Extending $ACO_{\mathbb{R}}$ to Solve Multi-Objective Problems

Abel Garcia-Najera
School of Computer Science
University of Birmingham
Birmingham, B15 2TT
UK
A.G.Najera@cs.bham.ac.uk

John A. Bullinaria
School of Computer Science
University of Birmingham
Birmignahm, B15 2TT
UK
J.A.Bullinaria@cs.bham.ac.uk

Abstract

Ant Colony Optimization (ACO) was first proposed to solve the Traveling Salesman Problem, and later applied to solve more problems of a combinatorial nature. Some research based on ACO to tackle continuous problems has been published, but this has not followed the original ACO metaheuristic exactly. Recently, $ACO_{\mathbb{R}}$ has been proposed to solve continuous function optimization problems. We have taken this work and extended it to solve multi-objective optimization problems. After an analysis of the results obtained, including comparisons with two other well-known methods, we conclude that $ACO_{\mathbb{R}}$ is a promising new technique for solving multi-objective problems.

1 Introduction

The Ant Colony Optimization (ACO) metaheuristic was first proposed to solve combinatorial problems like the traveling salesman problem [8, 6, 7], vehicle routing [1, 2, 9] and scheduling [15], among others. Even though ACO has been used widely to solve combinatorial problems efficiently, its use on continuous function problems has been limited due to the fact that there is not a straightforward extension.

Nevertheless, some methods based on ACO have been proposed to tackle continuous problems [14, 12, 13], but these have not followed the original metaheuristic exactly [11].

Recently, a new state-of-the-art technique has been proposed, extending ACO to continuous domains without the need to make any major conceptual change to its structure. This has been called $ACO_{\mathbb{R}}$ [11]. In that work, the authors deal with single-objective problems, comparing their solutions with other previously published results. These solutions suggest that $ACO_{\mathbb{R}}$ might usefully be extended to perform well on multi-objective problems too.

Multi-objective problems can be found in most engineering domains in the real world, and most of the time the various objectives are inconsistent. This means that, when we try to optimize one objective, we will probably not optimize them all. So, instead of looking for optimized parameter values to give an optimal solution, we must search for values that provide appropriate trade-offs.

In this paper we describe how we have extended the $ACO_{\mathbb{R}}$ approach to deal with multi-objective problems, and provide a visual analysis of comparative results obtained on some standard benchmark problems.

The remainder of this paper is organized as follows. In section 2 we review the standard ACO metaheuristic. A brief description of the $ACO_{\mathbb{R}}$ approach is given in section 3. In section 4 we define what a multi-objective optimization problem is. Our proposal for extending $ACO_{\mathbb{R}}$ to tackle multi-objective problems is described in section 5. The experimental setup and results are described in section 6. Finally, in section 7, we give our conclusions about this work and a few ideas for future work in this area.

2 The ACO metaheuristic

Ant Colony Optimization makes use of agents, called *ants*, which mimic the behavior of real ants in how they manage to establish shortest-route paths from their colony to feeding sources and back [8]. Ants communicate information through *pheromone trails*, which influence which routes the ants follow, and eventually lead to a solution route.

ACO was initially designed to solve the Traveling Salesman Problem (TSP) and works as follows. The salesman must visit a number of cities exactly once each by the shortest total path possible. The cities and routes between them can be represented as a connected graph, and the ants move from one city to another following the pheromone trails on the edges. Let τ_{ij} be the trail intensity on edge (i, j). Then, each ant chooses the next city to visit depending on the intensity of the associated trail. When the ants have completed their city tours, the trail intensity is updated according to:

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij} \tag{1}$$

where ρ is a coefficient such that $1-\rho$ represents the evaporation of trail, which must be set to a value less than one to avoid unlimited accumulation of trail, and

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

where $\Delta\tau_{ij}^{k} = 1/W_k$ is the pheromone quantity laid on edge (i, j) by the k-th ant, if edge (i, j) is in the trajectory of the k-th ant, and W_k is the trajectory cost (i.e. total path length) of the k-th ant [5].

The transition probability p_{ij}^{k} from city i to city j for the k-th ant is given by

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\ \eta_{ij}^{\beta}}{\sum_{l \in\ \mathrm{allowed}_k} \tau_{il}^{\alpha}\ \eta_{il}^{\beta}}, \quad \forall j \in\ \mathrm{allowed}_k \tag{3}$$

where $\eta_{ij} = 1/d_{ij}$ is called *visibility* and d_{ij} is the associated cost to travel from city i to city j, α and β are parameters that control the relative importance of trail versus cost, and allowed$_k$ is the set of allowed cities the k-th ant can move to from city i [5].

3 ACO for continuous domains

Recently, Ant Colony Optimization for continuous domains has been proposed by Socha and Dorigo [11], which they call $\mathrm{ACO}_{\mathbb{R}}$. The main difference between ACO and $\mathrm{ACO}_{\mathbb{R}}$ is that the first uses a discrete probability distribution to move from one node to another using equation (3), while the second considers a probability density function for each dimension. In their work, Socha and Dorigo [11] use a Gaussian function. In both versions we have to consider one distribution/function per step/dimension, i.e. n.

In ACO, the pheromone intensity τ_{ij} between nodes i and j is seen as a matrix entry, and each ant contributes to it with a quantity relative to the quality of the solution it is proposing, if edge (i, j) is in their chosen route. After the pheromone levels are updated, all routes are dropped and a new cycle begins.

$\mathrm{ACO}_{\mathbb{R}}$ uses a solution archive of size k, where the best solutions, after each cycle, are stored and sorted according to their rank. If we consider this archive as a matrix, each entry can be referred to as s_j^i, $i = 1, 2, ..., n$, $j = 1, 2, ..., k$. Each ant is going to stochastically select one solution from this archive in order to build its own solution.

A solution s_l in the archive is given a weight ω_l according to the its quality (rank)

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} \mathrm{e}^{\frac{(l-1)^2}{2q^2k^2}} \tag{4}$$

where q is a parameter of the algorithm. So, the probability of solution s_l being chosen is

$$p_l = \frac{\omega_l}{\sum_{r=1}^{k} \omega_r} \tag{5}$$

If q in (4) is small, the best-ranked solutions are strongly preferred, and when it is large, the probability becomes more uniform [11].

For every dimension, all ants have to sample a Gaussian function. For each ant we have a Gaussian function $g^i(x)$ at dimension i

$$g^i(x) = \frac{1}{\sigma^i\sqrt{2\pi}} \mathrm{e}^{\frac{(x-\mu^i)^2}{2(\sigma^i)^2}} \tag{6}$$

so we need to define μ^i and σ^i.

If an ant has selected solution l to construct its own solution, then

$$\mu^i = s_l^i \tag{7}$$

and for the standard deviation

$$\sigma^i = \xi \sum_{e=1}^{k} \frac{|s_e^i - s_l^i|}{k-1} \tag{8}$$

where ξ is a parameter of the algorithm that has the same effect as the pheromone evaporation parameter ρ in ACO, and the higher the value, the lower the resultant convergence speed of the algorithm [11].

4 Multi-objective optimization problems

We can define a multi-objective optimization problem, without loss of generality, as the minimization problem [10, 16]:

$$\text{minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})) \tag{9}$$

subject to the constraints:

$$g_i(\mathbf{x}) \leq 0, \qquad \forall\, i = 1, 2, ..., m \tag{10}$$

$$h_j(\mathbf{x}) = 0, \qquad \forall\, j = 1, 2, ..., p \tag{11}$$

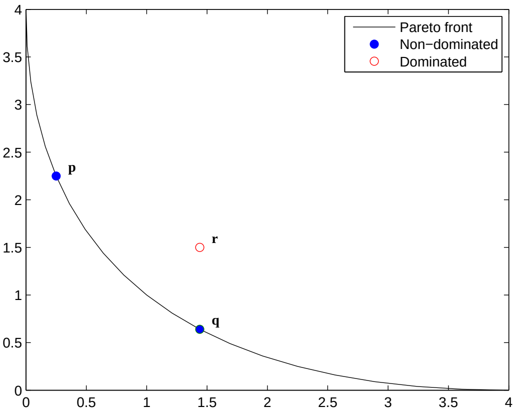

Figure 1: Definitions: $\mathbf{q} \prec \mathbf{r}$, $\mathbf{p}$ and $\mathbf{q}$ are non-dominated and belong to the Pareto front.

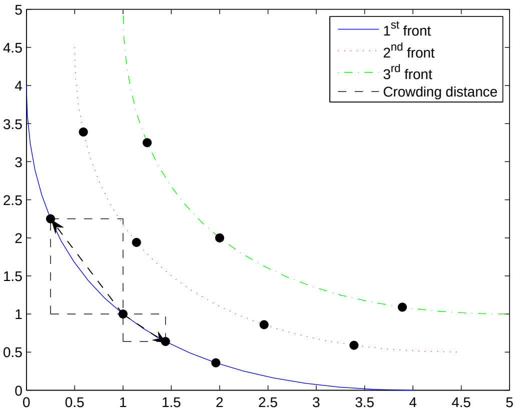

Figure 2: Selection criteria to store solutions in the archive.

where $\mathbf{x} = (x_1, x_2, ..., x_n) \in X$ is the vector of decision variables, X is the parameter space, and $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., k$ are the objective functions. Functions $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$ in (10) and (11) are the constraint functions of the problem.

To make this clearer, let us consider the following simple example. Minimize the objective functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$:

$$f_1(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2 \tag{12}$$

$$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2 \tag{13}$$

subject to the constraints:

$$g_1(\mathbf{x}) = x_1^2 + x_2^2 - 225 \leq 0 \tag{14}$$

$$g_2(\mathbf{x}) = x_1 - 3x_2 + 10 \leq 0 \tag{15}$$

In this case we have $k = 2$, $n = 2$, $m = 2$, $p = 0$, and the aim of this paper is to develop an algorithm that will find values of the decision variables x_1 and x_2 which minimize the functions f_i while satisfying the constraints g_i. The difficulty is that minimizing one f_i will often be incompatible with minimizing the others.

We need a terminology and notation to specify the trade-offs between minimizing the various f_i. A decision vector $\mathbf{x} \in X$ is said to *dominate* a decision vector $\mathbf{y} \in X$ (written as $\mathbf{x} \prec \mathbf{y}$) if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})\ \forall\ i = 1, 2, ..., n$ and $\exists\ j \in \{1, 2, ..., n\} : f_j(\mathbf{x}) < f_j(\mathbf{y})$. Conversely, we say that a decision vector $\mathbf{x} \in S \subset X$ is *nondominated* with respect to S if there is no decision vector $\mathbf{y} \in S$ such that $\mathbf{y} \prec \mathbf{x}$.

A decision vector $\mathbf{x} \in X$ is said to be *Pareto optimal* if it is nondominated with respect to X. The *Pareto optimal set* is defined as $P_s = \{\mathbf{x} \in X \mid \mathbf{x} \text{ is Pareto optimal}\}$. Finally, the *Pareto front* is defined as $P_f = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n \mid \mathbf{x} \in P_s\}$. These definitions are represented graphically in Figure 1.

5 ACO$_\mathbb{R}$ for multi-objective problems

The main difference we are proposing between ACO$_\mathbb{R}$ for single- and multi-objective problems lies in how to store solutions in the archive. In ACO$_\mathbb{R}$, the best solutions after each cycle are stored, with the best solutions defined as those that are closer to the optimal solution. In multi-objective problems we do not have an optimal solution, but a set of solutions that are trying to reach the Pareto optimal set [16]. In this case, we have to define which solutions are better than which others.

There are a number of criteria we could use to say that one solution is better than another. In this paper we have chosen to use the concept of *dominance depth* [4, 17]. Dominance depth involves grouping solutions into several fronts. Once the solutions are grouped, we prefer those solutions belonging to the outer-most front, i.e. the front closest to the Pareto front, to be stored in the archive.

If the number of solutions exceed the size k of the archive, they are selected according to the *density information*, which means that a solution's chance of being archived is decreased the greater the density of solutions in its neighborhood [17]. We will call this information *crowding distance* [4]. The criteria used to store solutions in the archive are represented graphically in Figure 2.

All the other processes in standard ACO$_\mathbb{R}$ are preserved for multi-objective problems.

Problem	n	Variable Bounds	Objective Functions	Comments
SCH1	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	convex
SCH2	1	$[-5, 10]$	$f_1(x) = \begin{cases} -x & \text{if } x \le 1 \\ -2+x & \text{if } 1 < x \le 3 \\ 4-x & \text{if } 3 < x \le 4 \\ 4+x & \text{if } x > 4 \end{cases}$ $f_2(x) = (x-5)^2$	disconnected
DEB1	2	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = (1+10x_2)\left(1 - \left(\frac{x_1}{1+10x_2}\right)^{\alpha} - \frac{x_1 \sin(2\pi q x_1)}{1+10x_2}\right)$ $\alpha = 2, q = 4$	disconnected
DEB2	2	$x_1 \in [0, 1]$ $x_2 \in [-30, 30]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})h(\mathbf{x})$ $g(\mathbf{x}) = 11 + x_2^2 - 10\cos(2\pi x_2)$ $h(\mathbf{x}) = \begin{cases} 1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})} & \text{if } f_1(\mathbf{x}) \le g(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}$	multimodal
POL	2	$[-\pi, \pi]$	$f_1(\mathbf{x}) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(\mathbf{x}) = (x_1+3)^2 + (x_2-1)^2$ $A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2$ $A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2$ $B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2$ $B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2$	nonconvex, disconnected
FON	3	$[-4, 4]$	$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$ $f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$	nonconvex
KUR	3	$[-5, 5]$	$f_1(\mathbf{x}) = \sum_{i=1}^{n-1}\left(-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right)$ $f_2(\mathbf{x}) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin x_i^3)$	nonconvex
ZDT1	30	$[0, 1]$	$f_1(\mathbf{x}) = 1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \sqrt{x_1/g(\mathbf{x})}\right)$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	convex
ZDT2	30	$[0, 1]$	$f_1(\mathbf{x}) = 1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - (x_1/g(\mathbf{x}))^2\right)$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	nonconvex
ZDT3	30	$[0, 1]$	$f_1(\mathbf{x}) = 1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \sqrt{x_1/g(\mathbf{x})}\right) - (x_1/g(\mathbf{x}))\sin(10\pi x_1)$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	convex, disconnected
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \ldots, n$	$f_1(\mathbf{x}) = 1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \sqrt{x_1/g(\mathbf{x})}\right)$ $g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^{n}\left(x_i^2 - 10\cos(4\pi x_i)\right)$	nonconvex
ZDT6	10	$[0, 1]$	$f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right)$ $g(\mathbf{x}) = 1 + 9\left((\sum_{i=2}^{n} x_i)/(n-1)\right)^{0.25}$	nonconvex, nonuniformly spaced

Table 1: The test problems studied.

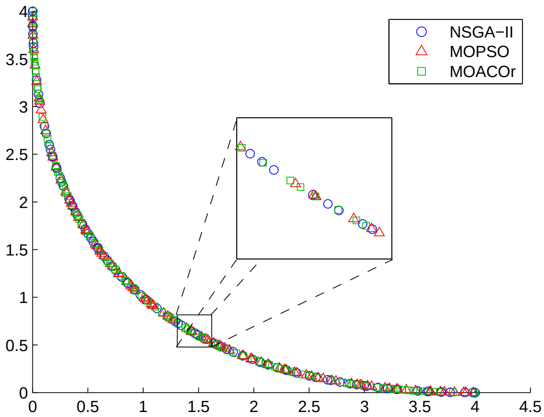

Figure 3: Problem SCH1

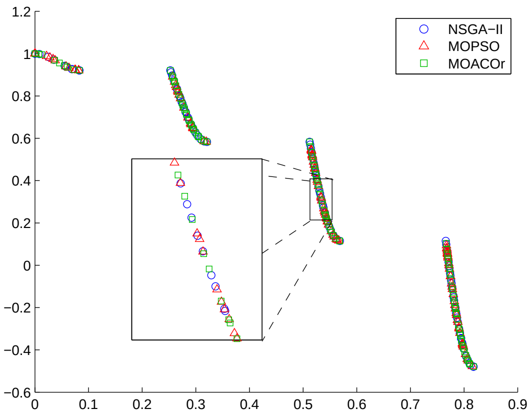

Figure 5: Problem DEB1

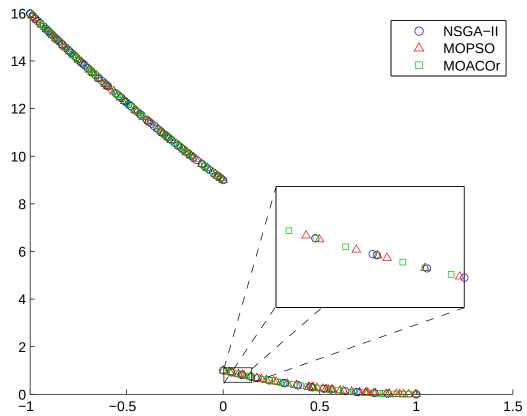

Figure 4: Problem SCH2

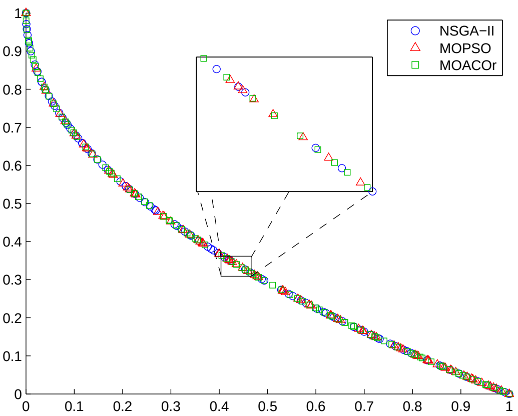

Figure 6: Problem DEB2

6 Experimental setup and results

Naturally we need to test our new approach against existing techniques for multi-objective optimization. To do this, we have considered the multi-objective test problems analyzed in [4] and [3]. These works apply NSGA-II and MOPSO respectively, which are well-known methods for solving multi-objective problems. The test problems studied are specified in Table 1.

Our results are compared directly with those obtained using NSGA-II [4] and MOPSO [3]. We have used the code for both of these that is available from the EMOO repository[1]. We have set the population size = 100 and the number of iterations = 250 for the three methods, and $k = 100$, $q = 0.0001$ and $\xi = 0.85$ in our proposal. The results comparisons are displayed in Figures 3 to 14.

In Figures 3 to 6 we can see that all three methods have obtained similar results, and the solutions are uniformly distributed over the Pareto approximation.

For problem POL, shown in Figure 7, our $ACO_{\mathbb{R}}$ approach obtained similar results to MOPSO, but these solutions are not as good as those obtained by NSGA-II.

Analyzing Figures 8 and 9, we see that $ACO_{\mathbb{R}}$ is not performing as well here as the other two techniques, as solutions from $ACO_{\mathbb{R}}$ are not uniformly distributed over the Pareto approximation.

For problems ZDT1, ZDT2 and ZDT3, shown in Figures 10, 11 and 12, we can observe that the solutions obtained using $ACO_{\mathbb{R}}$ and NSGA-II are uniformly distributed over the Pareto approximations. In these cases MOPSO was out-performed.

We can see in Figure 13, showing problem ZDT4, that the MOPSO Pareto approximation

1. `http://www.lania.mx/~ccoello/EMOO/`

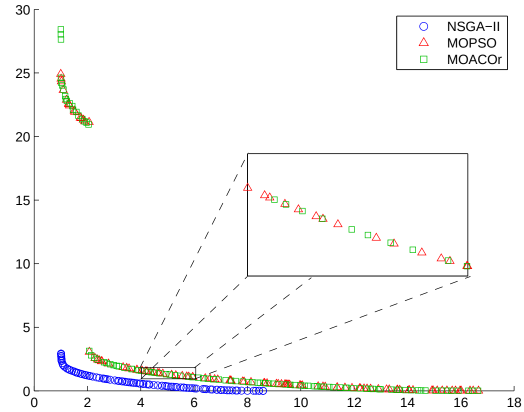

Figure 7: Problem POL

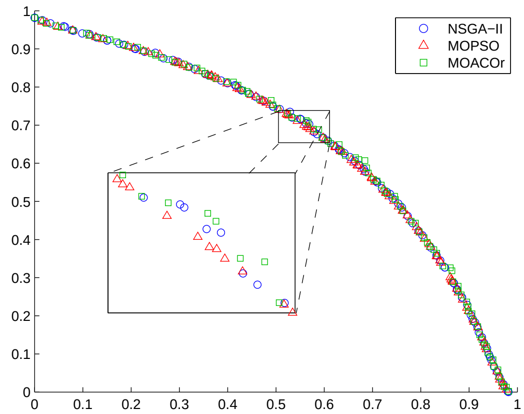

Figure 8: Problem FON

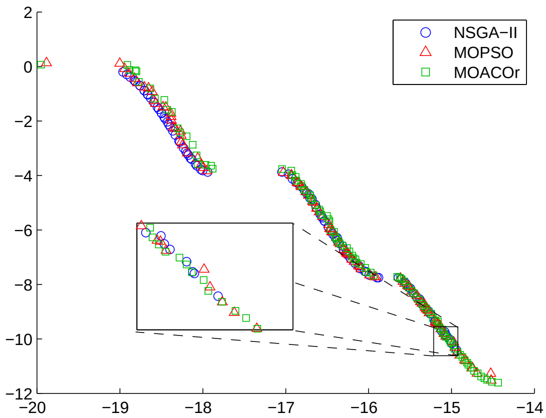

Figure 9: Problem KUR

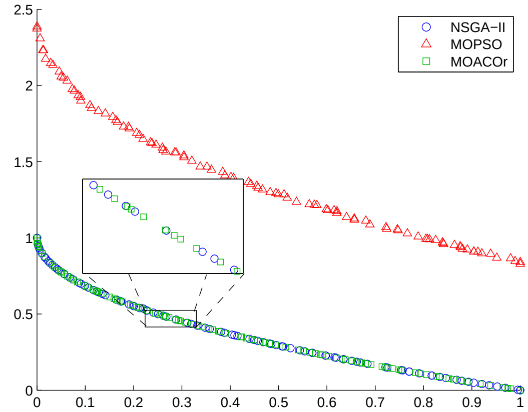

Figure 10: Problem ZDT1

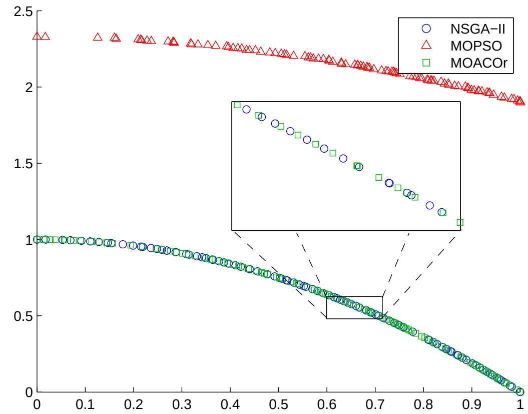

Figure 11: Problem ZDT2

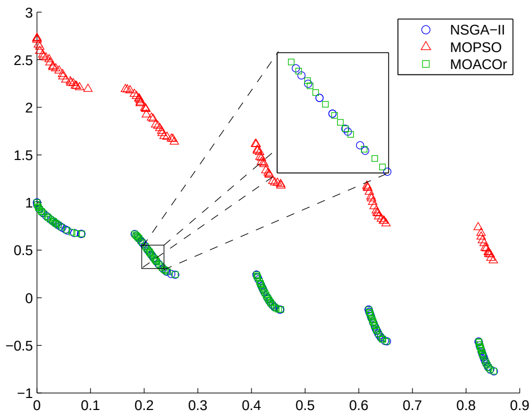

Figure 12: Problem ZDT3

in this case is considerably distant from that discovered by NSGA-II. The $ACO_{\mathbb{R}}$ solutions lie between these two sets.

The last problem, ZDT6, shown in Figure 14, is particularly interesting. MOPSO found solutions that are far from being a good Pareto approximation. NSGA-II solutions are very well distributed over the Pareto approximation.

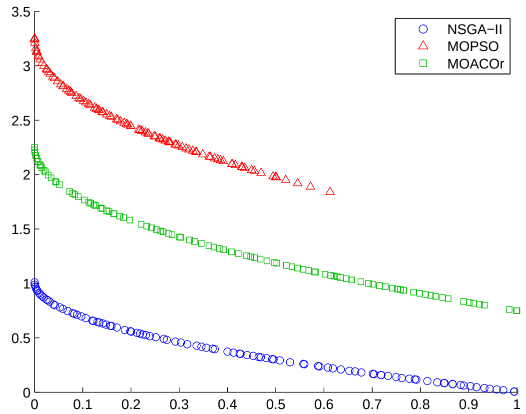

Figure 13: Problem ZDT4

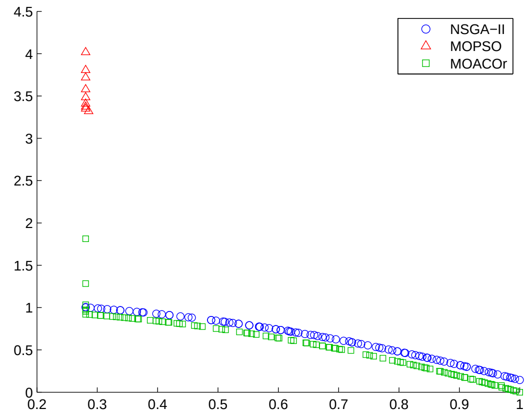

Figure 14: Problem ZDT6

$ACO_{\mathbb{R}}$ could find a better Pareto approximation than NSGA-II and has solutions uniformly distributed over it.

We can summarize our empirical results as follows: The new $ACO_{\mathbb{R}}$ algorithm presented here was able to find a Pareto approximation as good as that obtained with NSGA-II in seven of the twelve standard test problems considered, and found a better Pareto approximation in one of them.

7 Conclusions

We have presented an extension of the $ACO_{\mathbb{R}}$ algorithm that can be used for multi-objective optimization problems. Tests on a series of benchmark problems have indicated that this approach may provide a promising new approach for this type of problem. We have demonstrated that, overall, the solutions obtained using $ACO_{\mathbb{R}}$ are comparable with those obtained using two other well-known methods, namely NSGA-II and MOPSO.

We can conjecture, that the $ACO_{\mathbb{R}}$ algorithm's good performance is due to the fact that each ant has to select one of the promising solutions in the archive to construct its own solution. This means that there is a variety of Gaussian functions that are available to the ants to be selected and sampled in order for the ants to try to construct even better solutions.

This is the first work using a variation of $ACO_{\mathbb{R}}$ for solving multi-objective problems, and it is clear that further research would be worthwhile. For example, we should investigate more carefully the solution sets that are stored in the archive, and our conjecture concerning the reason for the algorithm's good performance. We need to consider more carefully the properties of the test problems that result in the different performance levels found across the various algorithms compared in this study. We also need to look at various performance metrics to establish whether there are significant differences in the performance levels obtained by the various methods, beyond those that are clearly visible in the graphs presented.

References

[1] Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss. Applying the ant system to the vehicle routing problem. In *2nd International Conference on Metaheuristics*, pages 1–12, 1997.

[2] Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89:319–328, 1999.

[3] Carlos A. Coello Coello and Maximino Salazar Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation*, volume 2, pages 1051–1056, 2002.

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[5] Marco Dorigo, Gianni Di Caro, and Luca Maria Gambardella. Ant algorithms

for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.

[6] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*, 43(1):73–81, 1997.

[7] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, April 1997.

[8] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transaction on Systems, Man and Cybernetics - Part B: Cybernetics*, 26(1):29–41, 1996.

[9] Silvia Mazzeo and Irene Loiseau. An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics*, 18:181–186, 2004.

[10] Margarita Reyes Sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[11] Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, doi:10.1016/j.ejor.2006.06.046, 2006.

[12] Shigeyoshi Tsutsui. Ant colony optimisation for continuous domains with aggregation pheromones metaphor. In Ahmad Lotfi, editor, *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, pages 207–212, Nottingham, United Kingdom, December 16-18 2004. Nottingham Trent University.

[13] Shigeyoshi Tsutsui, Martin Pelikan, and Ashish Ghosh. Performance of aggregation pheromone system on unimodal and multimodal problems. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005.

[14] Wei-Qing Xiong and Ping Wei. A kind of ant colony algorithm for function optimization. In *Proceedings of the 2002 International Conference on Machine Learning and Cybernetics*, pages 552–555. IEEE Press, 2002.

[15] Jun Zhang, Xiaomin Hu, X. Tan, J. H. Zhong, and Q. Huang. Implementation of an ant colony optimization technique for job shop scheduling problem. *Transactions of the Institute of Measurement and Control*, 28(1):93–108, 2006.

[16] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[17] Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A tutorial on evolutionary multiobjective optimization. In *Metaheuristics for Multiobjective Optimisation*, pages 3–38. Springer-Verlag, Berlin, Germany, 2004.