

Reaching Pareto Front Shape Invariance with a Continuous Multi-Objective Ant Colony Optimization Algorithm ^{*}

Rodolfo Humberto Tamayo¹ and Jesús Guillermo Falcón-Cardona¹[0000-0003-1131-098X] and Carlos A. Coello Coello²[0000-0002-8435-680X]

- ¹ Tecnológico de Monterrey, School of Engineering and Sciences, Ave. Eugenio Garza Sada 2501, Monterrey, N.L., México, 64849
A01282633@tec.mx, jfalcon@tec.mx
- ² CINVESTAV-IPN, Departamento de Computación, Av. IPN No. 2508, Col. San Pedro Zacatenco, Ciudad de México, México, 07360
carlos.coellocoello@cinvestav.mx

Abstract. Generating Pareto Front Approximations with good convergence, uniformity, and spread regardless of the geometry of the Pareto Front remains as an open problem. Many Multi-Objective Evolutionary Algorithms (MOEAs) have been proposed for this aim achieving remarkable results. However, the utilization of Swarm Intelligence algorithms such as Multi-Objective Ant Colony Optimization Algorithms (MOACOs) has been scarcely studied. In this paper, we propose a Geometric-Invariant MOACO _{\mathbb{R}} (GI-MOACO _{\mathbb{R}}) designed to tackle multi-objective optimization problems with a continuous decision space. According to our experimental results, GI-MOACO _{\mathbb{R}} outperforms the existing MOACOs for continuous search spaces and it is competitive with respect to state-of-the-art MOEAs on several test suites with regular and irregular Pareto Front geometries. To the best of the author's knowledge, GI-MOACO _{\mathbb{R}} is the first Pareto-Front-Shape invariant MOACO.

Keywords: Multi-Objective Ant Colony Optimization, Pareto Front Shape Invariance, Continuous Decision Space, Pair-Potential Energy

1 Introduction

A Pareto Front (PF) is the image of the solution to a Multi-Objective Optimization Problem (MOP). A PF is a manifold of dimension at most $m - 1$ that represents the trade-offs among the $m \geq 2$ conflicting objectives $f_i : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ [22].

^{*} The first author acknowledges support given by Tecnológico de Monterrey and Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) to pursue graduate studies under the CVU number 1238924. Carlos A. Coello Coello is a member of the Faculty of Excellence at the School of Engineering and Sciences of Tecnológico de Monterrey as part of a sabbatical leave.

In benchmark problems and real-world applications, the geometric characteristics of a PF may drastically change. In other words, a PF could be convex, linear, concave, disconnected, degenerate, or mixed. Hence, Multi-Objective Evolutionary Algorithms (MOEAs), which are metaheuristic methods to approximate the solution to a MOP, should be capable of generating an accurate and finite representation of a PF, regardless of its geometry [2]. However, Ishibuchi *et al.* pointed out that the performance of some MOEAs depends on the PF geometry [16].

In recent years, two main design strategies have been proposed to alleviate the performance dependence of MOEAs. On the one hand, MOEAs with adaptive reference point sets transform a set of objective vectors throughout the evolutionary process, aiming to generate a Pareto Front Approximation (PFA) that properly represents the PF [19–21, 24]. On the other hand, Multi-Indicator MOEAs leverage the strengths of multiple Quality Indicators³ (QIs) to construct selection mechanisms that increase the selection pressure towards the PF, exhibiting a specific distribution of points [10, 11, 14, 17]. All these MOEAs have shown promising results when tackling MOPs with different PF shapes [15, 16, 26]. However, Swarm Intelligence metaheuristics, such as Multi-Objective Ant Colony Optimization algorithms (MOACOs) [12], have been scarcely studied with the aim of a PF shape invariance behavior.

MOACO_ℝ [13] and the Indicator-based MOACO_ℝ (iMOACO_ℝ) [7] were proposed to tackle MOPs with a continuous decision space. They have obtained competitive results against MOEAs at tackling MOPs with degenerate and disconnected PF shapes [7, 13]. Hence, there may be specific MOPs where MOACOs can potentially discover and exploit complex relations among the variables that are difficult to MOEAs [7, 13]. It is worth noting that both algorithms use the mechanisms of ACO_ℝ to explore the decision space and to create new solutions [23]. Furthermore, MOACO_ℝ and iMOACO_ℝ use an archive with k decision vectors as a pheromone matrix to model the decision space using n Gaussian-kernel Probability Density Functions (GKPDFs). Both MOACOs differ in the selection mechanisms adopted to update the pheromone matrix. On the one hand, MOACO_ℝ uses the selection mechanisms of NSGA-II, thus, it loses selection pressure in high-dimensional objective spaces [3]. On the other hand, iMOACO_ℝ replaces the selection operators of NSGA-II by a survival selection mechanism based on the $R2$ indicator [1, 14]. Unlike MOACO_ℝ, iMOACO_ℝ can solve MOPs with more than three objectives and it generates uniform PFAs if the geometry of the PF is correlated with an m -dimensional simplex. Thus, the performance of iMOACO_ℝ depends on the PF shape.

In addition to the issues mentioned above of MOACO_ℝ and iMOACO_ℝ, the use of ACO_ℝ raises other problems. ACO_ℝ shows a strong dependence on a parameter $\xi > 0$ that controls the evaporation of the pheromones and, thus, the convergence behavior. As a result, ACO_ℝ has shown an extremely poor performance on multi-modal single-objective optimization problems. However, if ξ could take large values without reducing its convergence capability, solutions

³ A unary Quality Indicator is a set function that assigns a real value to a PFA depending on its degree of proximity to the PF, spread, or uniformity [18].

with high diversity would be reached. Addressing these issues would enhance the performance of ACO_R and, possibly, that of MOACO_R and iMOACO_R.

In this paper, we enhance ACO_R to improve its exploration capabilities. Then, we propose the Geometric-Invariant MOACO_R that uses the enhanced ACO_R and a diversity-preserving mechanism based on pair-potential energy [6] to deal with MOPs regardless of their PF geometry. We tested GI-MOACO_R on several test suites with regular and irregular PF shapes on multiple QIs. Our experimental results indicate that GI-MOACO_R outperforms MOACO_R and iMOACO_R, and it is competitive with state-of-the-art MOEAs designed to tackle MOPs with regular and irregular PF geometries. Thus, to the authors' best knowledge, GI-MOACO_R is the first PF-shape-invariant MOACO, highlighting that this work shows that Swarm Intelligence-based algorithms can have this property.

The structure of this paper is organized as follows. Section 2 presents the background to make the paper self-contained. Section 3 outlines our proposed GI-MOACO_R. Section 4 describes our experimental settings and Section 5 discusses the results. Finally, Section 6 provides our main conclusions as well as some possible paths for future research.

2 Background

In this section, we first describe ACO_R which is the search engine of GI-MOACO_R. Then, we defined a MOP and some terms important in Multi-Objective Optimization. Finally, we briefly introduce Pair-Potential Energy.

2.1 ACO_R

ACO_R is an ACO for continuous search spaces proposed in 2008 [23]. A distinctive characteristic of ACO_R is the utilization of a pheromone matrix \mathcal{T} that stores the best k solutions sorted in ascending order by the objective function. Hence, each solution $\vec{x}_l \in \mathcal{T}$ has a rank l according to its position in the ordering. For each dimension $i = 1, \dots, n$ of Ω , where $\Omega \subseteq \mathbb{R}^n$ is the decision space⁴, ACO_R models the promising search region with a Gaussian-kernel Probability Density Function ($G^i(z)$), using the solutions in \mathcal{T} . Thus, the promising region is defined as $G^i(z) = \sum_{l=1}^k w_l g_l^i(z, \mu_l^i, \sigma_l^i)$, where $w_l \geq 0$ is a weight factor and $g_l^i(z, \mu_l^i, \sigma_l^i)$ is a Gaussian function with mean $\mu_l^i = x_{li}$ (the i^{th} decision variable of \vec{x}_l) and the standard deviation σ_l^i is defined as follows:

$$\sigma_l^i = \xi \sum_{r=1}^k \frac{|x_{ri} - x_{li}|}{k-1}, \quad (1)$$

where $\xi \geq 0$, known as evaporation rate, is a user-defined parameter that controls convergence. Small values of ξ increase the convergence behavior of ACO_R. To

⁴ $\Omega = [l_1, u_1] \times [l_2, u_2] \times \dots \times [l_n, u_n]$, where l_i and u_i define the lower and upper bounds, respectively, of the i^{th} decision variable.

generate a new candidate solution (\vec{x}_{new}), each $\text{ant}_j \in \mathcal{C} = \{\text{ant}_1, \dots, \text{ant}_M\}$ first selects a guiding pheromone $\vec{x}_l \in \mathcal{T}$ with probability p_l . Then, ant_j samples all $G_l^i(z), i = 1, \dots, n$ to construct a new candidate solution. The probability p_l of selecting \vec{x}_l is directly proportional to its weight w_l that is defined as follows:

$$w_l = \frac{1}{qk\sqrt{2\pi}} \cdot e^{-\frac{(\iota-1)^2}{2q^2k^2}}, \quad (2)$$

where $q \geq 0$ is a user-defined parameter that controls the diversification process of the search, where large values of q promote a higher degree of elitism.

2.2 Multi-Objective Optimization

In this paper, we solve unconstrained MOPs⁵ assuming, without loss of generality, the minimization of all the objectives:

$$\min_{\vec{x} \in \Omega} \{f(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})]^\top\}, \quad (3)$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^\top$ is an n -dimensional decision vector and $A = f(\Omega) \subseteq \mathbb{R}^m$ is the objective space. $f : \Omega \rightarrow A$ is a vector-valued function based on $m (\geq 2)$ objectives $f_i : \Omega \rightarrow \mathbb{R}, i = 1, 2, \dots, m$. In MOPs, the definition of optimality is commonly based on the Pareto dominance relation. Given $\vec{x}, \vec{y} \in \Omega$, we say that \vec{x} dominates \vec{y} (denoted as $\vec{x} \prec \vec{y}$) if $\forall i = 1, 2, \dots, m, f_i(\vec{x}) \leq f_i(\vec{y})$ and there exists at least an index $j \in \{1, 2, \dots, m\}$ such that $f_j(\vec{x}) < f_j(\vec{y})$. Then, $\vec{x}_* \in \Omega$ is Pareto optimal if there is no other $\vec{x} \in \Omega$ such that $\vec{x} \prec \vec{x}_*$. The solution to a MOP is the Pareto Set $\text{PS} = \{\vec{x}_* \in \Omega \mid \vec{x}_* \text{ is Pareto optimal}\}$ and its image $f(\text{PS})$ is the Pareto Front. A PFA is a set of solutions that approximate a PF where the solutions are mutually non-dominated, i.e., given $\vec{x}, \vec{y} \in \mathcal{A}$, $\vec{x} \not\prec \vec{y}$ and $\vec{y} \not\prec \vec{x}$.

2.3 Potential Energy

A Pair-Potential Energy function (PPF), denoted as \mathcal{K} , measures the interaction between particles at positions $\vec{u}, \vec{v} \in \mathbb{R}^m$. In physics, there is a wide range of PPFs but just a few of them have been utilized in EMOO [8]. Two representative PPFs are the Coulomb's law $\mathcal{K}^{\text{COU}}(\vec{u}, \vec{v}) = \frac{q_1 q_2}{4\pi\epsilon_0} \cdot \|\vec{u} - \vec{v}\|^{-2}$, where $\frac{1}{4\pi\epsilon_0}$ is the Coulomb's constant, being $\epsilon_0 \approx 8.8542 \times 10^{-12} [\text{F/m}]$ the vacuum permittivity, and the Riesz s -kernel, $\mathcal{K}^{\text{RSE}}(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|^{-s}$, where $s > 0$. In the context of EMOO, Falc3n-Cardona *et al.* [8] proposed to set $q_1 = \|\vec{u}\|$ and $q_2 = \|\vec{v}\|$. Given $\mathcal{K} : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$, the total potential energy (U) of an N -point set $\mathcal{A} = \{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_N\}$, with $N \geq 2$, where the lower the U of \mathcal{A} the higher the diversity of the set [8], is given by:

⁵ In the Evolutionary Multi-Objective Optimization (EMOO) community, the term Many-Objective Optimization Problem (MaOP) is used to denote problems having more than three objectives.

$$U^{\mathcal{K}}(\mathcal{A}) = \sum_{i=1}^N \sum_{\substack{j=i \\ j \neq i}}^N \mathcal{K}(\vec{a}_i, \vec{a}_j). \quad (4)$$

In recent years, U has been used in MOEAs to increase the diversity of PFAs, regardless of the PF geometry. MOEAs use selection mechanisms based on heuristic algorithms to approximate $\mathcal{A}^* = \arg \min_{\substack{\mathcal{A} \subset \mathcal{S} \\ |\mathcal{A}|=N}} U(\mathcal{A})$, where $N \ll |\mathcal{S}|$. These selection mechanisms have been used as pruning policies in external archives [9], density estimators [14], and for the generation of reference point sets [6]. Mainly, these selection mechanisms are based on a fast greedy removal algorithm that iteratively deletes from \mathcal{S} the point $\vec{a}_{\text{worst}} = \arg \max_{\vec{a} \in \mathcal{S}} \Delta(\mathcal{S}, \vec{a})$, where $\Delta(\mathcal{S}, \vec{a}) = U(\mathcal{S}) - U(\mathcal{S} \setminus \{\vec{a}\})$ is the so-called contribution to U [6].

3 Our Proposal: GI-MOACO_ℝ

This section describes our proposed GI-MOACO_ℝ, which aims to tackle regular and irregular MOPs. GI-MOACO_ℝ utilizes an enhanced version of ACO_ℝ that solves the dependency to ξ and the lack of diversification when selecting solutions from the pheromone matrix. GI-MOACO_ℝ also employs a PPF-based selection mechanism to promote diversity regardless of the PF geometry by performing a subset selection on \mathcal{T} . In the following sections, we describe three mechanisms (namely, **Classification**, **Construction**, and **Diversification**) to improve the performance of ACO_ℝ. Finally, we introduce our proposed GI-MOACO_ℝ.

Algorithm 1 Classification

Input: \mathcal{T} : Pheromone matrix; $\bar{\sigma}$: Deviation threshold
Output: CHOICES, STAGNATED.
1: CHOICES, STAGNATED $\leftarrow \emptyset, \emptyset$
2: **for** $i = 1$ to n **do**
3: $\sigma'_{x_i} \leftarrow \text{Standard-deviation}(\mathcal{T}, i)$
4: $\sigma'_{x_i} \leftarrow (\sigma'_{x_i} - l_i)/(u_i - l_i)$
5: **if** $\sigma'_{x_i} > \bar{\sigma}$ **then**
6: CHOICES \leftarrow CHOICES $\cup \{i\}$
7: **else**
8: STAGNATED \leftarrow STAGNATED $\cup \{i\}$
9: **end if**
10: **end for**
11: **return** CHOICES, STAGNATED

3.1 Detection of Stagnated Variables

In ACO_ℝ, each $\text{ant}_j \in \mathcal{C}$ selects a guiding pheromone $\vec{x}_l \in \mathcal{T}$ to generate a new candidate solution (\vec{x}_{new}) by sampling $g_l^i(z), i = 1 \dots, n$. The sampling is done without dispersion information of each dimension in the pheromone matrix. For some problems, especially multi-modal ones, pheromones in \mathcal{T} may converge to

a single solution. Consequently, $\sigma_l^i \approx 0$ for all $l = 1, \dots, k$ and $i = 1, \dots, n$ which causes a lack of exploration. To improve the exploration ability of $\text{ACO}_{\mathbb{R}}$, we propose a mechanism (see Algorithm 1) to classify the decision variables in \mathcal{T} into stagnated or not. This mechanism requires a threshold $\bar{\sigma} \in (0, 1)$ that indicates the minimum permissible deviation value. Line 1 initializes the sets **CHOICES** and **STAGNATED**. For each dimension i , we calculate the standard deviation (σ_{x_i}) of the i^{th} decision variable in \mathcal{T} and we normalize it using the lower and upper bounds, l_i and u_i , respectively, of this decision variable. In case that $\sigma'_{x_i} \geq \bar{\sigma}$, the dimension i is assigned to **CHOICES**, otherwise i is assigned to **STAGNATED**. Hence, **STAGNATED** will contain the stagnated decision variables and **CHOICES** the rest. The smaller the value of $\bar{\sigma}$, the stricter is the constraint to consider a variable as stagnated. According to our experimental observations, well-spread values have $0.2 \leq \sigma'_{x_i} \leq 0.4$. Hence, we recommend setting $\bar{\sigma} \leq 0.15$, as larger values might prematurely assign non-stagnated decision variables to set **STAGNATED**, compromising convergence.

3.2 Constructing New Candidate Solutions

To improve the exploration and exploitation ability of our approach, we propose here a new method to construct a candidate solution $\vec{x}_{\text{new}} \in \Omega$. To this aim, ant_j needs to provide as input a guiding pheromone \vec{x}_l . In contrast to the original $\text{ACO}_{\mathbb{R}}$ where an ant samples each $g_l^i(z, \mu_l^i, \sigma_l^i)$, in our proposal only \bar{n} decision variables from **CHOICES** are selected to construct \vec{x}_{new} . Algorithm 2 sets the value of \bar{n} by iteratively calculating $\bar{n} = \min(\bar{n}, \sim \mathcal{U}_{\mathbb{N}}[1, |\text{CHOICES}|])$, where $\sim \mathcal{U}_{\mathbb{N}}[1, |\text{CHOICES}|]$ generates a random natural number in the range $[1, |\text{CHOICES}|]$. Then, in line 5, \bar{n} random and distinct variables from **CHOICES** are assigned to \mathcal{I} . To construct $\vec{x}_{\text{new}} \in \Omega$, the Gaussian kernels associated with each $i \in \mathcal{I}$ are sampled. The remaining variables⁶ are copied directly from the guiding pheromone \vec{x}_l . It is worth emphasizing that modifying fewer decision variables allows the use of larger values of ξ . As a result, it maintains convergence while increasing diversity. In consequence, this reduces the dependency of $\text{ACO}_{\mathbb{R}}$ to ξ .

Algorithm 2 Construction

Input: \vec{x}_l : Guiding pheromone; ξ : Evaporation rate; q : Diversification rate; **CHOICES**: Set of non-stagnated variables

Output: \vec{x}_{new} .

```

1:  $\bar{n} \leftarrow n$ 
2: for objective  $j = 1$  to  $m$  do
3:    $\bar{n} \leftarrow \min(\bar{n}, \sim \mathcal{U}_{\mathbb{N}}[1, |\text{CHOICES}|])$ 
4: end for
5:  $\mathcal{I} \leftarrow$  Randomly select  $\bar{n}$  distinct variables from CHOICES
6: Calculate  $\sigma_l^i$  using  $\xi$  and  $q$ 
7:  $x_{\text{new},i} \leftarrow$  Sample  $g_l^i(z, \mu_l^i, \sigma_l^i), \forall i \in \mathcal{I}$ 
8:  $x_{\text{new},i} \leftarrow x_{li}, \forall i \in \{1, 2, \dots, n\} \setminus \mathcal{I}$ 
9: return  $\vec{x}_{\text{new}}$ 

```

⁶ Note that each $\text{ant}_j \in \mathcal{C}$ might modify a different number \bar{n} of decision variables.

Algorithm 3 Diversification

Input: \mathcal{K} : PPF; \mathcal{M} : External archive; \mathcal{T} : Pheromone matrix; α : Percentage of pheromone replacement; **STAGNATED**: Set of stagnated decision variables
Output: \mathcal{T} : Updated pheromone matrix

- 1: $k \leftarrow |\mathcal{T}|$
- 2: **if** $|\mathcal{M}| > k$ **then**
- 3: $\mathcal{T} \leftarrow U^{\mathcal{K}}\text{-Based Selection Mechanism}(\mathcal{M}, \mathcal{K}, k)$
- 4: **else**
- 5: $\mathcal{N} \leftarrow$ Randomly select $k - |\mathcal{M}|$ guiding pheromones $\vec{x} \in \mathcal{M}$
- 6: $\mathcal{T} \leftarrow \mathcal{M} \cup \mathcal{N}$
- 7: **end if**
- 8: $i \leftarrow$ Select a random decision variable from **STAGNATED**
- 9: Randomly permute the pheromones of \mathcal{T}
- 10: **for** $j = 1$ to $\lceil k \times \alpha \rceil$ **do**
- 11: $\mathcal{T}_{j_i} \leftarrow \sim \mathcal{U}_{\mathbb{R}}[l_i, u_i]$
- 12: **end for**
- 13: **return** \mathcal{T}

3.3 Avoiding Stagnation of Decision Variables

The original ACO_ℝ suffers from stagnation because the best solutions are stored in the pheromone matrix. When the solutions in \mathcal{T} are similar, $\sigma_l^i \approx 0$ (see Eq. (1)) for all $l = 1, \dots, k$ and $i = 1, \dots, n$. In consequence, an ant cannot explore more regions of Ω . Algorithm 3 avoids stagnation by increasing the diversity of solutions in \mathcal{T} , using an external archive \mathcal{M} . First, if \mathcal{M} has more than $k = |\mathcal{T}|$ solutions, the content of \mathcal{T} is replaced by a subset of k solutions from \mathcal{M} using a PPF-based subset selection as in [6]. Otherwise, $k - |\mathcal{M}|$ solutions are randomly selected from \mathcal{M} to be inserted in \mathcal{T} . Lines 8 and 9 randomly permute the pheromones of \mathcal{T} and randomly select a decision variable i from **STAGNATED**. Finally, $\lceil k \times \alpha \rceil$ solutions $\in \mathcal{T}$ are perturbed by assigning a random real number $\sim \mathcal{U}_{\mathbb{R}}[l_i, u_i]$ to the i^{th} component of the solutions. Thus, this injection of randomness aims to increase the exploration ability and diversification of ACO_ℝ. As a result, the i^{th} component of \mathcal{T} will be added to the set **CHOICES** by Algorithm 1.

3.4 GI-MOACO_ℝ

Algorithm 4 outlines the main loop of GI-MOACO_ℝ, which uses a bounded external archive \mathcal{M} of maximum size 3μ to store non-dominated solutions found during the search process. The size of the set \mathcal{M} is limited because larger sizes require more computational and time resources to update, with a complexity of $\mathcal{O}(|\mathcal{M}|^2)$ [3]. Additionally, previous experimentation has shown that performance is not compromised once the size reaches 3μ . In case that \mathcal{M} has more than 3μ solutions, we apply a $U^{\mathcal{K}}$ -based subset selection using the algorithm provided in [6]. Lines 7 to 24 sketch the main loop of GI-MOACO_ℝ. First, Algorithm 1 classifies the decision variables into **CHOICES** and **STAGNATED**. Then, each ant $j \in \mathcal{C}, j = 1, \dots, k$ generates a new candidate solution using Algorithm 2, adding it to \mathcal{T} . In line 14, \mathcal{M} and \mathcal{T} are joined and, then, \mathcal{T} is sorted with the Non-Dominated Sorting Algorithm [3] and pruned if necessary. In line 17, we ensure that \mathcal{M} has at most 3μ solutions. Since applying Algorithm 3 at each iteration

Algorithm 4 GI-MOACO_ℝ

Input: ξ_0 : Initial evaporation rate; q : Diversification rate; μ : Approximation size; $\bar{\sigma}$: Deviation threshold; α : Percentage of pheromone replacement; T_w : Replacement time window; \mathcal{K} : PPF

Output: \mathcal{M} : Pareto front approximation

- 1: $k \leftarrow \lceil \mu/4 \rceil$
- 2: Randomly initialize pheromone matrix \mathcal{T} of size k
- 3: Initialize external archive \mathcal{M} with non-dominated solutions from \mathcal{T}
- 4: $\mathcal{M}_{max} \leftarrow 3\mu$
- 5: $\mathcal{T} \leftarrow U^{\mathcal{K}}$ -Based Selection Mechanism ($\mathcal{T}, \mathcal{K}, k$)
- 6: $g, \xi \leftarrow 0, \xi_0$
- 7: **while** termination condition is not fulfilled **do**
- 8: CHOICES, STAGNATED \leftarrow Classification ($\mathcal{T}, \bar{\sigma}$)
- 9: **for** $j = 1$ to k **do**
- 10: ant_j selects a guiding pheromone $\bar{x}_i \in \mathcal{T}$
- 11: $\bar{x}_{\text{new}} \leftarrow$ Construction ($\bar{x}_i, \xi, q, \text{CHOICES}$)
- 12: $\mathcal{T} \leftarrow \mathcal{T} \cup \{\bar{x}_i\}$
- 13: **end for**
- 14: $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{T}$
- 15: $\mathcal{T} \leftarrow$ Non-Dominated Sorting(\mathcal{T})
- 16: $\mathcal{T} \leftarrow U^{\mathcal{K}}$ -Based Selection Mechanism ($\mathcal{T}, \mathcal{K}, k$)
- 17: Prune \mathcal{M} if $|\mathcal{M}| > \mathcal{M}_{max}$
- 18: **if** $g \bmod T_w == 0$ and $|\text{STAGNATED}| > 0$ **then**
- 19: $\mathcal{T} \leftarrow$ Diversification ($\mathcal{K}, \mathcal{M}, \mathcal{T}, \alpha, \text{STAGNATED}$)
- 20: **end if**
- 21: **if** 80% of process is completed **then**
- 22: $\xi \leftarrow \xi_0/2$
- 23: **end if**
- 24: $g \leftarrow g + 1$
- 25: **end while**
- 26: $\mathcal{M} \leftarrow U^{\mathcal{K}}$ -Based Selection Mechanism ($\mathcal{M}, \mathcal{K}, \mu$)
- 27: **return** \mathcal{M}

may produce a lack of convergence, we execute it every T_w iterations only if $|\text{STAGNATED}| > 0$. Finally, ξ is updated if 80% of the search process has been completed to encourage an exploitation behavior. GI-MOACO_ℝ returns a subset of μ solutions from \mathcal{M} using the $U^{\mathcal{K}}$ -based subset selection.

4 Experimental Settings

This section is devoted to test the performance of GI-MOACO_ℝ⁷. We compared our proposal against MOACO_ℝ [13] and iMOACO_ℝ⁸ [7], and five state-of-the-art MOEAs⁹ (designed to tackle both regular and irregular PF shapes): AdaW [19], AR-MOEA [24], SPEA2+SDE [17], RVEA-iGNG [20], and Two_Arch2 [28]. We performed 30 independent executions of each algorithm per test instance.

4.1 Benchmark Problems

We adopted the test suites Deb-Thiele-Laumanns-Zitzler (DTLZ) [4], Walking-Fish-Group (WFG) [15], their inverted versions DTLZ⁻¹ and WFG⁻¹ [16], re-

⁷ The source code of GI-MOACO_ℝ is available at <https://github.com/Humberto-Tamayo/GI-MOACOR.git>.

⁸ MOACO_ℝ and iMOACO_ℝ were coded in Python 3.10.12. Their source code is available at <https://github.com/Humberto-Tamayo/MOACOR-iMOACOR.git>.

⁹ We used the implementations from the PlatEMO platform [25].

Table 1. Characteristics of the PF geometries of the selected MOPs and the reference point for the HV calculation.

MOP	PF geometry	Simplex-like	Reference point (HV)
DTLZ1	Triangular (linear)	Yes	(1, . . . , 1)
DTLZ2 - DTLZ4	Concave	Yes	(2, . . . , 2)
DTLZ5 & DTLZ6	Concave ($m = 2$)	Yes	(2, . . . , 2)
	Degenerate ($m = 3$)	No	
	Unknown ($m > 3$)	No	
DTLZ7	Disconnected	No	(1, . . . , 1, 21)
DTLZ1 ⁻¹	Inverted triangular	No	(1, . . . , 1)
DTLZ2 ⁻¹ - DTLZ6 ⁻¹	Convex	No	(1, . . . , 1)
DTLZ7 ⁻¹	Disconnected	No	(0.1, . . . , 0.1, -10)
WFG1	Mixed	Yes	(3, 5, 7, . . . , 2m + 1)
WFG2	Disconnected	Yes	(3, 5, 7, . . . , 2m + 1)
WFG3	Linear ($m = 2$)	Yes	(3, 5, 7, . . . , 2m + 1)
	Degenerate ($m \geq 3$)	No	
WFG4 - WFG9	Concave	Yes	(3, 5, 7, . . . , 2m + 1)
WFG1 ⁻¹	Mixed	No	(1, . . . , 1)
WFG2 ⁻¹ , WFG4 ⁻¹ - WFG9 ⁻¹	Convex	No	(1, . . . , 1)
WFG3 ⁻¹	Inverted triangular	No	(1, . . . , 1)
VIE1	Convex	No	(4, 5, 4)
VIE2	Mixed	No	(5, -15, -11)
VIE3	Mixed	No	(10, 18, 1)
IMOP1	Convex	Yes	(1.2, 1.2)
IMOP2	Concave	Yes	(1.2, 1.2)
IMOP3	Disconnected	No	(1.5, 1.2)
IMOP4	Degenerate	No	(1.2, 1.2, 1.2)
IMOP5	Disconnected	No	(1, 1, 2)
IMOP6	Degenerate	No	(1.2, 1.2, 1.2)
IMOP7	Concave	No	(1.2, 1.2, 1.2)
IMOP8	Degenerate	No	(1.2, 1.2, 3.2)

spectively, with 2, 3, 5, and 7 objectives. Additionally, we employed the Irregular MOPs (IMOP) [26] and the Viennet problems (VIE) [27]. We selected these test suites because they cover a wide range of PF geometries and search difficulties.

Table 1 presents the main characteristics of the PF geometries of the selected MOPs, indicating if the geometry correlates with the simplex shape. For DTLZ and DTLZ⁻¹ problems, the number of variables was set to $n = m + K - 1$, where m denotes the number of objectives, $K = 5$ for DTLZ1, $K = 10$ for DTLZ2-DTLZ6, and $K = 20$ for DTLZ7. The inverted versions share the same values of K . Regarding WFG problems, we set the tuples $(n, m, k_{\text{position}})$, where k_{position} is the number of position-related parameters, to $(24, 2, 4)$, $(26, 3, 4)$, $(30, 5, 8)$, and $(34, 7, 12)$. For the IMOP test suite, the parameters K, L, a_1, a_2 , and a_3 are set to 5, 5, 0.05, 0.05, and 10 as indicated by its authors [26].

4.2 Parameter Settings

For a fair comparison, all the algorithms use the same population size (μ) and they consume the same number of function evaluations (FE_{max}). Hence, we set $(\mu, m, \text{FE}_{\text{max}})$ to $(120, 2, 40 \times 10^3)$, $(120, 3, 50 \times 10^3)$, $(126, 5, 70 \times 10^3)$, and

(210, 7, 90×10^3). These values are selected as they are standard values utilized in the EMOO literature [5]. Regarding GI-MOACO_ℝ, the parameters ξ_0 , q , $\bar{\sigma}$, α , and T_w , are set to 1.8, 0.5, 0.1, 0.5, and 4, respectively. The selection of values is based on previous experimentation. Additionally, \mathcal{K} is set to \mathcal{K}^{COU} for all the processes to ensure convergence and diversity as suggested by Falcón-Cardona et al. [6], except on Algorithm 3, where is set to \mathcal{K}^{RSE} to take advantage of its theoretical properties. In MOACO_ℝ and iMOACO_ℝ the parameters q and ξ are set to 0.1 and 0.5, respectively, as suggested by their authors [7,13]. In MOACO_ℝ, the number of ants is set to 2. For all the selected MOEAs, the parameter values are set to their default values as in the PlatEMO platform [25]. The crossover and mutation probabilities are set to 1.0 and $1/n$, respectively, while both crossover and mutation distribution indexes are set to 20.

4.3 Quality Indicators

We adopted the Hypervolume indicator (HV) and the Inverted Generational Distance plus (IGD⁺) to assess convergence towards the PF [18]. To measure the diversity of the PFAs, we used the Riesz s -energy $E_s = U^{\mathcal{K}^{\text{RSE}}}$ (see Eq. 4), where $s > 0$ is a user-supplied parameter. Table 1 lists the reference points used to calculate HV for each test problem. The calculation of IGD⁺ requires a reference point set, constructed by merging all the PFAs generated by all algorithms for a given test instance and filtering out the non-dominated solutions. Then, we performed an E_s -based subset selection to obtain $100 \times m$ solutions with $s = m + 1$ [6]. For E_s assessment, we set $s = m + 1$ to evaluate the normalized PFAs.

5 Discussion of Results

The main goal of this Section is to present the numerical comparison of GI-MOACO_ℝ based on HV, IGD⁺, and E_s . However, we first show how Algorithms 2 and 3 help to enhance the performance of the original ACO_ℝ. Then, we discuss here the comparison of GI-MOACO_ℝ with MOACO_ℝ and iMOACO_ℝ. Then, we focus on comparing GI-MOACO_ℝ with the selected MOEAs. Due to space constraints, the complete numerical results for HV are provided in Tables SM-1 to SM-6 of the Supplementary Material (SM) available at <https://github.com/Humberto-Tamayo/GI-MOACOR.git>. Tables SM-7 to SM-12 show the results for IGD⁺, and Tables SM-13 to SM-18 show the results for E_s . Table 2 presents a summary of HV-based results in Tables SM-1 to SM-6 from the SM. Additionally, Fig. 2 shows PFAs according to the first, second, third, and last rank in the comparison from Tables SM-1 to SM-6.

5.1 Discussion of the Enhanced ACO_ℝ

To show the effect of Algorithms 1 and 3 in ACO_ℝ, we measured the performance of ACO_ℝ when turning on and off these algorithms. Moreover, we used

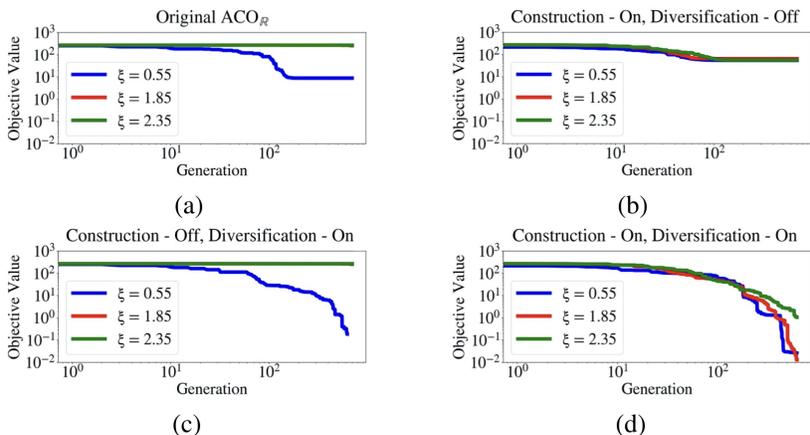


Fig. 1. Effect of Construction and Diversification algorithms in ACO_R on the Rastrigin Function. (a) Original ACO_R, (b) Construction on and Diversification off, (c) Construction off and Diversification on, (d) Construction On and Diversification off.

$\xi = 0.55, 1.85, 2.35$ to identify its effect. We tested the algorithm with the multimodal Rastrigin Function with 20 decision variables with 35 thousand function evaluations. Fig. 1(a) shows the performance of the original ACO_R when Algorithms 2 and 3 are turned off¹⁰. This illustrates that ACO_R fails to converge with high values of ξ and gets stuck with $\xi = 0.55$. The effect of Algorithm 2 is shown in Figure 1(b) for $\xi = 0.55, 1.85, 2.25$. In this case, all the algorithms have a similar behavior, showing a clear reduction in the dependency on ξ . On the other hand, the effect of Algorithm 3 is illustrated in Figure 1(c) for $\xi = 0.55$, obtaining better results than in Fig. 1(a). Finally, Figure 1(d) shows the effectiveness of combining Algorithms 2 and 3, outperforming the original ACO_R. The enhancements of ACO_R increase convergence and reduce the dependency to ξ , thus improving diversity which is beneficial for MaOPs.

5.2 Comparison against MOACOs

Table 2 shows that GI-MOACO_R outperforms MOACO_R and iMOACO_R at tackling multi-frontal MOPs such as DTLZ1 and WFG4. Additionally, the results associated with DTLZ4 and WFG8 show that GI-MOACO_R performs competitively against iMOACO_R in regular MOPs and MaOPs. In this regard, Tables SM-3, SM-5, SM-9, and SM-11 indicate that GI-MOACO_R outperforms iMOACO_R in most regular MaOPs from the DTLZ and WFG test suites. According to HV and IGD⁺, GI-MOACO_R obtained better approximations in 17 out of 26 regular MaOPs with 5 and 7 objectives.

Fig. 3 shows a set of heat maps comparing GI-MOACO_R with the selected MOACOs and MOEAs. The heat maps show the number of times an algo-

¹⁰ Note that for Algorithms 2 and 3 to work, Algorithm 1 must be turned on.

Table 2. Mean and standard deviation (in parentheses) of HV values. The two best values are highlighted in gray scale, where the darker tone corresponds to the best one. The rank in the comparison for each value is shown in parentheses. The symbol # is placed where the best-ranked algorithm performs statistically better according to a one-tailed Wilcoxon rank-sum test using a significance level of 0.05.

MOP	m	GI-MOACO _z	MOACO _z	iMOACO _z	AdaW	AR-MOEA	SPEA2+SDE	Two_Arch2	RVEA-iGNNG
VIE1	3	2.316e+01(4#) (1.995e-02)	2.306e+01(5#) (3.042e-02)	2.254e+01(8#) (6.831e-02)	2.320e+01(3) (1.655e-02)	2.289e+01(6) (7.504e-02)	2.289e+01(7#) (1.425e-01)	2.323e+01(1) (1.355e-02)	2.322e+01(2) (1.989e-02)
VIE3	3	3.161e+01(1) (1.509e-03)	3.152e+01(7#) (1.809e-02)	3.149e+01(8#) (3.691e-02)	3.160e+01(5) (1.624e-03)	3.161e+01(2) (1.366e-03)	3.160e+01(3) (7.785e-03)	3.160e+01(4) (5.184e-03)	3.159e+01(6) (4.069e-03)
IMOP6	3	9.959e-01(6#) (9.440e-03)	9.786e-01(8#) (1.035e-02)	9.842e-01(7#) (3.743e-03)	1.053e+00(1) (1.460e-03)	1.014e+00(4) (1.049e-01)	1.011e+00(5#) (1.067e-01)	1.051e+00(2#) (7.691e-04)	1.021e+00(3#) (1.277e-01)
IMOP7	3	9.662e-01(2#) (3.304e-02)	9.607e-01(4#) (9.753e-02)	9.623e-01(3#) (1.887e-02)	1.017e+00(1) (1.456e-01)	4.016e-01(7) (2.769e-01)	4.819e-01(6#) (3.449e-01)	3.798e-01(8#) (2.370e-01)	8.907e-01(5#) (2.059e-01)
DTLZ1	2	8.629e-01(6#) (8.770e-03)	0.000e+00(7#) (0.000e+00)	0.000e+00(8#) (0.000e+00)	8.736e-01(3) (3.530e-04)	8.736e-01(2) (4.751e-04)	8.733e-01(5#) (5.185e-04)	8.734e-01(4) (9.117e-04)	8.736e-01(1) (2.328e-04)
	3	9.663e-01(6#) (5.766e-03)	0.000e+00(7#) (0.000e+00)	0.000e+00(8#) (0.000e+00)	9.742e-01(2#) (1.546e-04)	9.743e-01(1) (1.254e-04)	9.700e-01(5#) (1.818e-03)	9.739e-01(4#) (2.098e-04)	9.741e-01(3#) (9.345e-05)
	5	9.946e-01(5#) (1.098e-02)	-	0.000e+00(7#) (0.000e+00)	9.987e-01(2#) (7.072e-05)	9.987e-01(1) (1.657e-05)	9.941e-01(6#) (1.837e-03)	9.984e-01(4#) (5.122e-05)	9.985e-01(3#) (8.291e-05)
	7	9.558e-01(6#) (8.0076e-02)	-	0.000e+00(7#) (0.000e+00)	9.999e-01(2#) (2.530e-05)	9.999e-01(1) (1.266e-06)	9.983e-01(5#) (5.641e-04)	9.998e-01(4#) (9.268e-06)	9.999e-01(3#) (3.129e-05)
DTLZ4	2	3.211e+00(2) (2.201e-04)	3.207e+00(4) (4.573e-03)	3.211e+00(1) (2.653e-05)	3.209e+00(3) (4.143e-03)	2.848e+00(7) (5.644e-01)	3.090e+00(5) (3.694e-01)	2.888e+00(6) (5.449e-01)	2.767e+00(8) (1.839e-01)
	3	7.408e+00(2) (2.165e-03)	7.370e+00(5) (1.471e-02)	7.420e+00(1) (3.813e-04)	7.382e+00(3) (1.953e-01)	6.960e+00(8) (9.104e-01)	7.288e+00(7) (3.457e-01)	7.307e+00(6) (6.246e-01)	7.375e+00(4) (1.839e-01)
	5	3.163e+01(4#) (2.249e-02)	-	3.164e+01(3#) (3.938e-03)	3.165e+01(1) (1.195e-02)	3.161e+01(5) (1.663e-01)	3.158e+01(6#) (2.018e-01)	3.152e+01(7#) (6.123e-02)	3.165e+01(2#) (1.147e-02)
	7	1.277e+02(6#) (1.134e-02)	-	1.277e+02(5#) (4.751e-03)	1.277e+02(4#) (1.006e-02)	1.278e+02(1) (4.495e-04)	1.278e+02(2#) (2.997e-03)	1.268e+02(7#) (4.048e-01)	1.278e+02(3#) (5.284e-03)
DTLZ7	2	1.773e+01(1) (9.318e-06)	1.770e+01(5#) (7.212e-02)	1.772e+01(2#) (3.993e-04)	1.771e+01(4) (2.985e-02)	1.766e+01(8) (1.414e-01)	1.771e+01(3) (1.479e-02)	1.768e+01(7) (1.203e-01)	1.769e+01(6) (1.060e-01)
	3	1.637e+01(1) (1.761e-03)	1.624e+01(7#) (3.756e-02)	1.624e+01(8#) (7.939e-03)	1.637e+01(2#) (1.179e-02)	1.629e+01(6) (1.282e-01)	1.631e+01(5) (7.380e-02)	1.633e+01(4) (1.020e-01)	1.636e+01(3) (5.546e-02)
	5	1.290e+01(3#) (2.086e-02)	-	1.177e+01(7#) (8.988e-02)	1.299e+01(2) (3.210e-02)	1.279e+01(4) (2.556e-02)	1.279e+01(5#) (2.630e-01)	1.274e+01(6#) (9.724e-02)	1.305e+01(1) (3.290e-02)
	7	9.180e+00(2#) (2.972e-02)	-	7.934e+00(7#) (1.534e-01)	9.055e+00(3) (1.926e-01)	8.847e+00(5) (4.012e-02)	8.947e+00(4#) (3.131e-01)	8.771e+00(6#) (4.031e-01)	9.423e+00(1) (7.067e-02)
DTLZ5 ⁻¹	2	1.758e+01(3#) (3.682e-02)	1.755e+01(6#) (4.488e-01)	1.754e+01(8#) (5.635e+01(8#))	1.758e+01(4) (4.175e-04)	1.757e+01(5) (5.013e-05)	1.755e+01(7#) (1.259e-02)	1.758e+01(1) (3.801e-04)	1.758e+01(2) (5.557e-04)
	3	5.888e+01(1) (4.033e+02(3#)) (7.404e-01)	5.637e+01(7#) (4.488e-01)	5.635e+01(8#) (4.177e+00)	5.886e+01(2) (1.278e+00)	5.816e+01(5) (4.687e+00)	5.703e+01(6) (5.860e+00)	5.869e+01(3) (2.210e+00)	5.859e+01(4) (2.224e+00)
	5	4.033e+02(3#) (7.404e-01)	-	2.929e+02(7#) (4.177e+00)	4.135e+02(1) (1.278e+00)	3.889e+02(5) (4.687e+00)	3.826e+02(6#) (5.860e+00)	4.040e+02(2#) (2.210e+00)	4.020e+02(4#) (2.224e+00)
	7	1.788e+03(5#) (6.238e+00)	-	1.191e+03(7#) (2.625e+01)	2.031e+03(1) (1.112e+01)	1.783e+03(6) (2.315e+01)	1.978e+03(3#) (2.773e+01)	1.970e+03(4#) (2.638e+01)	1.981e+03(2#) (1.663e+01)
WFG4	2	8.524e+00(6#) (2.342e-02)	8.071e+00(8#) (6.935e-02)	8.114e+00(7#) (2.326e-02)	8.571e+00(5) (3.198e-02)	8.583e+00(4) (3.405e-02)	8.638e+00(2#) (1.965e-02)	8.649e+00(1) (1.148e-02)	8.614e+00(3) (2.573e-02)
	3	7.533e+01(4#) (2.368e-01)	6.595e+01(8#) (7.651e-01)	6.942e+01(7#) (2.834e-01)	7.508e+01(6) (2.495e-01)	7.517e+01(5) (2.184e-01)	7.612e+01(1) (1.738e-01)	7.589e+01(3) (1.925e-01)	7.598e+01(2) (1.647e-01)
	5	8.915e+03(1) (3.170e+01)	-	7.829e+03(7#) (1.599e+02)	8.240e+03(6) (6.735e+01)	8.536e+03(4) (4.927e+01)	8.656e+03(2#) (5.303e+01)	8.288e+03(5) (5.584e+01)	8.593e+03(3) (3.763e+01)
	7	1.838e+06(1) (9.627e+03)	-	1.588e+06(6#) (2.822e+04)	1.496e+06(7) (3.291e+04)	1.704e+06(4) (1.347e+04)	1.739e+06(2) (1.287e+04)	1.592e+06(5) (1.822e+04)	1.726e+06(3) (1.149e+04)
WFG3	2	7.486e+00(5#) (4.485e-02)	7.452e+00(8#) (1.650e-01)	7.480e+00(7#) (5.244e-01)	7.652e+00(2) (3.484e-02)	7.540e+00(4) (4.556e-02)	7.624e+00(3#) (2.706e-02)	7.664e+00(1) (3.290e-02)	7.483e+00(6) (6.448e-02)
	3	6.763e+01(6#) (6.951e-01)	5.848e+01(8#) (1.092e+00)	6.471e+01(7#) (5.244e-01)	6.957e+01(4) (4.102e-01)	6.944e+01(5) (4.053e-01)	7.040e+01(1) (1.555e-01)	6.993e+01(3) (1.895e-01)	7.010e+01(2) (2.292e-01)
	5	7.640e+03(3#) (1.331e-02)	-	5.268e+03(7#) (3.124e+02)	7.239e+03(5#) (9.265e+01)	7.824e+03(1) (4.568e+01)	7.766e+03(2#) (5.206e+01)	7.172e+03(6#) (5.764e+01)	7.558e+03(4#) (7.764e+01)
	7	1.532e+06(3#) (2.850e+04)	-	9.383e+05(7#) (5.933e+04)	1.148e+06(6#) (4.279e+04)	1.577e+06(1) (1.491e+04)	1.560e+06(2#) (1.153e+04)	1.231e+06(5#) (2.321e+04)	1.441e+06(4#) (1.701e+04)
WFG7 ⁻¹	2	2.226e+01(1) (7.493e-04)	2.218e+01(8#) (1.163e-01)	2.223e+01(6#) (2.277e-02)	2.225e+01(3) (5.147e-03)	2.224e+01(4) (1.979e-03)	2.220e+01(7) (1.877e-02)	2.225e+01(2) (7.343e-03)	2.224e+01(5) (1.325e-02)
	3	1.441e+02(2#) (3.066e-01)	1.364e+02(8#) (1.170e+00)	1.384e+02(7#) (1.204e+00)	1.441e+02(3) (2.361e-01)	1.430e+02(5) (2.325e-01)	1.409e+02(6#) (9.857e-01)	1.444e+02(1) (1.135e-01)	1.440e+02(4) (2.331e-01)
	5	5.917e+03(6#) (4.950e+01)	-	3.748e+03(7#) (6.122e+01)	6.005e+03(5) (7.140e+01)	6.055e+03(3) (5.859e+01)	6.039e+03(4#) (2.122e+02)	6.357e+03(2#) (3.209e+01)	6.359e+03(1) (5.074e+01)
	7	1.946e+05(6#) (2.881e+03)	-	1.512e+05(7#) (3.547e+03)	2.196e+05(5) (5.651e+03)	2.488e+05(4) (7.363e+03)	2.847e+05(3#) (5.079e+03)	2.888e+05(2#) (3.445e+03)	2.958e+05(1) (4.001e+03)

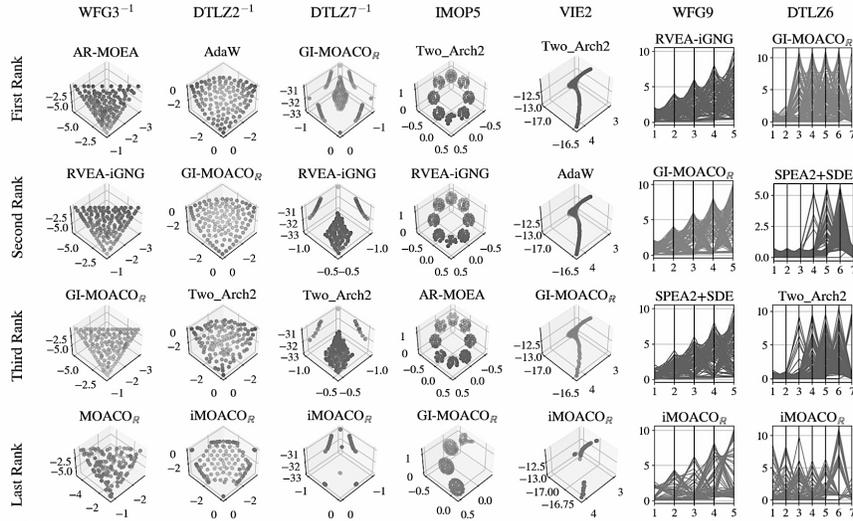


Fig. 2. PFAs generated by GI-MOACO_R and the selected MOACOs and MOEAs.

rithm is ranked first or second in the comparison based on HV, IGD⁺, and E_s , based on Tables SM-1 to SM-18. On the one hand, Fig. 3(a) shows that the PFAs generated by GI-MOACO_R are the best compared to MOACO_R and iMOACO_R according to HV, IGD⁺, and E_s in most of the adopted MOPs and MaOPs. GI-MOACO_R obtains the first rank in 80.89% of the MOPs and 89.58% of the MaOPs¹¹. Additionally, Fig. 2 shows several irregular MOPs such as WFG3⁻¹ DTLZ2⁻¹, DTLZ7⁻¹, and VIE2, where GI-MOACO_R outperformed MOACO_R and iMOACO_R, according to HV. For DTLZ2⁻¹, DTLZ7⁻¹, and VIE2, iMOACO_R shows difficulties to converge. These results indicate that GI-MOACO_R outperforms MOACO_R and iMOACO_R.

5.3 Comparison against state-of-the-art MOEAs

The goals of comparing GI-MOACO_R with state-of-the-art MOEAs are threefold: 1) determining if GI-MOACO_R is competitive against MOEAs designed to tackle regular and irregular PF geometries, 2) identifying if GI-MOACO_R outperforms these MOEAs in MOPs with specific properties, and 3) pointing out improvement areas for GI-MOACO_R.

Table 2 and Fig. 2 show that GI-MOACO_R is competitive with the selected MOEAs in multiple irregular MOPs such as DTLZ2⁻¹, DTLZ7⁻¹, VIE3, WFG7⁻¹ for 4 objectives, DTLZ6 for 7 objectives, and DTLZ7 regardless of the number of objectives, according to HV. Additionally, Tables SM-5 and SM-11 indicate that GI-MOACO_R obtains competitive results according to HV and IGD⁺

¹¹ Note that MOACO_R does not take part of the comparison of MaOPs since it does not properly scale.

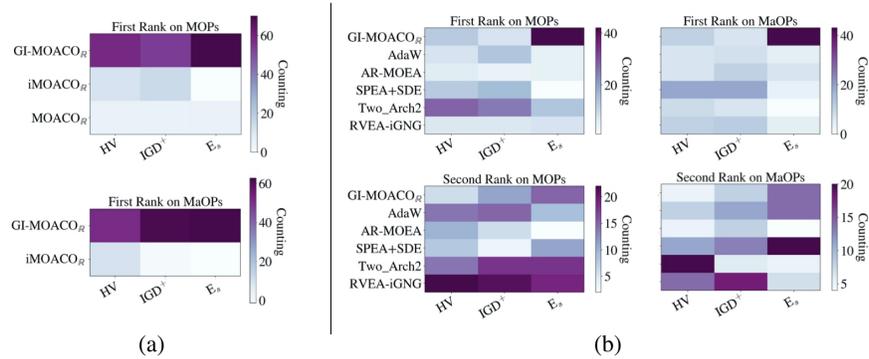


Fig. 3. (a) Heat map that shows the number of times a MOACO was ranked first according to HV, IGD⁺, and E_s. (b) Heat map that shows the number of times GI-MOACO_R or a MOEA was ranked first or second according to the three selected QIs.

in WFG2, WFG6, WFG7, and WFG9 for 5 and 7 objectives. These MaOPs are non-separable and simplex-like. These results show that GI-MOACO_R does not have a preference for specific PF geometries. However, it has a good performance on disconnected, non-separable, and deceptive problems. We believe that this behavior is due to the probabilistic mechanisms of ACO_R. In contrast to MOEAs where crossover might not easily generate offspring that jump across disconnected regions, or in deceptive landscapes, ACO_R is less constrained by the need for solutions to be adjacent or directly related in the search space.

Fig. 3(b) compares GI-MOACO_R with selected MOEAs. GI-MOACO_R shows superior PFA diversity (in terms of E_s) due to its use of PPFs. Two_Arch2 provides the best approximations for most MOPs, and SPEA+SDE excels in most MaOPs according to HV and IGD⁺. Despite some challenges in covering the entire PF in the IMOP test suite, GI-MOACO_R achieves top ranks in many MOPs and MaOPs, demonstrating its competitiveness. The issue likely stems from the sensitivity of ACO_R to the pheromone matrix composition, suggesting room for improving its probabilistic sampling and pheromone updating mechanisms.

6 Conclusions and Future Work

In this paper, we proposed GI-MOACO_R which is a Pareto-Front-Shape Invariant MOACO for solving continuous MOPs. GI-MOACO_R uses an enhanced ACO_R to improve its search capabilities and it has a diversity-preserving mechanism based on PPFs. According to our experimental results using several benchmark problems with regular and irregular Pareto Front geometries, it outperforms both MOACO_R and iMOACO_R and it is competitive concerning state-of-the-art MOEAs. It also showed significant performance in disconnected, non-separable, and deceptive MOPs. For future work, we want to develop a hybrid model and test PDFs to model more difficult and large-scale search spaces.

References

1. 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
2. Coello Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag, Berlin, Heidelberg (2006)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (April 2002)
4. 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 London (2005). https://doi.org/10.1007/1-84628-137-7_6
5. Falcón-Cardona, J.G.: New Findings on Indicator-based Multi-Objective Evolutionary Algorithms. Ph.D. thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Mexico, Mexico City (11 2020)
6. Falcón-Cardona, J., Covantes Osuna, E., Coello, C., Ishibuchi, H.: On the utilization of pair-potential energy functions in multi-objective optimization. *Swarm and Evolutionary Computation* **79**, 101308 (April 2023). <https://doi.org/10.1016/j.swevo.2023.101308>
7. Falcón-Cardona, J.G., Coello Coello, C.A.: A new indicator-based many-objective ant colony optimization optimizer for continuous search spaces. *Swarm Intelligence* **11**, 71–100 (2017). <https://doi.org/https://doi.org/10.1007/s11721-017-0133-x>
8. Falcón-Cardona, J.G., Covantes Osuna, E., Coello Coello, C.A.: An Overview of Pair-Potential Functions for Multi-objective Optimization. In: Ishibuchi, H., Zhang, Q., Cheng, R., Li, K., Li, H., Wang, H., Zhou, A. (eds.) *Evolutionary Multi-Criterion Optimization*. pp. 401–412. Springer International Publishing, Cham (2021)
9. Falcón-Cardona, J.G., Emmerich, M.T., Coello Coello, C.A.: On the Cooperation of Multiple Indicator-based Multi-Objective Evolutionary Algorithms. In: 2019 IEEE Congress on Evolutionary Computation (CEC). pp. 2050–2057 (2019). <https://doi.org/10.1109/CEC.2019.8790315>
10. Falcón-Cardona, J.G., Ishibuchi, H., Coello, C.A.C.: Exploiting the Trade-off between Convergence and Diversity Indicators. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 141–148 (2020). <https://doi.org/10.1109/SSCI47803.2020.9308469>
11. Falcón-Cardona, J.G., Ishibuchi, H., Coello Coello, C.A., Emmerich, M.: On the Effect of the Cooperation of Indicator-Based Multi-Objective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* **25**(4), 681–695 (2021). <https://doi.org/10.1109/TEVC.2021.3061545>
12. Falcón-Cardona, J.G., Leguizamón, G., Coello Coello, C.A., Castillo Tapia, M.G.: Multi-objective Ant Colony Optimization: An Updated Review of Approaches and Applications, pp. 1–32. Springer Nature Singapore, Singapore (2022). https://doi.org/10.1007/978-981-16-8930-7_1
13. García Nájera, A., Bullinaria, J.: Extending ACO_R to Solve Multi-Objective Problems. In: *Proceedings of the 2007 UK Workshop on Computation Intelligence (UKCI)* (2007)

14. Hernández Gómez, R., Coello Coello, C.A.: A Hyper-Heuristic of Scalarizing Functions. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 577–584. GECCO'17, Association for Computing Machinery (2017). <https://doi.org/10.1145/3071178.3071220>
15. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* **10**(5), 477–506 (2006). <https://doi.org/10.1109/TEVC.2005.861417>
16. Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation* **21**(2), 169–190 (April 2017)
17. Li, M., Yang, S., Liu, X.: Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **18**(3), 348–365 (2014). <https://doi.org/10.1109/TEVC.2013.2262178>
18. 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)
19. 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 (2020). <https://doi.org/10.1162/evco.a.00269>
20. 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>
21. Márquez-Vega, L.A., Falcón-Cardona, J.G., Covantes Osuna, E.: On the adaptation of reference sets using niching and pair-potential energy functions for multiobjective optimization. *Swarm and Evolutionary Computation* **83**, 101408 (2023). <https://doi.org/https://doi.org/10.1016/j.swevo.2023.101408>
22. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Springer New York, NY (1998). <https://doi.org/10.1007/978-1-4615-5563-6>
23. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. *European Journal of Operational Research* **185**(3), 1155–1173 (2008). <https://doi.org/https://doi.org/10.1016/j.ejor.2006.06.046>
24. Tian, Y., Cheng, R., Zhang, X., Cheng, F., Jin, Y.: An Indicator-Based Multiobjective Evolutionary Algorithm With Reference Point Adaptation for Better Versatility. *IEEE Transactions on Evolutionary Computation* **22**(4), 609–622 (2018). <https://doi.org/10.1109/TEVC.2017.2749619>
25. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Computational Intelligence Magazine* **12**(4), 73–87 (2017). <https://doi.org/10.1109/MCI.2017.2742868>
26. Tian, Y., Cheng, R., Zhang, X., Li, M., Jin, Y.: Diversity Assessment of Multi-Objective Evolutionary Algorithms: Performance Metric and Benchmark Problems [Research Frontier]. *IEEE Computational Intelligence Magazine* **14**(3), 61–74 (2019). <https://doi.org/10.1109/MCI.2019.2919398>
27. 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)

28. Wang, H., Jiao, L., Yao, X.: Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **19**(4), 524–541 (2015). <https://doi.org/10.1109/TEVC.2014.2350987>