Ant Algorithms
Solve Difficult Optimization Problems

Marco Dorigo

IRIDIA
Université Libre de Bruxelles
50 Avenue F. Roosevelt
B-1050 Brussels, Belgium
mdorigo@ulb.ac.be

Abstract. The *ant algorithms* research field builds on the idea that the study of the behavior of ant colonies or other social insects is interesting for computer scientists, because it provides models of distributed organization that can be used as a source of inspiration for the design of optimization and distributed control algorithms. In this paper we overview this growing research field, giving particular attention to ant colony optimization, the currently most successful example of ant algorithms, as well as to some other promising directions such as ant algorithms inspired by labor division and brood sorting.

1 Introduction

Models based on self-organization have recently been introduced by ethologists to study collective behavior in social insects [2, 3, 5, 14]. While the main motivation for the development of these models was to understand how complex behavior at the colony level emerges out of interactions among individual insects, computer scientists have recently started to exploit these models as an inspiration for the design of useful optimization and distributed control algorithms. For example, a model of cooperative foraging in ants has been transformed into a set of optimization algorithms, now known as ant colony optimization (ACO) [22, 24], capable of tackling very hard computational problems, such as the traveling salesman [28, 20, 29, 26, 60], the quadratic assignment problem [47, 46, 35, 61], the sequential ordering problem [33], the shortest common supersequence problem [49], various scheduling problems [57, 12, 48], and many others (see Table 1). More recently, ACO has also been successfully applied to distributed control problems such as adaptive routing in communications networks [54, 17]. Another model, initially introduced to explain brood sorting in ants, was used by computer scientists to devise a distributed algorithm for data clustering [43, 39]. And a model of flexible task allocation in wasps has become a distributed

[0] Published in *Advances in Artificial Life*, Proceedings of the Sixth European Conference on Artificial Life, LNAI 2159, Springer-Verlag, pp. 11–22.

algorithm for dynamic scheduling or resource allocation in a factory or a computer network [9]. This line of research, termed *ant algorithms* [21, 23, 27] or *swarm intelligence* [2, 4] has also met the interest of roboticists for the design of distributed algorithms for the control of swarms of robots.

In the following we will overview the main results obtained in the field of ant algorithms. In Section 2 we introduce the ACO metaheuristic, currently the most successful example of ant algorithms, using the classical traveling salesman problem as an example, and we overview the results obtained on a number of problems. In Section 3 we briefly present other types of ant algorithms that, although still in the phase of exploratory research, look promising and maybe one day will be as successful as ACO has shown to be.

2 From real ants to ant colony optimization

An important insight of early research on ants' behavior was that in many ant species the visual perceptive faculty is very rudimentarily developed (there are even ant species which are completely blind) and that most communication among individuals, or between individuals and their environment, is based on the use of chemicals, called *pheromones*, produced by the ants. Particularly important for the social life of some ant species is the *trail pheromone*, a pheromone that individuals deposit while walking in search for food. By sensing pheromone trails, foragers can follow the path to food discovered by other ants. This collective pheromone-laying/pheromone-following behavior whereby an ant is influenced by a chemical trail left by other ants was the inspiring source of ant colony optimization, as explained in the following by means of the double bridge experiment.

2.1 The double bridge experiment

The double bridge experiment [36] is an important experiment in the field of ant algorithms. In fact, it gave the initial inspiration [20, 28] to all the research work that led to the definition of the ACO metaheuristic. In the double bridge experiment, see Figure 1, an ant nest is connected to a food source via two paths of different length. At start time all ants are in the nest and they are left free to move. The experimental apparatus is built in such a way that the only way for the ants to reach the food is by using one of the two bridge branches. In the initial phase the ants move randomly and they choose between the shorter and the longer branch with equal probability. While walking ants deposit on the ground a pheromone trail; when choosing their way, ants choose with higher probability those directions marked by a stronger pheromone concentration. As those ants choosing the shorter branch will also be the first to find the food and to go back to the nest, the pheromone trail on the shorter branch will grow faster, increasing this way the probability that it will be used by forthcoming ants. This auto-catalytic (positive feedback) process is at the heart of the auto-organizing behavior that very quickly leads all the ants to choose the shortest

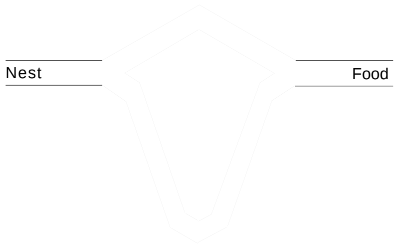

Fig. 1. Experimental setup for the double bridge experiment. Modified from [14, 36].

branch. A similar mechanism can be used by opportunely defined *artificial ants* to find minimum cost paths on graphs, as explained in the following.

2.2 Artificial ants for the traveling salesman problem

To illustrate how artificial ants can solve optimization problems, we consider an easy-to-formulate but hard-to-solve combinatorial problem: the well-known traveling salesman problem (TSP), a problem known to be $\mathcal{NP}$-hard.

Consider a set of cities and a set of weighted edges (the edges represent direct connections between pairs of cities and the weights may represent, for example, the distance between the connected cities) such that the induced graph is connected. The TSP is easily described as follows: find the shortest tour that visits each city in the set once and only once. The (constructive and stochastic) algorithm followed by the artificial ants exploits virtual pheromone concentrations associated to the edges connecting pairs of cities: artificial ants build tours, selecting which next city to hop to among a list of non-visited cities depending on city distance and on pheromone concentration. The shorter tours are reinforced by increasing the pheromone concentrations. Then evaporation, which consists in decreasing the pheromone values, is applied to all edges. After a number of iterations, very good solutions are discovered. In addition to finding a very good solution, the algorithm maintains a pool of alternative portions of solutions: this feature may become particularly interesting when the problem is dynamically changing (as most real-world problems are), since the algorithm can focus the search toward this pool of alternative portions of solutions.

The traveling salesman problem obviously lends itself to an ant-based description. But many other optimization problems can be solved with the same approach, because they can be formulated as minimum cost path problems on graphs: the ant-based approach has been shown to be extremely efficient on structured (real-world) instances of the quadratic assignment problem, on the sequential ordering problem, the vehicle routing problem, the shortest common supersequence problem, and many others (see Table 1).

```
procedure ACO metaheuristic
    ScheduleActivities {possibly in parallel}
        ManageAntsActivity()
        EvaporatePheromone()
        DaemonActions() {Optional}
    end ScheduleActivities
end ACO metaheuristic
```

Fig. 2. The ACO metaheuristic in pseudo-code. Comments are enclosed in braces. The **ScheduleActivities** construct may be executed sequentially, as typically happens in combinatorial optimization problems, or in parallel, as done for example in routing applications. The procedure DaemonActions() is optional and refers to centralized actions executed by a daemon possessing global knowledge.

2.3 The ant colony optimization metaheuristic

Although the algorithms developed for the above-mentioned applications differ in many details among themselves, still their artificial ants share the basic behavior and cooperation mechanisms as the TSP artificial ants explained above (see also [20, 28, 29]). This fact was recently captured in the definition of a common framework, called ACO metaheuristic [22, 24]. Informally, the ACO metaheuristic (see also Figure 2) can be defined as follows (the reader interested in a more formal definition should refer to [22]).

A colony of (artificial) ants concurrently and asynchronously build solutions to a given discrete optimization problem by moving on the problem's graph representation, where each feasible path encodes a solution of the problem. They move by applying a stochastic local decision rule that exploits pheromone trail values. By moving, ants incrementally build solutions to the optimization problem. Once an ant has built a solution, or while the solution is being built, the ant evaluates the (partial) solution and deposits pheromone on the graph components it used. This pheromone information directs the search of the ants in the future.

Besides ants' activity, an ACO algorithm includes two additional procedures: *pheromone trail evaporation* and *daemon actions* (the last component being optional). Pheromone evaporation is the process by means of which the pheromone trail intensity on the components decreases over time. From a practical point of view, pheromone evaporation is needed to avoid a too rapid convergence of the algorithm towards a sub-optimal region. It implements a useful form of *forgetting*, favoring the exploration of new areas of the search space. Daemon actions can be used to implement centralized actions which cannot be performed by single ants. Examples are the activation of a local optimization procedure, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a non-local

Table 1. Some of the current applications of ACO algorithms. Applications are listed by class of problems and in chronological order.

Problem name	*Authors*	*Algorithm name*	*Year*	*Main references*
Traveling salesman	Dorigo, Maniezzo & Colorni	AS	1991	[20, 28, 29]
	Gambardella & Dorigo	Ant-Q	1995	[30]
	Dorigo & Gambardella	ACS & ACS-3-opt	1996	[25, 26, 31]
	Stützle & Hoos	$\mathcal{MM}$AS	1997	[60, 58, 61]
	Bullnheimer, Hartl & Strauss	AS$_{rank}$	1997	[8]
Quadratic assignment	Maniezzo, Colorni & Dorigo	AS-QAP	1994	[47]
	Gambardella, Taillard & Dorigo	HAS-QAP[a]	1997	[35]
	Stützle & Hoos	$\mathcal{MM}$AS-QAP	1997	[56, 61]
	Maniezzo	ANTS-QAP	1998	[44]
	Maniezzo & Colorni	AS-QAP[b]	1999	[46]
Scheduling problems	Colorni, Dorigo & Maniezzo	AS-JSP	1994	[10]
	Stützle	AS-FSP	1997	[57]
	Bauer et *al.*	ACS-SMTTP	1999	[1]
	den Besten, Stützle & Dorigo	ACS-SMTWTP	1999	[12]
	Merkle, Middendorf & Schmeck	ACO-RCPS	2000	[48]
Vehicle routing	Bullnheimer, Hartl & Strauss	AS-VRP	1997	[6, 7]
	Gambardella, Taillard & Agazzi	HAS-VRP	1999	[34]
Connection-oriented network routing	Schoonderwoerd *et al.*	ABC	1996	[54, 53]
	Di Caro & Dorigo	AntNet-FS	1998	[18]
Connection-less network routing	Di Caro & Dorigo	AntNet & AntNet-FA	1997	[16, 17, 19]
Sequential ordering	Gambardella & Dorigo	HAS-SOP	1997	[32, 33]
Graph coloring	Costa & Hertz	ANTCOL	1997	[11]
Shortest common supersequence	Michel & Middendorf	AS-SCS	1998	[49, 50]
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998	[45]
Generalized assignment	Ramalhinho Lourenço & Serra	MMAS-GAP	1998	[52]
Multiple knapsack	Leguizamón & Michalewicz	AS-MKP	1999	[41]
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999	[51]
Redundancy allocation	Liang & Smith	ACO-RAP	1999	[42]
Constraint satisfaction	Solnon	Ant-P-solver	2000	[55]

[a] HAS-QAP is an ant algorithm which does not follow all the aspects of the ACO metaheuristic.

[b] This is a variant of the original AS-QAP.

perspective. As a practical example, the daemon can choose to deposit extra pheromone on the components used by the ant that built the best solution.

It is interesting to note that the ACO approach has a feature which makes it particularly appealing: it is explicitly formulated in terms of computational agents. While it may in principle be possible to get rid of the agents to focus on the core optimizing mechanism (reinforcement and evaporation), the agent-based formulation may prove to be a useful aid for designing problem-solving systems. Routing in communications networks is a very good example of this aspect. Routing is the mechanism that directs messages in a communications network from their source nodes to their destination nodes through a sequence of intermediate nodes or switching stations. Each switching station has a routing table that tells messages or portions of messages called packets where to go given their destinations. Because of the highly dynamic nature of communications networks due to the time-varying stochastic changes in network load, as well as to unpredictable failures of network components, areas of the network may become congested and new routes have to be discovered dynamically. In the ant-based approach [17, 54] ant-like agents reinforce routing table entries depending on their experience in the network: for example, if an agent has been delayed a long time because it went through a highly congested area of the network, it will only weakly, or not at all, reinforce routing table entries that send packets to that area of the network. A forgetting (or evaporation) mechanism is also applied regularly to refresh the system (to avoid obsolete solutions being maintained). AntNet [17], an ACO algorithm designed for routing in packet-switched networks, was shown to outperform (in realistically simulated conditions) all routing algorithms in widespread use, especially, but not only, in strongly variable traffic conditions. Implicitly maintaining a pool of alternative partial routes is the way the system copes with changing conditions and allows it to be flexible and robust.

3 Other applications inspired by social insects

As we said, ant colony optimization algorithms are only one, although the most successful, example of ant algorithms. Wagner et al. have proposed two algorithms also inspired by the foraging behavior of ant colonies for the exploration of a graph. Other researchers have taken inspiration from other social insect behaviors, such as division of labor, brood sorting and cemetery organization, to propose new types of distributed, multi-agent algorithms, as explained in the following.

3.1 Foraging and graph exploration

Taking inspiration from the pheromone-laying/pheromone-following behavior of ant colonies, Wagner et al. [62, 63] have proposed two algorithms for exploring a graph called respectively *Edge Ant Walk* [62] and *Vertex Ant Walk* [63] in which one or more artificial ants walk along the edges of the graph, lay a pheromone trail on the visited edges (respectively nodes) and use the pheromone trails deposited by previous ants to direct their exploration. Although the general idea behind the algorithm is similar to the one that inspired ant colony optimization, their goal and implementation are very different. In the work of Wagner et al., pheromone trail is used as a kind of distributed memory that directs the ants towards unexplored areas of the search space. In fact, their goal is to cover the graph, that is to visit all the nodes, without knowing the graph topology. They were able to prove a number of theoretical results, for example concerning the time complexity for covering a generic graph. Also, they recently extended their algorithms [64] so that they can be applied to dynamically changing graphs. A possible and promising application of this work is to Internet search, where the problem is to track down the hundreds of thousands of pages added every day [40] (as well as the ones that disappear).

3.2 Division of labor and dynamic task allocation in robots

Division of labor is an important and widespread feature of colonial life in many species of social insects. In ant colonies, for example, workers and soldier ants are typically concerned with nest maintenance and nest defense, respectively. Still, individuals of one class can, if necessary, perform activities typical of the other class. This self-organizing, adaptive aspect of labor division can be explained by a simple behavioral threshold model: although each ant is equipped with a complete set of behavioral responses, different ants have different threshold for different behaviors. For example, a soldier generally has a low threshold for defense related activities and a high threshold for nest maintenance duties, while for workers it is just the opposite. An high demand of workers, due for example to the sudden need for extra brood care, can lead to the involvement of soldiers in this, for them atypical, activity. Interestingly, ants in a same class have similar, but not identical, threshold levels for the same activity. This differentiation determines a continuum in the space of behavioral responses so that the ant colony as a whole can adapt to continuously changing environmental conditions. Krieger and Billeter [38] used the threshold model of ants to define a distributed labor division system for a group of robots that have the goal of collecting items dispersed in an arena. Their experiments have shown that robots governed by threshold-based behavior activation can accomplish the task, and that the system as a whole presents inherent fault-tolerance and graceful degradation of performance: the faulty behavior of one or more robots causes the automatic adaptation in the behavior of the other robots with only minor decreases in overall performance. Most important, this result was accomplished without designing any explicit mechanism of communication among robots, and without the faulty situations being explicitly included in the robots control programs. Although the experiments were run with small toy robots in university lab experimental conditions, it seems clear that the approach has great potentialities for the control of fleets of industrial robots in unstructured environments.

3.3 Brood sorting, cemetery organization and data clustering

Another typical activity that can be observed in ant colonies is clustering of objects. For example, ants cluster corpses of dead ants into cemeteries, or food items into the nest. The clustering activity is performed in a completely distributed way, that is, without any central control mechanism. The way ants accomplish this can be explained by a simple model, similar to the one used for labor division. In the model, ants are endowed with "pick up" and "drop" behaviors and these behaviors are activated with probabilities that are function of both a threshold and of environmental conditions [15]. The environmental condition is in this case given by the density of items in the neighborhood of the ants location. While moving around, an ant carrying an item has a high probability to drop it in zones where a high density of the same item is found, but a low probability to drop it in zones where there is a low density. On the contrary, when an unloaded ant meets an item, it will pick it up with high probability if the zone in which it is located has a low density of that same item, with low probability otherwise. If more kinds of items are present and ants use different thresholds for different items, these are sorted into different clusters. This simple mechanism has been put to work for data visualization and clustering. Kuntz et al. [39], building on an algorithm first proposed by Lumer and Faieta [43], consider the following problem. Given a set of n-dimensional data represented as points in a n-dimensional space, and a metric d which measures the distance between pairs of data items, project the points on a plane so that points in the plane belong to a same cluster if and only if the corresponding data items are similar in the n-dimensional space under the metric d. To let artificial ants solve this problem, the initial projection of data items on the plane is done randomly. Then artificial ants move randomly on the plane and pick up or drop projected data items using rules equivalent to those of the threshold model explained above. The results obtained are qualitatively comparable to those obtained by more classic techniques such as spectral decomposition or stress minimization, but at a much lower computational cost. Moreover, the technique can be easily extended to other difficult problems such as multidimensional scaling (i.e., the problem of transforming a squared matrix of distances among pairs of points into the coordinate of the original points), or data sorting.

4 Conclusions

The researchers' interest in ant algorithms has recently greatly increased, due to both the charm of the ant colony metaphor and the excitement caused by some very promising results obtained in practical applications. Although the approach is very promising, a more systematic comparison with other heuristics is required. We also need to better understand why ant algorithms work so well on certain types of problems, and to clearly identify the problem characteristics which make a problem susceptible of being successfully tackled by an ant algorithm. Finally, results on the theoretical properties of these algorithms are most of the times missing. A notable exception concerns some important instances of ACO algorithms, for which convergence to the optimal solution has recently been proved [37, 59].

Acknowledgments

This work was partially supported by the "Metaheuristics Network", a Research Training Network funded by the Improving Human Potential programme of the CEC, grant HPRN-CT-1999-00106. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Senior Research Associate.

References

1. A. Bauer, B. Bullnheimer, R. F. Hartl, and C. Strauss. An ant colony optimization approach for the single machine total tardiness problem. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pages 1445–1450. IEEE Press, Piscataway, NJ, 1999.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, NJ, 1999.
3. E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from social insect behavior. *Nature*, 406:39–42, 2000.
4. E. Bonabeau and G. Theraulaz. Swarm smarts. *Scientific American*, 282(3):54–61, 2000.
5. E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine. Self-organization in social insects. *Tree*, 12(5):188–193, 1997.
6. B. Bullnheimer, R. F. Hartl, and C. Strauss. Applying the Ant System to the vehicle routing problem. In S. Voß, S. Martello, I. H. Osman, and C. Roucairol, editors, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 285–296. Kluwer Academic Publishers, Dordrecht, 1999.
7. B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89:319–328, 1999.
8. B. Bullnheimer, R. F. Hartl, and C. Strauss. A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics*, 7(1):25–38, 1999.
9. M. Campos, E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg. Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, 8(3):83–96, 2000.
10. A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. Ant System for job-shop scheduling. *JORBEL – Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1):39–53, 1994.
11. D. Costa and A. Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1997.
12. M. L. den Besten, T. Stützle, and M. Dorigo. Ant colony optimization for the total weighted tardiness problem. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-S. Schwefel, editors, *Proceedings of PPSN-VI, Sixth International Conference on Parallel Problem Solving from Nature*, volume 1917 of *Lecture Notes in Computer Science*, pages 611–620. Springer Verlag, Berlin, Germany, 2000.

13. M. L. den Besten, T. Stützle, and M. Dorigo. Design of iterated local search algorithms: An example application to the single machine total weighted tardiness problem. In E. J. W. Boers, J. Gottlieb, P. L. Lanzi, R. E. Smith, S. Cagnoni, E. Hart, G. R. Raidl, and H. Tijink, editors, *Proceedings of EvoStim'01*, Lecture Notes in Computer Science, pages 441–452. Springer Verlag, Berlin, Germany, 2001.
14. J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3:159–168, 1990.
15. J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In J.-A. Meyer and S. W. Wilson, editors, *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–363. MIT Press, Cambridge, MA, 1991.
16. G. Di Caro and M. Dorigo. AntNet: A mobile agents approach to adaptive routing. Technical Report IRIDIA/97-12, IRIDIA, Université Libre de Bruxelles, Belgium, 1997.
17. G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.
18. G. Di Caro and M. Dorigo. Extending AntNet for best-effort Quality-of-Service routing. Unpublished presentation at *ANTS'98 – From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimization* http://iridia.ulb.ac.be/ants98/ants98.html, October 15–16 1998.
19. G. Di Caro and M. Dorigo. Two ant colony algorithms for best-effort routing in datagram networks. In Y. Pan, S. G. Akl, and K. Li, editors, *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pages 541–546. IASTED/ACTA Press, Anheim, 1998.
20. M. Dorigo. *Optimization, Learning and Natural Algorithms* (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992. 140 pages.
21. M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871, 2000.
22. M. Dorigo and G. Di Caro. The Ant Colony Optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw Hill, London, UK, 1999.
23. M. Dorigo, G. Di Caro, and T. Stützle (Editors). Special issue on "Ant Algorithms". *Future Generation Computer Systems*, 16(8), 2000. 104 pages.
24. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
25. M. Dorigo and L. M. Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*, 43:73–81, 1997.
26. M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
27. M. Dorigo, L. M. Gambardella, M. Middendorf, and T. Stützle (Editors). Special issue on "Ant Algorithms and Swarm Intelligence". *IEEE Transactions on Evolutionary Computation*, 2002.
28. M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.
29. M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1):29–41, 1996.

30. L. M. Gambardella and M. Dorigo. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning (ML-95)*, pages 252–260. Morgan Kaufmann Publishers, Palo Alto, CA, 1995.
31. L. M. Gambardella and M. Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 622–627. IEEE Press, Piscataway, NJ, 1996.
32. L. M. Gambardella and M. Dorigo. HAS-SOP: An hybrid Ant System for the sequential ordering problem. Technical Report IDSIA-11-97, IDSIA, Lugano, Switzerland, 1997.
33. L. M. Gambardella and M. Dorigo. Ant Colony System hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3):237–255, 2000.
34. L. M. Gambardella, È. D. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 63–76. McGraw Hill, London, UK, 1999.
35. L. M. Gambardella, È. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2):167–176, 1999.
36. S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76:579–581, 1989.
37. W. J. Gutjahr. A graph-based Ant System and its convergence. *Future Generation Computer Systems*, 16(8):873–888, 2000.
38. M. J. B. Krieger and J.-B. Billeter. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30:65–84, 2000.
39. P. Kuntz, D. Snyers, and P. Layzell. A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning. *Journal of Heuristics*, 5:327–351, 1999.
40. S. Lawrence and C. L. Giles. Searching the world wide web. *Science*, 280:98–100, 1998.
41. G. Leguizamón and Z. Michalewicz. A new version of Ant System for subset problems. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pages 1459–1464. IEEE Press, Piscataway, NJ, 1999.
42. Y.-C. Liang and A. E. Smith. An Ant System approach to redundancy allocation. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1478–1484. IEEE Press, Piscataway, NJ, 1999.
43. E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In J.-A. Meyer and S. W. Wilson, editors, *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, pages 501–508. MIT Press, Cambridge, MA, 1994.
44. V. Maniezzo. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal on Computing*, 11(4):358–369, 1999.
45. V. Maniezzo and A. Carbonaro. An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, 16(8):927–935, 2000.
46. V. Maniezzo and A. Colorni. The Ant System applied to the quadratic assignment problem. *IEEE Transactions on Data and Knowledge Engineering*, 11(5):769–778, 1999.

47. V. Maniezzo, A. Colorni, and M. Dorigo. The Ant System applied to the quadratic assignment problem. Technical Report IRIDIA/94-28, IRIDIA, Université Libre de Bruxelles, Belgium, 1994.
48. D. Merkle, M. Middendorf, and H. Schmeck. Ant colony optimization for resource-constrained project scheduling. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 893–900. Morgan Kaufmann Publishers, San Francisco, CA, 2000.
49. R. Michel and M. Middendorf. An island model based Ant System with lookahead for the shortest supersequence problem. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 692–701. Springer Verlag, Berlin, Germany, 1998.
50. R. Michel and M. Middendorf. An ACO algorithm for the shortest supersequence problem. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 51–61. McGraw Hill, London, UK, 1999.
51. G. Navarro Varela and M. C. Sinclair. Ant colony optimisation for virtual-wavelength-path routing and wavelength allocation. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pages 1809–1816. IEEE Press, Piscataway, NJ, 1999.
52. H. Ramalhinho Lourenço and D. Serra. Adaptive approach heuristics for the generalized assignment problem. Technical Report Technical Report Economic Working Papers Series No.304, Universitat Pompeu Fabra, Dept. of Economics and Management, Barcelona, Spain, 1998.
53. R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunications networks. In *Proceedings of the First International Conference on Autonomous Agents*, pages 209–216. ACM Press, 1997.
54. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2):169–207, 1996.
55. C. Solnon. Solving permutation constraint satisfaction problems with artificial ants. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 118–122. IOS Press, Amsterdam, The Netherlands, 2000.
56. T. Stützle. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System for the quadratic assignment problem. Technical Report AIDA-97-4, FG Intellektik, FB Informatik, TU Darmstadt, July 1997.
57. T. Stützle. An ant approach to the flow shop problem. In *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, volume 3, pages 1560–1564. Verlag Mainz, Wissenschaftsverlag, Aachen, 1998.
58. T. Stützle. *Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications.* Infix, Sankt Augustin, Germany, 1999.
59. T. Stützle and M. Dorigo. A short convergence proof for a class of aco algorithms. Technical Report IRIDIA/2000-35, IRIDIA, Université Libre de Bruxelles, Belgium, 2000.
60. T. Stützle and H. H. Hoos. The $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System and local search for the traveling salesman problem. In T. Bäck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 309–314. IEEE Press, Piscataway, NJ, 1997.
61. T. Stützle and H. H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. *Future Generation Computer Systems*, 16(8):889–914, 2000.
62. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Smell as a computational resource – a lesson we can learn from the ant. In *Proceedings of the Fourth Israeli Symposium on Theory of Computing and Systems (ISTCS-99)*, pages 219–230, 1996.

63. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Efficient graph search by a smell-oriented vertex process. *Annals of Mathematics and Artificial Intelligence*, 24:211–223, 1998.

64. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. ANTS: Agents, networks, trees and subgraphs. *Future Generation Computer Systems*, 16(8):915–926, 2000.