iMOACO$_{\mathbb{R}}$: A New Indicator-Based Multi-objective Ant Colony Optimization Algorithm for Continuous Search Spaces

Jesús Guillermo Falcón-Cardona(✉) and Carlos A. Coello Coello

Computer Science Department, CINVESTAV-IPN (Evolutionary Computation Group), Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07300, Mexico
jfalcon@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx

Abstract. Ant colony optimization (ACO) is a metaheurisitc which was originally designed to solve combinatorial optimization problems. In recent years, ACO has been extended to tackle continuous single-objective optimization problems, being ACO$_{\mathbb{R}}$ one of the most remarkable approaches of this sort. However, there exist just a few ACO-based algorithms designed to solve continuous multi-objective optimization problems (MOPs) and none of them has been tested with many-objective problems (i.e., multi-objective problems having four or more objectives). 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. Our proposed approach is the first specifically designed to tackle many-objective optimization problems. Moreover, we present a comparative study of our proposal with respect to NSGA-III, MOEA/D, MOACO$_{\mathbb{R}}$ and SMS-EMOA using standard test problems and performance indicators adopted in the specialized literature. Our preliminary results indicate that iMOACO$_{\mathbb{R}}$ is very competitive with respect to state-of-the-art multi-objective evolutionary algorithms and is also able to outperform MOACO$_{\mathbb{R}}$.

1 Introduction

In artificial intelligence, the social behavior of animals and insects has been a prominent source of inspiration for several metaheuristics which are part of the broad concept of Swarm Intelligence. Ant Colony Optimization (ACO), was originally proposed by Dorigo [1], and it 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. The ants tend to take, with a higher probability, those paths where there is a larger amount of pheromone. In fact, after some time, the shortest path is the one with the largest amount of pheromone [2]. Due to this property, ACO was originally applied to the solution of combinatorial optimization problems (COPs).

J.G. Falcón-Cardona—acknowledges support from CINVESTAV-IPN.
C.A. Coello Coello—acknowledges support from CONACyT project no. 221551.

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 389–398, 2016.
DOI: 10.1007/978-3-319-45823-6_36

Over the years, the ACO metaheuristic has been extended to continuous search spaces, being the proposal of Bilchev and Parmee [3] the first of this sort. According to [4], there are several ACO-based optimizers for continuous domains although the ACO algorithm for continuous domains ($ACO_{\mathbb{R}}$) [5] is possibly the most remarkable. In spite of the relatively large amount of ACO-based algorithms currently available for continuous domains, there are just a few oriented to solve multi-objective optimization problems (MOPs). In [4] only two proposals are reported: the Population-based ACO Algorithm for Multi-Objective Function Optimization (PACO-MOFO) [6] and the Multi-Objective Ant Colony Optimizer ($MOACO_{\mathbb{R}}$) [7], both based on $ACO_{\mathbb{R}}$. Furthermore, in the specialized literature no multi-objective ant colony optimizer (MOACO) had been reported so far as being able to solve many-objective problems [8].

In this paper, we propose a novel indicator-based Multi-Objective Ant Colony Optimizer based on $ACO_{\mathbb{R}}$, called $iMOACO_{\mathbb{R}}$. To the authors' best knowledge, this is the first MOACO algorithm that is able to tackle many-objective optimization problems.

The remainder of this paper is organized as follows. Section 2 presents an overview of the previous work on ACO in continuous optimization problems. Section 3 briefly describes $ACO_{\mathbb{R}}$. The detailed description of our proposal is presented in Sect. 4. Then, we provide our experimental results in Sect. 5. Finally, Sect. 6 provides our conclusions and some possible paths for future research.

2 Previous Related Work

The first ACO algorithm designed for continuous search spaces was proposed by Bilchev and Parmee [3]. In this approach, each ant incrementally explores the search space from a single nest, defined as a promising point, trying different search directions at a radius not greater than R. At choosing a search direction, each ant's decision was biased by a trail quantity which was incremented if and only if the direction resulted in an improvement of the objective function; otherwise, the search direction was not taken into account. This process was repeated until a termination condition was met.

Socha and Dorigo proposed the $ACO_{\mathbb{R}}$ [5] algorithm whose fundamental idea is the use of a continuous probability density function (PDF) instead of a discrete one as in traditional ACOs. $ACO_{\mathbb{R}}$ uses a constant-size archive as its pheromone model where the best-so-far solutions are stored. For each dimension, a Gaussian-kernel PDF is defined using the corresponding elements of every stored solution. An ant incrementally constructs a new solution via the sampling of each Gaussian-kernel PDF. Once all ants have constructed a new solution, only the best ones are kept in the archive and the same number are removed from it. A detailed description of $ACO_{\mathbb{R}}$ will be provided in the next section.

The use of ACO in continuous MOPs has been scarcely explored [4]. We are only aware of two approaches. The first of them is PACO-MOFO, which is based on the Crowding Population-based ACO algorithm (CPACO) [9] and $ACO_{\mathbb{R}}$. PACO-MOFO applies a replacement operator based on crowding distance in order to maintain diversity and fitness sharing in furtherance of a uniform sampling of the objective space. The second proposal is MOACO$_\mathbb{R}$ [7], which is a direct extension of ACO$_\mathbb{R}$. The concept of *dominance depth* of NSGA-II [10] is used in this case to preserve at each iteration those solutions closer to the Pareto Front. Moreover, if the number of solutions exceed the size of the archive, those with a higher crowding distance value are removed in order to maintain constant the size of the archive.

3 ACO$_\mathbb{R}$ Overview

The pheromone model of ACO$_\mathbb{R}$ [5] is represented by an archive $\mathcal{T}$ that stores the k best-so-far solutions. For the i^{th} dimension, a Gaussian-kernel PDF is defined using the corresponding components of all stored solutions as follows:

$$G^i(x) = \sum_{j=1}^{k} w_j g_j^i(x) = \sum_{j=1}^{k} w_j \frac{1}{\sigma_j^i \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_j^i)^2}{2{\sigma_j^i}^2}} \tag{1}$$

where $i = 1, \ldots, n$ and n is the number of decision variables. Each archive's solution j stores a vector of decision variables $s_j = (s_j^1, \ldots, s_j^n)$, an objective value $u(s_j)$ and the weight ω_j. The solutions are sorted by their quality, i.e., $u(s_1) \leq u(s_2) \leq \cdots \leq u(s_k)$, for a minimization problem.

Equation (1) depends on three vectors of parameters: μ^i is the vector of means, σ^i is the vector of standard deviations, and ω is the vector of weights. The vector of means μ^i is defined as follows:

$$\mu^i = \{\mu_1^i, \ldots, \mu_k^i\} = \{s_1^i, \ldots, s_k^i\} \tag{2}$$

The elements of σ^i have to be independently calculated for each Gaussian-kernel using the following formula:

$$\sigma_j^i = \xi \sum_{r=1}^{k} \frac{|s_r^i - s_j^i|}{k-1} \tag{3}$$

where $\xi > 0$ is a parameter of the algorithm that controls the way the long term memory is used, i.e., the speed of convergence. When ξ is large, the speed of convergence is slower and in case its value is close to zero, the speed of convergence is increased. Finally, each $\omega_j \in \omega$ is calculated as follows:

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} \cdot e^{-\frac{(rank(s_j)-1)^2}{2q^2k^2}} \tag{4}$$

where $rank(\cdot)$ returns the solution's rank in $\mathcal{T}$ according to the established order and $q > 0$ is a parameter that controls the diversification of the search. As $q \to 0$, the best-ranked solutions are preferred to guide the search, and when it takes a large value, the weights tend to be more uniform.

In order to generate a new solution, first, each ant $a_i \in \mathcal{A}$ chooses, with probability $p_j = \omega_j / \sum_{r=1}^{k} \omega_r$, a guiding pheromone s_j from $\mathcal{T}$. Then, a_i samples $g_j^i(x)$, $i = 1, \ldots, n$, with the purpose of creating a new solution.

4 Our Proposed Approach

The hypervolume (HV) [18] and the $R2$ indicator [11] are two recommended unary performance indicators which simultaneously evaluate all the desired aspects of a Pareto Front approximation [11]. However, the $R2$ indicator requires less computational effort and it produces a more uniform distribution than HV. Given a Pareto Front approximation A, the unary version of the $R2$ indicator [11] is defined as follows:

$$R2(A, U) = \frac{1}{|U|} \sum_{u \in U} \min_{a \in A} \{u(a)\} \tag{5}$$

where U is a set of utility functions $u : \mathbb{R}^m \rightarrow \mathbb{R}$ 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, Hernández and Coello proposed in [13] a ranking algorithm based on it, called $R2$-ranking. This mechanism groups solutions which optimize a set of utility functions, and place 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.

Concerning the choice of the utility function u in Eq. (5), we use the achievement scalarizing function (ASF) [12] defined as:

$$u_{asf}(v \mid r, \lambda) = \max_{i \in \{1,\ldots,m\}} \frac{1}{\lambda_i} |v_i - r_i| \tag{6}$$

where r is a reference vector and λ is a convex weight vector, both of dimension m. The set $U = \{\lambda^i \mid i = 1, \ldots, N\}$ ($N = C_{m-1}^{H+m-1}$, H is a parameter of the algorithm) is computed using Simple-Lattice-Design (SLD). Moreover, we normalize each objective function $f_i(x)$ (the $R2$-ranking algorithm requieres this normalization) using the following formula:

$$f_i'(x) = \frac{f_i(x) - z_i^{min}}{z_i^{max} - z_i^{min}}, \forall i \in \{1, \ldots, m\} \tag{7}$$

where z^{min} and z^{max} are statistical approximations to the ideal and nadir vectors [12], respectively. These vectors are updated using a data structure called *RECORD*, which was proposed by Hernández and Coello [13].

When we deal with MOPs there is not a unique solution but a set of solutions which represent the best possible trade-offs among the objectives. Due to this fact, $ACO_{\mathbb{R}}$'s pheromone model has to be slightly modified in order to store the best solutions according to some criterion. A Pareto-based scheme is not a good choice if we aim to solve many-objective problems. Thus, we propose to use the $R2$-ranking algorithm because of its good performance in many-objective problems.

Each record of the archive $\mathcal{T}$ stores the same information as in ACO$_\mathbb{R}$, although in this case the objective value is treated as an objective vector $F(s_i)$. Additionally, it is added a field $rank(s_i)$. Once the solutions in $\mathcal{T}$ have been processed by the $R2$-ranking, the rank assigned to each solution s_j is stored in $rank(s_j)$. In order to create a new solution, we applied the standard process of ACO$_\mathbb{R}$ using Eqs. (1) to (4).

The underlying idea of the pheromone update is to promote a competition between the newly created solutions $\mathcal{A}$ and the pheromones in $\mathcal{T}$. Let $\Psi = \mathcal{A} \cup \mathcal{T}$. The union set is ranked by the $R2$-ranking and is immediately sorted, in increasing order, by the following criteria: (1) rank, (2) utility value, (3) L_2-norm. Finally, all pheromones in $\mathcal{T}$ are substituted by the first k solutions of Ψ.

In Algorithm 1, we describe our proposed iMOACO$_\mathbb{R}$.[1] The algorithm only requires three parameters: (1) the set of $N = C_{m-1}^{H+m-1}$ weight vectors, (2) the diversification parameter q, and (3) the convergence speed factor, ξ. The population size (M) and the archive size (k) are equal to N due to the optimal μ-distributions of the $R2$ indicator [11]. In lines 1 to 3, k random solutions are generated to initialize $\mathcal{T}$ and the RECORD structure is created. At each iteration, the $R2$-ranking is applied on $\mathcal{T}$ and afterwards every ant generates a new solution. Then, in line 8, the RECORD is updated using the ants' solutions with the aim of producing new values of z^{min} and z^{max}. From lines 9 to 13, the pheromone update is performed. This process is repeated until a termination condition is fulfilled and then the solutions in $\mathcal{T}$ are returned in line 14.

Algorithm 1. Main loop of iMOACO$_\mathbb{R}$.

Require: MOP, set of $N = C_{m-1}^{H+m-1}$ convex weight vectors, q, ξ
Ensure: Pareto front approximation
1: Randomly initialize archive $\mathcal{T}$
2: Initialize RECORD R
3: Initialize z^{min} and z^{max}
4: **while** termination condition is not fulfilled **do**
5:     $\mathcal{T} \leftarrow R2ranking(\mathcal{T})$
6:     **for all** $ant \in \mathcal{A}$ **do**
7:         $generateSolution(ant, z^{min}, z^{max})$
8:     Update reference points (z^{min}, z^{max}) using R
9:     $\Psi \leftarrow \mathcal{A} \cup \mathcal{T}$
10:     $\Psi' \leftarrow R2ranking(\Psi)$
11:     Remove all elements from $\mathcal{T}$
12:     $\Psi' \leftarrow sortReduction(\Psi')$
13:     Copy the first k elements from Ψ' to $\mathcal{T}$
14: **return** $\mathcal{T}$

[1] The source code of our approach is available at: http://computacion.cs.cinvestav.mx/~jfalcon/iMOACOR/imoacor.html.

5 Experimental Results

In order to assess the performance of our proposed approach, we used the Zitzler-Deb-Thiele (ZDT) test suite, the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite, and the Walking-Fish-Group (WFG) test suite. However, due to space limitations, only the results for the ZDT and DTLZ test problems are included here. Our proposed approach was compared with respect to: NSGA-III[2] [15], MOEA/D [14], SMS-EMOA [16] (using HypE to estimate the hypervolume values [17]) and MOACO$_{\mathbb{R}}$[3] [7]. Results were compared using the hypervolume (HV), inverted generational distance plus (IGD+[4]) [19] and spacing (S) [18].

Attending the original papers, the common parameter settings for NSGA-III, MOEA/D and SMS-EMOA have been set as follows: $N_c = 20$, $P_c = 1.0$, $N_m = 20$ and $P_m = 1/n$. The neighborhood size of MOEA/D was set to 20. The number of samples in the HypE algorithm was set to 10,000. Based on an experimental study, the parameters (q, ξ) of iMOACO$_{\mathbb{R}}$ and MOACO$_{\mathbb{R}}$ were set as $(0.1, 0.5)$ for low and high dimensionality. In all cases, we performed a maximum number of 50,000 function evaluations. We used $N = 120$ weight vectors, which implies $H = 119$ and $H = 14$, for two and three dimensions, respectively.

For the scalability test, we employed DTLZ2 from four to nine objectives. All parameter values remained the same except for $N_c = 30$, as suggested in [15]. The maximum number of function evaluations remained the same as before. The experimental configurations $(m, N(H))$ are as follows: (4, 120(7)), (5, 126(5)), (6, 126(4)), (7, 84(3)), (8, 120(3)), (9, 165(3)) and (10, 220(3)).

5.1 Discussion of Results

This section compares iMOACO$_{\mathbb{R}}$ with three state-of-the-art MOEAs and a MOACO that was designed for continuous MOPs. The comparison is performed in terms of convergence and diversity of the solutions obtained. We perform 30 independent runs of each of the 5 algorithms on all the test instances adopted. Tables 1 and 2 show the average HV, IGD+ and S values, as well as the standard deviations (shown in parentheses) obtained by all the algorithms compared. The two best values among the algorithms are emphasized in gray scale, where the darker tone corresponds to the best value. A sharp symbol (#) is placed when a result is statistically different from iMOACO$_{\mathbb{R}}$'s result based on a single-tail Wilcoxon test (WT) using a significance level of 95 %.

Table 1 shows that NSGA-III yields the best HV results in three of four of the ZDT test problems and that iMOACO$_{\mathbb{R}}$ is the best in one of them. However, iMOACO$_{\mathbb{R}}$ obtained the second best HV value in the problems where NSGA-III wins. Moreover, iMOACO$_{\mathbb{R}}$ outperformed MOEA/D, SMS-EMOA (HypE)

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

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

[4] For each problem, the reference set is constructed joining the results from all algorithms and then applying the k-means clustering algorithm in order to reduce its cardinality to k.

and MOACO$_{\mathbb{R}}$ in all the ZDT problems and the differences are statistically significant. With respect to IGD+, iMOACO$_{\mathbb{R}}$ obtained the second place in 50 % of the problems and outperformed MOACO$_{\mathbb{R}}$ and SMS-EMOA(HypE) in 75 % of the problems (the differences were statistically significant).

Table 1. Comparison of iMOACO$_{\mathbb{R}}$ with respect to SMS-EMOA, MOEA/D, NSGA-III and MOACO$_{\mathbb{R}}$ in the ZDT test problems with two objectives. The symbol # is placed when the difference with respect to iMOACO$_{\mathbb{R}}$'s result is statistically significant, based on Wilcoxon's test. The two best values are shown in gray scale, where the darker tone corresponds to the best value. NC stands for a not computable result due to an algorithm's error.

Problem	Algorithm	HV	IGD+	S
ZDT1	iMOACO$_{\mathbb{R}}$	120.650592(0.002695)	0.006982(0.000262)	0.022642(0.000638)
	MOACO$_{\mathbb{R}}$	120.647992(0.001834)#	0.004849(0.000213)	0.005066(0.000462)
	SMS-EMOA	115.056548(0.204942)#	0.110323(0.006696)#	0.002271(0.002064)
	MOEA/D	120.556524(0.029634)#	0.002429(0.000194)	0.004075(0.000219)
	NSGA-III	120.662065(0.000361)	0.001986(0.000024)	0.008623(0.000175)
ZDT2	iMOACO$_{\mathbb{R}}$	120.319499(0.002065)	0.004579(0.000105)	0.022031(0.000424)
	MOACO$_{\mathbb{R}}$	NC	NC	NC
	SMS-EMOA	111.557974(0.298732)#	0.155868(0.009460)#	0.002802(0.000375)
	MOEA/D	120.303458(0.009442)#	0.002511(0.000320)	0.003736(0.000130)
	NSGA-III	120.328489(0.000541)	0.001919(0.000008)	0.003460(0.000067)
ZDT3	iMOACO$_{\mathbb{R}}$	128.746630(0.006519)	0.002151(0.000116)	0.017216(0.000907)
	MOACO$_{\mathbb{R}}$	128.718532(0.008340)#	0.003121(0.000213)#	0.005641(0.000496)
	SMS-EMOA	125.892282(1.903359)#	0.032616(0.025295)#	0.028416(0.019151)#
	MOEA/D	128.214272(0.946685)#	0.005800(0.006714)#	0.014828(0.001110)
	NSGA-III	128.774980(0.000181)	0.001413(0.000068)	0.011243(0.000922)
ZDT6	iMOACO$_{\mathbb{R}}$	117.381093(0.023285)	0.013011(0.001239)	0.023835(0.001014)
	MOACO$_{\mathbb{R}}$	NC	NC	NC
	SMS-EMOA	113.246727(1.377160)#	0.217514(0.070758)#	0.008383(0.016202)
	MOEA/D	116.763019(0.071014)#	0.050276(0.005115)#	0.003248(0.000604)
	NSGA-III	116.418956(0.002804)#	0.003126(0.000080)	0.001221(0.000018)

Table 2 shows the HV, IGD+ and S values in the DTLZ test problems with 3 objectives. MOACO$_{\mathbb{R}}$ obtained the best HV results in 40 % of the problems. iMOACO$_{\mathbb{R}}$ and NSGA-III performed similarly in HV with only one best value and in 40 % of the problems it ranked second. Moreover, iMOACO$_{\mathbb{R}}$ outperformed NSGA-III, MOEA/D and MOACO$_{\mathbb{R}}$ in 40 % of the problems and outperformed SMS-EMOA(HypE) in a statistically significant way, in all problems. On the other hand, both MOACO$_{\mathbb{R}}$ and MOEA/D outperformed, in terms of IGD+, the rest of the algorithms in 40 % of the problems in a statistically significant way. Finally, it is worth emphasizing that iMOACO$_{\mathbb{R}}$ obtained the best results in DTLZ6 for every indicator and outperformed the other MOEAs in a statistically significant way. However, iMOACO$_{\mathbb{R}}$ could not outperform MOACO$_{\mathbb{R}}$ in a statistically significant way.

It is worth indicating that, although SMS-EMOA(HypE) obtained the best S values and iMOACO$_{\mathbb{R}}$ the worst, we observed that the solutions obtained by SMS-EMOA(HypE) are not well spread and they tend to concentrate on a small region of objective function space. This is not reflected in the S values, because the solutions are all generated in the same small region. In contrast, iMOACO$_{\mathbb{R}}$ provides a better coverage along the Pareto front, but presents a non-uniform distribution in some cases, which is the explanation for its poor values.

Table 2. Comparison of iMOACO$_{\mathbb{R}}$ with respect to SMS-EMOA, MOEA/D, NSGA-III and MOACO$_{\mathbb{R}}$ in the DTLZ test suite with three objectives. The symbol # is placed when the difference with respect to iMOACO$_{\mathbb{R}}$'s result is statistically significant, based on Wilcoxon's test. The two best values are shown in gray scale, where a darker tone corresponds to the best value.

Problem	Algorithm	HV	IGD+	S
DTLZ2	iMOACO$_{\mathbb{R}}$	7.420386(0.000218)	0.020631(0.000148)	0.051706(0.000954)
	MOACO$_{\mathbb{R}}$	7.396275(0.005367)#	0.027855(0.001570)#	0.049300(0.004648)
	SMS-EMOA	4.096654(0.078739)#	0.327737(0.006341)#	0.015061(0.004850)
	MOEA/D	7.421695(0.000110)	0.019927(0.000004)	0.048915(0.000023)
	NSGA-III	7.421721(0.000480)	0.020182(0.000256)	0.048387(0.000899)
DTLZ4	iMOACO$_{\mathbb{R}}$	7.419849(0.000499)	0.031649(0.000353)	0.059475(0.003305)
	MOACO$_{\mathbb{R}}$	7.397087(0.004471)	0.037138(0.001348)	0.047430(0.004064)
	SMS-EMOA	4.540085(0.510681)#	0.244637(0.072285)#	0.020506(0.023548)
	MOEA/D	7.421583(0.000095)	0.029946(0.000007)	0.048923(0.000022)
	NSGA-III	7.219506(0.405047)	0.066563(0.073020)	0.040964(0.015405)
DTLZ5	iMOACO$_{\mathbb{R}}$	59.838732(0.006907)	0.002099(0.000280)	0.004906(0.003023)
	MOACO$_{\mathbb{R}}$	59.868424(0.001271)#	0.001168(0.000229)#	0.007250(0.000658)#
	SMS-EMOA	50.323056(0.664565)#	0.001539(0.000362)	0.006650(0.003698)
	MOEA/D	59.734700(0.001057)#	0.004707(0.000008)#	0.220014(0.005494)#
	NSGA-III	59.831769(0.008471)#	0.001708(0.000447)	0.011428(0.001740)#
DTLZ6	iMOACO$_{\mathbb{R}}$	1318.921707(0.019110)	0.006341(0.000682)	0.007476(0.003907)
	MOACO$_{\mathbb{R}}$	1315.603646(18.281311)	0.067062(0.338421)	0.028337(0.103803)
	SMS-EMOA	1179.647918(18.251463)#	0.260869(0.044008)#	0.012619(0.001664)#
	MOEA/D	1317.080995(0.438458)#	0.117785(0.030957)#	0.241082(0.002048)#
	NSGA-III	1317.572393(0.378312)#	0.081541(0.023995)#	0.060883(0.039139)#
DTLZ7	iMOACO$_{\mathbb{R}}$	1.481848(0.128161)	0.208907(0.064111)	0.110911(0.042411)
	MOACO$_{\mathbb{R}}$	1.955759(0.012511)	0.033899(0.002521)	0.066972(0.006164)
	SMS-EMOA	1.481553(0.151019)	0.192244(0.050467)	0.046280(0.016713)
	MOEA/D	1.827781(0.211815)	0.096353(0.150199)	0.166670(0.034483)#
	NSGA-III	1.937277(0.011787)	0.037603(0.001834)	0.058137(0.004748)

Table 3. Comparison of iMOACO$_{\mathbb{R}}$ with respect to three MOEAs in DTLZ2 with four to nine objectives. The symbol # is placed when the difference with respect to iMOACO$_{\mathbb{R}}$'s result is statistically significant, based on Wilcoxon's test. The two best values are shown in gray scale, where the darker tone corresponds to the best value.

Problem	Algorithm	HV	IGD+
DTLZ2 4D	iMOACO$_{\mathbb{R}}$	15.560885(0.000752)	0.050234(0.004783)
	SMS-EMOA	10.249730(0.677489)#	0.291283(0.009656)#
	MOEA/D	15.567068(0.000241)	0.037633(0.000019)
	NSGA-III	15.566456(0.000668)	0.038679(0.000926)
DTLZ2 5D	iMOACO$_{\mathbb{R}}$	31.650513(0.001900)	0.079425(0.004944)
	SMS-EMOA	21.358261(0.676573)#	0.363571(0.006165)#
	MOEA/D	31.667626(0.000250)	0.057564(0.000070)
	NSGA-III	31.665300(0.000589)	0.059863(0.000627)
DTLZ2 6D	iMOACO$_{\mathbb{R}}$	63.714682(0.002503)	0.091338(0.008945)
	SMS-EMOA	47.221717(1.575810)#	0.395902(0.009846)#
	MOEA/D	63.738154(0.000667)	0.048382(0.000030)
	NSGA-III	63.737999(0.001056)	0.051075(0.000791)
DTLZ2 7D	iMOACO$_{\mathbb{R}}$	127.695926(0.008977)	0.147653(0.008183)
	SMS-EMOA	82.448331(3.777842)#	0.501245(0.006396)#
	MOEA/D	127.747411(0.001454)	0.088905(0.000024)
	NSGA-III	127.749053(0.001358)	0.092568(0.001253)
DTLZ2 8D	iMOACO$_{\mathbb{R}}$	255.731810(0.060472)	0.166126(0.010043)
	SMS-EMOA	184.360111(8.860506)#	0.530043(0.005534)#
	MOEA/D	255.819317(0.001518)	0.093164(0.000159)
	NSGA-III	255.815238(0.001521)	0.099347(0.001020)
DTLZ2 9D	iMOACO$_{\mathbb{R}}$	511.711415(0.161806)	0.177908(0.009877)
	SMS-EMOA	414.480972(10.937714)#	0.548778(0.006587)#
	MOEA/D	511.866089(0.003034)	0.087360(0.000135)
	NSGA-III	511.870831(0.001247)	0.092867(0.001409)

Regarding our scalability test, in Table 3 we provide the HV and IGD+ values in DTLZ2 having from four to nine objectives. Clearly, MOEA/D and NSGA-III present better HV and IGD+ results than iMOACO$_{\mathbb{R}}$. However, the maximum observed difference, in relation to HV, is of order 10^{-1}, which is not very significant. iMOACO$_{\mathbb{R}}$ outperforms SMS-EMOA (HypE) in 100 % of the cases.

6 Conclusions and Future Work

In this paper, we have proposed a new ACO-based multi-objective optimizer for continuous search spaces, called iMOACO$_{\mathbb{R}}$. Our 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 the others. This allows our approach to tackle many-objective problems.

Our experimental results indicate that iMOACO$_{\mathbb{R}}$ had a competitive performance with respect to NSGA-III and MOEA/D and that is able to outperform SMS-EMOA (HypE) 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 multi-objective optimizer based on ACO. However, one aspect that must be emphasized is the difficulty that iMOACO$_{\mathbb{R}}$ has on multi-frontal problems such as ZDT4, DTLZ1 and DTLZ3. Our proposed approach has difficulties to maintain diversity in these problems and more work in this direction is still required.

It is worth noticing that the solutions produced by iMOACO$_{\mathbb{R}}$ are similar to those generated by NSGA-III and MOEA/D in terms of distribution and it also achieves a competitive performance in terms of convergence. Furthermore, our proposed approach requires much less computational effort than SMS-EMOA.

As part of our future work, we are interested in studying different diversity mechanisms that allow us to maintain the biological metaphor of the ACO algorithm. Additionally, the pheromone structure still has a lot of room for improvement. Finally, we also aim to improve the performance of our approach in many-objective problems.

References

1. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano, Italy (1992)
2. Dorigo, M., Stuetzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
3. Bilchev, G., Parmee, I.C.: The ant colony metaphor for searching continuous design spaces. In: Fogarty, T.C. (ed.) Evolutionary Computing. LNCS, vol. 993, pp. 232–244. Springer, Heidelgberg. (1995)
4. Leguizamón, G., Coello, C.A.C.: Multi-objective ant colony optimization: a taxonomy and review of approaches. In: Integration of Swarm Intelligence and Artificial, Neural Networks, pp. 67–94 (2011)
5. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. Eur. J. Oper. Res. **185**(3), 1155–1173 (2008)
6. Angus, D.: Population-based ant colony optimisation for multi-objective function optimisation. In: Randall, M., Abbass, H.A., Wiles, J. (eds.) ACAL 2007. LNCS (LNAI), vol. 4828, pp. 232–244. Springer, Heidelberg (2007)

7. Garcia-Najera, A., Bullinaria, J.A.: Extending $ACO_{\mathbb{R}}$ to solve multi-objective problems. In: Proceedings of the UK Workshop on Computational Intelligence (UKCI 2007), London, UK (2007)
8. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: a short review. In: IEEE Congress on Evolutionary Computation (2008)
9. Angus, D.: 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), pp. 333–340. IEEE Press, Honolulu (2007)
10. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
11. Brockhoff, D., Wagner, T., Trautmann, H.: On the properties of the R2 indicator. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, pp. 465–472. ACM (2012)
12. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publisher, Boston (1999)
13. Hernández Gómez, R., Coello, C.A.C.: Improved metaheuristic based on the $R2$ indicator for many-objective optimization. In: Silva, S. (ed.) Proceedings of the 2015 Anual Conference on Genetic and Evolutionary Computation, pp. 679–686. ACM, Madrid (2015)
14. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
15. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
16. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur. J. Oper. Res. **181**(3), 1653–1669 (2007)
17. Bader, J., Zitzler, E.: HypE: an algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. **19**(1), 45–76 (2011)
18. Coello, C.A.C., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-objective Problems, vol. 242. New York Kluwer Academic, New York (2002)
19. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. In: IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making (MCDM), pp. 170–177. IEEE (2014)