ANT COLONY ALGORITHMS APPLIED TO DISCRETE OPTIMIZATION PROBLEMS

Priscila V. S. Z. Capriles, **Leonardo Goliatt da Fonseca**, **Helio J. C. Barbosa**,
Laboratório Nacional de Computação Científica - LNCC/MCT
Ave. Getúlio Vargas, 333 - Quitandinha
25651-075, Petrópolis, RJ
{capriles, goliatt, hcbm}@lncc.br

Afonso C. C. Lemonge
Universidade Federal de Juiz de Fora, Faculdade de Engenharia, Departamento de Estruturas
Campus Universitário, Bairro Martelos
36036-330, Juiz de Fora, MG
lemonge@numec.ufjf.br

Abstract

Five variants of the ant colony optimization metaheuristic, namely Ant System, Ant Colony System, Max-Min Ant System, Rank Based Ant System, and Best-Worst Ant System, were implemented and applied to discrete function as well as structural optimization problems in order to compare the different implementations. The results obtained suggest the Rank Based Ant System as the best implementation of the ACO in relation to the efficacy and the quality of the final solution.

1 Introduction

The Ant Colony Optimization (ACO), inspired by the observation of the foraging behavior of real ant colonies, is a metaheuristic which uses the concept of stigmergy (indirect communication mediated by pheromone rates) in order to construct increasingly better candidate solutions in a discrete search space. The ACO algorithm can be summarized as a set of computational agents (a colony of ants) that move through states of the problem, using a stochastic local decision policy. After each ant completes a trail (which defines a candidate solution), the pheromone rates in that particular trail are modified according to the evaluation of such solution. The decision of the subsequent ants will be affected for the amount of pheromone deposited by the previous ants during the construction of their trails.

The Ant Colony Algorithm was first proposed by Dorigo (1992) in order to solve difficult combinatorial optimization problems such as the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP). Since then, many variants of the basic principle have been reported in the literature. Five of them were implemented here: Ant System, Ant Colony System, Max-Min Ant System, Rank Based Ant System, and Best-Worst Ant System. These variants differ from each other in the form of pheromone update in the environment. These different implementations were applied to some discrete optimization problems and the results are compared by means of numerical experiments.

The paper is organized as follows. Section 2 contains the description of the ACO as its five variants implemented. Section 3 describes the formulation of the discrete optimization problem. Section 4 reports the numerical experiments and Section 5 provides a discussion of the results obtained. Conclusions are presented in Section 6.

2 Ant Colony Algorithms

The ACO metaheuristic consists of a colony of artificial ants with the characteristics to search good solutions to discrete optimization problems. Each ant builds a solution, starting from an initial state selected according to some criteria. Ants can act concurrently and independently, showing a cooperative behavior, using the stigmergy, an indirect mechanism for their communication mediated by pheromone rates, that governs the information exchange among the ants. Each ant, using a stochastic local decision policy, moves through states of the problem, and is able to construct a (feasible or infeasible) candidate solution.

After each ant completes a trail, defining a candidate solution, the pheromone rates in that particular trail are modified according to the evaluation of such solution. The decision of the subsequent ants will be affected by the amount of pheromone deposited by the previous ants during the construction of their trails.

Although real ants are simple agents with limited capabilities, the colony as a whole is capable of executing several complex tasks.

Artificial ants have been inspired by the real ones although they can be enriched with some capabilities which are not found in real ants. Some similarities with real ants are the existence of a colony of cooperating individuals, pheromone trail and stigmergy, and the task of finding the shortest (minimum cost) path joining origin and destination sites. The main differences are: (i) artificial ants live in a discrete world and their move are from one discrete state to another, and (ii) they have an internal memory of the search procedure. Furthermore, artificial ants deposit an amount of pheromone which is a function of the quality of the solution found. To make them more effective and efficient in solving difficult optimization problems it is reasonable to give artificial ants some capabilities not present in real ants.

In the following the ACO algorithms implemented are summarized.

2.1 Ant System (AS)

Developed by Dorigo, Maniezzo and Colorni (Dorigo et al., 1991), it was the first algorithm based on ACO metaheuristic to solve the TSP. In the AS algorithm, the evaporation occurs all over the trail constructed by each ant. Besides that, pheromone is deposited along the trail proportionally to the quality of the corresponding solution.

The search space is represented by the pheromone matrix τ, which has as dimensions the number of variables and the number of values that each one can assume.

Solutions in AS are constructed as follows: at each step, an ant i, located at position j, moves to the next position $j+1$ with probability given by

$$p_{jk} = \left[\frac{(\tau_{jk})^{\alpha} \left(\frac{1}{\eta_{jk}}\right)^{\beta}}{\sum (\tau_{jk})^{\alpha} \left(\frac{1}{\eta_{jk}}\right)^{\beta}} \right] \tag{1}$$

where η_{jk} is the heuristic desirability of transition and α and β are two parameters that weigh the relative importance of the pheromone trail and the heuristic information, respectively.

The pheromone trail is evaporated by reducing all pheromone trails

$$\tau_{jk} = \left[(1-\rho) \frac{\tau_{jk}}{\sum \tau_{jk}} \right] \tag{2}$$

where $\rho \in (0, 1]$ is the evaporation rate. Next, each ant i deposits an amount of pheromone in its route

$$\tau_{jk} = \tau_{jk} + Q \tag{3}$$

where

$$Q = \frac{1}{F(x)} \tag{4}$$

and $F(x)$ is the fitness function of the i-th ant. The fitness function is a measure of the quality of the solution, which will be defined in the Section 3.

A pseudo-code of the AS algorithm is displayed in the Algorithm 1.

Algorithm 1 Ant System (AS)

```
for each colony do
    for each ant do
        generate route
        evaluate route
        evaporate pheromone in trails
        deposit pheromone on trails
    end for
end for
```

2.2 Ant Colony System (ACS)

Also developed by Dorigo and Maniezzo (Maniezzo et al., 2004) to solve the TSP it was the first successor and improvement of the AS implementation.

In this case, besides the pheromone matrix updating, before each ant completes its route the ACS makes an extra update in the best global route. This happens with an evaporation rate ρ_2.

To add pheromone, ACS uses the same formulation presented in Eq. (3). To update pheromone, the Eq. (4) is modified by an extra constant σ, as presented in Eq. 5.

$$Q = \frac{1}{\sigma F(x)} \tag{5}$$

In ACS, the choice of the route occurs in two ways, using q_0 as determination index. In each iteration a q value is chosen randomly, if $q > q_0$ the ant chooses the route by Eq. (1), otherwise it chooses the route by Eq. (6):

$$k = \text{argmax}_l \; \tau_{jl} \tag{6}$$

Algorithm 2 shows a pseudo-code of the ACS.

Algorithm 2 Ant Colony System (ACS)

```
for each colony do
    for each ant do
        generate route
        evaluate route
        evaporate pheromone in all trails (ρ rate)
        deposit pheromone on all trails
    end for
    evaporate pheromone in best global route (ρ2 rate)
    deposit pheromone on best global route
end for
```

2.3 Max-Min Ant System (MMAS)

Developed by Stützle and Hoos (1996), as another variation for the TSP, the MMAS algorithm shows differences in the steps of pheromone deposition and evaporation, that occur only after the i-th ant for each colony stablish its trail.

In this case, the evaporation happens in all trails for the current colony followed by deposition on the best global route, as presented in Eq. (3) and Eq. (5).

For the choice of the route, the MMAS uses the same principle of the ACS.

The pseudo-code of the MMAS algorithm is presented in the Algorithm 3.

Algorithm 3 Max-Min Ant System (MMAS)

```
for each colony do
    for each ant do
        generate route
        evaluate route
    end for
    verify for global or local best
    evaporate pheromone in all trails
    deposit pheromone on best global route
end for
```

2.4 Rank Based Ant System (RBAS)

Proposed in 1997 by Bullnheimer and collaborators (Bullnheimer et al., 1999), RBAS is an AS variation that uses a rank idea to update the pheromone matrix.

After the i-th ant for current colony stablish its trail, all trails suffer pheromone evaporation, the fitness $F(x)$ are calculated and ants are sorted according to the quality of the solutions constructed.

In this way, a percentage (in general 25%) from the best routes are chosen for the addition of pheromone according to Eq. (5), where the parameter σ is replaced by a parameter σ_2. Besides that, the extra deposition in the best global route is made using Eq. (5).

With this procedure, exploration among the elite routes is introduced instead of favoring only the best global route.

The pheromone matrix update occurs as presented in Eq. (3) and Eq. (5). The choice of the route follows as in ACS and MMAS.

The pseudo-code for the RBAS algorithm is displayed in the Algorithm 4.

Algorithm 4 Rank Based Ant System (RBAS)

```
for each colony do
    for each ant do
        generate route
        evaluate route
    end for
    verify for global or local optimum
    evaporate pheromone in all trails
    generate an elite from current colony
    deposit a Y amount of pheromone on elite trails
    deposit an X amount of pheromone on best global route
end for
```

2.5 Best-Worst Ant System (BWAS)

Considered for the first time in 1999 (Cordón et al., 2000), the BWAS algorithm is the AS variation which presents more modifications.

The BWAS introduces two series of pheromone evaporation, one at the end of each colony, with evaporation rate ρ in all trails of the current colony, and an extra evaporation with rate ρ_2 in the worst route of the current colony.

The pheromone deposition occurs only on the best global route, providing a lesser degree of exploration amongst the possible routes.

The pheromone matrix update occurs as presented in Eq. (3) and Eq. (5). The choice of the route, follows as in ACS, MMAS and RBAS.

The Algorithm 5 describes the pseudo-code for the BWAS algorithm.

Furthermore, the BWAS can be implemented using evolutionary computation concepts, such as mutation in the pheromone matrix to introduce diversity and increase the exploration in the search process. However, mutation was not implemented here.

Algorithm 5 Best-Worst Ant System (BWAS)

```
for each colony do
    for each ant do
        generate route
        evaluate route
    end for
    verify for global or local optimum
    evaporate pheromone in all trails (ρ rate)
    verify the worst route from current colony
    deposit pheromone on best global route
    evaporate pheromone in edges of the worst
    route from current colony which do not be-
    long to the best global route (ρ2 rate)
end for
```

3 The Discrete optimization Problem

The discrete optimization problem can written as:

$$\text{minimize} \quad f(x) \tag{7}$$
$$\text{subject to}$$
$$g_k(x) \le b_k \qquad k = 1, ..., m$$
$$x_i^L \le x_i \le x_i^U, \quad i = 1, ..., n$$

where:
$x \in \mathbb{R}^n$ is the vector of discrete variables,
$f : \mathbb{R}^n \to \mathbb{R}$ is the objective function,
$g : \mathbb{R}^m \to \mathbb{R}$ is the constraint function on x,
x_i^L and x_i^U are the lower and upper bounds of the search space.

The problem (7) is a constrained optimization problem, and one way to treat the constraints is to replace the constrained optimization problem by an unconstrained one by adding a penalty.

Consider the following relaxation of (7):

$$\text{minimize} \quad F(x) \tag{8}$$
$$x_i^L \le x_i \le x_i^U, \quad x_i \text{ integer}, \ i = 1, ..., n$$

where

$$F(x) = f(x) + \xi \sum_{k=1}^{m} [s_k^+]^2 \tag{9}$$

$$s_k = \frac{|g_k(x)|}{|b_k|} - 1, \qquad k = 1, ..., m \tag{10}$$

The function $F(x)$ is named fitness function and $s_k^+ = s_k$ if s_k is positive and 0 otherwise.

Allowing the variable s_k to become positive, the objective function $f(x)$ is increased by a penalization term ξs_k^2 in the k-th constraint, proportional to the square of the violation. In the experiments performed here, the parameter ξ is constant in each problem.

4 Numerical Experiments

In order to investigate the performance of the ACO procedures, some discrete optimization problems from literature are solved.

4.1 Design of a Gear Train

The compound gear train, from Kannan and Kramer (1994), is shown in the Fig. 1. The objective of the problem is to find the number of teeth in each of the four gears, such that the gear ratio is as close to 1/6.931 as possible.

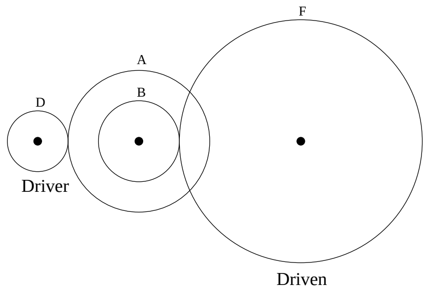

Figure 1: Compound gear train.

The problem variables are the number of teeth in the gears A, B, D and F denoted by x_1, x_2, x_3 and x_4, that can assume values between 12 and 60.

The problem can be stated as:

$$\text{minimize} \quad f(x) = \left[\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right]^2 \tag{11}$$
$$\text{subject to:} \quad 12 \le x_i \le 60$$
$$x_i \text{ integer}, \qquad i = 1, 2, 3, 4$$

The problem was solved using the ACO metaheuristic implemented, for a colony of 20 ants and 10000 function evaluations in 500 cycles. In this experiment 30 independent runs were performed.

The results, including the parameters used, are displayed in the Table 1.

4.2 Ten Bar Truss

This test-problem corresponds to the weight minimization of the classic ten-bar truss (Lemonge and Barbosa, 2004). The planar truss is shown in the Fig. 2.

The constraints involve the stress in each member and the displacements at the nodal points. The design variables are the cross-sectional areas of the

	AS	ACS	MMAS	RBAS	BWAS
Best	2.3578e-09	2.7071e-12	2.4597e-07	2.7071e-12	2.7267e-08
Average	6.4482e-04	5.5613e-04	1.0700e-01	4.5709e-03	3.2102e-02
Worst	1.9196e-02	1.6139e-02	1.4877e+00	6.8187e-02	2.5088e-01
α	1.00	1.00	1.00	0.20	1.00
β	–	–	–	–	–
σ	0.75	0.68	0.70	0.68	0.69
σ_2	–	–	–	0.10	–
ρ	0.92	0.20	0.10	0.20	0.20
ρ_2	–	0.20	–	–	0.20
q_0	–	0.60	0.50	0.50	0.50

Table 1: Gear train results for five ACO algorithms using a colony of 20 ants and 10000 evaluations in 30 independent runs.

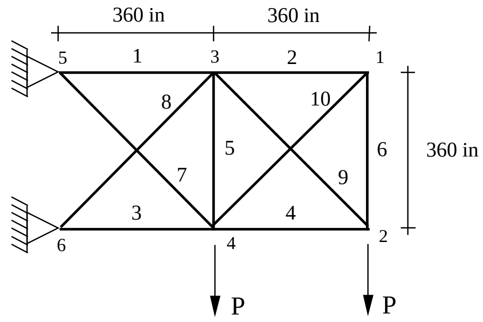

Figure 2: Ten bar truss.

bars $A = (A_1, A_2, ..., A_{10})$. The normalized stress and displacements constraints are given by

$$\frac{|\sigma_j|}{\sigma_{adm}} - 1 \leq 0 \qquad 1 \leq j \leq 10 \qquad (12)$$

$$\frac{|u_i|}{u_{adm}} - 1 \leq 0 \quad 1 \leq i \leq 8 \qquad (13)$$

The allowable stress is $\sigma_{adm} = 25$ ksi and displacements are limited to $u_{adm} = 2$ in, in the x and y directions. Other relevant information are: the density of the material is 0.10 lb/in^3, Young modulus is $E = 10^4$ ksi, vertical nodal loads of 100 kips are applied at nodes 2 and 4, and the penalty parameter ξ is set to 10^7.

The values of cross-sectional areas A_i are chosen from the set $\mathcal{L} = \{$ 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50 $\}$ resulting in 42 options for each member in the structure.

In Table 2 are shown, for each algorithm, the best and worst solutions found, and the average in 30 independent runs. Is important to notice that the average was calculated considering only feasible solutions. The number of infeasible solutions and the percentage of feasible solutions are displayed together with the set of parameters used in the algorithms. An infeasible solution is such that $f(x) < F(x)$. In this experiment a colony of 20 ants is used with 96000 function evaluations in 4800 cycles.

4.3 Twenty-five Bar Truss

This classical problem (Lemonge and Barbosa, 2004) is the weight minimization of a truss with 25 bars shown in the Fig. 3.

	AS		ACS		MMAS		RBAS		BWAS	
	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$
Best	6016.863	6016.863	5523.905	5523.905	5505.834	5505.834	5499.354	5499.354	5509.717	5509.717
Average	6952.226	6952.226	6015.056	6015.056	5961.313	5961.313	5707.715	5707.715	5936.286	5936.286
Worst	8030.234	8030.234	10237.784	10237.784	7849.908	7849.908	6339.127	6339.127	7747.449	7747.449
Infeasible (#)	5	5	1	1	0	0	0	0	0	0
Feasible (%)	83.33	83.33	96.67	96.67	100.00	100.00	100.00	100.00	100.00	100.00
α	0.20		0.20		0.20		0.20		0.20	
β	0.43		0.43		0.43		0.43		0.43	
σ	–		1.00		0.10		1.00		1.00	
σ_2	–		–		–		0.50		–	
ρ	0.50		0.60		0.60		0.10		0.60	
ρ_2	–		0.10		–		–		0.70	
q_0	–		0.70		0.70		0.60		0.70	

Table 2: Results for the ten-bar truss for the ACO algorithms using a colony of 20 ants and 96000 evaluations in 30 independent runs.

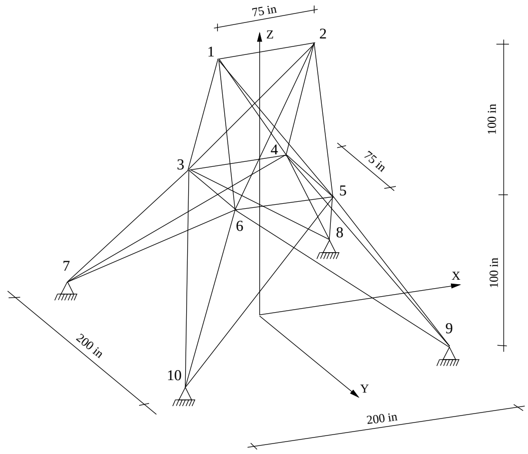

Figure 3: Twenty-five bar truss.

The allowable stress for each member is $\sigma_{adm} = 40$ ksi and the displacements must not exceed $u_{adm} = 0.35$ in, in the x and y directions. The material has a Young modulus $E = 10^7$ psi and density of 0.10 lb/in^3. The loads applied in the structure are displayed in the Table 3.

Node	F_x(kip)	F_y(kip)	F_z(kip)
1	0.0	20.0	-5.0
2	0.0	-20.0	-5.0

Table 3: Loading data for twenty-five bar truss.

The design variables are the cross-sectional areas $A = (A_1, A_2, ..., A_8)$, organized into eight groups, where all members in the groups share the same area. This arrangement results in a constrained optimization problem with eight discrete variables, to be chosen from the set $\mathcal{L} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4\}$ with 34 values in square inches. The penalty parameter ξ is set to 10^7.

The results obtained for 30 independent runs are presented in Table 4 together with parameters used in each algorithm. In this experiment are used a colony of 20 ants and 60000 evaluations in 3000 cycles.

5 Discussion

The presented results show that the best algorithm for the set of problems considered is the Rank Based Ant System. According to the literature, the BWAS should have presented equally good or better results than RBAS. That is probably due to the fact

	AS		ACS		MMAS		RBAS		BWAS	
	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$	$f(A)$	$F(A)$
Best	503.197	503.197	485.574	485.574	486.100	486.100	484.854	484.854	486.100	486.100
Average	525.122	525.122	492.880	492.880	495.088	495.088	488.226	488.226	497.690	497.690
Worst	540.730	540.730	511.531	511.531	515.947	515.947	504.736	504.736	537.402	537.402
Infeasible (#)	0	0	0	0	0	0	0	0	0	0
Feasible (%)	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
α	0.20		0.20		0.20		0.20		0.20	
β	0.43		0.43		0.43		0.43		0.43	
σ	–		0.92		0.10		1.10		0.90	
σ_2	–		–		–		0.80		–	
ρ	0.92		0.60		0.60		0.92		0.70	
ρ_2	–		0.10		–		–		0.70	
q_0	–		0.70		0.70		0.60		0.70	

Table 4: Results for the twenty-five bar truss for the ACO algorithms, corresponding to a colony of 20 ants and 60000 evaluations in 30 independent runs.

that the mutation in the pheromone matrix, which has not been implemented, would have favored the exploration of the search space.

The results for the gear train design problem are considered satisfactory when compared with other results found in literature, such as Deb and Goyal (1998), using Genetic Adaptive Search algorithm (GeneAS), and Kannan and Kramer (1994) where the Augmented Lagrange Multiplier (ALM) was used to solve the problem. The Table 5 shows the comparison of the results among ACS, RBAS, GeneAS and ALM for the gear train problem. This problem has four global optimal solutions, which two of thus were found by ACS, GeneAS and RBAS.

x_i	**ACS**	**RBAS**	**GeneAS**	**ALM**
x_1	19	16	19	13
x_2	16	19	16	15
x_3	49	49	49	33
x_4	43	43	43	41
$f(x)$	2.707e-12	2.707e-12	2.707e-12	2.125e-08

Table 5: Results for gear train problem. ACS and RBAS results corresponding to 10000 evaluations. The number of evaluations for GeneAS and ALM are not available.

For the cases of the ten-bar and twenty-five bar trusses, the MMAS algorithm turned out to be competitive with BWAS and the RBAS algorithms, producing good results.

Table 6 displays the comparison between RBAS and a genetic algorithm with an adaptive penalty technique (APM) (Lemonge and Barbosa, 2004) for the ten-bar truss when 96000 evaluations were used for RBAS versus 90000 evaluations for APM. In the twenty-five bar truss, RBAS using 60000 evaluations produced the same solution as that found by the APM using 20000 evaluations. This is probably due to the very simple penalty scheme adopted here as compared to that used by APM.

The ACO algorithms are an interesting research field which can produce efficient techniques to solve discrete optimization problems.

However, some care must be taken when chosing the parameters in the algorithm. An option for parameter setting is to use values established in similar problems in the literature.

6 Conclusion

Such as genetic algorithms, simulated annealing, and neural networks, the ACO metaheuristic is an algorithm built around some basic principles taken from the observation of a natural phenomenon.

This paper presented the main variants of ACO metaheuristic: Ant System, Ant Colony System,

Variables	RBAS	APM
A_1	33.50	33.50
A_2	1.62	1.62
A_3	22.90	22.90
A_4	15.50	14.20
A_5	1.62	1.62
A_6	1.62	1.62
A_7	7.22	7.97
A_8	22.00	22.90
A_9	22.90	22.00
A_{10}	1.62	1.62
$F(A)$	5499.354	5490.738

Table 6: Results for ten-bar truss optimization. RBAS results corresponding to 96000 evaluations and APM corresponding to 90000 evaluations.

Max-Min Ant System, Rank Based Ant System and Best-Worst Ant System. These five implementations were compared by means of numerical experiments, in order to investigate their performance in solving discrete constrained and unconstrained optimization problems.

The results obtained suggest the Rank Based Ant System as the best implementation of the ACO in relation to the efficacy and the quality of the final solution.

Parallelization, hybridization with other available techniques, and the introduction of a better penalty technique for constrained problems constitute interesting and potentially rewarding research lines.

References

Bullnheimer, B., Hartl, R. F., and Strauss, C. (1999). A new rank-based version of the ant system: A computational study. *Central European Journal for Operations Research and Economics*, 7(1):25–38.

Cordón, O., de Vianna, I. F., Herrera, F., and Moreno, L. (2000). A new aco model integrating evolutionary computation concepts: The best-worst ant system. In Dorigo, M., Middendorf, M., and Stützle, T., editors, *Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms*, pages 22–29. IRIGIA, Université Libre de Bruxelles, Belgium.

Deb, K. and Goyal, M. (1998). A robust optimization procedure for mechanical component design based on genetic adaptive search. *Transactions of the ASME: Journal of Mechanical Design*, 120(2):162–164.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica e Informazione - Politecnico di Milano/ Italia.

Dorigo, M., Maniezzo, V., and Colorni, A. (1991). Positive feedback as a search strategy. Technical Report 91-012, Dipartimento di Elettronica e Informazione - Politecnico di Milano/ Italia.

Kannan, B. K. and Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 116:405–411.

Lemonge, A. and Barbosa, H. (2004). An adaptive penalty scheme for genetic algorithms in structural optimization. *Int. J. Num. Meth. in Engineering*, 59:703–736.

Maniezzo, V., Gambardella, L. M., and Luigi, F. (2004). Ant colony optimization. In Onwubolu, G. C. and Babu, B. V., editors, *New Optimization Techniques in Engineering*, chapter 5, pages 101–117. Springer-Verlag Berlin Heidelberg.

Stützle, T. and Hoos, H. (1996). Improving the ant system: A detailed report on the max-min ant system. Technical Report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt/ Germany.