worst}}$  is the sole solution in  $\mathcal{L}_p$ . In line 12,  $\vec{x}_{\text{worst}}$  is deleted from  $Q$  to determine the population for the next iteration  $t + 1$ . The algorithm outputs the last population as the approximation set.

---

### Algorithm 1. Generic Steady-State IB-EMOA

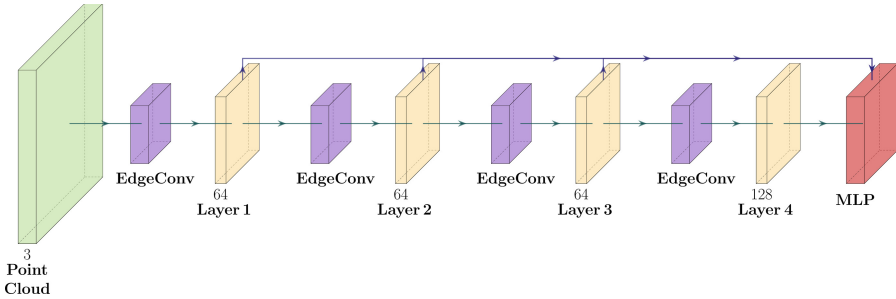
---

**Input:** Indicator  $\mathcal{I}$   
**Output:** Approximation set  $\mathcal{A}$   
1: Randomly initialize population  $P_0$   
2: Set  $t \leftarrow 0$   
3: **while** Stopping criterion is not fulfilled **do**  
4:    $\vec{y} \leftarrow \text{VARIATION}(P_t)$   
5:    $Q \leftarrow P_t \cup \{\vec{y}\}$   
6:    $\{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_p\} \leftarrow \text{NONDOMINATEDSORTING}(Q)$   
7:   **if**  $|\mathcal{L}_p| > 1$  **then**  
8:      $\vec{x}_{\text{worst}} \leftarrow \underset{\vec{x} \in \mathcal{L}_p}{\text{argmin}} C_{\mathcal{I}}(\vec{x}, \mathcal{L}_p)$   
9:   **else**  
10:      $\vec{x}_{\text{worst}} \leftarrow$  the sole individual in  $\mathcal{L}_p$   
11:   **end if**  
12:    $P_{t+1} \leftarrow Q \setminus \{\vec{x}_{\text{worst}}\}$   
13:    $t \leftarrow t + 1$   
14: **end while**  
15: **return**  $\mathcal{A} \leftarrow P_t$

---

Algorithm 1 follows the framework of the SMS-EMOA, which is a steady-state IB-EMOA. To reproduce the SMS-EMOA behavior with Algorithm 1, we have to set  $\mathcal{I} = \text{HV}$ . So, HV is to be maximized; the worst-contributing solution to HV is the one with the minimum contribution value. Depending on the definition of  $\vec{r}$ , the preferences of SMS-EMOA may change. For instance, if  $\vec{r}$  is approximately equal to the nadir point, SMS-EMOA generates uniform Pareto front approximations in linear triangular Pareto fronts, or it can produce solutions in the boundary and around the Pareto front's knee when the geometry

is concave triangular. Since SMS-EMOA has to perform multiple calculations of HV (which increases super-polynomially with the number of objectives), it is computationally expensive. Other less computationally expensive but weaker QIs have been used to avoid this issue. For instance, Brockhoff *et al.* proposed  $R2$ -EMOA that uses the  $\mathcal{I} = R2$  indicator [5]. Unlike SMS-EMOA,  $R2$ -EMOA generates uniform Pareto front approximations in both linear triangular and concave triangular Pareto fronts. However, it has issues when tackling disconnected or degenerate Pareto fronts. Finally, in case that  $\mathcal{I} = E_s$ , we can generate an IB-EMOA that will show the preferences of  $E_s$ , and we denote it as  $E_s$ -EMOA.



**Fig. 1.** Dynamic Graph Convolutional Neural Network (DGCNN) architecture.

## 2.4 Dynamic Graph Convolutional Neural Network (DGCNN)

DGCNN [33] is a point cloud classifier inspired by similar works like PointNet [6]. Its main feature is its ability to capture local geometric structures while maintaining permutation invariance. This is achieved through an operation called *edge convolution* (EdgeConv). Given a point cloud, EdgeConv constructs a directed graph using the  $k$ -Nearest Neighbors ( $k$ -NN) algorithm, similar to graph CNNs. According to the authors, DGCNN outperforms other point cloud classifiers because the EdgeConv process is recomputed after each layer of the CNN. Hence, the graph is dynamically updated and not fixed like in traditional graph CNNs. [33]

Due to the DNN architecture employed, the hidden layers work in the feature space created by the previous layer. DGCNN features four hidden layers and the input and output layers, as shown in Fig. 1. The first three hidden layers are made up of 64 neurons, while the last hidden layer is made up of 128 neurons. The input layer of DGCNN consists of a set of  $N$  three-dimensional real-valued points. Hence, we could feed DGCNN with  $f(\mathcal{A})$ , where  $\mathcal{A}$  is the approximation set generated by an EMOA for a three-objective MOP. At each layer of DGCNN, EdgeConv constructs a directed graph, extracting local geometric information by connecting neighboring points. The graph's edges are then used to compute *edge features* via a nonlinear function  $h_{\Theta}$  with parameters  $\Theta$ . The edge features

are then fed into a *max-pooling* operation with a *ReLU* activation function that captures global shape structure and local neighborhood information. The features outputted by the last EdgeConv layer are then globally aggregated by another *max-pooling* operator, forming a 1D global descriptor used to generate the  $c$  classification label in the output layer.

### 3 Proposed Approach

Our proposal, called DeepEMO, is a steady-state MIB-EMOA that employs a heuristic selection mechanism (based on the classification label produced by DGCNN) to execute the best-fitted indicator-based selection mechanism according to specific rules. The following sections introduce DeepEMO’s general framework and how we incorporate DGCNN into an EMOA.

---

#### Algorithm 2. DeepEMO General Framework

---

**Input:** Certainty threshold  $\beta$   
**Output:** Approximation set  $\mathcal{A}$

- 1: Randomly initialize population  $P_0$
- 2: Set  $t \leftarrow 0$
- 3: **while** Stopping criterion is not fulfilled **do**
- 4:    $\vec{y} \leftarrow \text{VARIATION}(P_t)$
- 5:    $Q \leftarrow P_t \cup \{q\}$
- 6:   Normalize  $Q$
- 7:    $\{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_p\} \leftarrow \text{NONDOMINATEDSORTING}(Q)$
- 8:   **if**  $|\mathcal{L}_p| > 1$  **then**
- 9:     **geometry, certainty**  $\leftarrow \text{DGCNN}(f(Q))$
- 10:     **if** geometry is *convex* **and** certainty  $\geq \beta$  **then**
- 11:        $\vec{x}_{\text{worst}} \leftarrow \underset{\vec{x} \in \mathcal{L}_p}{\text{argmin}} C_{\text{HV}}(\mathcal{L}_p, \vec{z}_{\text{ref}})$
- 12:     **else if** geometry is *concave* **and** certainty  $\geq \beta$  **then**
- 13:        $\vec{x}_{\text{worst}} \leftarrow \underset{\vec{x} \in \mathcal{L}_p}{\text{argmin}} C_{\text{R2}}(\mathcal{L}_p, W)$
- 14:     **else**
- 15:        $\vec{x}_{\text{worst}} \leftarrow \underset{\vec{x} \in \mathcal{L}_p}{\text{argmax}} C_{E_s}(\mathcal{L}_p)$
- 16:     **end if**
- 17:   **else**
- 18:      $\vec{x}_{\text{worst}} \leftarrow$  the sole individual in  $\mathcal{L}_p$
- 19:   **end if**
- 20:    $P_{t+1} \leftarrow Q \setminus \{\vec{x}_{\text{worst}}\}$
- 21:    $t \leftarrow t + 1$
- 22: **end while**
- 23: **return**  $\mathcal{A} = P_t$

---

#### 3.1 General Framework

The general framework of DeepEMO is presented in Algorithm 2. It follows a similar structure to Algorithm 1. Lines 8 to 17 encompass the core idea of DeepEMO. Our proposed EMOA employs a hyper-heuristic that uses a set of predefined rules to select the best-fitted indicator-based density estimator. The selection rules are based on previous studies on the convergence and diversity properties



of indicator-based density estimators [23]. We used HV,  $R2$ , and  $E_s$  for this proof-of-concept to define individual density estimators. According to the literature, we know that an HV-based density estimator has a good performance on MOPs whose Pareto front geometry is convex. This is because HV rewards solutions around the Pareto front’s knee and on the boundaries.  $R2$  is suitable for triangular concave Pareto front shapes because of the utilization of the simplex-like weight vectors.  $E_s$  is an appropriate strategy for other Pareto front geometries [11]. Hence, in line 9 of Algorithm 2, we feed a previously trained DGCNN (described in the next section) with the approximation set  $Q$  image. DGCNN returns the classification label and a `certainty` value. We use the degree of certainty in tandem with the geometric classification because the model might not be entirely sure of the Pareto front geometry. In such a case, applying a more general QI (*e.g.*, the Riesz  $s$ -energy) would be preferable to other more specialized indicators. If the `geometry` is convex and `certainty` is greater than or equal to a user-supplied threshold ( $\beta$ ), then the HV-based density estimator is performed in line 11. In case the `geometry` is concave and `certainty`  $\geq \beta$ , the  $R2$ -based density estimator is executed in line 13. Otherwise, the  $E_s$ -based density estimator is performed by default in line 15. It is worth noting that we set  $\beta = 10\%$  based on previous experiments. A limitation of DeepEMO is that it can only tackle two- and three-objective MOPs. This problem stems from using DGCNN, which can only classify two- and three-dimensional point clouds. This is unsurprising since point clouds usually represent real-world objects; therefore, DGCNN cannot classify point clouds of dimension four or more.

### 3.2 Using DGCNN in DeepEMO

To use DGCNN in DeepEMO, training the model with data related to Pareto front approximations is mandatory. Hence, we constructed a special dataset (using the format required by DGCNN) that contains  $m$ -dimensional points from normalized Pareto front approximations of size 50, varying the related geometries. We obtained the data from thirteen EMOAs, available in PlatEMO [29], with distinct preferences: NSGA-II [9], MOEA/D [36], MOEA/DD [18], MOMBI-II [13], AdaW [22], BiGE [20], SPEA2+SDE [19], RPEA [25], RVEA-iGNG [24], SRA [17], SPEA-R [16], t-DEA [35], and Two\_Arch2 [32]. Aiming to maximize the range of geometries, we selected problems from the following test suites: Deb-Thiele-Laumanns-Zitzler (DTLZ) [10], Irregular MOPs (IMOPs) [30], Viennet test suite (VIE) [31], and the Walking-Fish-Group (WFG) [14]. Specifically, we chose the problems DTLZ1, DTLZ2, DTLZ5, DTIZ7, WFG1, WFG2, and WFG3 with two and three objectives, and IMOP1-IMOP8 and VIE1-VIE3 using the given fixed number of objectives. By default, DGCNN can only process three-dimensional point clouds; thus, we added a fictional variable with a zero value to two-objective Pareto front approximations to make them compatible with DGCNN. Finally, the dataset size was then augmented by rotating the Pareto fronts  $360^\circ$  in  $10^\circ$  intervals over the  $45^\circ$  azimuth. After data curation, we obtained a dataset of 75,600 Pareto front approximations. Then, we use a simple validation with 80% of the instances for

the training set and the rest for the test set. The model we use in DeepEMO in line 9 of Algorithm 2 is produced using the training set.

## 4 Experimental Results

We compared DeepEMO with three IB-EMOAs resulting from setting  $\mathcal{I} = \text{HV}$ ,  $R2$ , or  $E_s$  in Algorithm 1. We denote these IB-EMOAs as SMS-EMOA,  $R2$ -EMOA, and  $E_s$ -EMOA. To determine if the DGCNN-based heuristic selection is better than a simple random selection, we conducted a comparative analysis of DeepEMO with a random version, which we denote as Rand-DeepEMO. Since the five algorithms are genetic steady-state EMOAs, we used the simulated binary crossover (SBX) and polynomial-based mutation (PBM). We set the crossover and mutation probabilities equal to 0.9 and  $1/n$ , where  $n$  is the number of decision variables, respectively. Both crossover and mutation distribution indexes are equal to 20. For a fair comparison, we employed a population size of 55 solutions and a stopping criterion of 50,000 function evaluations for all the algorithms. The population size equals the number of weight vectors  $R2$ -EMOA uses, employing the Simplex-Lattice-Design (SLD) method. To calculate  $R2$ , we implemented the Achievement Scalarizing Function (ASF). Plus, for  $E_s$ -EMOA, we set the parameter  $s$  to  $m - 1$ , and for DGCNN, we established a  $g = 5$  parameter to construct the local graph via  $k$ -NN. For each algorithm in each instance, we performed 20 independent executions.

### 4.1 Test Problems

To test DeepEMO and the selected EMOAs, we used DTLZ1, DTLZ2, and DTLZ7 with three objective and their inverted variants, denoted as  $\text{DTLZ1}^{-1}$ ,  $\text{DTLZ2}^{-1}$ , and  $\text{DTLZ7}^{-1}$  [15]. We used the inverted DTLZ problems because they were not employed when training the DGCNN model. We set  $n = m + k - 1$  as the number of decision variables for these problems, where  $k = 5, 10, \text{ or } 20$  for DTLZ1, DTLZ2, and DTLZ7, and their corresponding inverted versions, respectively. The IMOP problems were also used in our comparative study because they test the ability of an EMOA to maintain diversified solutions. We employed ten decision variables for these problems, as suggested by the authors [30]. Finally, we also considered VIE1-VIE3 problems, with two-dimensional decision spaces. We must emphasize that all the selected problems have different Pareto front shapes. It is worth mentioning that DGCNN was trained using Pareto front approximations of the selected MOPs to classify the geometry of the point clouds. However, throughout the evolutionary process, DeepEMO feeds DGCNN with points not even close to the Pareto front. Hence, the training process of DGCNN does not provide DeepEMO and advantage over other EMOAs in terms of convergence behavior.

### 4.2 Performance Assessment

To measure the performance of the selected EMOAs, we used multiple QIs, *i.e.*, HV,  $R2$ ,  $E_s$ , Inverted Generational Distance (IGD) [7],  $\text{IGD}^+$ , Averaged

Hausdorff Distance ( $\Delta_p$ ) [27], additive  $\epsilon$  indicator ( $\epsilon^+$ ) [21], and the Solow-Polasky Diversity indicator (SPD) [1]. Table 1 specifies the reference point we used for HV. A set of 55 weight vectors produced by SLD was employed to define the same number of utility values based on the vector angle distance scaling function to calculate  $R2$ . Moreover, we considered  $s = m - 1$  for  $E_s$  and  $\theta = 10$  for SPD. Due to IGD, IGD $^+$ ,  $\Delta_p$ , and  $\epsilon^+$  requiring a reference point set, we obtained the image of 500 Pareto optimal solutions for each problem from PlatEMO. Plus, we conducted a Wilcoxon rank-sum test with a significance level  $\alpha = 0.05$  to get statistical confidence.

Table 2 shows the numerical comparison based on HV. Due to space limitations, Tables 2 to 9 from the Supplementary Material (freely available at <https://github.com/eBernalZ/DeepEMO>) show the numerical results of  $R2$ ,  $E_s$ , IGD, IGD $^+$ ,  $\Delta_p$ ,  $\epsilon^+$ , and SPD.

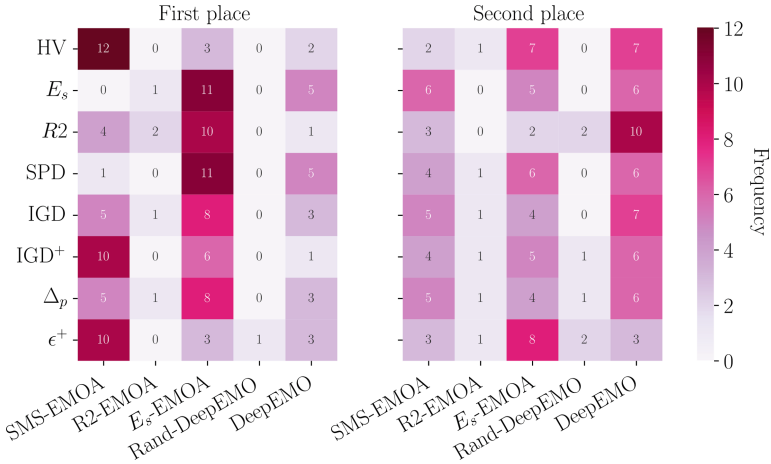
**Table 1.** Reference points employed for calculating HV per each MOP.

MOP	Reference point	MOP	Reference point
DTLZ1	(1, 1, 1)	DTLZ1 $^{-1}$	(0.1, 0.1, 0.1)
DTLZ2	(1.2, 1.2, 1.2)	DTLZ2 $^{-1}$	(0.1, 0.1, 0.1)
DTLZ7	(1, 1, 21)	DTLZ7 $^{-1}$	(0.1, 0.1, 0.1)
IMOP1 IMOP2 IMOP3	(2, 2)	IMOP4 IMOP6 IMOP7	(2, 2, 2)
IMOP5	(1.5, 1.5, 2)	IMOP8	(2, 2, 4)
VIE1	(5, 6, 5)	VIE2	(6, 6, 6)
VIE3	(19, 19, 19)		

### 4.3 Discussion

An *a posteriori* EMOA should have a *robust* performance when tackling real-world problems. By robust performance, we mean that its performance should be good for different quality measures. This is why multiple QIs are used to evaluate the performance of DeepEMO. Moreover, the core idea of DeepEMO is to compensate for the weaknesses of a given QI with the strengths of others by using the DGCNN-based heuristic selector. Figure 2 depicts the number of times that each algorithm obtained either the first or second place in the comparison for all the selected QIs. This figure reveals that SMS-EMOA and  $E_s$ -EMOA often obtain the first position in the comparisons, followed by DeepEMO. Regarding the right-hand side of the figure, we can see that DeepEMO consistently obtains the second place for all QIs. From these observations, we can argue the following. First, the outstanding performance of SMS-EMOA comes with a high computational cost (as expected) and difficulty in setting the reference point to obtain uniform Pareto front approximations. Regarding  $E_s$ -EMOA, it produces Pareto

front approximations with good diversity, but since  $E_s$  is a diversity indicator,  $E_s$ -EMOA would lose convergence pressure in MOPs with more than three objectives.

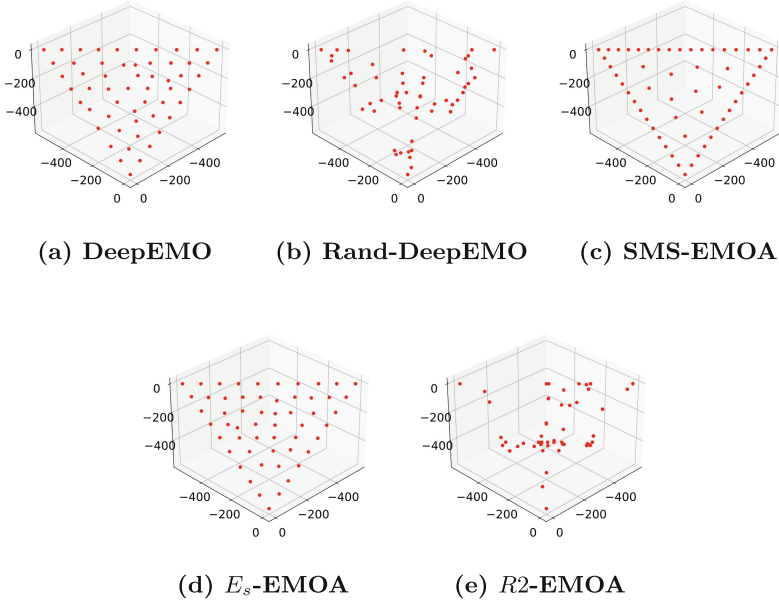


**Fig. 2.** Heatmap from the number of times an IB-EMOA was ranked first or second according to the HV,  $E_s$ , R2, SPD, IGD, IGD<sup>+</sup>,  $\Delta_p$ , and  $\epsilon^+$  indicators.

DeepEMO can be employed to compensate for the difficulties of always using a single QI in an IB-EMOA. By analyzing Table 2 related to the HV comparison, we can see that DeepEMO presents good convergence results. This is because DeepEMO crushes solutions towards the Pareto front by taking advantage of its baseline indicator-based mechanisms depending on the geometry classification of the current Pareto front approximation. Hence, in most cases, DeepEMO is less computationally expensive than SMS-EMOA because the probability of constantly applying the HV-based selection is close to zero. In this regard, due to the switching between selection mechanisms, DeepEMO generates more selection pressure, which makes it possible to scale its performance to MOPs with three or more objectives (once DGCNN scales too). By consistently obtaining the second place in the comparison as shown in Fig. 2, DeepEMO reveals that its Pareto front approximations are not biased to fulfill the preferences of a single QI (as in the case of SMS-EMOA or  $E_s$ -EMOA). This behavior is because DeepEMO generates Pareto front approximations with good diversity as illustrated in Fig. 3 for the three-objective DTLZ1<sup>-1</sup>. DeepEMO inherits this diversity property due to utilizing  $E_s$ , HV, and R2. Finally, by comparing DeepEMO and Rand-DeedEMO, we can conclude that using the rule-based heuristic selection in DeepEMO produces better results than randomly selecting indicator-based mechanisms.

**Table 2.** Mean and standard deviation (in parentheses) of HV results. A symbol # is placed when the outperforming EMOA performed significantly better than the other EMOAs based on a one-tailed Wilcoxon test using a significance level of  $\alpha = 0.05$ . The two best values are shown in grayscale, where the darkest tone corresponds to the best.

MOP	Dim.	SMS-EMOA	R2-EMOA	$E_s$ -EMOA	Rand-DeepEMO	DeepEMO
DTLZ1	3	9.717413e-01 <sup>1</sup> (6.986066e-05)	9.715930e-01 <sup>2</sup> # (8.853934e-05)	9.715261e-01 <sup>4</sup> # (1.570679e-04)	9.713076e-01 <sup>5</sup> # (2.113887e-04)	9.715414e-01 <sup>3</sup> # (1.349688e-04)
DTLZ1 <sup>-1</sup>	3	1.634825e+07 <sup>4</sup> # (1.660599e+05)	1.398368e+07 <sup>5</sup> # (1.232136e+06)	1.897918e+07 <sup>2</sup> (3.385136e+05)	1.786897e+07 <sup>3</sup> # (5.839311e+05)	1.909096e+07 <sup>1</sup> <sup>5</sup> (2.877119e+05)
DTLZ2	3	7.408208e+00 <sup>1</sup> (9.329663e-05)	7.394442e+00 <sup>4</sup> # (3.066330e-04)	7.395424e+00 <sup>2</sup> # (2.093433e-03)	7.395173e+00 <sup>3</sup> # (2.893253e-03)	7.394185e+00 <sup>5</sup> # (5.033179e-04)
DTLZ2 <sup>-1</sup>	3	5.750160e+01 <sup>1</sup> (1.960825e-02)	4.866969e+01 <sup>5</sup> # (9.853446e-01)	5.725490e+01 <sup>3</sup> # (8.940140e-02)	5.483055e+01 <sup>4</sup> # (7.272273e-01)	5.738631e+01 <sup>2</sup> # (7.059462e-02)
DTLZ7	3	1.624788e+01 <sup>1</sup> (1.716603e-01)	1.574660e+01 <sup>5</sup> # (1.278959e-01)	1.619470e+01 <sup>3</sup> # (1.926495e-01)	1.611579e+01 <sup>4</sup> # (1.483026e-01)	1.619684e+01 <sup>2</sup> # (1.378072e-01)
DTLZ7 <sup>-1</sup>	3	2.698252e+01 <sup>2</sup> (1.027667 e-01)	2.679963e+01 <sup>5</sup> # (1.101505e-01)	2.698817e+01 <sup>1</sup> (3.117033e-03)	2.692513e+01 <sup>4</sup> # (4.969799e-02)	2.695391e+01 <sup>3</sup> # (6.691855e-02)
IMOP1	2	3.984844e+00 <sup>1</sup> (8.313591e-06)	2.017798e+00 <sup>5</sup> # (2.297379e-02)	3.983902e+00 <sup>3</sup> # (7.384897e-05)	3.747995e+00 <sup>4</sup> # (2.823840e-01)	3.984837e+00 <sup>2</sup> (2.613447e-05)
IMOP2	2	3.065830e+00 <sup>1</sup> (3.212894e-03)	2.000071e+00 <sup>4</sup> # (4.775736e-05)	3.064269e+00 <sup>2</sup> # (3.320674e-03)	2.094344e+00 <sup>3</sup> # (2.732107e-01)	1.999785e+00 <sup>5</sup> # (1.329149e-03)
IMOP3	2	3.639962e+00 <sup>1</sup> (2.229995e-02)	2.379765e+00 <sup>5</sup> # (2.478631e-02)	3.634140e+00 <sup>2</sup> (2.272153e-02)	2.636026e+00 <sup>4</sup> # (8.415924e-02)	3.633433e+00 <sup>3</sup> (1.747447e-02)
IMOP4	3	6.329276e+00 <sup>1</sup> (3.019870e-02)	2.040948e+00 <sup>5</sup> # (1.462829e-02)	6.327558e+00 <sup>2</sup> (2.733316e-02)	2.364078e+00 <sup>3</sup> # (5.546803e-02)	6.324291e+00 <sup>3</sup> (3.217523e-02)
IMOP5	3	6.860433e+00 <sup>1</sup> (2.711298e-03)	6.192728e+00 <sup>5</sup> # (1.151343e+00)	6.811442e+00 <sup>3</sup> # (9.720637e-03)	6.681764e+00 <sup>4</sup> # (2.125056e-02)	6.813510e+00 <sup>2</sup> # (1.347183e-02)
IMOP6	3	6.796313e+00 <sup>3</sup> (3.095751e-01)	3.906444e+00 <sup>5</sup> # (4.870139e-01)	6.835016e+00 <sup>2</sup> # (6.571779e-03)	3.981992e+00 <sup>4</sup> # (6.573286e-01)	6.838866e+00 <sup>1</sup> (3.501328e-03)
IMOP7	3	4.688555e+00 <sup>2</sup> # (1.703501e+00)	4.001614e+00 <sup>5</sup> # (1.939027e-03)	7.356681e+00 <sup>1</sup> (4.169080e-03)	4.193473e+00 <sup>3</sup> # (7.405203e-01)	4.187927e+00 <sup>4</sup> # (1.074160e-01)
IMOP8	3	1.467495e+01 <sup>3</sup> (1.703501e+00)	7.379639e+00 <sup>5</sup> # (1.644353e+00)	1.490112e+01 <sup>1</sup> (3.675314e-02)	1.211802e+01 <sup>4</sup> (4.037208e+00)	1.485823e+01 <sup>2</sup> (1.074160e-01)
VIE1	3	6.141673e+01 <sup>1</sup> (9.511744e-03)	5.842468e+01 <sup>4</sup> # (9.156291e-01)	6.072238e+01 <sup>2</sup> # (7.808997e-02)	6.015306e+01 <sup>3</sup> # (6.614788e-01)	6.141673e+01 <sup>1</sup> (9.511744e-03)
VIE2	3	1.309430e+03 <sup>1</sup> (3.082634e-03)	1.304450e+03 <sup>5</sup> # (4.795875e+00)	1.309306e+03 <sup>2</sup> # (2.438912e-02)	1.307448e+03 <sup>4</sup> # (1.934688e+00)	1.309305e+03 <sup>3</sup> # (2.176253e-02)
VIE3	3	1.420648e+03 <sup>1</sup> (1.081860e-02)	1.192055e+03 <sup>5</sup> # (1.719101e+02)	1.413235e+03 <sup>3</sup> # (2.216500e+00)	1.410666e+03 <sup>4</sup> # (3.997438e+00)	1.413311e+03 <sup>2</sup> # (1.811991e+00)



**Fig. 3.** Graphical performance comparison between (a) DeepEMO, (b) Rand-DeepEMO, (c) SMS-EMOA, (d)  $E_s$ -EMOA, and (e)  $R2$ -EMOA in  $DTLZ1^{-1}$ .

## 5 Conclusions

This paper proposed DeepEMO, the first Multi-Indicator-based EMOA that uses a CNN to detect the Pareto front geometry and choose the most appropriate indicator-based selection mechanism. Our proposal was compared with SMS-EMOA,  $R2$ -EMOA,  $E_s$ -EMOA, and a random version of DeepEMO. Our experimental results show that DeepEMO consistently obtains evenly distributed approximation sets, regardless of the Pareto front shape, with good convergence regarding multiple state-of-the-art QIs. These results prove that DeepEMO can compensate for the weaknesses of a single indicator-based selection method with the strengths of others. In other words, DeepEMO can tackle different MOPs without sacrificing convergence and diversity performance across different QIs. A current drawback of DeepEMO is that its CNN can only classify three-dimensional point clouds, making it unable to scale in objective space naturally. For future work, we plan to refine the rule-based hyper-heuristic method of DeepEMO to improve its performance in more MOPs. Furthermore, because of our current limitation to two- and three-objective MOPs, we are interested in expanding the capabilities of DeepEMO to four or more dimensional MOPs, *i.e.*, the so-called Many-objective Optimization Problems (MaOPs). We believe this will allow DeepEMO to outperform the  $E_s$ -EMOA, as the Riesz  $s$ -energy function loses selection pressure when tackling MaOPs.

**Acknowledgments.** The authors wish to acknowledge the financial support of the Writing Lab, Institute for the Future of Education, Tecnológico de Monterrey, Mexico, in the production of this work. This work was produced during the Research Internship of Tec Semester thanks to the educational innovation of Tecnológico de Monterrey.

## References

1. Basto-Fernandes, V., Yevseyeva, I., Deutz, A., Emmerich, M.: A Survey of Diversity Oriented Optimization: Problems, Indicators, and Algorithms. In: Emmerich, M., Deutz, A., Schütze, O., Legrand, P., Tantar, E., Tantar, A.-A. (eds.) *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation VII*, pp. 3–23. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-49325-1\\_1](https://doi.org/10.1007/978-3-319-49325-1_1)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (16 Sept 2007)
3. Borodachov, S.V., Hardin, D.P., Saff, E.B.: *Discrete Energy on Rectifiable Sets*. SMM, Springer, New York (2019). <https://doi.org/10.1007/978-0-387-84808-2>
4. Brockhoff, D., Wagner, T., Trautmann, H.: On the properties of the  $R2$  indicator. in: 2012 genetic and evolutionary computation conference (GECCO'2012). pp. 465–472. ACM Press, Philadelphia, USA (July 2012), iSBN: 978-1-4503-1177-9
5. Brockhoff, D., Wagner, T., Trautmann, H.:  $R2$  Indicator-based multiobjective search. *evolutionary computation* Vol. **23**(3), pp. 369–395 (Fall 2015)
6. Charles, R., Su, H., Kaichun, M., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85. IEEE Computer Society, Los Alamitos, CA, USA (July 2017). <https://doi.org/10.1109/CVPR.2017.16>, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.16>
7. Coello Coello, C.A., Cruz Cortés, N.: Solving multiobjective optimization problems using an artificial immune system. *Genet. Program Evolvable Mach.* **6**(2), 163–190 (2005)
8. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edn. (September 2007), iSBN 978-0-387-33254-3
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
10. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pp. 105–145. Springer, USA (2005)
11. Falcón-Cardona, J.G., Covantes Osuna, E., Coello Coello, C.A., Ishibuchi, H.: On the utilization of pair-potential energy functions in multi-objective optimization. *Swarm and Evolutionary Computation* **79**, 101308 (2023). <https://doi.org/10.1016/j.swevo.2023.101308>, <https://www.sciencedirect.com/science/article/pii/S2210650223000810>
12. Falcón-Cardona, J.G., Ishibuchi, H., Coello Coello, C.A., Emmerich, M.: On the Effect of the Cooperation of Indicator-Based Multiobjective Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **25**(4), 681–695 (2021). <https://doi.org/10.1109/TEVC.2021.3061545>

13. Hernández Gómez, R., Coello Coello, C.A.: Improved Metaheuristic Based on the  $R2$  Indicator for Many-Objective Optimization. In: 2015 Genetic and Evolutionary Computation Conference (GECCO 2015). pp. 679–686. ACM Press, Madrid, Spain (July 11–15 2015), ISBN 978-1-4503-3472-3
14. Huband, S., Hingston, P., Barone, L., While, L.: A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006)
15. Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Trans. Evol. Comput.* **21**(2), 169–190 (2017)
16. Jiang, S., Yang, S.: A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Trans. Evol. Comput.* **21**(3), 329–346 (2017). <https://doi.org/10.1109/TEVC.2016.2592479>
17. Li, B., Tang, K., Li, J., Yao, X.: Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Trans. Evol. Comput.* **20**(6), 924–938 (2016). <https://doi.org/10.1109/TEVC.2016.2549267>
18. Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **19**(5), 694–716 (2015). <https://doi.org/10.1109/TEVC.2014.2373386>
19. Li, M., Yang, S., Liu, X.: Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput.* **18**(3), 348–365 (2014). <https://doi.org/10.1109/TEVC.2013.2262178>
20. Li, M., Yang, S., Liu, X.: Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence* 228, 45–65 (2015). <https://doi.org/10.1016/j.artint.2015.06.007>, <https://www.sciencedirect.com/science/article/pii/S0004370215000995>
21. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys* 52(2), 26:1–26:38 (Mar 2019)
22. Li, M., Yao, X.: What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evolutionary Computation* **28**(2), 227–253 (Jun 2020). <https://doi.org/10.1162/evco.a.00269>
23. Liefoghe, A., Derbel, B.: A Correlation Analysis of Set Quality Indicator Values in Multiobjective Optimization. In: 2016 Genetic and Evolutionary Computation Conference (GECCO'2016). pp. 581–588. ACM Press, Denver, Colorado, USA (20–24 July 2016), ISBN 978-1-4503-4206-3
24. Liu, Q., Jin, Y., Heiderich, M., Rodemann, T., Yu, G.: An Adaptive Reference Vector-Guided Evolutionary Algorithm Using Growing Neural Gas for Many-Objective Optimization of Irregular Problems. *IEEE Transactions on Cybernetics* **52**(5), 2698–2711 (2022). <https://doi.org/10.1109/TCYB.2020.3020630>
25. Liu, Y., Gong, D., Sun, X., Zhang, Y.: Many-objective evolutionary optimization based on reference points. *Applied Soft Computing* 50, 344–355 (2017). <https://doi.org/10.1016/j.asoc.2016.11.009>, <https://www.sciencedirect.com/science/article/pii/S1568494616305786>
26. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 5105–5114. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
27. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **16**(4), 504–522 (2012)



28. Tian, Y., Cheng, R., Zhang, X., Cheng, F., Jin, Y.: An Indicator-Based Multiobjective Evolutionary Algorithm With Reference Point Adaptation for Better Versatility. *IEEE Trans. Evol. Comput.* **22**(4), 609–622 (2018). <https://doi.org/10.1109/TEVC.2017.2749619>
29. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Comput. Intell. Mag.* **12**(4), 73–87 (2017)
30. Tian, Y., Cheng, R., Zhang, X., Li, M., Jin, Y.: Diversity Assessment of Multi-Objective Evolutionary Algorithms: Performance Metric and Benchmark Problems. *IEEE Comput. Intell. Mag.* **14**(3), 61–74 (2019)
31. Veldhuizen, D.A.V.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA (May 1999)
32. Wang, H., Jiao, L., Yao, X.: Two\_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **19**(4), 524–541 (2015). <https://doi.org/10.1109/TEVC.2014.2350987>
33. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics* **38**(5), 1–12 (Oct 2019). <https://doi.org/10.1145/3326362>, <https://doi.org/10.1145/3326362>
34. Yuan, J., Liu, H.L., Gu, F., Zhang, Q., He, Z.: Investigating the Properties of Indicators and an Evolutionary Many-Objective Algorithm Using Promising Regions. *IEEE Trans. Evol. Comput.* **25**(1), 75–86 (2021). <https://doi.org/10.1109/TEVC.2020.2999100>
35. Yuan, Y., Xu, H., Wang, B., Yao, X.: A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **20**(1), 16–37 (2016). <https://doi.org/10.1109/TEVC.2015.2420112>
36. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)