An Extension of the iMOACO$_{\mathbb{R}}$ Algorithm Based on Layer-Set Selection

Ashraf M. Abdelbar[1][0000-0002-7921-1892], Thomas Humphries[2][0000-0003-0844-6771], Jesús Guillermo Falcón-Cardona[3][0000-0003-1131-098X], and Carlos A. Coello Coello[4][0000-0002-8435-680X]

[1] Department of Mathematics & Computer Science, Brandon University, Brandon, MB, Canada abdelbara@brandonu.ca
[2] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada thomas.humphries@uwaterloo.ca
[3] Tecnológico de Monterrey, School of Engineering and Sciences, Monterrey, Nuevo León, Mexico jfalcon@tec.mx
[4] CINVESTAV-IPN, Computer Science Department, Mexico City, Mexico ccoello@cs.cinvestav.mx

Abstract. iMOACO$_{\mathbb{R}}$ is an ant colony optimization algorithm designed to tackle multi-objective optimization problems in continuous search spaces. It is built on top of ACO$_{\mathbb{R}}$ and uses the $R2$ indicator (to improve its performance on high-dimensional objective function spaces) to rank the pheromone archive of the best previously-explored solutions. Due to the utilization of an $R2$-based selection mechanism, there are typically a large number of tied-ranks in iMOACO$_{\mathbb{R}}$'s pheromone archive. It is worth noting that the solutions of a specific layer share the same importance based on the $R2$ indicator. A critical issue due to the large number of tied-ranks is a reduction of the algorithm's exploitation ability. In consequence, in this paper, we propose replacing iMOACO$_{\mathbb{R}}$'s probabilistic solution selection mechanism with a mechanism tailored to these layer-sets. Our proposed layer-set selection uses rank-proportionate (roulette wheel) selection to select a layer, with all the solutions in the layer sharing equally in the layer's probability. Our experimental evaluation indicates that our proposal, which we call iMOACO$'_{\mathbb{R}}$, performs better than iMOACO$_{\mathbb{R}}$ to a statistically significant extent on a large number of benchmark problems having from 3 to 10 objective functions.

1 Overview

Multi-objective optimization problems (MOP) [3,16] are a class of problems that require the simultaneous optimization of multiple objective functions which are mutually conflicting. Due to this conflict, the solution of a MOP is composed of a set of solutions that represent the best possible trade-offs among the objective functions. The bio-inspired metaheuristics are promising techniques to solve MOPs. Among these techniques, those based on the behavior of colonies of ants have recently attracted the attention of the community to solve continuous

MOPs. iMOACO$_{\mathbb{R}}$ [7] is an ant colony algorithm [6] for multi-objective optimization in continuous spaces, and is designed specifically for problems with four or more objective functions. iMOACO$_{\mathbb{R}}$ uses the $R2$ indicator [2] to rank solutions, and is built on top of ACO$_{\mathbb{R}}$ [13], a well-established ant colony algorithm for continuous-domain optimization.

As a consequence of using an $R2$-based selection mechanism, there are typically a large number of tied-ranks in iMOACO$_{\mathbb{R}}$'s archived population. A tie would require the fitness function values for two archived solution to be exactly the same, which is unlikely to happen very often in a typical real optimization problem with a real-valued objective function. For this reason, the handling of tied ranks is not a very important issue for most single-objective optimization applications of ACO$_{\mathbb{R}}$, and does not seem to have received much attention in the literature. However, in iMOACO$_{\mathbb{R}}$, the archive is typically made up of a number of layers, with the set of solutions at each layer having the same $R2$ indicator value. The number of distinct layers can be much smaller than the population size. For example, we have found that in a population of 220 solutions, it is not uncommon for the number of layer-sets (distinct ranks) to be no more than 50 for most of the computation. The problem is that tied ranks smooth out the probability distribution used for selection. This results in an algorithm with lower exploitation than the same algorithm with a uniquely ranked population.

In this paper, we propose replacing iMOACO$_{\mathbb{R}}$'s probabilistic solution selection mechanism with a mechanism tailored to layer-sets. Our proposed layer-set selection mechanism uses roulette wheel selection to select a layer, with all the solutions in the layer sharing equally in the layer's probability. We evaluate our proposal with respect to standard iMOACO$_{\mathbb{R}}$ using the same suite of problems and experimental settings adopted in [7]. Our results indicate that our proposal, which we call iMOACO$'_{\mathbb{R}}$, performs better than standard iMOACO$_{\mathbb{R}}$ to a statistically significant extent in several state-of-the-art benchmark problems, with the number of objective functions varying from 3 to 10.

2 Background

The unconstrained multi-objective optimization problem is mathematically defined as follows: $\min_{x\in\Omega} f(x) := [f_1(x), f_2(x), \ldots, f_m(x)]^T$, where $x \in \Omega$ is an n-dimensional vector of decision variables and $\Omega \subseteq \mathbb{R}^n$ is the decision space. $f_i : \Omega \to \mathbb{R}$, $i = 1, \ldots, m$ are the objective functions. When solving a MOP, the aim is to find in Ω a subset of solutions x^* that yield the optimum values for all the objective functions (i.e., the particular set that represents the best possible trade-offs among the objective functions). In furtherance of determining which solutions are optimal, the most common binary order relation used in multi-objective optimization is the Pareto dominance relation. Given two vectors of decision variables $x, y \in \Omega$, we say that x **dominates** y (denoted by $x \prec y$) if $f_i(x) \leq f_i(y)$ for $i = 1, \ldots, m$ and there exists at least an index $j \in \{1, \ldots, m\}$ such that $f_j(x) < f_j(y)$. Based on the Pareto dominance relation, we say that a vector of decision variables $x^* \in \Omega$ is **Pareto optimal** if there does not exist another $x \in \Omega$ such that $x \prec x^*$. The set that contains all the Pareto optimal solutions is known as the **Pareto Optimal Set** and its image in the objective functions space is known as the **Pareto Front**.

In order to assess the performance of MOEAs, a wide variety of quality indicators (QIs) have been proposed in the specialized literature [17]. Among the plethora of available QIs, the most relevant are those that assess the convergence of a Pareto front approximation to the true Pareto front $\mathcal{PF}^*$. One of these QIs is the discrete unary $R2$ indicator [2] that assesses the convergence of an approximation set $\mathcal{A}$ (containing a finite set of objective vectors that approximate $\mathcal{PF}^*$), using scalarizing functions. The discrete unary $R2$ indicator is defined as follows:

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

where W is a set of m-dimensional convex weight vectors and $u_w : \mathbb{R}^m \to \mathbb{R}$ is a scalarizing function, parameterized by $w \in W$, that assigns a real value to each objective vector in $\mathcal{A}$.

3 The iMOACO$_\mathbb{R}$ Algorithm

In 2017, Falcón-Cardona and Coello Coello proposed the indicator-based many-objective ant colony optimizer for continuous search spaces (iMOACO$_\mathbb{R}$) [7] which is based on the ACO$_\mathbb{R}$ [13–15] search engine. The most important element of every ACO-based algorithm is the design of the pheromone matrix since it stores knowledge throughout the search process to solve the optimization problem [6]. The pheromone matrix of ACO$_\mathbb{R}$ is an archive that stores the best N solutions found so far and it sorts them according to the quality of the objective function. However, this scheme cannot be directly implemented in iMOACO$_\mathbb{R}$ since the Pareto dominance relation does not establish a total order. Hence, Falcón-Cardona and Coello Coello proposed to use the $R2$ indicator [2] to transform the multi-objective problem into a single-objective one and, thus, imposing a total order. For this purpose, the $R2$-ranking algorithm [10] was employed to rank the population in a similar fashion to the nondominated sorting algorithm [5] and, then, the best N solutions are stored according to the rank assigned. For each solution $x^j, j = 1, \ldots, N$, the auxiliary fields store its vector of objective values, the rank assigned and a weight value ω_j.

For each solution x^j in the archive, let r_j denote the rank of x^j. At each iteration, the weights $\omega_j, j = 1, \ldots, N$ are computed using the following formula:

$$\omega_j = \gamma(r_j - 1; 0, qN) \tag{2}$$

where $q > 0$ is a parameter that controls the diversification process of the search, r_j denotes the rank of archived solution x^j where a rank of 1 denotes the best solution, and $\gamma(a; b, c) = \frac{1}{c\sqrt{2\pi}} e^{-\frac{(a-b)^2}{2c^2}}$ denotes the Gaussian function.

To create new solutions, all k^{th} components of the N solutions are employed to define a Gaussian-kernel probability density function $G^k(y) = \sum_{j=1}^{N} \omega_j g_j^k(y) =$

$\sum_{j=1}^{N} \omega_j \gamma(y; \mu_k^j, \sigma_k^j)$, where $k = 1, \ldots, n$, and $G^k(y)$ depends on three parameter vectors: ω is the vector of weights associated with the individual Gaussian functions, μ_k is the vector of means, and σ_k is the vector of standard deviations. $\mu_k = \{\mu_k^1, \mu_k^2 \ldots, \mu_k^N\} = \{x_k^1, x_k^2, \ldots, x_k^N\}$, and each $\sigma_k^j \in \sigma_k$ is computed as follows: $\sigma_k^j = \xi \sum_{l=1}^{N} \frac{|x_k^l - x_k^j|}{N-1}$, where $\xi > 0$ is a parameter that controls the convergence rate, simulating the evaporation of pheromones.

After computing the weights, each of the M ants performs n construction steps to create a new solution x^{new}, where each component x_k^{new} is drawn by sampling the b^{th} Gaussian function that is part of G^k. The index $b \in \{1, \ldots, N\}$ is selected with probability $\Pr(\text{select } b) = \frac{\omega_b}{\sum_{l=1}^{N} \omega_l}$. Finally, the M newly created solutions compete with the ones in the pheromone matrix to be part of the pheromone matrix in the next iteration.

4 Our Proposed Approach

In typical continuous-domain single-objective optimization applications of $\text{ACO}_{\mathbb{R}}$, tied ranks in the archive are usually quite rare. For this reason, the handling of tied ranks is not a very important issue for most single-objective optimization applications of $\text{ACO}_{\mathbb{R}}$, and does not seem to have received much attention in the literature. But, in $\text{iMOACO}_{\mathbb{R}}$, there will typically be many tied ranks since any set of solutions with the same $R2$ value will have the same rank. The solution archive can be thought of as being made up of a number of layers, where each layer consists of a set of solutions that are tied for the same rank—and thus, have the same value of ω_r and the same probability of selection.

We recorded the number of distinct ranks in the population at each iteration for a single run of $\text{iMOACO}_{\mathbb{R}}$ on the `DTLZ5` problem instance with 10 objectives, using the experimental settings described in Section 5, and used this data to construct the plot shown in Fig. 1. In this figure, the x-axis represents the iteration number and the y-axis represents the number of distinct ranks in the population for that iteration. The figure indicates that for most of the computation, the number of distinct ranks is around 50 (in a population of size 220). Hence, this

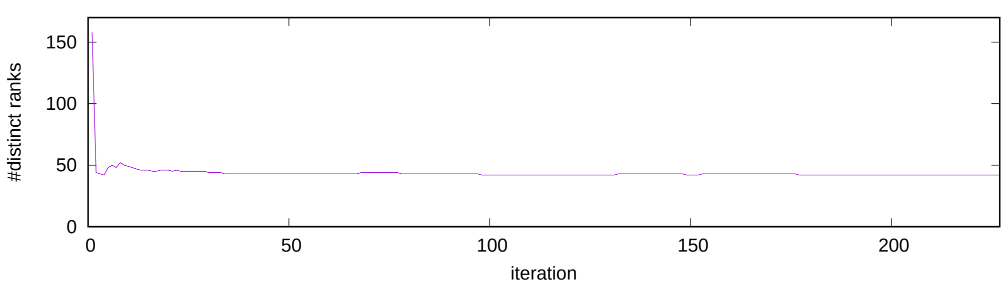

Fig. 1. Plot of the number of distinct ranks (y-axis) versus iteration number (x-axis), in a single run of $\text{iMOACO}_{\mathbb{R}}$ on the `DTLZ5` problem instance with (10 objectives).

is a clear drawback of iMOACO$_{\mathbb{R}}$ that results in a decrease of its exploitation ability.

We propose iMOACO$'_{\mathbb{R}}$, a variation in which the ACO$_{\mathbb{R}}$'s rank-proportionate selection mechanism is applied at the level of the layers rather than at the level of individual solutions, with all the solutions in a given layer sharing the probability of selection of their layer. Specifically, we propose replacing Eq. (2) with the following:

$$\omega_j = \frac{\gamma(r_j - 1, 0, cqR)}{N_{r_j}} \tag{3}$$

where R is the number of distinct ranks in the population, N_r is the number of solutions tied for rank r, and c is an additional parameter that is needed to compensate for the fact that a value of q that is appropriate for standard iMOACO$_{\mathbb{R}}$ may not be the most appropriate for the modified iMOACO$'_{\mathbb{R}}$. (We use a value of $c = 2$.)

Eq. (3) computes the weight ω_j of a solution of rank b (i.e., $r_j = b$). Let us assume that there is a set of solutions of size N_b tied for rank b. All solutions in that set will have equal weight. That weight is determined first by calculating the weight of selection of the set (which is the numerator of the formula in Eq. (3)), then dividing that weight by the size of the set (the denominator N_b).

Consider the following numerical example. Suppose we have a population of 30 solutions, consisting of: 6 solutions tied for rank 1, 4 tied for rank 2, 7 for rank 3, 6 for rank 4, and 7 for rank 5. Table 1 compares selection probabilities under iMOACO$_{\mathbb{R}}$ and iMOACO$'_{\mathbb{R}}$ for this population. Each row corresponds to a rank layer-set. The first two columns show the rank and the number of solutions in that rank-set. The next two columns show the individual probability of selection of each of the solutions at that rank. The last two columns show the overall probability that one of the solutions in that layer-set will be selected. The table indicates that the probability of selection of layer 1 is much higher under iMOACO$'_{\mathbb{R}}$, the probability of selection of layer 2 is similar, and the probability of subsequent layers is much smaller under iMOACO$'_{\mathbb{R}}$ and drops rapidly as k increases.

The rapid decline, under iMOACO$'_{\mathbb{R}}$, of the probability of selection of a layer k, as k increases, is not specific to the given numerical example. In general, if we define define z_k as the ratio of the probability of selection of layer k to the

Table 1. Numerical example showing a population of 30 solutions.

layer	#sols	prob. of sol.		prob. of layer	
		iMOACO$_{\mathbb{R}}$	iMOACO$'_{\mathbb{R}}$	iMOACO$_{\mathbb{R}}$	iMOACO$'_{\mathbb{R}}$
1	6	0.055	0.134	0.327	0.805
2	4	0.046	0.044	0.185	0.180
3	7	0.034	0.002	0.245	0.015
4	6	0.024	7.4E-05	0.142	4.5E-04
5	7	0.014	7.1E-07	0.101	4.9E-06

probability of selection of layer 1, then it is possible to obtain

$$z_k = \frac{\gamma(k-1, 0, cqR)}{\gamma(0, 0, cqR)} = e^{-\frac{k-1}{cqR}} \tag{4}$$

indicating that z_k decays exponentially with k. This is consistent with the spirit of $\text{ACO}_{\mathbb{R}}$. If we assume that ties in a typical single-objective application of $\text{ACO}_{\mathbb{R}}$ are negligibly rare, and define z_k for $\text{ACO}_{\mathbb{R}}$ as the ratio of the probability of selection of the k^{th} best solution in the archive to the probability of selection of the best solution in the archive, then it is possible to obtain $z_k = e^{-\frac{k-1}{qN}}$ indicating that z_k also decays exponentially with k in $\text{ACO}_{\mathbb{R}}$.

In terms of Holland's classical exploitation-exploration trade-off [11], $\text{iMOACO}'_{\mathbb{R}}$ is more exploitative (in the same spirit as $\text{ACO}_{\mathbb{R}}$) than $\text{iMOACO}_{\mathbb{R}}$.

5 Experimental Methodology and Discussion of Results

Our experimental methodology is based on that of Falcón-Cardona and Coello Coello [7]. We used the test suites Deb-Thiele-Laumanns-Zitzler (DTLZ) [4] and Walking-Fish-Group (WFG) [12]. For each problem, we set the number of objective functions (m) to 3, 5, 7, and 10. With $m = 3$, we set the population size N to 120, the maximum number of generations G_{max} to 416, and h to 14; with $m = 5$, we set: $N = 126, G_{max} = 396$, and $h = 5$; with $m = 7$: we set $N = 85, G_{max} = 595$, and $h = 7$; with 10 objectives, we set: $N = 220, G_{max} = 227$, and $h = 19$. Moreover, we set $q = 0.1$, $\xi = \alpha = 0.5$, and $\epsilon = 0.5$. For each instance, we performed 30 independent runs of each of $\text{iMOACO}_{\mathbb{R}}$ and $\text{iMOACO}'_{\mathbb{R}}$.

In our comparison, performance is assessed with the hypervolume (HV) indicator [1]. We used the HV implementation of [8], available in [9]. Computing the HV requires that a reference vector be supplied by the user. This was set to $(1, 1, \dots)$ for `DTLZ1`, $(2, 2, \dots)$ for `DTLZ2` and `DTLZ4`, $(7, 7, \dots)$ for `DTLZ3`, $(4, 4, \dots)$ for `DTLZ5`, $(11, 11, \dots)$ for `DTLZ6`, $(1, 1, \dots, 21)$ for `DTLZ7`, and $(3, 5, 7, \dots, 2m + 1)$ for all WFG problems. Occasionally, particularly for `DTLZ1` and `DTLZ3`, the reference vector dominates all the solutions returned by the algorithm under evaluation; in such cases, HV is taken as zero.

We ran $\text{iMOACO}_{\mathbb{R}}$ and $\text{iMOACO}'_{\mathbb{R}}$ for 30 independent trials on each of the 64 problem instances in our test suite, and computed the value of the hypervolume (HV) indicator in each case. Table 2 reports the mean and standard deviation of HV for each algorithm for each problem instance. In each row, the better mean HV value is underlined.

The table indicates that $\text{iMOACO}'_{\mathbb{R}}$ had better performance on 36 instances, and worse on 20 instances, with 8 ties. Considering the 28 `DTLZ` instances alone: $\text{iMOACO}'_{\mathbb{R}}$ had 13 wins, 7 losses, and 8 ties; for the 36 WFG instances: $\text{iMOACO}'_{\mathbb{R}}$ had 23 wins, 13 losses, and 0 ties. For the 3-objective instances alone: $\text{iMOACO}'_{\mathbb{R}}$ had 8 wins, 6 losses, and 2 ties; for the 5-objective instances: 10 wins, 4 losses, and 2 ties; for the 7-objective instances: 9 wins, 5 losses, and 2 ties; for the 10-objective: 9 wins, 5 losses, and 2 ties. Thus, $\text{iMOACO}'_{\mathbb{R}}$ performs better on each of these subgroups of the test suite.

Table 2. The mean and standard deviation of HV for the original iMOACO$_{\mathbb{R}}$ and our proposed modified iMOACO$'_{\mathbb{R}}$.

prob.	m	mean		std. dev.		prob.	m	mean		std. dev.	
		mod.	orig.	mod.	orig.			mod.	orig.	mod.	orig.
DTLZ1	3	0	0	0	0	WFG2	3	9.786e1	9.744e1	7.9e-1	5.5e-1
	5	0	0	0	0		5	9.947e3	9.707e3	1.0e2	9.1e1
	7	0	0	0	0		7	1.742e6	1.694e6	3.3e4	2.8e4
	10	0	0	0	0		10	9.882e9	9.467e9	2.3e8	1.3e8
DTLZ2	3	7.420	7.420	2.5e-4	3.1e-4	WFG3	3	7.256e1	7.245e1	2.6e-1	2.5e-1
	5	3.165e1	3.165e1	2.6e-3	2.0e-3		5	5.202e3	5.391e3	2.8e2	2.5e2
	7	1.277e2	1.272e2	9.8e-3	1.4		7	7.793e5	7.839e5	2.8e4	2.0e4
	10	1.023e3	1.014e3	4.7e-1	3.0e1		10	4.688e9	4.800e9	1.1e8	2.4e8
DTLZ3	3	0	0	0	0	WFG4	3	7.060e1	7.067e1	3.5e-1	3.5e-1
	5	0	0	0	0		5	7.611e3	7.615e3	1.7e2	1.6e2
	7	0	0	0	0		7	1.271e6	1.242e6	4.6e4	5.1e4
	10	0	0	0	0		10	7.437e9	7.219e9	3.2e8	3.1e8
DTLZ4	3	7.419	7.419	1.1e-3	9.2e-4	WFG5	3	6.847e1	6.831e1	7.0e-1	7.1e-1
	5	3.164e1	3.163e1	4.4e-3	5.2e-3		5	4.838e3	4.786e3	2.0e2	1.7e2
	7	1.277e2	1.265e2	7.1e-3	4.2		7	6.784e5	6.938e5	3.6e4	3.4e4
	10	1.024e3	1.003e3	4.0e-3	4.2e1		10	4.271e9	4.326e9	1.9e8	2.1e8
DTLZ5	3	5.984e1	5.984e1	1.0e-2	7.9e-3	WFG6	3	7.425e1	7.414e1	4.7e-1	3.6e-1
	5	9.379e2	9.374e2	1.5	9.1e-1		5	7.201e3	6.674e3	5.9e2	3.4e2
	7	1.434e4	1.438e4	9.3e1	1.1e2		7	8.779e5	8.466e5	7.3e4	7.4e4
	10	9.291e5	9.362e5	4.7e3	6.3e3		10	4.847e9	4.796e9	3.4e8	2.4e8
DTLZ6	3	1.318e3	1.316e3	3.8e-1	1.3	WFG7	3	7.545e1	7.522e1	2.6e-1	2.6e-1
	5	1.562e5	1.568e5	1.6e3	1.0e3		5	7.419e3	7.214e3	2.4e2	2.7e2
	7	1.783e7	1.734e7	3.2e5	1.9e6		7	1.074e6	1.086e6	6.3e4	6.0e4
	10	2.425e10	2.386e10	3.7e8	1.4e9		10	6.903e9	6.961e9	3.4e8	2.7e8
DTLZ7	3	1.624e1	1.625e1	1.0e-1	6.0e-2	WFG8	3	6.547e1	6.541e1	5.1e-1	3.0e-1
	5	1.259e1	1.256e1	1.2e-1	1.1e-1		5	5.272e3	5.158e3	3.2e2	2.7e2
	7	8.278	8.239	1.5e-1	1.9e-1		7	7.571e5	7.797e5	8.2e4	6.8e4
	10	2.414	1.464	1.8e-1	7.5e-1		10	5.037e9	5.181e9	5.7e8	4.1e8
WFG1	3	4.420e1	4.417e1	8.8e-1	7.0e-1	WFG9	3	6.594e1	6.594e1	2.5e-1	1.8e-1
	5	3.973e3	3.923e3	1.3e2	9.2e1		5	5.828e3	5.851e3	4.4e2	4.0e2
	7	6.776e5	6.693e5	4.0e4	3.3e4		7	7.405e5	7.114e5	1.3e5	1.0e5
	10	3.992e9	3.969e9	3.8e7	2.1e7		10	4.400e9	4.162e9	5.0e8	3.5e8

A one-tailed Wilcoxon signed-rank test applied to the results of Table 2 produced a p-value of 0.031, indicating a statistically significant difference.

Finally, we note that our proposed layer-set selection mechanism can generally be applied to other situations where ACO$_{\mathbb{R}}$ is used in an application with a non-negligible frequency of tied-ranks.

Acknowledgements. The last author acknowledges support from CONACyT project no. 1920.

References

1. Brockhoff, D., Friedrich, T., Neumann, F.: Analyzing hypervolume indicator based algorithms. In: Proceedings Parallel Problem Solving from Nature (PPSN-2008). pp. 651–660 (2008)
2. Brockhoff, D., Wagner, T., Trautmann, H.: On the properties of the R2 indicator. In: Proceedings 2012 Genetic and Evolutionary Computation Conference (GECCO-2012). pp. 465–472. ACM Press, Philadelphia, PA, USA (July 2012)
3. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer (2007)
4. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings 2002 IEEE Congress on Evolutionary Computation (CEC'2002). vol. 1, pp. 825–830. IEEE Press, Piscataway, NJ, USA (2002)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (April 2002)
6. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, MA, USA (2004)
7. Falcón-Cardona, J.G., Coello Coello, C.A.: A new indicator-based many-objective ant colony optimizer for continuous search spaces. Swarm Intelligence **11**, 71–100 (2017)
8. Foncesca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proceedings 2006 IEEE Congress on Evolutionary Computation (CEC-2006). pp. 1157–1163. IEEE Press, Piscataway, NJ, USA (2016)
9. Fonsesca, C.M., López-Ibáñez, M., Paquete, L., Guerreiro, A.P.: Computation of the hypervolume indicator. `http://iridia.ulb.ac.be/manuel/hypervolume`, accessed: May 2017
10. Hernández Gómez, R., Coello Coello, C.A.: Improved metaheuristic based on the R2 indicator for many-objective optimization. In: Proceedings 2015 Genetic and Evolutionary Computation Conference (GECCO-2015). pp. 679–686. ACM Press, Madrid, Spain (July 2015)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. University of Michigan Press, Ann Arbor, MI, USA (1975)
12. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**, 477–506 (2006)
13. Liao, T., Socha, K., Montes de Oca, M., Stützle, T., Dorigo, M.: Ant colony optimization for mixed-variable optimization problems. IEEE Transactions on Evolutionary Computation **18**, 503–518 (2014)
14. Socha, K., Blum, C.: An ant colony optimization algorithm for continuous optimization: Application to feed-forward neural network training. Neural Computing & Applications **16**, 235–247 (2007)
15. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. European Journal of Operational Research **185**, 1155–1173 (2008)
16. Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K.C., Jin, Y.: Evolutionary large-scale multi-objective optimization: A survey. ACM Computing Surveys **54**(8), 1–34 (November 2021)

17. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (April 2003)