Swarm Intelligence manuscript No.
(will be inserted by the editor)

A New Indicator-Based Many-Objective Ant Colony Optimizer for Continuous Search Spaces

Jesús Guillermo Falcón-Cardona · Carlos A. Coello Coello

Received: date / Accepted: date

Abstract In this paper, we propose a novel multi-objective ant colony optimizer (called iMOACO$_{\mathbb{R}}$) for continuous search spaces, which is based on ACO$_{\mathbb{R}}$ and the $R2$ performance indicator. iMOACO$_{\mathbb{R}}$ is the first multi-objective ant colony optimizer (MOACO) specifically designed to tackle continuous many-objective optimization problems (i.e., multi-objective optimization problems having four or more objectives). Our proposed iMOACO$_{\mathbb{R}}$ is compared to three state-of-the-art multi-objective evolutionary algorithms (NSGA-III, MOEA/D and SMS-EMOA) and a MOACO algorithm called MOACO$_{\mathbb{R}}$ using standard test problems and performance indicators taken from the specialized literature. Our experimental results indicate that iMOACO$_{\mathbb{R}}$ is very competitive with respect to NSGA-III and MOEA/D and it is able to outperform SMS-EMOA and MOACO$_{\mathbb{R}}$ in most of the test problems adopted.

Keywords Ant Colony Optimization · Many-Objective Optimization · Continuous Optimization · Performance Assessment

1 Introduction

Many of the existing applications in engineering, science and industry require the solution of problems involving several, normally conflicting, objective functions, which must be simultaneously optimized. In such problems, due to the conflict

Jesús Guillermo Falcón-Cardona
CINVESTAV-IPN (Evolutionary Computation Group)
Av. IPN No. 2508, Col. San Pedro Zacatenco, México City 07300, MEXICO
Tel.: +52 55 5747 3800 ext. 6564
Fax : +52 55 5747 3757
E-mail: jfalcon@computacion.cs.cinvestav.mx

Carlos A. Coello Coello
CINVESTAV-IPN (Evolutionary Computation Group)
Av. IPN No. 2508, Col. San Pedro Zacatenco, México City 07300, MEXICO
Tel.: +52 55 5747 3800 ext. 6564
Fax : +52 55 5747 3757
E-mail: ccoello@cs.cinvestav.mx

among the objectives, an objective function cannot be improved without worsening another one. These are the so-called multi-objective optimization problems (MOPs). Unlike single-objective optimization problems in which we look for a single solution, when solving a MOP, we aim to find a set of solutions which represents the best possible trade-offs among all objectives. Such solutions constitute the so-called "Pareto Optimal Set" and the image of this set is known as the "Pareto Optimal Front".

A number of mathematical programming techniques (Miettinen, 1999) have been developed to solve MOPs. However, and in spite of their efficiency, mathematical programming techniques have several limitations (e.g., most of them generate a single solution per run and many of them are very sensitive to the shape and continuity of the Pareto front). These limitations have motivated the development of alternative solution techniques, from which metaheuristics (particularly evolutionary algorithms) have become the most popular choice (Coello Coello et al., 2007).

Multi-Objective Evolutionary Algorithms (MOEAs) are population-based metaheuristics, inspired by the natural evolution of organisms, which have been found to be a very good choice for solving highly complex MOPs (Coello Coello et al., 2007). Currently, a wide variety of MOEAs are available in the specialized literature, such as the Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2000) or the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) (Zhang and Li, 2007), among many others.

For many years, MOEAs have adopted selection mechanisms based on Pareto ranking, i.e. solutions are ranked based on Pareto optimality, such that nondominated solutions are given a higher probability of being selected. However, in the last few years, it has been shown that Pareto-based MOEAs do not perform properly when dealing with MOPs having four or more objectives, the so-called many-objective optimization problems (Ishibuchi et al., 2008; Knowles and Corne, 2007). Many-objective problems give rise to a number of challenging issues such as: (1) deterioration of the search ability (dilution of the selection pressure) due to the increase in the number of nondominated solutions, (2) exponential increase in the number of solutions required for properly sampling the entire Pareto front, and (3) difficulty of the visualization of solutions because they belong to high-dimensional spaces. Overall, this has motivated an important amount of research related to the development of new MOEAs.

Ant Colony Optimization (ACO) is inspired by colonies of real ants that deposit a chemical substance, called pheromone, on the ground with the aim of tracing paths to a source of food. Initially, ACO was designed for solving combinatorial optimization problems. Over the years, ACO has been extended to solve continuous optimization problems. However, according to Leguizamón and Coello Coello (2011) the extension of ACO-based algorithms for solving continuous MOPs has been scarcely explored and the development of multi-objective ant colony optimizers (MOACOs) for continuous domains has remained practically unexplored unlike MOACOs for combinatorial MOPs which have been extensively studied (Angus and Woodward, 2009; Lopez-Ibanez and Stützle, 2012a,b). In this paper, we present an extended version of the preliminary work reported in Falcón-Cardona and Coello (2016), in which the so-called indicator-based Multi-Objective Ant Colony Optimization Algorithm for Continuous Search Spaces (iMOACO$_{\mathbb{R}}$) was introduced. To the best of the authors' knowledge, this is the only MOACO algorithm that has been explicitly designed to solve many-objective continuous optimization problems and which has a competitive performance with respect to state-of-the-art MOEAs.

The remainder of this paper is organized as follows. Section 2 provides some basic concepts related to multi-objective optimization and ant colony optimization. Section 3 presents a general overview of ACO$_{\mathbb{R}}$ (an ant colony optimizer for continuous search spaces). Section 4 describes in detail our proposed approach. Section 5 outlines the experimental setup adopted to validate our algorithm, and Section 6 presents our results. Finally, Section 7 concludes and provides some possible paths for future research in the area.

2 Background

In this section, we first introduce some mathematical concepts about multi-objective optimization. Then, we briefly review the principles of the ACO metaheuristic. Finally, we provide a short introduction to some MOACO algorithms that have been proposed for continuous domains.

2.1 Multi-Objective Optimization

We are interested in solving problems of the type[1]:

$$\text{minimize } \mathbf{F}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})] \tag{1}$$

subject to:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, k \tag{2}$$

$$h_j(\mathbf{x}) = 0 \quad j = 1, 2, \ldots, p \tag{3}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, ..., m$ are the objective functions and $g_i, h_j : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, ..., k$, $j = 1, ..., p$ are the constraint functions of the problem. These are the so-called multi-objective optimization problems (MOPs).

To describe the concept of optimality in which we are interested, we will introduce next a few definitions.

Definition 1. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we say that $\mathbf{u} \leq \mathbf{v}$ if $u_i \leq v_i$ for $i = 1, \ldots, n$.

Definition 2. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we say that **u dominates v** (denoted by $\mathbf{u} \prec \mathbf{v}$) if $u_i \leq v_i$ for $i = 1, \ldots, n$ and there exists at least an index j such that $u_j < v_j$.

This definition indicates that the vector **u** dominates **v** if the former is as good as the latter in every element and it is better in at least one of them. Evidently,

[1] Without loss of generality, we will assume only minimization problems.

this same definition can be also used with sets.

Definition 3. We say that a vector of decision variables $\mathbf{x}^* \in \mathcal{F}$ ($\mathcal{F}$ is the feasible region) is **Pareto optimum** (or **Pareto optimal**) if there does not exist another $\mathbf{x} \in \mathcal{F}$ such that $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x^*})$.

Thus, a solution is Pareto optimal if it cannot be dominated (see Definition 2) by any other feasible solution.

Definition 4. The **Pareto Optimal Set** $\mathcal{P}^*$ is defined by:

$$\mathcal{P}^* = \{\mathbf{x}^* \in \mathcal{F} \mid \mathbf{x} \text{ is Pareto optimum or Pareto optimal}\}$$

The elements of the Pareto Optimal Set represent the best possible trade-offs among the objectives. Such solutions cannot be improved in one objective without being worsened in another one.
The vectors $\mathbf{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *nondominated*.

Definition 5. The **Pareto Front** $\mathcal{PF}^*$ is defined by:

$$\mathcal{PF}^* = \{\mathbf{F}(\mathbf{x}^*) \in \mathbb{R}^m \mid \mathbf{x}^* \in \mathcal{P}^*\}$$

The Pareto Front is the image of the Pareto Optimal Set. In other words, the Pareto front denotes the objective function values of the solutions contained in the Pareto Optimal Set.

We thus wish to determine the Pareto optimal set from the set $\mathcal{F}$ of all decision variable vectors that satisfy (2) and (3).

Two special vectors bound the Pareto optimal front of a MOP:

Definition 6. The *Ideal Objective Vector* ($\mathbf{z}^* \in \mathbb{R}^m$) contains the minimum of each of the objective functions, considered separately. Each i^{th}-component of the ideal vector is defined as $z_i^* = \min_{\mathbf{x}} f_i(\mathbf{x})$.

Definition 7. The *Nadir Objective Vector* ($\mathbf{z}^{nad} \in \mathbb{R}^m$) is constructed using the worst values of $\mathcal{PF}^*$. Each i^{th}-component is defined as $z_i^{nad} = \max_{\mathbf{x} \in \mathcal{P}^*} f_i(\mathbf{x})$.

To illustrate these two concepts, let's imagine a problem with two objectives. In this case, the Pareto front is a curve. If this curve is placed inside a rectangular box, the lower lefthand corner of the box is the ideal objective vector, and upper righthand corner of the box is the nadir objective vector.

Definition 8. A (unary) performance indicator is a function $\mathcal{I} : \mathcal{Z} \subset \mathbb{R}^m \mapsto \mathbb{R}$ that assigns each approximation set $\mathcal{Z}$ a real number.

When defining a performance indicator, we want to be able to compare the outcome of two or more multi-objective optimization algorithms using a single numerical value. Thus, a performance indicator takes the outcome of a multi-objective optimizer (i.e., an approximation of the Pareto front) and produces a single real number that indicates how "good" this approximation is. The term *unary* is associated to performance indicators that take as their input to the outcome of a single multi-objective optimizer. There are also *binary* performance indicators that take as their input the outcome of two multi-objective optimizers, but such indicators won't be discussed in this paper.

Definition 9. An indicator $\mathcal{I}$ is said to be strictly monotonic if and only if whenever a Pareto set approximation entirely dominates another one, then the indicator value of the dominant set will also be better. Considering maximization of $\mathcal{I}$, this can be formally expressed as:

$$\forall A, B \in \mathcal{Z} \;:\; A \prec B \Rightarrow \mathcal{I}(A) > \mathcal{I}(B).$$

Definition 10. An indicator $\mathcal{I}$ is said to be weakly monotonic if and only if for any Pareto set approximation that is compared to another Pareto set approximation, it holds that being at least as good in terms of the dominance relation implies being at least as good in terms of the indicator values. Considering maximization of $\mathcal{I}$, this can be formally expressed as:

$$\forall A, B \in \mathcal{Z} \;:\; A \preceq B \Rightarrow \mathcal{I}(A) \geq \mathcal{I}(B).$$

These two definitions are important, because they will tell us if the outcome of a certain unary performance indicator is reliable or not. A unary performance indicator is expected to be at least weakly monotonic in order to be reliable.

2.2 Ant Colony Optimization

Ant Colony Optimization (ACO), which was originally proposed by Dorigo (1992), is a metaheuristic inspired by the foraging behavior of real ants (Bonabeau et al., 1999; Dorigo and Stützle, 2004). ACO-based algorithms are stochastic-search procedures. In every ACO algorithm, the pheromone model is the central component (Dorigo and Blum, 2005). It is a probabilistic representation of the search space that is exploited by the ants to create new solutions. Algorithm 1 outlines the general framework of an ACO-based algorithm. First, the pheromone model is randomly initialized. At each cycle t, M ants incrementally build solutions for the problem $\mathcal{P}$ via the pheromone model. Then, some of the newly created solutions are chosen with the aim of updating the pheromone model. Optionally, centralized operations, called Daemon actions, that cannot be individually executed by ants (e.g., local-search methods or the recollection of global heuristic information), are performed before a new cycle begins. These operations are repeated while a termination condition is not fulfilled.

Primarily, the utilization of ACO-based algorithms has been oriented to the solution of combinatorial optimization problems, being the traveling salesman problem (TSP) the most representative of them (Applegate et al., 2007). The Ant System (AS), proposed by Dorigo et al. (1996), was the first ACO-based algorithm used to solve the TSP. Due to its promising results, different AS variants were later proposed, from which the most relevant are: Ant Colony System (ACS) (Dorigo and Gambardella, 1997) and $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) (Stützle and Hoos, 2000). Also, in recent years, the ACO metaheuristic has been extended in order to tackle continuous optimization problems. Bilchev and Parmee (1995) proposed the first ant-related algorithm for continuous optimization, however, it does not follow the pheromone model. Leguizamón and Coello Coello (2011) mention that there are several algorithms based on the behavior of real ants for continuous optimization (Dréo and Siarry, 2004; Chen et al., 2003; Kong and Tian, 2006), however, one of the most representative approaches using a pheromone model is the Ant Colony Optimizer for continuous domains ($\text{ACO}_{\mathbb{R}}$) presented by Socha and Dorigo (2008) . A detailed description of $\text{ACO}_{\mathbb{R}}$ will be provided in Section 3.

Algorithm 1 General ACO metaheuristic

```
1: t ← 0
2: InitializePheromones()
3: while termination condition not satisfied do
4:     AntBasedSolutionConstruction()
5:     PheromoneUpdate()
6:     DaemonActions() {Optional}
7:     t ← t + 1
8: end while
```

2.3 MOACO algorithms for continuous domains

The use of MOACOs in continuous optimization has been scarcely explored (Angus and Woodward, 2009; Leguizamón and Coello Coello, 2011). We are only aware of two approaches of this sort: the Population-based ACO Algorithm for Multi-Objective Function Optimization (PACO-MOFO) (Angus, 2007b) and the Multi-Objective Ant Colony Optimizer ($\text{MOACO}_{\mathbb{R}}$) proposed by Garcia-Najera and Bullinaria (2007).

PACO-MOFO is based on $\text{ACO}_{\mathbb{R}}$ and the Crowding Population-based ACO (CPACO) (Angus, 2007a). The pheromone model is similar to that of $\text{ACO}_{\mathbb{R}}$. PACO-MOFO applies a replacement operator based on the crowding distance in order to maintain diversity in the population and fitness sharing in furtherance of a uniform sampling of the objective space.

Garcia-Najera and Bullinaria (2007) proposed an extension of $\text{ACO}_{\mathbb{R}}$ with the aim of solving multi-objective optimization problems. Since the solution of a MOP is a set of mutually nondominated points instead of a global optimum, the fundamental question underlying $\text{MOACO}_{\mathbb{R}}$ is to determine which solutions should be stored in the pheromone archive. In $\text{ACO}_{\mathbb{R}}$, after each iteration, the best newly created solutions, according to their objective function values, are kept in the archive. However, the authors decided to use the concept of *dominance depth* (Deb et al., 2000) in order to preserve at each iteration those solutions closer to the Pareto Front. Moreover, as the pheromone archive has a constant size, if the number of solutions exceeds such size, a density mechanism based on crowding distance is activated. This mechanism calculates the crowding distance per solution, and those with a higher value are removed to keep the archive size below some maximal

s_1	s_1^1	s_1^2	$\cdots$	s_1^j	$\cdots$	s_1^n	$f(s_1)$	w_1
s_2	s_2^1	s_2^2	$\cdots$	s_2^j	$\cdots$	s_2^n	$f(s_2)$	w_2
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
s_k	s_k^1	s_k^2	$\cdots$	s_k^j	$\cdots$	s_k^n	$f(s_k)$	w_k
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
s_N	s_N^1	s_N^2	$\cdots$	s_N^j	$\cdots$	s_N^n	$f(s_N)$	w_N
	G^1	G^2		G^j		G^n		

Fig. 1 Pheromone archive kept by ACO$_{\mathbb{R}}$. The solutions are sorted according to their quality. Each weight value w_k is proportional to the solution's quality. G^j represents a Gaussian-kernel PDF that is constructed using only the j^{th} coordinates of all N solutions from the archive (Socha and Dorigo, 2008).

size. MOACO$_{\mathbb{R}}$ employs the same ACO$_{\mathbb{R}}$ mechanisms in order to generate new solutions.

3 ACO$_{\mathbb{R}}$

ACO$_{\mathbb{R}}$ implements a direct extension of the ACO metaheuristic without any conceptual change in its general framework (Socha and Dorigo, 2008). The fundamental idea underlying ACO$_{\mathbb{R}}$ is the use of a pheromone model based on *probability density functions* (PDFs) instead of *discrete probability distributions*. In ACO$_{\mathbb{R}}$, the pheromone model is represented using an archive $\mathcal{T}$ that stores the N best solutions that have been obtained so far. For each solution s_k to an n-dimensional problem, ACO$_{\mathbb{R}}$ stores in $\mathcal{T}$ the values of its n decision variables (the j^{th} variable of the k^{th} solution is denoted by s_k^j), the objective function value $f(s_k)$ and a weight value w_k, as it is shown in Figure 1. The solutions are sorted by their quality, i.e., for a minimization problem: $f(s_1) \leq f(s_2) \leq \cdots \leq f(s_N)$.

Based on the solutions stored in $\mathcal{T}$, n Gaussian-kernel PDFs, one per dimension of the problem, are dynamically generated. Socha and Dorigo (2008) decided to use Gaussian-kernel PDFs in order to be able to describe disjoint areas of the search space that are promising for each dimension of the problem. The Gaussian-kernel PDF G^j, for the j^{th} dimension of the problem, is defined as follows:

$$G^j(x) = \sum_{k=1}^{N} w_k g_k^j(x) = \sum_{k=1}^{N} w_k \frac{1}{\sigma_k^j \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_k^j)^2}{2*{\sigma_k^j}^2}} \tag{4}$$

where $j = 1, \ldots, n$ and N is the total number of solutions stored in $\mathcal{T}$. Each G^j depends on three parameter vectors: $\mathbf{w}$ is the vector of weights associated with the individual Gaussian functions, $\boldsymbol{\mu}^j$ is the vector of means, and $\boldsymbol{\sigma}^j$ is the vector of standard deviations. Only the j^{th} decision variables of all solutions are used to calculate the values of $\boldsymbol{\mu}^j$ and $\boldsymbol{\sigma}^j$.

Each $w_k \in \mathbf{w}$ is a value of a Gaussian function according to the following formula:

$$w_k = \frac{1}{qN\sqrt{2\pi}} \cdot e^{-\frac{(\text{rank}(s_k)-1)^2}{2q^2N^2}}, \tag{5}$$

where rank$(\cdot)$ returns the solution's rank in $\mathcal{T}$ and q (> 0) is a parameter that controls the diversification process of the search. In fact, as q tends to zero, the best ranked solutions are preferred; otherwise, the weight values tend to be uniformly distributed.

The vectors $\boldsymbol{\mu}^j$ and $\boldsymbol{\sigma}^j$ are calculated as follows. First, the elements of $\boldsymbol{\mu}^j$ correspond to the j^{th} decision variables of all solutions:

$$\boldsymbol{\mu}^j = \{\mu_1^j, \ldots, \mu_N^j\} = \{s_1^j, \ldots, s_N^j\}. \tag{6}$$

Then, each $\sigma_k^j \in \boldsymbol{\sigma}^j$ is defined by:

$$\sigma_k^j = \xi \sum_{r=1}^{N} \frac{|s_r^j - s_k^j|}{N-1}, \tag{7}$$

where $\xi > 0$ is a parameter of the algorithm that controls the convergence rate by simulating the pheromone evaporation. The higher the value of ξ, the lower the convergence rate.

ACO$_{\mathbb{R}}$ is based on the framework of Algorithm 1. At the beginning, N random solutions are generated using a uniform distribution in order to initialize $\mathcal{T}$. At each iteration, the weight vector $\mathbf{w}$ is calculated using eq. (5) and then every ant $i = 1, \ldots, M$ performs n construction steps in order to generate a new solution. Before performing the n construction steps, each ant i randomly chooses an index $r \in \{1, \ldots, N\}$ with probability $p_r = w_r / \sum_{l=1}^{N} w_l$ in order to use it through the sampling process. At construction step j, ant i samples the r^{th} Gaussian-kernel $g_r^j(x) = g_r^j(x; \mu_r^j, \sigma_r^j)$ of $G^j(x)$ using, for example, the Box-Müller method (Box and Müller, 1958), in order to assign a value to the j^{th} decision variable. It is worth mentioning that $G^j(x)$, Eq. (4), is not completely computed, that is, all elements of the vectors $\boldsymbol{\mu}^j$ and $\boldsymbol{\sigma}^j$ are not calculated; instead, only μ_r^j and σ_r^j need to be known in order to characterize the Gaussian-kernel being sampled. Once all ants have created their solutions, the pheromone update process is executed. From the set of newly created solutions, we select those that improve the solutions stored in $\mathcal{T}$ with the aim of replacing them. It must be emphasized that the number of solutions stored remains constant. At the end of the search process, the best-ranked solution in $\mathcal{T}$ corresponds to the algorithm's outcome.

Optionally, ACO$_{\mathbb{R}}$ is able to exploit correlation information between different decision variables. At each step of the construction process, each ant chooses a direction $\mathbf{s_u s_r}$ where s_u is randomly chosen and it is reasonably far away from the solution s_r chosen earlier as the mean of the Gaussian-Kernel. Based on all these directions, a new orthogonal basis for the ant's coordinate system is computed using the Gram-Schmidt process (Anton, 2010). Then, all current coordinates of all solutions in the archive are rotated and recalculated according to this new orthogonal basis.

4 Our proposed approach

In this section, we describe our proposed approach called *indicator-based Many-Objective Ant Colony Optimizer for Continuous Search Spaces* (iMOACO$_{\mathbb{R}}$) that was originally introduced in (Falcón-Cardona and Coello, 2016). First, it is discussed why we chose ACO$_{\mathbb{R}}$ (Socha and Dorigo, 2008) as our search engine. Next, we outline its selection scheme based on the $R2$ indicator, which allows iMOACO$_{\mathbb{R}}$ to solve many-objective problems. Then, we describe a statistical mechanism whose purpose is the approximation of the ideal and nadir vectors. Finally, we introduce iMOACO$_{\mathbb{R}}$, including a brief analysis of its computational complexity.

4.1 Search engine: ACO$_{\mathbb{R}}$

iMOACO$_{\mathbb{R}}$ is a many-objective optimizer based on the ACO metaheuristic. This decision is based on the remarkable success of ACO in combinatorial single- and multi-objective optimization problems (Angus and Woodward, 2009; Lopez-Ibanez and Stützle, 2012a,b) and on the promising results shown in continuous single-objective optimization problems (Leguizamón and Coello Coello, 2011). A remarkable version of ACO for continuous problems is ACO$_{\mathbb{R}}$. According to Socha and Dorigo (2008), ACO$_{\mathbb{R}}$ has shown favorable results compared to genetic algorithms, probability-learning methods and other ant-related approaches. It has also been used as the search engine of the only two MOACO algorithms for continuous domains that we are aware of. This is the main motivation to propose a multi-objective optimizer using ACO$_{\mathbb{R}}$ as its search engine.

In order to use ACO$_{\mathbb{R}}$ as iMOACO$_{\mathbb{R}}$' seach engine, we only have to do a simple modification. Initially, ACO$_{\mathbb{R}}$ sorts the solutions in the pheromone archive $\mathcal{T}$ according to their objective function value, obtaining in the first rank the best-so-far solution. In case of MOPs, there is no straightforward way to determine the quality of the objective function and to impose a total order due to the existence of several nondominated solutions. In consequence, we propose to use an indicator-based ranking algorithm (described in the next section) with the aim of establishing different quality levels among solutions produced by ants, in a similar way to the nondominated levels generated by the nondominated sorting procedure proposed by Deb et al. (2000). In this case, L_r is the r^{th} level where $r = 1, \ldots, \mathcal{K}$ and $\mathcal{K}$ is the total number of levels. Having the solutions ranked, they are stored in $\mathcal{T}$ together with their assigned rank and objective vector $\mathbf{F}$. Now, those solutions in L_1 (the ones closer to the true Pareto front) are the set of best-so-far vectors. Once $\mathcal{T}$ contains the solutions, the mechanisms to generate new solutions, as defined in ACO$_{\mathbb{R}}$, can be applied.

4.2 $R2$-ranking algorithm

The hypervolume (HV) (Zitzler, 1999; Brockhoff et al., 2008) and the $R2$ indicator (Brockhoff et al., 2012) are two recommended unary performance indicators which simultaneously evaluate all desired aspects of a Pareto Front approximation (Coello Coello et al., 2007). However, the $R2$ indicator requires less computational effort and it produces a more uniform distribution than HV (Brockhoff et al., 2012;

Hernández Gómez and Coello Coello, 2013). Given a Pareto Front approximation A, the unary version of the $R2$ indicator is defined as follows:

$$R2(A, U) = \frac{1}{|U|} \sum_{u \in U} \min_{a \in A}\{u(a)\}, \tag{8}$$

where U is a set of utility functions $u : \mathbb{R}^m \mapsto \mathbb{R}$ that are a model of the decision maker's preference that maps each objective vector into a scalar value.

Motivated by the nice properties of the $R2$ indicator and the poor performance of algorithms based on Pareto ranking (Ishibuchi et al., 2008), several selection schemes based on this indicator have been developed (Phan and Suzuki, 2013; Hernández Gómez and Coello Coello, 2013; Brockhoff et al., 2015). Hernández Gómez and Coello Coello (2013) proposed the $R2$-ranking algorithm that has been implemented in MOMBI (Hernández Gómez and Coello Coello, 2013) and MOMBI-II (Hernández Gómez and Coello Coello, 2015) showing very competitive results in many-objective problems. The results obtained by MOMBI and MOMBI-II and the bad performance of the nondominated sorting procedure in MOACO$_{\mathbb{R}}$ have encouraged us to use the $R2$ indicator in the selection scheme.

The $R2$-ranking algorithm, whose main purpose is the creation of a nondominated sorting scheme as the one of NSGA-II (Deb et al., 2000), works as follows. It groups solutions which optimize a set of utility functions, and places them on top, such that they get the first rank. Then, such points are removed and a second rank is assigned in the same way and so on until there are no more points left to be ranked. One of the advantages of this scheme is its good performance on many-objective problems due to a higher selection pressure.

Concerning the choice of the utility function u in Eq. (8), we use the achievement scalarizing function (ASF) (Miettinen, 1999). The use of ASF is based on an experiment made by Hernández Gómez and Coello Coello (2015) who discovered that it has a better performance on many-objective problems than the Weighted Tchebycheff and the Penalty-based Boundary Intersection (PBI) methods. The ASF is defined as follows:

$$u_{asf}(\mathbf{v} \mid \mathbf{r}, \boldsymbol{\lambda}) = \max_{i \in \{1,\dots,m\}} \frac{1}{\lambda_i} |v_i - r_i| \tag{9}$$

where $\mathbf{r}$ is a reference vector and $\boldsymbol{\lambda}$ is a convex weight vector, both of dimension m. In order to define the set of utility functions, it is necessary to create a set of convex weight vectors $\Lambda = \{\boldsymbol{\lambda}^i \mid i = 1, \dots, N\}$. Λ is computed using the Simplex-Lattice Design (SLD) (Scheffe, 1958, 1963) that creates a simplex-lattice $\{m, h\}$, where m is the number of objectives and h is a proportional parameter. This structure consists of all possible combinations of proportions of each objective function. Hence, each coefficient of the weight vector takes $h + 1$ evenly spaced values between 0 and 1, that is:

$$\lambda_j^i = \epsilon, \frac{1}{h}, \frac{2}{h}, \dots, 1 \tag{10}$$

where ϵ is a value close to zero (usually 10^{-4}) employed with the aim of avoiding cancellation in the calculations. The cardinality of Λ is represented by a combinatorial number $N = C_{m-1}^{h+m-1}$. We use the algorithm of Scott D. Chasalow (1995) for the calculation of the SLD.

In Algorithm 2, we show the ranking algorithm proposed by Hernández Gómez and Coello Coello (2013). For this algorithm, we assume that each solution p in population P has the following structure:

- $p.\mathbf{F}$: objective vector
- $p.\alpha$: current utility value for the weight vector λ
- $p.u^*$: best utility value obtained
- $p.rank$: solution's rank assigned by the algorithm

Algorithm 2 $R2$-ranking (Hernández Gómez and Coello Coello, 2015)

Require: Population P, set of weight vectors Λ
Ensure: Ranking of the population

$p.rank \leftarrow p.\alpha \leftarrow \infty \ \ \forall p \in P$
for all $\mathbf{u} \in U$ **do**
  for all $p \in P$ **do**
    $p.\alpha \leftarrow u_{asf}(p.\mathbf{F} \mid \mathbf{z}^*, \boldsymbol{\lambda})$
  end for
  Sort P with regard to the values α and L_2-norm in increasing order
  $rank \leftarrow 1$
  for all $p \in P$ **do**
    if $rank < p.rank$ **then**
      $p.rank = rank$
      $p.u^* = p.\alpha$
    end if
    $rank \leftarrow rank + 1$
  end for
end for

According to its authors, the computational complexity of the $R2$-ranking algorithm is $\mathcal{O}(|\Lambda||P|(\log |P| + m))$.

4.3 Update of reference points

In Section 2, we introduced two important reference vectors: the ideal vector $\mathbf{z}^*$ and the nadir vector $\mathbf{z}^{nad}$. The former preserves the minimum value per objective; the latter is composed by the maximum objective values, obtained from the solutions in the Pareto optimal front. These vectors are specially relevant in the normalization process of the objectives.

The $R2$-ranking algorithm requires the objective vectors to be normalized in order to produce the ranking. This normalization is done using the following formula:

$$f'_j(\mathbf{x}) = \frac{f_j(\mathbf{x}) - z^*_j}{z^{nad}_j - z^*_j}, \forall j \in \{1, \ldots, m\}. \tag{11}$$

Hernández Gómez and Coello Coello (2015) found some problems using this approach, e.g., the ideal vector of the feasible region was never retained, and the nadir vector was always an outlier in multifrontal MOPs, among others. For this reason, they proposed a mechanism that generates statistical approximations $\mathbf{z}^{min}$ and $\mathbf{z}^{max}$ of the ideal and nadir vectors, respectively. The idea is to monitor the nadir point of the current population at each iteration, determining how close are the solutions from the true Pareto front. A high variance means that solutions are far away, which strongly biases the location of the point.

In Algorithm 3, we show the pseudo-code of this mechanism. This algorithm requires a data structure called RECORD that stores the nadir vector of a few generations. Based on these registers, it calculates a variance vector $\mathbf{v}$ where each $v_j \in \mathbf{v}$ represents the variance for the j^{th} dimension of the stored nadir vectors. The mechanism needs the parameter α that determines the variance threshold for $\mathbf{z}^{nad}$ and a parameter ϵ which is the tolerance threshold. Hernández Gómez and Coello Coello (2015) recommend using: $\alpha = 0.5$, $\epsilon = 0.001$ and to store the nadir vectors of 5 generations. The underlying idea of this mechanism is to examine the variance vector $\mathbf{v}$. If the maximum variance of all objectives is high, then $\mathbf{z}^{max}$ is updated using the maximum objective of the nadir vector (lines 5 and 6). Otherwise, from lines 8 to 18, each component of $\mathbf{z}^{max}$ is examined and we have three cases: (1) if $|z_i^{max} - z_i^{min}| < \epsilon$, then z_i^{max} is updated using the maximum objective value of $\mathbf{z}^{max}$ and it is marked, (2) if $z_i^{nad} < z_i^{max}$, then z_i^{max} is expanded and marked, (3) if the variance is zero and z_i^{max} has not been recently marked, z_i^{max} is set as an average of its previous value and the maximum value stored for it in RECORD. In all cases, the mark lasts the same number of generations that RECORD is kept. The computational complexity of this algorithm is $\mathcal{O}(|P|m)$.

Algorithm 3 Update Reference Points (Hernández Gómez and Coello Coello, 2015)

Require: $\mathbf{z}^{min}, \mathbf{z}^{max}$, population P, number of objectives m
Ensure: $\mathbf{z}^{min}, \mathbf{z}^{max}$
1: Update $\mathbf{z}^*$ and $\mathbf{z}^{nad}$ using definitions 7 and 8
2: $z_i^{min} \leftarrow \min z_i^{min}, z_i^* \;\; \forall i \in \{1, \ldots, m\}$
3: Add $\mathbf{z}^{nad}$ to RECORD
4: Obtain variance vector $\mathbf{v} \in \mathbb{R}^m$ for $\mathbf{z}^{nad}$ from RECORD
5: **if** $\max_{j \in \{1,\ldots,m\}} v_j > \alpha$ **then**
6: $\quad z_i^{max} \leftarrow \max_{j \in \{1,\ldots,m\}} z_j^{nad} \;\; \forall i \in \{1, \ldots, m\}$
7: **else**
8: $\quad$ **for all** $i \in \{1, \ldots, m\}$ **do**
9: $\quad\quad$ **if** $|z_i^{max} - z_i^{min}| < \epsilon$ **then**
10: $\quad\quad\quad z_i^{max} \leftarrow \max_{j \in \{1,\ldots,m\}} z_j^{max}$
11: $\quad\quad\quad$ Mark z_i^{max}
12: $\quad\quad$ **else if** $z_i^{nad} > z_i^{max}$ **then**
13: $\quad\quad\quad z_i^{max} \leftarrow 2z_i^{nad} - z_i^{max}$
14: $\quad\quad\quad$ Mark z_i^{max}
15: $\quad\quad$ **else if** $v_i = 0 \wedge z_i^{max}$ has not been recently marked **then**
16: $\quad\quad\quad$ Get the maximum value a for z_i^{nad} from RECORD
17: $\quad\quad\quad z_i^{max} \leftarrow (z_i^{max} + a)/2$
18: $\quad\quad\quad$ Mark z_i^{max}
19: $\quad\quad$ **end if**
20: $\quad$ **end for**
21: **end if**
22: **return** $\{\mathbf{z}^{min}, \mathbf{z}^{max}\}$

4.4 iMOACO$_{\mathbb{R}}$

In the previous sections, we described the basic mechanisms that are used in the construction of iMOACO$_{\mathbb{R}}$ (see Algorithm 4). This proposal is an extension of the ACO metaheuristic for the solution of continuous MOPs that uses ACO$_{\mathbb{R}}$ as its search engine without the variable correlation mechanism. The two main aspects of iMOACO$_{\mathbb{R}}$ are the following. Firstly, it uses a slightly different pheromone archive, which stores the best solutions according to the $R2$-ranking algorithm. Secondly, the pheromone update process promotes a competition between the newly created solutions and the current solutions stored in the archive.

Each ant $a \in \mathcal{A}$ and pheromone $p \in \mathcal{T}$, where $\mathcal{A}$ denotes the set of ants and $\mathcal{T}$ the pheromone archive, have the following field structure:

- $\mathbf{x}$: decision variables vector,
- $\mathbf{F}$: objective vector,
- $\mathbf{F}_{norm}$: normalized objective vector,
- α: current utility value, and
- u^*: best utility value.

iMOACO$_{\mathbb{R}}$ requires six parameters: $G_{max}, q, \xi, \alpha, \epsilon$ and h. G_{max} is the maximum number of generations. The parameters q and ξ are employed by the ACO$_{\mathbb{R}}$-based search engine (see Section 3). The update of the reference vectors requires the parameters α and ϵ, which are the variance threshold and the tolerance threshold, respectively. Finally, h is the proportional parameter utilized for the construction of the simplex-lattice on the SLD in order to create the set of N convex weight vectors. N is equally used as the number of ants and the cardinality of the pheromone archive $\mathcal{T}$. This decision is based on the μ-optimal distributions of the $R2$ indicator that claims that if we have μ solutions and N weight vectors with $\mu \geq N$, the $\mu - N$ solutions will not contribute to the $R2$ indicator value (Brockhoff et al., 2012).

Next, we describe iMOACO$_{\mathbb{R}}$ in Algorithm 4. In line 1, the set of N weight vectors is computed using the SLD method. In lines 2 to 4, the pheromone archive $\mathcal{T}$ is initialized using a uniform distribution and, immediately, $\mathbf{z}^*$ and $\mathbf{z}^{nad}$ are computed with the aim of normalizing $\mathcal{T}$. The RECORD instance is created and initialized in lines 5 and 6. In line 7, the solutions in $\mathcal{T}$ are ranked using the $R2$-ranking algorithm in furtherance of determining their quality. From lines 9 to 21, the main loop of iMOACO$_{\mathbb{R}}$ is executed until the maximum number of iterations is exceeded. In lines 10 to 12, each ant generates a new solution using the standard mechanisms of ACO$_{\mathbb{R}}$. The statistical approximations of the ideal and nadir vectors are calculated in line 13 using Algorithm 3. The pheromone update process is described from lines 14 to 19. Let $\Psi = \mathcal{A} \bigcup \mathcal{T}$. Then, in line 15, the objective vectors of all solutions in Ψ are normalized with the aim of being ranked by the $R2$-ranking algorithm. Ψ is later sorted with regard to the fields: (1) rank, (2) u^*, and (3) the L_2-norm, in increasing order. The sorting will ensure to have at the top those solutions closer to the Pareto optimal set. In line 18, all solutions of $\mathcal{T}$ are removed and the first N of Ψ are copied into the archive. As we explained above, the pheromone process promotes a competition between the newly created pheromones and the older ones with the purpose of preserving those solutions that maximize the $R2$ indicator value. At the end of the search process, the content of $\mathcal{T}$ is returned as the Pareto front approximation.

Algorithm 4 iMOACO$_{\mathbb{R}}$ main loop

Require: h, q, ξ, α, ϵ, G_{max}
Ensure: Pareto Front approximation
Generate a set Λ of weight vectors using the SLD method
Randomly initialize pheromones in $\mathcal{T}$
Initialize $\mathbf{z}^*$ and $\mathbf{z}^{nad}$ using solutions in $\mathcal{T}$
Normalize($\mathcal{T}, \mathbf{z}^*, \mathbf{z}^{nad}$)
Create instance R of RECORD
Add $\mathbf{z}^{nad}$ to R
$\mathcal{T} \leftarrow$ R2-ranking($\mathcal{T}, \Lambda$)
$t \leftarrow 0$
while $t < G_{max}$ **do**
  for all $a \in \mathcal{A}$ **do**
    Generate solution for a based on $\mathcal{T}$
  end for
  $\{\mathbf{z}^*, \mathbf{z}^{nad}\} \leftarrow$ UpdateRefPoints($\mathbf{z}^*, \mathbf{z}^{nad}, \mathcal{A}, m$) {Algorithm 3}
  $\Psi \leftarrow \mathcal{A} \cup \mathcal{T}$
  Normalize($\Psi, \mathbf{z}^*, \mathbf{z}^{nad}$)
  $\Psi \leftarrow$ R2ranking(Ψ, Λ)
  Sort Ψ, in increasing order, with regards to: (1) $rank$, (2) u^*, y (3) L_2-norm
  $\mathcal{T} \leftarrow \emptyset$
  Copy into $\mathcal{T}$ the first N elements of Ψ
  $t \leftarrow t + 1$
end while
return $\mathcal{T}$

The computational complexity of iMOACO$_{\mathbb{R}}$ is analyzed next. The generation of new solutions takes $\mathcal{O}(N^2 n)$. Updating the reference vectors requires $\mathcal{O}(Nm)$. Both the union of $\mathcal{A}$ and $\mathcal{T}$ and the objective vector normalization are performed in $\mathcal{O}(Nm)$. The $R2$-ranking algorithm has complexity $\mathcal{O}(N^2(\log N + m))$ (as $N = |P|$). Removing solutions from $\mathcal{T}$ is performed in $\mathcal{O}(1)$. Copying the best N solutions of Ψ into $\mathcal{T}$ is done in $\mathcal{O}(N(n+m))$. Finally, as we execute one more time the $R2$-ranking algorithm, this requires $\mathcal{O}(N^2(\log N + m))$ meanwhile its sorting takes $\mathcal{O}(N \log N)$. Therefore, the overall complexity of iMOACO$_{\mathbb{R}}$ at each iteration is $\mathcal{O}(N^2(\log N + m + n))$ and the storage requires $\mathcal{O}(N(m+n))$.

5 Experimental Setup

In this section, we describe the experimental setup used for the performance assessment of our proposed iMOACO$_{\mathbb{R}}$[2]. We compared our proposal with respect to three state-of-the-art MOEAs and one MOACO algorithm for continuous domains. The first is the Multi-Objective Evolutionay Algorithm based on Decomposition (MOEA/D)[3] (Zhang and Li, 2007) which decomposes the MOP using a set of weight vectors to generate a set of subproblems which are simultaneously optimized. The second approach is the Nondominated Sorting Genetic Algorithm III (NSGA-III)[4], proposed by Deb and Jain (2014), that is capable to scale its per-

[2] The source code and the complete study of iMOACO$_{\mathbb{R}}$ are available at: http://computacion.cs.cinvestav.mx/~jfalcon/iMOACOR/imoacor.html.

[3] We used the implementation from 2007 for continuous search spaces: http://dces.essex.ac.uk/staff/zhang/webofmoead.htm

[4] We used the implementation available at: http://web.ntnu.edu.tw/~tcchiang/publicstions/nsga3cpp/nsga3cpp.htm

Table 1 Common parameters used for the algorithms adopted in our comparative study. m is the dimension of the objective space; SBX crossover needs P_c and N_c while polynomial-based mutation requires P_m and N_m; h is the proportional parameter for the generation of the set of weight vectors and G_{max} is related to the maximum number of generations. FE indicates the total number of function evaluations executed (FE = Pop. size $\times$ G_{max}).

m	P_c	N_c	P_m	N_m	h	**Pop. size**	G_{max}	FE
3	1.0	20	$1/n$	20	14	120	416	49,920
5		30			5	126	396	49,896
7					3	84	595	49,980
10					3	220	227	49,940

formance to high dimensionality due to the utilization of a set of evenly spread reference points across objective space. In third instance, we adopted the S Metric Selection-Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) (Beume et al., 2007) which is a hypervolume-based MOEA. Since SMS-EMOA requires a considerably large computational time in high-dimensional problems, we decided to use the hypervolume estimation algorithm introduced by the algorithm HypE (Bader and Zitzler, 2011) in order to reduce the computational cost of SMS-EMOA. Finally, the selected MOACO algorithm is MOACO$_{\mathbb{R}}$[5] (Garcia-Najera and Bullinaria, 2007), which is described in Section 2.3.

Next, we describe the parameters adopted to compare the performance on 3, 5, 7 and 10 objectives of MOEA/D, NSGA-III, SMS-EMOA(HypE), MOACO$_{\mathbb{R}}$ and iMOACO$_{\mathbb{R}}$. Then, we introduce the performance assessment measures used to validate our results and, finally, we describe the test problems adopted.

5.1 Parameters settings

MOEA/D, NSGA-III and SMS-EMOA are based on genetic algorithms and all of them adopt the same variation operators: simulated binary crossover (SBX) and polynomial-based mutation (Deb and Agrawal, 1995). This variation operators depend on four parameters: crossover rate (P_c), crossover distribution index (N_c), mutation rate (P_m) and mutation distribution index (N_m). In Table 1 we show the adopted values of these parameters for different dimensions of the objective space. We must emphasize that the adopted values are based on the recommendations of Deb et al. (2000) and Deb and Jain (2014) which coincide with those used in the original papers of each algorithm. For MOEA/D the neighborhood size T was set to 20 as proposed by Zhang and Li (2007). Concerning SMS-EMOA, we decided to use 10,000 samples for HyPE's hypervolume estimation algorithm.

As MOACO$_{\mathbb{R}}$ and iMOACO$_{\mathbb{R}}$ are based on ACO$_{\mathbb{R}}$ (see Section 3), they need the parameters q and ξ that control the diversification process and the speed of convergence, respectively. For all problems, regardless of their dimensionality, these parameters have been set as follows: $q = 0.1$ and $\xi = 0.5$. This decision was based on the results obtained by ACO$_{\mathbb{R}}$ on an empirical performance study that we conducted. As suggested by Hernández Gómez and Coello Coello (2015), the parameters α and ϵ, which are required to update the reference points in Algorithm 3, were set to 0.5 and 0.001, respectively.

[5] The source code was provided by its author, Abel García-Nájera.

Table 2 Properties of the test problems.

Problem	Separability	Frontality	Geometry
DTLZ1	separable	multifrontal	linear
DTLZ2	separable	unifrontal	concave
DTLZ3	separable	multifrontal	concave
DTLZ4	separable	unifrontal	concave
DTLZ5	unknown	unifrontal	degenerated
DTLZ6	unknown	unifrontal	degenerated
DTLZ7	not applicable separable	$f_{1:m-1}$ unimodal f_m multimodal	disconected mixed
WFG1	separable	unifrontal	convex, mixed
WFG2	non-separable	$f_{1:m-1}$ unimodal f_m multimodal	convex disconected
WFG3	non-separable	unifrontal	linear, degenerated
WFG4	separable	multifrontal	concave
WFG5	separable	deceptive	concave
WFG6	non-separable	unifrontal	concave
WFG7	separable	unifrontal	concave
WFG8	non-separable	unifrontal	concave
WFG9	non-separable	multifrontal, deceptive	concave

The population size and the maximum number of generations (G_{max}) adopted in the experiments reported next are shown in Table 1. In case of MOACO$_{\mathbb{R}}$ and iMOACO$_{\mathbb{R}}$, the population size is equal to the pheromone archive size. Since MOEA/D, NSGA-III and iMOACO$_{\mathbb{R}}$ use a set of weight vectors (created by the SLD method), they need the parameter h whose value changes depending on the dimensionality of the objective space. The total number of function evaluations (FEs) is calculated as follows: FE = Population size $\times$ G_{max}. For all algorithms in all problems, the total number of FEs was set in such a way that it does not exceed 50,000, as shown in Table 1.

5.2 Test problems

For comparison purposes, we adopted the Deb-Thiele-Laumanns-Zitzler (DTLZ) (Deb et al., 2002) and the Walking-Fish-Group (WFG) (Huband et al., 2005, 2006) test suites. Table 2 summarizes the main features of these test problems (Huband et al., 2006). All minimization problems adopted are scalable with respect to the number of objectives and have a variety of features (e.g., linearity, convexity, concavity, multifrontality, bias, separability, etc.) which are summarized by Huband et al. (2006).

The standard methodology to construct the test instances is the following (Coello Coello et al., 2007). For the DTLZ test suite, the total number of decision variables is given by $n = m + k - 1$, where m is the number of objectives. k has been set to 5 for DTLZ1, 10 for DTLZ2-6 and 20 for DTLZ7. In case of the WFG test suite, Table 3 shows the number of decision variables and position-related parameters employed, as suggested by Huband et al. (2005, 2006).

Table 3 Adopted configuration of position-related parameters and decision variables for the WFG test problems.

Parameters	**Objective space dimension**			
	3D	5D	7D	10D
Position-related	4	8	12	18
Decision variables	24	47	70	105

5.3 Performance assessment

For comparing results, we selected three state-of-the-art performance indicators. The first is the hypervolume (or S metric) (HV) (Zitzler, 1999; Brockhoff et al., 2008) that is the only unary indicator known to be strictly monotonic with respect to Pareto dominance (Zitzler et al., 2003). HV determines the volume dominated by an approximation to the Pareto optimal front (PF_{known}), bounded by some reference point. Since this reference point affects the calculated hypervolume value, we provide the values that we adopted for each test problem in Table 4. It is worth noting that maximizing HV guarantees converging to the true Pareto optimal front of the problem being solved (Zitzler et al., 2003; Knowles and Corne, 2002). Mathematically, the hypervolume can be described using the next formula:

$$HV(A) = \left\{ \bigcup volume(v) \mid v \in A \right\}. \tag{12}$$

In order to calculate the hypervolume in all the considered dimensions, we employed the algorithm proposed by While et al. (2012).

Our second choice is the Spacing indicator (S) (Coello Coello et al., 2007) which measures the variance of the distance between neighboring vectors in PF_{known}. S is defined by Eqs. (13) and (14) as follows:

$$S := \sqrt{\frac{1}{|PF_{known}| - 1} \sum_{i=1}^{|PF_{known}|} (\bar{d} - d_i)^2} \tag{13}$$

and

$$d_i = \min_j \left(\sum_{k=1}^{m} \left| f_k^i(\mathbf{x}) - f_k^j(\mathbf{x}) \right| \right) \tag{14}$$

where d_i, $i, j = 1, \ldots, N$, $\bar{d}$ is the average of all d_i's, N is the cardinality of PF_{known} and m is the dimension of the objective space. When $S = 0$, all points are evenly spaced. S does not require the true Pareto optimal front to be known, although, it is assumed that the algorithm has converged before applying this indicator.

Finally, the third selected indicator is the modified Inverted Generational Distance (IGD$^+$) (Ishibuchi et al., 2014, 2015). IGD$^+$ solves some difficulties related to the reference sets in high dimensionality, of the Inverted Generational Distance indicator (IGD) (Coello Coello et al., 2007; Coello Coello and Cruz Cortés, 2005). The main idea of IGD$^+$ is to consider the dominance relation between a solution and a reference point when the distance is calculated. Due to this fact,

Table 4 Reference points used in the calculation of the hypervolume indicator.

Test problem	Reference point
DTLZ1	$(1, 1, 1, \ldots)$
DTLZ2, DTLZ4	$(2, 2, 2, \ldots)$
DTLZ3	$(7, 7, 7, \ldots)$
DTLZ5	$(4, 4, 4, \ldots)$
DTLZ6	$(11, 11, 11, \ldots)$
DTLZ7	$(1, 1, 1, \ldots, 21)$
WFG1-9	$(3, 5, 7, \ldots, 2m + 1)$

IGD$^+$ is weakly Pareto compliant. For a minimization problem, IGD$^+$ is defined by Eqs. (15) and (16):

$$\mathrm{IGD}^+(A) := \frac{1}{|Z|} \sum_{j=1}^{|Z|} \min_{a_i \in A} d_{\mathrm{IGD}^+}(a_i, z_j) \tag{15}$$

and

$$d_{\mathrm{IGD}^+} = \sqrt{\sum_{k=1}^{m} \left(\max\{a_k - z_k, 0\}\right)^2} \tag{16}$$

where A is a Pareto front approximation and the reference set is denoted by Z. For a specific problem, we calculate Z by joining the Pareto front approximations (keeping only nondominated solutions) of the algorithms and then applying k-means clustering (Jain et al., 1999) in order to reduce the number of solutions. When IGD$^+$ is closer to zero, A is closer to Z. In consequence, IGD$^+$ simultaneously evaluates convergence and spread of an approximation set.

6 Discussion of results

This section aims to present and discuss the statistical results of iMOACO$_\mathbb{R}$ compared to MOACO$_\mathbb{R}$, MOEA/D, NSGA-III and SMS-EMOA, using the DTLZ and WFG test suites. We performed 30 independent runs of each of the 5 metaheuristics compared on all test instances adopted. For our statistical assessment of results, we adopted the Wilcoxon rank sum test (one-tailed), with a significance level of 5%. A comparison of the results of the five algorithms in terms of HV, IGD$^+$ and S values are presented in Tables 5, 6 and 7. In these tables, for each m-objective test instance, the two best values are shown in gray scale, where the darker tone corresponds to the best value. Furthermore, the symbol # is placed when a result is statistically different from iMOACO$_\mathbb{R}$'s result based on the one-tailed Wilcoxon test.

All experiments reported here were performed on a PC having an Intel(R) Xeon(R) running at 3.00 GHz and with 64 GBytes in RAM. All metaheuristics adopted in our comparative study were implemented in C/C++ under Linux (Fedora 15), using real-numbers encoding.

Table 5 Statistical results (mean and standard deviation) of the HV values obtained by iMOACO$_{\mathbb{R}}$, MOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on the DTLZ and WFG test suites. The two best values are shown in gray scale, where the darker tone corresponds to the best value. The symbol # is placed when a result is statistically different from iMOACO$_{\mathbb{R}}$'s result based on a one-tailed Wilcoxon test with a significance level of 5%. N/A (not applicable) means that the experiment was not executed. A zero HV value implies that the algorithm was unable to converge to the true Pareto front.

Problem	m	iMOACO$_{\mathbb{R}}$	MOACO$_{\mathbb{R}}$	MOEA/D	NSGA-III	SMS-EMOA
DTLZ1	3	0.000000e+00(0.000e+00)	0.000000e+00(0.000e+00)	9.740945e-01(2.619e-04)	9.741172e-01(3.045e-04)	6.766570e-01(1.240e-02)
	5	0.000000e+00(0.000e+00)	N/A	9.985205e-01(6.682e-05)	1.017165e+00(9.991e-02)	4.138286e+00(1.554e+01)
	7	0.000000e+00(0.000e+00)	N/A	9.996951e-01(7.351e-05)	1.967483e+00(3.623e+00)	8.228500e-01(7.216e-02)
	10	0.000000e+00(0.000e+00)	N/A	9.850897e-01(4.702e-03)	9.997797e-01(4.081e-04)	N/A
DTLZ2	3	7.420231e+00(2.561e-04)	7.393920e+00(5.114e-03)#	7.421715e+00(1.371e-04)	7.421572e+00(6.064e-04)	4.111960e+00(9.221e-02)#
	5	3.165018e+01(2.529e-03)	N/A	3.166768e+01(3.208e-04)	3.166519e+01(8.761e-04)	2.143628e+01(7.673e-01)#
	7	1.277012e+02(7.756e-03)	N/A	1.277477e+02(1.907e-03)	1.277489e+02(1.372e-03)	8.263293e+01(4.223e+00)#
	10	1.023614e+03(9.715e-02)	N/A	1.023857e+03(1.823e-02)	1.023849e+03(1.252e-02)	N/A
DTLZ3	3	0.000000e+00(0.000e+00)	0.000000e+00(0.000e+00)	3.420659e+02(1.134e+00)	3.417475e+02(1.559e+00)	1.420606e+04(4.967e+04)
	5	0.000000e+00(0.000e+00)	N/A	1.718503e+04(4.197e+03)	2.990271e+04(5.922e+04)	5.327954e+05(3.461e+05)
	7	0.000000e+00(0.000e+00)	N/A	7.426452e+05(5.279e+04)	7.172101e+06(3.423e+07)	1.907858e+06(3.530e+06)
	10	0.000000e+00(0.000e+00)	N/A	2.563130e+08(1.884e+07)	2.599402e+09(6.664e+09)	N/A
DTLZ4	3	7.419261e+00(9.233e-04)	7.397218e+00(4.404e-03)#	7.421635e+00(1.148e-04)	7.218780e+00(4.062e-01)	4.478782e+00(3.855e-01)#
	5	3.163519e+01(3.624e-03)	N/A	3.166851e+01(1.701e-04)	3.162929e+01(1.453e-01)	2.323799e+01(2.179e+00)#
	7	1.277266e+02(7.245e-03)	N/A	1.277444e+02(2.351e-03)	1.277541e+02(4.924e-04)	9.249773e+01(3.292e+00)#
	10	1.023870e+03(4.029e-03)	N/A	1.023912e+03(2.478e-03)	1.023902e+03(3.014e-03)	N/A
DTLZ5	3	5.983806e+01(7.194e-03)	5.986894e+01(1.342e-03)	5.973486e+01(1.291e-03)#	5.982962e+01(8.365e-03)#	5.027973e+01(5.019e-01)#
	5	9.372627e+02(9.321e-01)	N/A	9.465698e+02(4.026e-01)	9.447159e+02(6.332e+00)	7.232389e+02(5.931e+00)#
	7	1.444221e+04(1.115e+02)	N/A	1.492758e+04(1.705e+01)	1.486426e+04(9.418e+01)	1.080888e+04(1.253e+02)#
	10	9.357044e+05(6.615e+03)	N/A	9.477366e+05(6.757e+02)	9.411728e+05(8.763e+03)	N/A
DTLZ6	3	1.315942e+03(1.175e+00)	1.318611e+03(1.107e+00)	1.316428e+03(8.424e-01)	1.316830e+03(4.802e-01)	1.186353e+03(1.891e+01)#
	5	1.563606e+05(1.388e+03)	N/A	1.584583e+05(8.812e+02)	1.542282e+05(4.384e+02)#	1.417357e+05(7.095e+02)#
	7	1.801024e+07(2.869e+05)	N/A	1.919747e+07(3.721e+04)	1.849564e+07(7.311e+04)	1.672813e+07(1.029e+05)#
	10	2.416417e+10(3.315e+08)	N/A	2.529085e+10(2.087e+08)	2.322341e+10(2.063e+08)#	N/A
DTLZ7	3	1.624747e+01(3.431e-02)	1.633222e+01(1.027e-02)	1.620770e+01(1.240e-01)#	1.631926e+01(1.253e-02)	8.253488e+00(3.300e+00)#
	5	1.256712e+01(8.935e-02)	N/A	5.594105e+00(1.696e+00)#	1.283906e+01(3.913e-02)	2.089669e+00(3.188e+00)#
	7	8.283139e+00(2.093e-01)	N/A	1.367398e-01(2.729e-01)#	8.446224e+00(3.724e-01)	9.710104e-03(2.028e-02)#
	10	2.152074e+00(6.544e-01)	N/A	2.947743e-03(7.802e-03)#	1.534016e+00(4.499e-01)#	N/A
WFG1	3	4.425029e+01(5.984e-01)	4.482325e+01(2.679e-01)	4.862674e+01(3.106e+00)	4.353844e+01(2.491e+00)#	4.561803e+01(1.162e+00)
	5	3.937020e+03(1.049e+02)	N/A	3.884123e+03(1.727e+02)#	3.246474e+03(7.198e+01)#	3.688324e+03(5.756e+01)#
	7	6.762082e+05(3.201e+04)	N/A	6.618675e+05(1.448e+04)#	5.478707e+05(1.352e+04)#	5.913484e+05(1.588e+04)#
	10	3.961333e+09(2.039e+07)	N/A	3.802791e+09(5.153e+07)#	3.273061e+09(5.361e+07)#	N/A
WFG2	3	9.706166e+01(5.756e-01)	9.980343e+01(2.557e-01)	8.220310e+01(6.485e+00)#	9.556901e+01(6.631e+00)	7.472122e+01(4.189e+00)#
	5	9.736942e+03(9.487e+01)	N/A	7.225543e+03(5.552e+02)#	9.245978e+03(8.013e+02)	8.281549e+03(1.282e+02)#
	7	1.701444e+06(2.296e+04)	N/A	1.227180e+06(6.399e+04)#	1.672969e+06(1.575e+05)	1.527222e+06(6.119e+04)#
	10	9.437415e+09(1.712e+08)	N/A	8.086769e+09(3.775e+08)#	1.117947e+10(8.590e+08)	N/A
WFG3	3	7.243938e+01(2.404e-01)	7.084552e+01(4.362e-01)#	6.879431e+01(2.038e+00)#	7.405609e+01(4.437e-01)	4.710888e+01(8.093e-01)#
	5	5.400104e+03(2.486e+02)	N/A	5.236481e+03(1.518e+02)#	6.410663e+03(6.333e+01)	3.196907e+03(1.283e+02)#
	7	7.904484e+05(2.199e+04)	N/A	7.264922e+05(5.165e+04)#	5.312858e+05(2.120e+05)#	4.979576e+05(2.701e+04)#
	10	4.742821e+09(7.999e+07)	N/A	2.394173e+09(1.504e+08)#	5.662479e+09(3.371e+08)	N/A
WFG4	3	7.068614e+01(3.010e-01)	6.887500e+01(4.965e-01)#	7.267887e+01(5.718e-01)	7.501456e+01(2.272e-01)	3.822859e+01(1.458e+00)#
	5	7.617384e+03(1.162e+02)	N/A	7.466392e+03(2.892e+02)	8.063859e+03(8.255e+01)	3.544988e+03(7.739e+01)#
	7	1.244395e+06(4.907e+04)	N/A	1.019393e+06(1.131e+05)#	1.583968e+06(2.412e+04)	6.203148e+05(1.688e+04)#
	10	7.170558e+09(2.850e+08)	N/A	6.385061e+09(7.696e+08)#	8.950776e+09(2.529e+08)	N/A
WFG5	3	6.834469e+01(8.124e-01)	6.346205e+01(1.672e+00)#	7.065721e+01(2.755e-01)	7.197491e+01(4.053e-01)	3.537145e+01(1.463e+00)#
	5	4.835114e+03(1.938e+02)	N/A	7.629797e+03(1.728e+02)	8.011001e+03(6.378e+01)	3.485080e+03(1.106e+02)#
	7	6.887508e+05(4.105e+04)	N/A	1.159399e+06(1.543e+05)	1.588530e+06(2.191e+04)	6.047207e+05(2.297e+04)#
	10	4.317068e+09(1.650e+08)	N/A	6.192197e+09(7.343e+08)	8.883282e+09(1.562e+08)	N/A
WFG6	3	7.424075e+01(4.159e-01)	7.257026e+01(7.573e-01)#	7.065243e+01(7.719e-01)#	7.242177e+01(5.130e-01)#	3.732450e+01(2.274e+00)#
	5	6.768850e+03(3.912e+02)	N/A	6.431094e+03(9.169e+02)	8.230034e+03(6.976e+01)	3.568933e+03(2.393e+02)#
	7	8.324711e+05(5.551e+04)	N/A	8.176655e+05(9.898e+03)	1.693310e+06(1.419e+04)	6.188628e+05(1.439e+04)#
	10	4.790208e+09(2.524e+08)	N/A	5.170353e+09(1.858e+08)	8.885712e+09(2.249e+08)	N/A
WFG7	3	7.531982e+01(2.689e-01)	7.394943e+01(4.090e-01)#	7.254121e+01(1.567e+00)#	7.615301e+01(1.096e-01)	3.670642e+01(1.643e+00)#
	5	7.261774e+03(2.611e+02)	N/A	6.982161e+03(4.230e+02)#	8.313109e+03(1.061e+02)	3.463331e+03(7.242e+01)#
	7	1.094708e+06(5.602e+04)	N/A	8.603106e+05(1.645e+04)#	1.664359e+06(2.987e+04)	6.225210e+05(6.574e+03)#
	10	6.963838e+09(2.330e+08)	N/A	5.243514e+09(2.561e+08)#	9.386685e+09(1.513e+08)	N/A
WFG8	3	6.535407e+01(5.311e-01)	6.298557e+01(6.187e-01)#	6.717480e+01(6.432e-01)	6.862097e+01(3.072e-01)	3.428887e+01(1.102e+00)#
	5	5.109791e+03(2.478e+02)	N/A	6.203692e+03(5.497e+02)	7.603183e+03(7.756e+01)	2.934883e+03(1.141e+02)#
	7	7.712374e+05(6.224e+04)	N/A	4.876062e+05(1.312e+05)#	1.476134e+06(2.868e+04)	4.547833e+05(4.406e+04)#
	10	5.339008e+09(3.843e+08)	N/A	2.549207e+09(8.836e+08)#	8.394377e+09(1.694e+08)	N/A
WFG9	3	6.597174e+01(1.815e-01)	6.554430e+01(4.531e-01)#	6.533818e+01(1.699e+00)	7.002323e+01(1.791e+00)	3.708994e+01(2.585e+00)#
	5	5.768426e+03(3.253e+02)	N/A	6.362462e+03(9.840e+02)	7.476119e+03(2.474e+02)	3.629624e+03(2.498e+02)#
	7	7.309045e+05(1.035e+05)	N/A	8.943812e+05(2.363e+05)	1.352921e+06(5.646e+04)	6.987414e+05(1.173e+05)#
	10	4.188941e+09(3.585e+08)	N/A	3.122641e+09(1.626e+09)#	7.329193e+09(3.111e+08)	N/A

6.1 Performance comparisons on the DTLZ test suite

DTLZ1 and DTLZ3 are two multifrontal problems (as shown in Table 2) which introduce $(11^k - 1)$ and $(3^k - 1)$ local Pareto fronts, respectively. Due to the multifrontality of these two problems, iMOACO$_{\mathbb{R}}$ got stuck in a local Pareto front in both of them. This behavior is also detected in MOACO$_{\mathbb{R}}$. This issue arises as a natural consequence of the use of ACO$_{\mathbb{R}}$ as the search engine of both optimizers. Unfortunately, ACO$_{\mathbb{R}}$ presents some difficulties when dealing with multimodal problems (Leguizamón and Coello, 2010). This lack of convergence to the true Pareto front is reported in Tables 5 with a zero hypervolume value, and in Tables 6 and 7 with an ∞ value.

Table 6 Statistical results (mean and standard deviation) of the IGD$^+$ values obtained by iMOACO$_\mathbb{R}$, MOACO$_\mathbb{R}$, MOEA/D, NSGA-III and SMS-EMOA on the DTLZ and WFG test suites. The two best values are shown in gray scale, where the darker tone corresponds to the best value. The symbol # is placed when a result is statistically different from iMOACO$_\mathbb{R}$'s result based on a one-tailed Wilcoxon test with a significance level of 5%. N/A (not applicable) means that the experiment was not executed. An ∞ value implies that the algorithm was unable to converge to the true Pareto front.

Problem	m	iMOACO$_\mathbb{R}$	MOACO$_\mathbb{R}$	MOEA/D	NSGA-III	SMS-EMOA
DTLZ1	3	∞	∞	1.352432e-02(7.357e-04)	1.354202e-02(9.167e-04)	1.044670e-01(2.295e-03)
	5	∞	N/A	1.770441e-02(1.492e-03)	2.092960e-02(8.441e-03)	8.781075e-02(2.831e-02)
	7	∞	N/A	4.057104e-02(1.569e-03)	5.422202e-02(4.656e-02)	1.100855e-01(4.155e-03)
	10	∞	N/A	9.850897e-01(4.702e-03)	5.412175e-02(4.205e-02)	N/A
DTLZ2	3	2.073739e-02(1.372e-04)	2.857186e-02(1.446e-03)#	1.352432e-02(7.357e-04)	1.354202e-02(9.167e-04)	1.044670e-01(2.295e-03)
	5	9.001354e-02(5.425e-03)	N/A	6.668343e-02(8.980e-05)	6.850192e-02(7.848e-04)	3.404108e-01(6.097e-03)#
	7	1.386139e-01(9.467e-03)	N/A	8.300738e-02(2.521e-05)	8.636540e-02(1.142e-03)	5.246910e-01(7.837e-03)#
	10	1.601365e-01(7.690e-03)	N/A	3.045827e-02(9.151e-04)	4.832999e-02(3.458e-03)	N/A
DTLZ3	3	∞	∞	8.482280e-02(1.843e-01)	1.769805e-01(3.064e-01)	4.360171e-01(5.285e-01)
	5	∞	N/A	3.623186e-01(9.500e-01)	7.170602e-01(8.917e-01)	3.887638e+00(2.129e+00)
	7	∞	N/A	5.469882e-01(2.872e-01)	1.518148e+00(1.529e+00)	1.213168e+00(9.903e-01)
	10	∞	N/A	4.583286e-01(2.665e-01)	1.334163e+00(8.890e-01)	N/A
DTLZ4	3	3.193750e-02(5.791e-04)	3.724734e-02(1.161e-03)#	2.996432e-02(7.542e-06)	6.666353e-02(7.313e-02)	2.445605e-01(6.890e-02)
	5	9.464995e-02(3.952e-03)	N/A	5.104591e-02(4.149e-05)	5.894254e-02(2.509e-02)	3.921806e-01(3.884e-02)#
	7	1.480956e-01(1.241e-02)	N/A	7.252125e-02(1.831e-05)	7.330909e-02(4.118e-04)	5.534455e-01(1.991e-02)#
	10	1.024742e-01(3.222e-03)	N/A	3.666230e-02(1.242e-03)	5.058520e-02(3.238e-03)	N/A
DTLZ5	3	2.129263e-03(3.711e-04)	1.225832e-03(2.675e-04)	4.703606e-03(1.009e-05)#	1.694491e-03(4.522e-04)	1.629077e-03(4.585e-04)
	5	1.056244e-01(6.993e-03)	N/A	6.643330e-02(1.089e-03)	9.998197e-02(2.243e-02)	9.516339e-02(2.970e-03)
	7	1.487053e-01(1.088e-02)	N/A	1.120602e-01(2.013e-03)	2.126830e-01(3.582e-02)#	1.581263e-01(7.660e-03)#
	10	1.357425e-01(1.717e-02)	N/A	1.496687e-01(1.917e-03)#	1.814031e-01(2.168e-02)#	
DTLZ6	3	1.111736e-01(4.320e-02)	2.407273e-02(5.620e-02)	1.256803e-01(3.448e-02)	1.257182e-01(2.720e-02)	2.437784e-01(4.500e-02)#
	5	2.026485e-01(1.348e-01)	N/A	2.446304e-01(7.294e-03)	1.284266e+00(9.024e-02)#	3.054495e-01(1.515e-02)#
	7	1.102604e+00(3.119e-01)	N/A	2.096216e-01(2.663e-02)	1.819214e+00(1.212e-01)#	2.890078e-01(2.385e-02)
	10	9.391129e-01(2.690e-01)	N/A	4.568281e-01(2.082e-01)	2.374775e+00(1.287e-01)#	N/A
DTLZ7	3	8.585564e-02(3.574e-02)	3.434431e-02(3.130e-03)	9.662943e-02(1.501e-01)	3.709021e-02(2.104e-03)	1.937586e-01(5.030e-02)#
	5	1.152563e+00(3.334e-01)	N/A	2.219557e-01(3.948e-01)	8.262998e-01(3.633e-01)	7.051361e-01(3.533e-01)
	7	1.899385e+00(2.490e-01)	N/A	1.280057e+00(1.990e-01)	1.893459e+00(1.170e+00)	1.180358e+00(5.177e-02)
	10	1.587637e+00(1.843e-01)	N/A	3.060151e+00(6.544e-01)	4.572206e+00(6.019e-01)	N/A
WFG1	3	1.430117e+00(9.456e-03)	1.447081e+00(1.012e-02)#	1.294346e+00(4.620e-02)	1.363994e+00(2.860e-02)	1.326351e+00(3.493e-02)
	5	6.179398e-01(2.081e-02)	N/A	1.155149e-01(2.703e-02)	3.577268e-01(2.199e-02)	2.341036e-01(2.241e-02)
	7	1.028239e+00(1.473e-01)	N/A	1.115985e-01(2.944e-02)	3.822505e-01(2.072e-02)	4.146121e-01(4.105e-02)
	10	1.038277e+00(1.221e-01)	N/A	6.107899e-02(1.435e-02)	3.946059e-01(1.701e-02)	
WFG2	3	1.114796e-01(1.584e-02)	7.510881e-02(1.379e-02)	2.507221e-01(6.124e-02)#	4.831741e-02(7.710e-03)	6.071864e-01(1.551e-01)#
	5	2.753754e-01(2.809e-02)	N/A	1.217884e+00(4.161e-01)#	5.133961e-01(4.987e-01)	7.239465e-01(4.892e-01)#
	7	8.755361e-01(1.499e-01)	N/A	1.680729e+00(4.983e-01)#	7.917200e-01(5.896e-01)	9.216179e-01(5.588e-01)
	10	9.497545e-01(7.075e-02)	N/A	2.368147e+00(3.475e-01)#	8.249208e-01(9.370e-01)	
WFG3	3	1.485811e-01(9.103e-03)	2.411323e-01(2.320e-02)#	1.888430e-01(4.826e-02)#	7.228996e-02(1.314e-02)	6.850529e-01(1.817772e-02)#
	5	6.671848e-01(8.818e-02)	N/A	2.547736e-01(1.713e-02)	6.035394e-01(5.807e-02)	6.095891e-01(1.412e-02)
	7	9.882048e-01(8.973e-02)	N/A	6.327384e-01(2.945e-02)	7.781914e+00(2.409e+00)#	9.318176e-01(5.209e-02)
	10	2.039118e+00(9.069e-02)	N/A	1.311043e+00(2.552e-02)	3.069705e+00(7.192e-01)#	N/A
WFG4	3	2.048992e-01(5.533e-03)	2.450116e-01(1.176e-02)#	1.977936e-01(1.199e-02)	1.315811e-01(4.272e-03)	1.184868e+00(6.830e-03)#
	5	3.923978e-01(2.091e-02)	N/A	3.511958e-01(3.000e-02)	2.703007e-01(1.533e-02)	9.282835e-01(2.034e-02)#
	7	1.065930e+00(2.533e-01)	N/A	7.705341e-01(3.280e-02)	3.730912e-01(6.329e-02)	1.224622e+00(2.313e-02)#
	10	2.815883e+00(2.478e-01)	N/A	1.240225e+00(2.006e-02)	9.860570e-01(1.236e-01)	N/A
WFG5	3	1.963753e-01(1.223e-02)	2.926745e-01(3.711e-02)#	2.057112e-01(6.869e-03)#	1.725189e-01(2.703e-03)	8.767424e-01(3.600e-02)#
	5	1.197580e+00(9.484e-02)	N/A	2.585594e-01(8.855e-03)	2.193627e-01(1.334e-02)	7.810527e-01(2.091e-02)
	7	2.450597e+00(2.191e-01)	N/A	5.237272e-01(6.163e-02)	2.674830e-01(2.846e-02)	1.090338e+00(3.707e-02)
	10	3.836405e+00(2.880e-01)	N/A	8.299703e-01(1.921e-02)	6.795388e-01(4.928e-02)	N/A
WFG6	3	1.104239e-01(5.556e-03)	1.428572e-01(1.477e-02)#	2.002454e-01(2.172e-02)#	1.399133e-01(9.798e-03)#	1.080563e+00(1.993e-02)#
	5	5.469506e-01(9.212e-02)	N/A	3.980014e-01(8.110e-02)	2.223761e-01(1.321e-02)	8.117315e-01(4.400e-02)#
	7	1.343287e+00(1.042e-01)	N/A	6.406209e-01(5.949e-03)	2.173142e-01(1.622e-02)	1.013691e+00(1.940e-02)
	10	2.174708e+00(1.602e-01)	N/A	1.153931e+00(1.470e-02)	5.371587e-01(5.946e-02)	N/A
WFG7	3	1.062657e-01(3.269e-03)	1.452924e-01(1.385e-02)#	1.596908e-01(1.044e-02)#	9.364718e-02(2.070e-03)	5.564280e-01(2.119e-02)#
	5	4.764515e-01(5.880e-02)	N/A	3.607015e-01(3.271e-02)	2.436847e-01(2.408e-02)	8.468314e-01(1.661e-02)#
	7	1.029619e+00(4.873e-02)	N/A	7.712396e-01(7.232e-03)	2.739815e-01(3.385e-02)	1.223627e+00(1.849e-02)#
	10	1.555666e+00(1.116e-01)	N/A	1.120330e+00(1.898e-02)	5.434857e-01(5.101e-02)	N/A
WFG8	3	2.363381e-01(9.816e-03)	2.904208e-01(1.485e-02)#	2.122931e-01(1.056e-02)	1.729058e-01(6.910e-03)	9.814335e-01(1.599e-02)#
	5	7.648970e-01(5.604e-02)	N/A	4.274071e-01(1.142e-01)	2.406510e-01(1.567e-02)	7.894058e-01(2.615e-02)#
	7	1.392503e+00(1.029e-01)	N/A	1.243237e+00(1.979e-01)	3.524058e-01(2.377e-01)	1.294171e+00(8.067e-02)
	10	2.049588e+00(1.115e-01)	N/A	1.670862e+00(1.425e-01)	5.230700e-01(8.613e-02)	N/A
WFG9	3	1.841420e-01(1.479e-03)	2.001286e-01(8.352e-03)#	2.358542e-01(4.167e-02)#	1.309404e-01(3.243e-02)	1.021225e+00(1.309e-01)#
	5	8.974369e-01(1.704e-01)	N/A	4.452596e-01(2.242e-01)	2.607509e-01(5.326e-02	7.080046e-01(3.199e-02)
	7	2.301408e+00(3.109e-01)	N/A	9.136098e-01(3.396e-01)	4.106712e-01(1.066e-01)	9.305680e-01(8.084e-02)
	10	3.543821e+00(2.839e-01)	N/A	1.517764e+00(4.994e-01	6.791161e-01(6.389e-02)	N/A

DTLZ2 is a unifrontal and separable problem with a concave Pareto front, where the objective functions of a Pareto optimal solution $\mathbf{x}^*$ needs to satisfy: $\sum_{i=1}^{m} f_i^2(\mathbf{x}^*)$. As can be observed in Figure 2, iMOACO$_\mathbb{R}$ shows promising performance in terms of convergence and distribution for the 3-objective DTLZ2. The $R2$-ranking algorithm is responsible of the obtained distribution, which is similar to that of NSGA-III and MOEA/D, because it tries to get the optimal solution for each subproblem defined by a weight vector. The HV values in Table 5 indicate that the performance of iMOACO$_\mathbb{R}$, for all dimensions, is very similar to that of MOEA/D and NSGA-III and it outperforms MOACO$_\mathbb{R}$ and SMS-EMOA in all cases. These results are also supported by IGD$^+$'s statistical results. Concerning

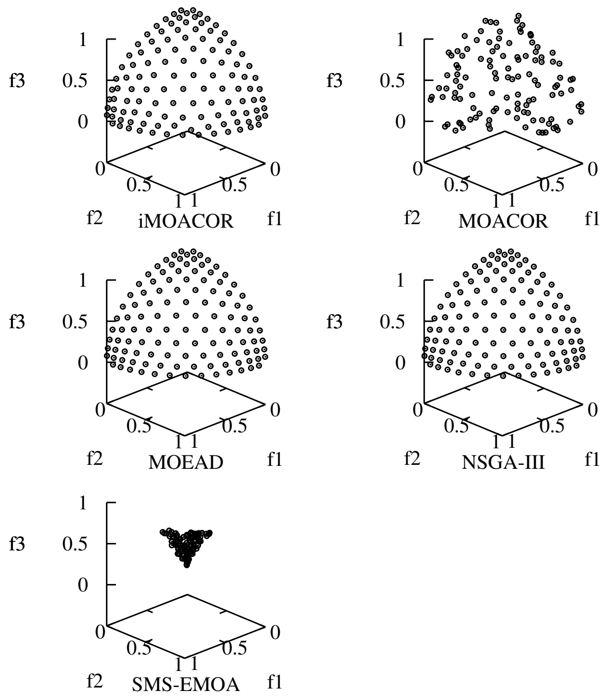

Fig. 2 Pareto fronts produced by iMOACO$_{\mathbb{R}}$, MOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on DTLZ2 with 3 objectives. All fronts correspond to the execution in the median of the HV.

the S indicator, iMOACO$_{\mathbb{R}}$ obtains the worst value as shown in Table 6. Regarding this fact, we must emphasize that the S indicator measures the variance of the distance between neighboring vectors, thus, it is evident that Pareto front approximations where the solutions are closer, will produce a better S value. In Figure 2, we can observe that SMS-EMOA's Pareto front only covers a small objective region and its vectors are very close from each other. Consequently, SMS-EMOA gets the best S value but offering a bad coverage. Similarly, MOACO$_{\mathbb{R}}$ is also benefited by the S indicator due to its bad distribution as seen in Figure 2.

DTLZ4 tests the optimizers' ability to maintain a good distribution of solutions because it introduces a strong search bias. From the IGD$^+$ values in Table 6, iMOACO$_{\mathbb{R}}$ presents a good distribution for 3 dimensions and, in the many-objective cases, it is outperformed by MOEA/D and NSGA-III. For the hypervolume, iMOACO$_{\mathbb{R}}$ gets the second best result for the 3- and 5-objective case, while MOEA/D is the best optimizer for this test instance. iMOACO$_{\mathbb{R}}$ is only outperformed by MOEA/D and NSGA-III. In all cases, iMOACO$_{\mathbb{R}}$ outperforms SMS-EMOA.

DTLZ5 results of a modification to DTLZ2 in order to generate a unifrontal and degenerated hypersurface. Based on Figure 3 and the IGD$^+$ values of Table 6, it can be claimed that iMOACO$_{\mathbb{R}}$ produces a good distribution of solutions. For 3-dimensional instances, MOACO$_{\mathbb{R}}$ produces a distribution of solutions that is better

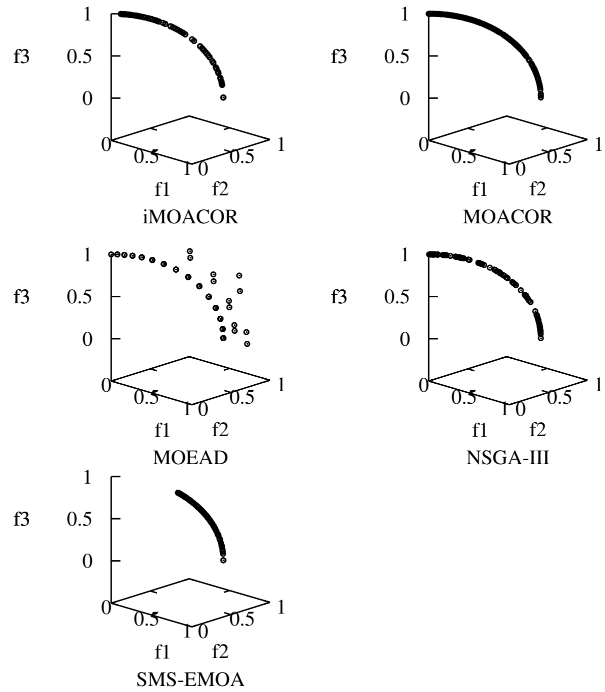

Fig. 3 Pareto fronts produced by iMOACO$_\mathbb{R}$, MOACO$_\mathbb{R}$, MOEA/D, NSGA-III and SMS-EMOA on DTLZ5 with 3 objectives. All fronts correspond to the execution in the median of the HV.

only than that generated by iMOACO$_\mathbb{R}$. However, our proposal obtaines the best S value in such 3-objectives instances. iMOACO$_\mathbb{R}$ obtains the second best value in 7 dimensions and the best one for 10 dimensions regarding IGD$^+$, which supports our previous claim about its capability to generate good distributions of solutions. Regarding the hypervolume, our algorithm only obtains the second best value for the 3-dimensional case while in the rest of the problems, it is outperformed by MOEA/D and NSGA-III.

DTLZ6 is a more complex version of DTLZ5, which results in a unifrontal, degenerated and biased problem. According to the IGD$^+$ results, iMOACO$_\mathbb{R}$ presents a good distribution and convergence for all dimensions, being only outperformed by MOEA/D in 7 and 10 dimensions and by MOACO$_\mathbb{R}$ in the 3-dimensional case. Regarding IGD$^+$, iMOACO$_\mathbb{R}$ statistically outperforms NSGA-III and SMS-EMOA. In terms of the hypervolume, our algorithm only obtains the second best value for 7 and 10 objectives. MOEA/D obtained the best results for the many-objective case while MOACO$_\mathbb{R}$ obtained the best results in low dimensionality. Finally, iMOACO$_\mathbb{R}$ had a better S value than MOEA/D and it is competitive with respect to NSGA-III.

DTLZ7 has a set of 2^{m-1} disconnected Pareto-optimal regions which tests the algorithms' capacity to maintain different subpopulations. The performance of iMOACO$_\mathbb{R}$ in terms of convergence is very competitive, obtaining the third place

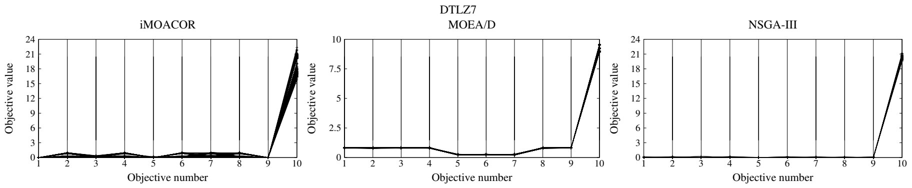

Fig. 4 Parallel coordinates of nondominated fronts produced by iMOACO$_{\mathbb{R}}$, MOEA/D and NSGA-III on the 10-objective DTLZ7. All fronts correspond to the execution in the median of the HV.

in 3 dimensions, the second place in 5 and 7 dimensions and the best value for the 10-objective instance of DTLZ7. It is worth noting that the difference in the 10-objective case is very significant with respect to the other algorithms. This result can be explained using the parallel coordinates plot[6] depicted in Figure 4. From the plot, we can observe that iMOACO$_{\mathbb{R}}$'s approximation has better spread of solutions than MOEA/D and NSGA-III because it covers a wide range of objective function values. Due to its better coverage, it can dominate a larger region of objective function space, which is reflected by the better HV value.

Summarizing, the performance of iMOACO$_{\mathbb{R}}$ is regular in this benchmark. First of all, iMOACO$_{\mathbb{R}}$ is unable to converge when dealing with highly multifrontal problems such as DTLZ1 and DTLZ3. iMOACO$_{\mathbb{R}}$' spread of solutions is quite similar to that of NSGA-III and MOEA/D due to the use of the $R2$-ranking algorithm. iMOACO$_{\mathbb{R}}$ is not severely affected by parametric bias as indicated by its results in DTLZ4. The use of our proposal is recommended when dealing with MOPs whose Pareto fronts are degenerated such as DTLZ5 and DTLZ6. Furthermore, iMOACO$_{\mathbb{R}}$ has shown its ability to maintain different subpopulations in order to solve MOPs with disconnected Pareto fronts such as DTLZ7.

6.2 Performance comparisons on the WFG test suite

WFG1 tests the optimizer's ability to tackle flat bias and mixed structure of the Pareto front (including concave and convex). These characteristics make WFG1 a very difficult MOP. Based on the HV results in Table 5, iMOACO$_{\mathbb{R}}$ is the best optimizer in high-dimensionality. Figures 5 and 6 show the parallel coordinates plots of WFG1 in 7 and 10 objectives, respectively. In both plots, we can observe that iMOACO$_{\mathbb{R}}$ produces a better spread of solutions and it covers more objective

[6] According to Deb (2001), the *parallel coordinates plot* (also called *value path plot*), is a graphical method to show the objective values of high-dimensional nondominated fronts. The horizontal axis marks the identity of the objective function, and thus, must be ticked only at integers starting from 1 to m and a bar is put in each tick. The vertical axis will mark the objective function values. The plot provides the following information:

1. Qualitative assessment of the spread of the obtained solutions. An algorithm which spreads its solutions over the entire objective value axis is considered to be good at finding diverse solutions.
2. The extent to which the cross-lines 'zig-zag' shows the trade-off among the objective functions captured by the obtained nondominated solutions. An algorithm having a large change of slope between two objective function bars is considered to be good in terms of finding good trade-off nondominated solutions.

Table 7 Statistical results (mean and standard deviation) of the S values obtained by iMOACO$_{\mathbb{R}}$, MOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on the DTLZ and WFG test suites. The two best values are shown in gray scale, where the darker tone corresponds to the best value. The symbol # is placed when a result is statistically different from iMOACO$_{\mathbb{R}}$'s result based on a one-tailed Wilcoxon test with a significance level of 5%. N/A (not applicable) means that the experiment was not executed. An ∞ value implies that the algorithm was unable to converge to the true Pareto front.

Problem	iMOACO$_{\mathbb{R}}$	MOACO$_{\mathbb{R}}$	MOEA/D	NSGA-III	SMS-EMOA
DTLZ1	∞	∞	5.665252e-04(2.658e-04)	9.639678e-03(1.597e-02)	4.668093e-03(3.088e-03)
DTLZ2	5.256836e-02(1.370e-03)	4.739753e-02(4.342e-03)	4.890725e-02(3.032e-05)	4.842788e-02(8.740e-04)	1.478099e-02(4.202e-03)
DTLZ3	∞	∞	5.521024e-02(3.130e-02)	1.316527e-01(8.429e-02)	7.033163e+00(1.058e+01)
DTLZ4	6.107899e-02(3.996e-03)	4.797831e-02(3.209e-03)	4.891094e-02(2.596e-05)	4.074251e-02(1.582e-02)	2.099415e-02(1.977e-02)
DTLZ5	4.268713e-03(3.265e-03)	7.027390e-03(5.236e-04)#	2.164089e-01(5.072e-03)#	1.239829e-02(1.881e-03)#	6.198692e-03(1.705e-03)#
DTLZ6	7.331618e-02(7.146e-03)	1.623979e-02(2.086e-02)	2.380922e-01(2.572e-03)#	6.772126e-02(4.153e-02)	1.313191e-02(3.313e-03)
DTLZ7	1.234456e-01(3.631e-02)	6.621065e-02(6.538e-03)	1.668798e-01(3.452e-02)#	6.030668e-02(4.979e-03)	4.503081e-02(1.319e-02)
WFG1	1.055275e-01(2.969e-02)	1.615551e-01(9.895e-03)#	8.217338e-02(1.171e-02)	6.741428e-02(1.741e-02)	4.239153e-02(9.209e-03)
WFG2	3.382511e-01(7.371e-02)	2.390982e-01(4.452e-02)	8.975647e-02(1.658e-02)	1.317904e-01(1.410e-02)	4.702230e-02(5.693e-02)
WFG3	1.323829e-01(1.711e-02)	1.172149e-01(1.024e-02)	1.940602e-01(2.327e-02)#	9.726909e-02(9.199e-03)	3.303524e-02(1.201e-02)
WFG4	3.039401e-01(1.250e-02)	1.971638e-01(1.690e-02)	2.330489e-01(2.605e-03)	1.884123e-01(4.541e-03)	5.557515e-02(6.816e-03)
WFG5	3.104586e-01(1.350e-02)	1.784433e-01(1.516e-02)	2.295599e-01(1.726e-03)	1.907428e-01(8.703e-03)	6.326510e-02(3.617e-02)
WFG6	3.103782e-01(9.858e-03)	1.892799e-01(1.339e-02)	2.355838e-01(5.291e-03)	1.875025e-01(4.058e-03)	6.161465e-02(1.452e-02)
WFG7	3.146193e-01(9.699e-03)	1.939828e-01(1.636e-02)	2.360618e-01(3.696e-03)	1.867679e-01(2.335e-03)	5.220852e-02(1.183e-02)
WFG8	3.055681e-01(7.712e-03)	1.843525e-01(1.700e-02)	2.654930e-01(5.887e-03)	2.031002e-01(1.030e-02)	6.640425e-02(1.162e-02)
WFG9	3.240877e-01(7.407e-03)	1.746757e-01(1.131e-02)	2.401901e-01(4.054e-03)	1.849227e-01(7.993e-03)	8.047164e-02(9.723e-02)

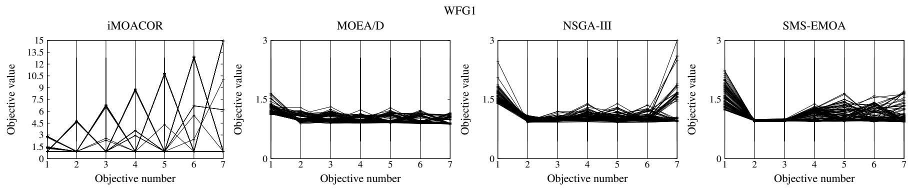

Fig. 5 Parallel coordinates of nondominated fronts produced by iMOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on the 7-objective WFG1. All fronts correspond to the execution in the median of the HV.

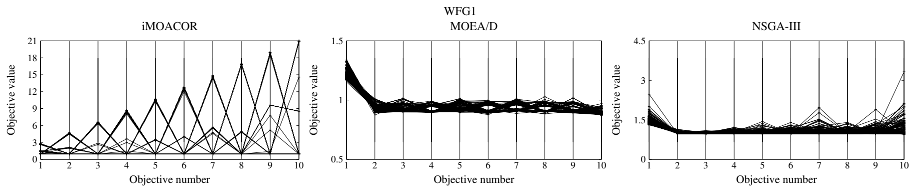

Fig. 6 Parallel coordinates of nondominated fronts produced by iMOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on the 10-objective WFG1. All fronts correspond to the execution in the median of the HV.

values unlike the other algorithms that produce solutions whose objective values are within a very small range. Additionally, we can observe that the cross-lines have a large change of slope which reflects better trade-offs among solutions. Regarding low dimensionality, iMOACO$_{\mathbb{R}}$ is outperformed by MOEA/D. Regarding IGD^{+}, Table 6 shows that MOEA/D presents the best results in all cases.

WFG2 is a separable and multifrontal MOP whose Pareto front is disconnected which implies that it intends to test the optimizer's ability to maintain subpopulations. iMOACO$_{\mathbb{R}}$ is very competitive in this problem as it obtained the best HV value in 5 and 7 objectives and the second best HV value for 3 and 10 objectives. Figure 7 reveals that in the 3-objective case, iMOACO$_{\mathbb{R}}$ generates more solutions around the knee of the Pareto front. Considering Figure 8, which shows the parallel coordinates plot of the 7-objective WFG2, it is possible to see that iMOACO$_{\mathbb{R}}$

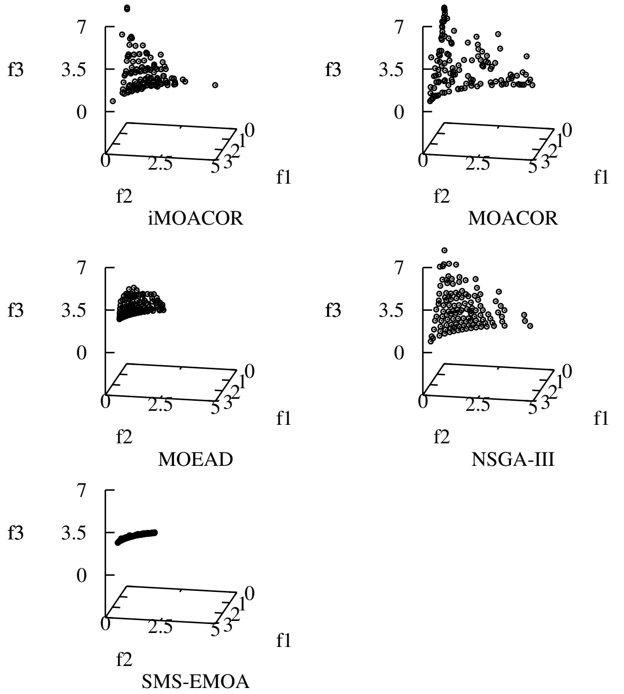

Fig. 7 Pareto fronts produced by iMOACO$_{\mathbb{R}}$, MOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on WFG2 with 3 objectives. All fronts correspond to the execution in the median of the HV.

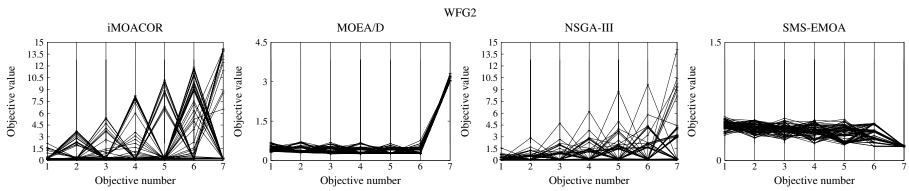

Fig. 8 Parallel coordinates of nondominated fronts produced by iMOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on the 7-objective WFG2. All fronts correspond to the execution in the median of the HV.

generates a better spread of solutions compared to the other algorithms, especially to NSGA-III. Regarding the IGD^{+} values in Table 6, iMOACO$_{\mathbb{R}}$ had a competitive performance with respect to NSGA-III. For this indicator, our algorithm obtained the best value in the 5-objective case and the second place in 7 and 10 objectives. In low dimensionality, the S indicator penalizes iMOACO$_{\mathbb{R}}$ because of the outliers that can be seen in Figure 7. Despite the multifrontality of this MOP, iMOACO$_{\mathbb{R}}$ had a very good performance in contrast to its performance in DTLZ1 and DTLZ3, which are both multifrontal MOPs.

WFG3 is separable and unifrontal but its main source of difficulty is that it has a linear degenerated Pareto front. In terms of convergence using the HV indi-

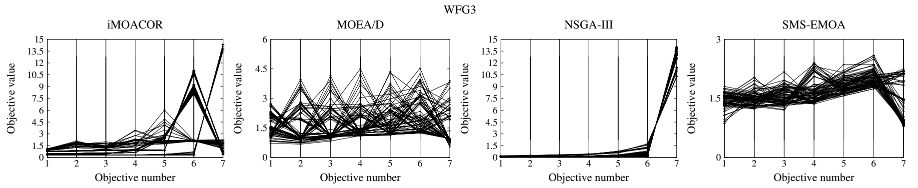

Fig. 9 Parallel coordinates of nondominated fronts produced by iMOACO$_\mathbb{R}$, MOEA/D, NSGA-III and SMS-EMOA on the 7-objective WFG3. All fronts correspond to the execution in the median of the HV.

cator, iMOACO$_\mathbb{R}$ had a competitive performance with respect to NSGA-III. Our algorithm obtained the best HV value in 7 dimensions of the objective space, and the second best value in the remaining considered dimensions. Figure 9 shows the parallel coordinates of the 7-objective WFG3. From this figure, it is evident that iMOACO$_\mathbb{R}$ had a better spread of solutions than the other algorithms. However, this is not reflected by the IGD$^+$ values in Table 6, where MOEA/D outperforms our proposed approach. Concerning the S values (Table 7), our results are competitive with respect to MOEA/D and NSGA-III.

WFG4 is a highly multifrontal MOP with a concave Pareto front. The performance of iMOACO$_\mathbb{R}$ is competitive compared to NSGA-III's performance in high-dimensionality considering the hypervolume. In these cases, iMOACO$_\mathbb{R}$ always gets the second best value. Unfortunately, the IGD$^+$ and S results indicate that the distribution of solutions of our approach is not very good because it is outperformed by MOEA/D and NSGA-III. However, it is worth noting that iMOACO$_\mathbb{R}$ converges in all cases in spite of the multifrontality of this problem.

WFG5 is similar to WFG4 except for the introduction of deceptiveness. Table 5 shows that NSGA-III obtains the best HV values and MOEA/D the second best values in all cases. iMOACO$_\mathbb{R}$ is able to outperform only MOACO$_\mathbb{R}$ and SMS-EMOA in all dimensions of the objective space. These results are confirmed by the IGD$^+$ values in Table 6 except for the 3-objective case, in which iMOACO$_\mathbb{R}$ obtained the second best value. Discarding SMS-EMOA for the S indicator, our algorithm gets the second place with respect to this performance measure.

WFG6 introduces a nonseparable and unifrontal MOP with a concave Pareto front. Figure 9 shows the approximation fronts produced by all algorithms for the 3-objecive instance of WFG6 where it is evident that MOEA/D and NSGA-III obtained the best distributions. iMOACO$_\mathbb{R}$ produced a good distribution on the knee of the front while the solutions towards the extremes of the front are not very well positioned. This fact directly affects the S value of our algorithm although it obtained the best IGD$^+$ value for the 3-objective case. Regarding the hypervolume, iMOACO$_\mathbb{R}$ outperformed all the other algorithms in low-dimensionality and it obtained the second best value in 5 and 7 objectives, outperforming MOEA/D and SMS-EMOA.

WFG7 is a separable and unifrontal MOP, but with parameter dependency whose Pareto front is concave. In terms of the hypervolume, iMOACO$_\mathbb{R}$ obtained the second best value in all cases, outperforming MOEA/D, MOACO$_\mathbb{R}$ and SMS-EMOA. The IGD$^+$ values indicate that in low-dimensionality our algorithm ranks second while in the remaining cases, iMOACO$_\mathbb{R}$ is able to outperform only SMS-EMOA.

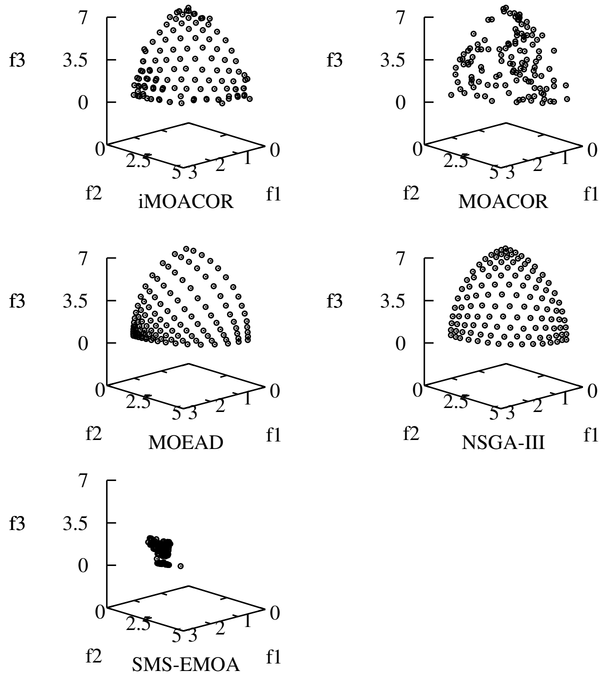

Fig. 10 Pareto fronts produced by iMOACO$_{\mathbb{R}}$, MOACO$_{\mathbb{R}}$, MOEA/D, NSGA-III and SMS-EMOA on WFG6 with 3 objectives. All fronts correspond to the execution in the median of the HV.

WFG8 presents a higher parameter dependency and it is also unifrontal. Considering the HV indicator (Table 5), iMOACO$_{\mathbb{R}}$ is able to outperform only MOACO$_{\mathbb{R}}$ and SMS-EMOA in the 3- and 5-objective instances of WFG8. For 7 and 10 dimensions of the objective space, iMOACO$_{\mathbb{R}}$ ranked second, outperforming MOEA/D and SMS-EMOA. NSGA-III showed the best results for this indicator in all cases. Regarding the IGD$^+$ values in Table 6, iMOACO$_{\mathbb{R}}$ was always outperformed by MOEA/D and NSGA-III. This means that the degree of convergence and spread of them is better than the one of our algorithm. The S value supports this claim for the 3-objective case.

WFG9 is a difficult problem because it presents nonseparability, multifrontality, deceptiveness and a parametric-bias. Despite these characteristics, iMOACO$_{\mathbb{R}}$ had a competitive performance in terms of HV. In fact, for the 3- and 10-objective case, our approach obtained the second best value, outperforming MOACO$_{\mathbb{R}}$, MOEA/D and SMS-EMOA for the former case and outperforming MOEA/D for the latter case. Considering 5 and 7 dimensions of the objective space, iMOACO$_{\mathbb{R}}$ is able to outperform only to SMS-EMOA. The IGD$^+$ statistical results indicate a regular performance of our approach where in the low-dimensional case it gets the second best value, just being outperformed by NSGA-III. In the rest of cases, it is able to outperform only to SMS-EMOA. The S value shows that iMOACO$_{\mathbb{R}}$ outperforms both MOEA/D and NSGA-III.

Table 8 Resulting combinations due to the different values for the parameters q and ξ.

Name	q	ξ
C1	0.0001	0.5
C2	0.0001	0.85
C3	0.1	0.5
C4	0.1	0.85
C5	0.25	0.5
C6	0.25	0.85
C7	1.0	0.5
C8	1.0	0.85

In summary, the performance of iMOACO$_{\mathbb{R}}$, considering all high-dimensional cases, is very competitive in this benchmark. Our approach is widely recommended to solve problems similar to WFG1 because in this test instance, it produced the best convergence results in high-dimensionality because of its good spread of solutions that no other algorithm could match. Furthermore, iMOACO$_{\mathbb{R}}$ shows a good performance on WFG2 and WFG3 where it always obtained the best or the second best HV value. For the case of WFG4 and WFG6 to WFG9, our algorithm had a competitive performance while in WFG5, it did not behave very well. In conclusion, we recommend our algorithm in order to tackle MOPs with a strong bias, regardless of separability and frontality and, especially, for problems with a Pareto front having a mixed structure: disconnected and degenerated.

6.3 Parameter sensitivity analysis

Regarding the parameter sensitivity of iMOACO$_{\mathbb{R}}$, we conducted a single-tail analysis of variance (ANOVA) (Scheffe, 1999). There are two major parameters in iMOACO$_{\mathbb{R}}$ which are part of the search engine: q that controls the diversification process and ξ that regulates the convergence rate. The main purpose of the ANOVA is to determine the effect of these two parameters in terms of hypervolume values. To study how iMOACO$_{\mathbb{R}}$ is sensitive to these two parameters, we tried to cover some special[7] values for each parameter: $q \in \{0.0001, 0.1, 0.25, 1.0\}$ and $\xi \in \{0.5, 0.85\}$. Based on these values, we formed eight combinations as shown in Table 8. We employed the 3-objective instances of DTLZ2, DTLZ7, WFG2 and WFG6 in order to compare the performance of all 8 different parameter combinations. Thirty independent runs were executed for each configuration on each test instance.

Figure 4 shows the median HV values obtained by these 8 different configurations on each selected test instance. In the light of these four plots, one can observe that for DTLZ2, WFG2 and WFG6, the combinations C1, C3, C5 and C7 produced the best results, while for DTLZ7, the combinations C2, C4, C6 and C8 are preferred. In the former case, it is clear that the HV value is dependent on ξ because in all odd combinations (which give the best results) $\xi = 0.5$. Furthermore, we can claim that in all these problems, the value of q does not have influence on the performance of iMOACO$_{\mathbb{R}}$. On the other hand, iMOACO$_{\mathbb{R}}$ has produced best results on DTLZ7 when $\xi = 0.85$, not affecting the value of q. Based

[7] The election of these values is based on the analysis of diversification versus intensification of the search, made by Socha and Dorigo (2008) for ACO$_{\mathbb{R}}$.

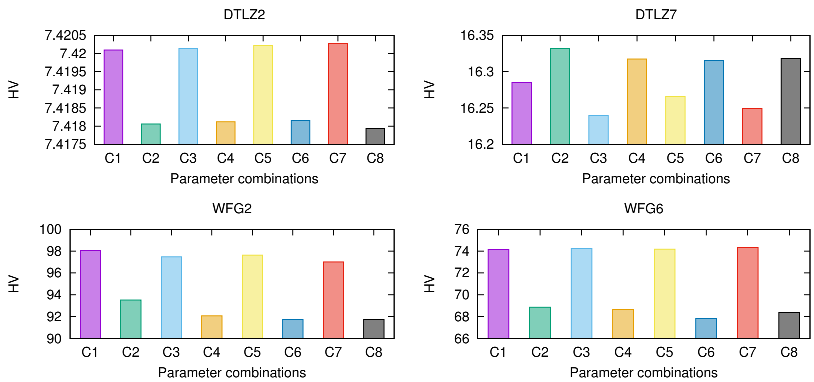

Fig. 11 Median HV values found by iMOACO$_{\mathbb{R}}$ with 8 different combinations of q and ξ on DTLZ2, DTLZ7, WFG2 and WFG6 with three objectives. The combinations C_i are given in Table 8.

on these results, we claim that iMOACO$_{\mathbb{R}}$ is only sensitive to the parameter ξ that controls the rate of convergence of the search engine. In Eq. 7, ξ has the ability to reduce or to increase the standard deviation value which directly affects the exploitation capacity of the underlying search engine. For ACO$_{\mathbb{R}}$, Socha and Dorigo (2008) proposed to set $\xi = 0.85$ in furtherance of a robust search for the single-objective case. However, we have found that a lower value of ξ is better for the multi-objective case. Additionally, we observed that our algorithm is not sensitive to the parameter q. In spite of this, we have adopted $q = 0.1$ with the aim of strengthening the diversification (or exploratory) capacity of iMOACO$_{\mathbb{R}}$. In conclusion, the results obtained by the sensitivity analysis based on ANOVA support the election of $q = 0.1$ and $\xi = 0.5$ for all the test instances for iMOACO$_{\mathbb{R}}$, as illustrated in Section 5.1.

7 Conclusions and future work

In this paper, we have proposed a new ACO-based many-objective optimizer for continuous search spaces, called iMOACO$_{\mathbb{R}}$. Our proposed approach uses ACO$_{\mathbb{R}}$ as its search engine and employs a ranking algorithm based on the $R2$ indicator in order to define which solutions are better than others. This allows our approach to tackle many-objective problems.

Our experimental results indicate that iMOACO$_{\mathbb{R}}$ has a competitive performance with respect to NSGA-III and MOEA/D and that it is able to outperform SMS-EMOA (using HypE's approximation scheme) and MOACO$_{\mathbb{R}}$ in most of the test problems adopted. Therefore, we consider that iMOACO$_{\mathbb{R}}$ is a good starting point for having a highly competitive many-objective optimizer based on ACO. However, one aspect that must be emphasized is the difficulty that iMOACO$_{\mathbb{R}}$ has on multifrontal problems such as DTLZ1 and DTLZ3. Our proposed approach has difficulties to maintain diversity in these problems and more work in this direction is still required.

Additionally, iMOACO$_{\mathbb{R}}$ presents problems to refine good solutions. For example, in Figure 2, the Pareto front approximation produced by iMOACO$_{\mathbb{R}}$ for

DTLZ2 is similar in distribution and convergence to those generated by NSGA-III and MOEA/D, however, it is outperformed by them according to HV, IGD^+ and S. The generating-solutions operators of iMOACO$_{\mathbb{R}}$, taken from ACO$_{\mathbb{R}}$, cannot exploit good solutions quickly. The critical part of the mechanism is the standard deviation (see Eq. (7)) associated with the normal distribution which is sampled in order to generate a new solution. This standard deviation represents the average error, for a certain dimension, of all decision variables with respect to one of them. This error is then weighted by ξ that controls the rate of convergence, i.e., the exploration range. On the basis of DTLZ2, we observed that at the end of the search, ξ is too large. In other words, iMOACO$_{\mathbb{R}}$ cannot explore areas nearby a good solution. Consequently, iMOACO$_{\mathbb{R}}$'s exploitation ability is significantly diminished, which is reflected on the HV values. This exploitation issue is inherent to ACO$_{\mathbb{R}}$. In fact, there have been a number of research efforts to extend ACO$_{\mathbb{R}}$ in order to improve its performance (Leguizamón and Coello, 2010; Liao et al., 2011).

As part of our future work, we are interested in studying different mechanisms that can enhance diversity, while maintaining the ACO metaphor.This can be done using a different selection mechanism based on other indicator or a combination of several of them, e.g., the ϵ-indicator, $R2$ and IGD^+ because they are weakly monotonic. Furthermore, an adaptive mechanism for the ξ parameter is also needed in order to validate our hypothesis indicated above. Additionally, the pheromone structure adopted in our proposed approach still has lots of room for improvement.

Acknowledgements The first author acknowledges support from CONACyT and CINVESTAV-IPN to pursue graduate studies in Computer Science. The second author gratefully acknowledges support from CONACyT project no. 221551. Both authors thank the Referees for their valuable comments.

References

Daniel Angus. Crowding Population-based Ant Colony Optimization for the Multi-objective Travelling Salesman Problem. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM'2007)*, pages 333–340, Honolulu, Hawaii, USA, April 2007a. IEEE Press.

Daniel Angus. Population-Based Ant Colony Optimisation for Multi-objective Function Optimisation. In Marcus Randall, Hussein A. Abbass, and Janet Wiles, editors, *Progress in Artificial Life, Third Australian Conference (ACAL'2007)*, pages 232–244. Springer. Lecture Notes in Computer Science, Vol. 4828, Gold Coast, Australia, 2007b.

Daniel Angus and Clinton Woodward. Multiple Objective Ant Colony Optimisation. *Swarm Intelligence*, 3(1):69–85, 2009.

H. Anton. *Elementary Linear Algebra*. John Wiley & Sons, 2010.

David L. Applegate, Robert E. Bixby, Vasek Chvatal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, USA, 2007.

Johannes Bader and Eckart Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring 2011.

Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

George Bilchev and Ian C. Parmee. The Ant Colony Metaphor for Searching Continuous Design Spaces. In *Selected Papers from AISB Workshop on Evolutionary Computing*, pages 25–39, London, UK, UK, 1995. Springer-Verlag.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA, 1999.

George EP Box and Mervin E Müller. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.

Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing Hypervolume Indicator Based Algorithms. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 651–660. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.

Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.

Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. R2 Indicator-Based Multiobjective Search. *Evolutionary Computation*, 23(3):369–395, Fall 2015.

Ling Chen, Jie Shen, Ling Qin, and Hongjian Chen. An improved ant colony algorithm in continuous optimization. *Journal of Systems Science and Systems Engineering*, 12(2):224–235, 2003.

Carlos A. Coello Coello and Nareli Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.

Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9(3):115–148, 1995.

Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014.

Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.

M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, Apr 1997.

Marco Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2):243–278, 2005.

Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Company, 2004.

Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1):29–41, February 1996.

J. Dréo and P. Siarry. Continuous Interacting Ant Colony Algorithm Based on Dense Heterarchy. *Future Generation Computer Systems*, 20(5):841–856, june 2004.

Jesús Guillermo Falcón-Cardona and Carlos A. Coello Coello. iMOACO$_{\mathbb{R}}$: A New Indicator-Based Multi-Objective Ant Colony Optimization Algorithm for Continuous Search Spaces. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV, 14th International Conference*, pages 389–398. Springer. Lecture Notes in Computer Science Vol. 9921, Edinburgh, UK, September 17-21 2016. ISBN 978-3-319-45822-9.

Abel Garcia-Najera and John A. Bullinaria. Extending ACO$_R$ to Solve Multi-Objective Problems. In G. M. Coghill, editor, *Proceedings of the UK Workshop on Computational Intelligence (UKCI 2007)*, London, UK, 2007. Imperial College United Kingdom.

Raquel Hernández Gómez and Carlos A. Coello Coello. MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the $R2$ Indicator. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 2488–2495, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

Raquel Hernández Gómez and Carlos A. Coello Coello. Improved Metaheuristic Based on the $R2$ Indicator for Many-Objective Optimization. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 679–686, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.

Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A Scalable Multi-objective Test Problem Toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 280–295, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Difficulties in Specifying Reference Points to Calculate the Inverted Generational Distance for Many-Objective Optimization Problems. In *2014 IEEE Symposium* *on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'2014)*, pages 170–177, Orlando, Florida, USA, 9-12 December 2014. IEEE Press. ISBN 978-1-4799-4467-5.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, sep 1999.

Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002. IEEE Service Center.

Joshua Knowles and David Corne. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 757–771, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

Min Kong and Peng Tian. A Direct Application of Ant Colony Optimization to Function Optimization Problem in Continuous Domain. In *Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS'06)*, pages 324–331, Berlin, Heidelberg, 2006. Springer-Verlag.

Guillermo Leguizamón and Carlos A. Coello Coello. An alternative aco$_{\mathbb{R}}$ algorithm for continuous optimization problems. In *Proceedings of the 7th International Conference on Swarm Intelligence*, ANTS'10, pages 48–59, Berlin, Heidelberg, 2010. Springer-Verlag.

Guillermo Leguizamón and Carlos A. Coello Coello. Multi-Objective Ant Colony Optimization: A Taxonomy and Review of Approaches. In Satchidanada Dehuri, Susmita Ghosh, and Sung Bae Cho, editors, *Integration of Swarm Intelligence and Artificial Neural Network*, chapter 3, pages 67–94. World Scientific, Singapore, 2011. ISBN 978-981-4280-14-3.

Tianjun Liao, Marco A. Montes de Oca, Dogan Aydin, Thomas Stützle, and Marco Dorigo. An Incremental Ant Colony Algorithm with Local Search for Continuous Optimization. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 125–132, New York, NY, USA, 2011. ACM.

Manuel Lopez-Ibanez and Thomas Stützle. The Automatic Design of Multiobjective Ant Colony Optimization Algorithms. *IEEE Transactions on Evolutionary Computation*, 16(6):861–875, December 2012a.

Manuel Lopez-Ibanez and Thomas Stützle. An experimental analysis of design choices of multi-objective ant colony optimization algorithms. *Swarm Intelligence*, 6(3):207–232, September 2012b.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publisher, 1999.

Dũng H. Phan and Junichi Suzuki. R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 1836–1845, Cancún, México, 20-23 June

2013. IEEE Press. ISBN 978-1-4799-0454-9.

Henry Scheffe. Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 20(2):344–360, 1958.

Henry Scheffe. The Simplex-Centroid Design for Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 25(2):235–263, 1963.

Henry Scheffe. *The Analysis of Variance*. Wiley, 1999.

Richard J. Brand Scott D. Chasalow. Algorithm as 299: Generation of simplex lattice points. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(4):534–545, 1995.

Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3):1155–1173, 2008.

Thomas Stützle and Holger H. Hoos. Max-min ant system. *Future Generation Computer Systems*, 16(9):889–914, june 2000.

Lyndon While, Lucas Bradstreet, and Luigi Barone. A Fast Way of Calculating Exact Hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1): 86–95, February 2012.

Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11 (6):712–731, December 2007.

Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.