Ant Colony Optimization

Article *in* IEEE Computational Intelligence Magazine · December 2006

DOI: 10.1109/MCI.2006.329691

3 authors:

Marco Dorigo
Université Libre de Bruxelles

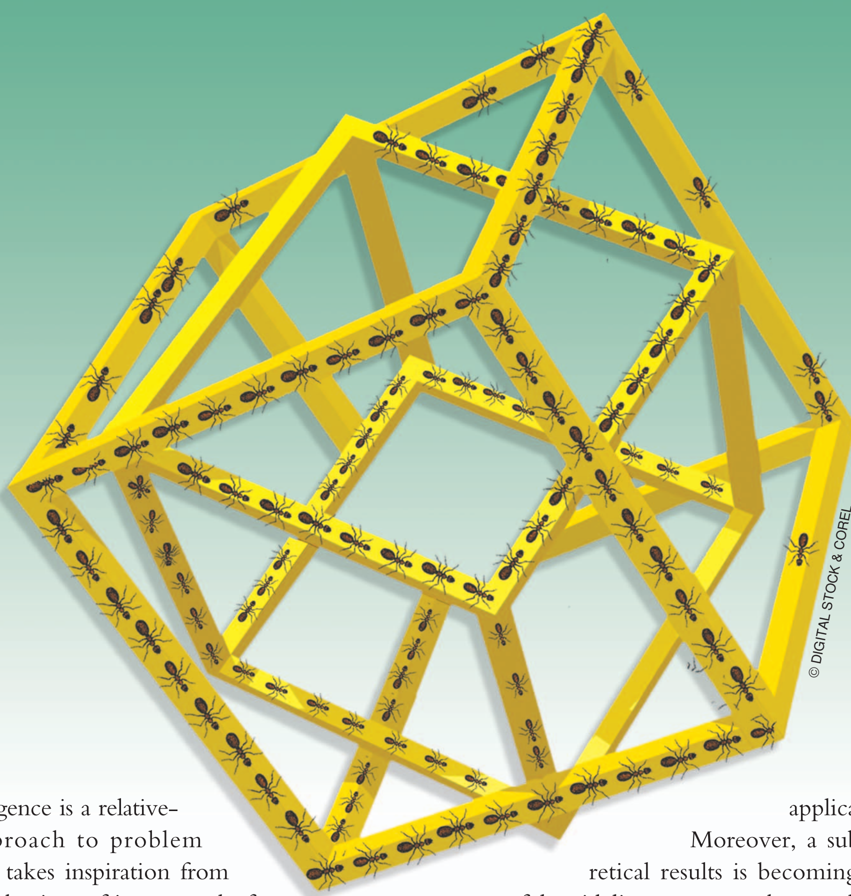
Mauro Birattari
Université Libre de Bruxelles

Thomas Stützle
Université Libre de Bruxelles

Marco Dorigo, Mauro Birattari, and Thomas Stützle
Université Libre de Bruxelles, BELGIUM

Ant Colony Optimization

Artificial Ants as a Computational Intelligence Technique



Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects and of other animals. In particular, ants have inspired a number of methods and techniques among which the most studied and the most successful is the general purpose optimization technique known as ant colony optimization.

Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems.

From the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of increasing numbers of researchers and many successful applications are now available. Moreover, a substantial corpus of theoretical results is becoming available that provides useful guidelines to researchers and practitioners in further applications of ACO.

The goal of this article is to introduce ant colony optimization and to survey its most notable applications. Section I provides some background information on the foraging behavior of ants. Section II describes ant colony optimization and its main variants. Section III surveys the most notable theoretical results concerning ACO, and Section IV illustrates some of its most successful applications. Section V highlights some currently active research topics, and Section VI provides an overview of some other algorithms that, although not directly related to ACO, are nonetheless inspired by the behavior of ants. Section VII concludes the article.

1556-603X/06/$20.00©2006IEEE

I. Biological Inspiration

In the forties and fifties of the twentieth century, the French entomologist Pierre-Paul Grassé [1] observed that some species of termites react to what he called "significant stimuli". He observed that the effects of these reactions can act as new significant stimuli for both the insect that produced them and for the other insects in the colony. Grassé used the term *stigmergy* [2] to describe this particular type of communication in which the "workers are stimulated by the performance they have achieved".

The two main characteristics of stigmergy that differentiate it from other forms of communication are the following.

- Stigmergy is an indirect, non-symbolic form of communication mediated by the environment: insects exchange information by modifying their environment; and
- Stigmergic information is local: it can only be accessed by those insects that visit the locus in which it was released (or its immediate neighborhood).

Examples of stigmergy can be observed in colonies of ants. In many ant species, ants walking to and from a food source deposit on the ground a substance called *pheromone*. Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a remarkably effective way.

Deneubourg et al. [3] thoroughly investigated the pheromone laying and following behavior of ants. In an experiment known as the "double bridge experiment", the nest of a colony of Argentine ants was connected to a food source by two bridges of equal lengths [see Figure 1(a)]. In such a setting, ants start to explore the surroundings of the nest and eventually reach the food source. Along their path between food source and nest, Argentine ants deposit pheromone. Initially, each ant randomly chooses one of the two bridges. However, due to random fluctuations, after some time one of the two bridges presents a higher concentration of pheromone than the other and, therefore, attracts more ants. This brings a further amount of pheromone on that bridge making it more attractive with the result that after some time the whole colony converges toward the use of the same bridge.[1]

This colony-level behavior, based on autocatalysis, that is, on the exploitation of positive feedback, can be used by ants to find the shortest path between a food source and their nest. Goss et al. [4] considered a variant of the double bridge experiment in which one bridge is significantly longer than the other [see Figure 1(b)]. In this case, the stochastic fluctuations in the initial choice of a bridge are much reduced and a second mechanism plays an important role: the ants choosing by chance the short bridge are the first to reach the nest. The short bridge receives, therefore, pheromone earlier than the long one and this fact increases the probability that further ants select it rather than the long one. Goss et al. [4] developed a model of the observed behavior: assuming that at a given moment in time m_1 ants have used the first bridge and m_2 the second one, the probability p_1 for an ant to choose the first bridge is:

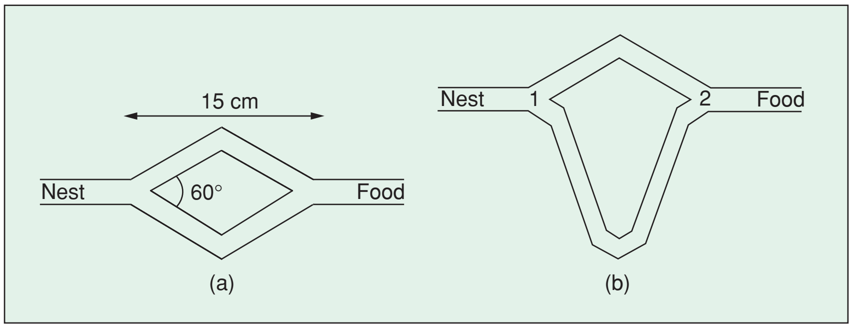


FIGURE 1 Experimental setup for the double bridge experiment. (a) Branches have equal lengths [3]. (b) Branches have different lengths [4].

TABLE 1 A non-exhaustive list of successful ant colony optimization algorithms (in chronological order).

ALGORITHM	AUTHORS	YEAR	REFERENCES
ANT SYSTEM (AS)	DORIGO ET AL.	1991	[6]–[8]
ELITIST AS	DORIGO ET AL.	1992	[7], [8]
ANT-Q	GAMBARDELLA & DORIGO	1995	[9]
ANT COLONY SYSTEM	DORIGO & GAMBARDELLA	1996	[10]–[12]
$\mathcal{MAX}$-$\mathcal{MIN}$ AS	STÜTZLE & HOOS	1996	[13]–[15]
RANK-BASED AS	BULLNHEIMER ET AL.	1997	[16], [17]
ANTS	MANIEZZO	1999	[18]
BWAS	CORDON ET AL.	2000	[19]
HYPER-CUBE AS	BLUM ET AL.	2001	[20], [21]

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}, \qquad (1)$$

where parameters k and h are to be fitted to the experimental data—obviously, $p_2 = 1 - p_1$. Monte Carlo simulations showed a very good fit for $k \approx 20$ and $h \approx 2$ [5].

II. The Optimization Technique

The model proposed by Deneubourg and co-workers for explaining the foraging behavior of ants was the main source of inspiration for the development of ant colony optimization. In ACO, a number of artificial ants build solutions to the considered optimization problem at hand and exchange

[1]Deneubourg and co-workers repeated the experiment a number of times and observed that each of the two bridges is used in about 50% of the cases.

information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants.

Different ant colony optimization algorithms have been proposed. The original ant colony optimization algorithm is known as Ant System [6]–[8] and was proposed in the early nineties. Since then, a number of other ACO algorithms were introduced. (See Table 1 for a non-exhaustive list of successful variants.) All ant colony optimization algorithms share the same idea, which is best illustrated through an example of how ACO algorithms can be applied. Section II-A describes in simple terms how a generic ACO algorithm is applied to the well-known traveling salesman problem, and Section II-B gives a more formal description of ACO.

In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is reminiscent of the one adopted by real ants.

A. ACO for the Traveling Salesman Problem

In the traveling salesman problem, a set of cities is given and the distance between each of them is known. The goal is to find the shortest tour that allows each city to be visited once and only once. In more formal terms, the goal is to find a Hamiltonian tour of minimal length on a fully connected graph.

In ant colony optimization, the problem is tackled by simulating a number of artificial ants moving on a graph that encodes the problem itself: each vertex represents a city and each edge represents a connection between two cities. A variable called pheromone is associated with each edge and can be read and modified by ants.

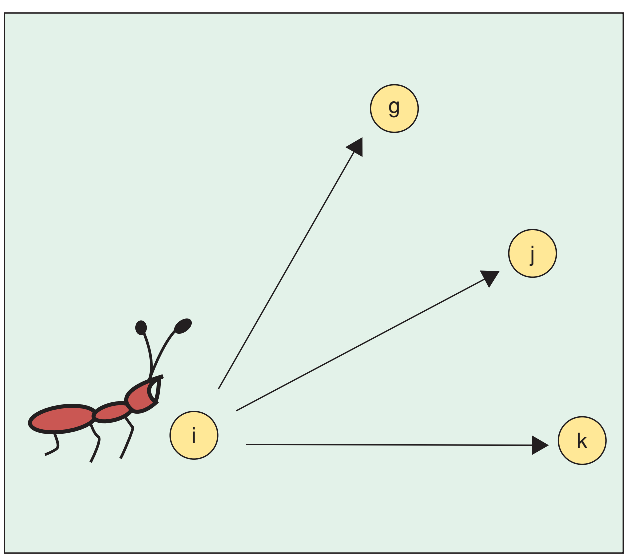


FIGURE 2 An ant in city *i* chooses the next city to visit via a stochastic mechanism: if *j* has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge (i, j).

Ant colony optimization is an iterative algorithm. At each iteration, a number of artificial ants are considered. Each of them builds a solution by walking from vertex to vertex on the graph with the constraint of not visiting any vertex that she has already visited in her walk. At each step of the solution construction, an ant selects the following vertex to be visited according to a stochastic mechanism that is biased by the pheromone: when in vertex i, the following vertex is selected stochastically among the previously unvisited ones (see Figure 2). In particular, if j has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge (i, j).

At the end of an iteration, on the basis of the quality of the solutions constructed by the ants, the pheromone values are modified in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed.

B. The Ant Colony Optimization Metaheuristic

Ant colony optimization has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo and co-workers [22], [23]. A *metaheuristic* is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. Examples of metaheuristics include simulated annealing [24], [25], tabu search [26]–[28], iterated local search [29], evolutionary computation [30]–[33], and ant colony optimization [8], [22], [23], [34].

In order to apply ACO to a given a combinatorial optimization problem, an adequate model is needed:

A combinatorial optimization problem

A *model* $P = (\mathbf{S}, \Omega, f)$ *of a combinatorial optimization problem consists of:*

- a *search space* $\mathbf{S}$ *defined over a finite set of discrete decision variables* X_i, $i = 1, \ldots, n$;
- *a set* Ω *of constraints among the variables; and*
- *an objective function* $f : \mathbf{S} \to \mathbb{R}_0^+$ *to be minimized.*[2]

The generic variable X_i *takes values in* $\mathbf{D}_i = \{v_i^1, \ldots, v_i^{|\mathbf{D}_i|}\}$. *A feasible solution* $s \in \mathbf{S}$ *is a complete assignment of values to variables that satisfies all constraints in* Ω. *A solution* $s^* \in \mathbf{S}$ *is called a global optimum if and only if:* $f(s^*) \le f(s)\ \forall s \in \mathbf{S}$.

The model of a combinatorial optimization problem is used to define the pheromone model of ACO. A pheromone value is associated with each possible *solution component*; that is, with each possible assignment of a value to a variable. Formally, the

[2] Any maximization problem can be trivially reduced to a minimization problem: maximizing a given function g is clearly equivalent to minimizing $f = -g$.

pheromone value τ_{ij} is associated with the solution component c_{ij}, which consists of the assignment $X_i = v_i^j$. The set of all possible solution components is denoted by $\mathbf{C}$.

In ACO, an artificial ant builds a solution by traversing the fully connected *construction graph* $G_C(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of vertices and $\mathbf{E}$ is a set of edges. This graph can be obtained from the set of solution components $\mathbf{C}$ in two ways: components may be represented either by vertices or by edges. Artificial ants move from vertex to vertex along the edges of the graph, incrementally building a *partial solution*. Additionally, ants deposit a certain amount of pheromone on the components; that is, either on the vertices or on the edges that they traverse. The amount $\Delta\tau$ of pheromone deposited may depend on the quality of the solution found. Subsequent ants use the pheromone information as a guide toward promising regions of the search space.

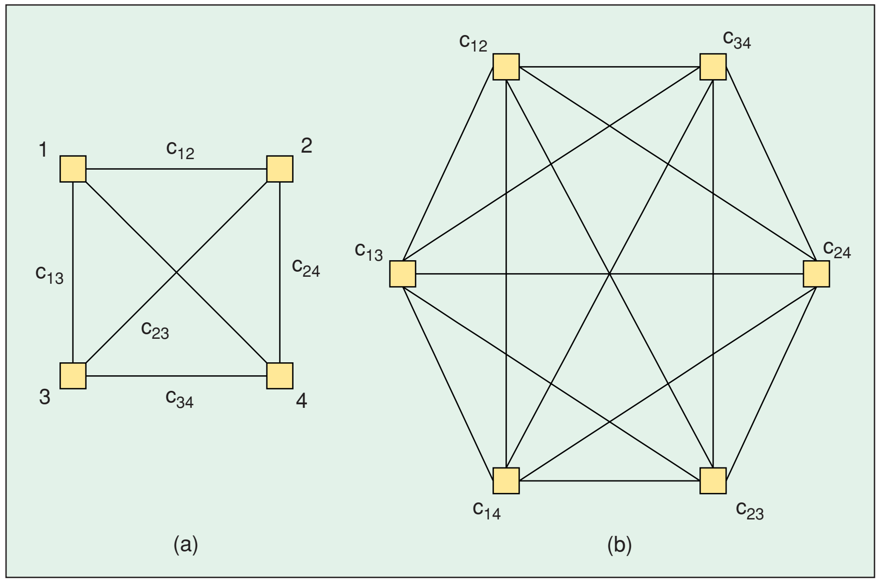


FIGURE 3 Example of possible construction graphs for a four-city TSP where components are associated with (a) the edges or with (b) the vertices of the graph.

In the traveling salesman problem, a solution can be represented through a set of n variables, where n is the number of cities. Each of these variables is associated with a city. The variable X_i indicates the city to be visited after city i. Here, solution components are pairs of cities to be visited one after the other, in the given order: the solution component $c_{ij} = (i, j)$ indicates that the solution under analysis prescribes that city j should be visited immediately after city i. In this case, the construction graph is a graph in which the vertices are the cities of the original traveling salesman problem, and the edges are solution components. As a consequence, ants deposit pheromone on the edges of the construction graph.

It should be noticed that the construction graph could be obtained by representing solution components as vertices on which pheromone is deposited. Although this second way of obtaining a construction graph seems less natural for the traveling salesman problem, it is nonetheless correct. The two ways of defining the construction graph for a four-city traveling salesman problem are represented in Figure 3.

The ACO metaheuristic is shown in Algorithm 1. After initialization, the metaheuristic iterates over three phases: at each iteration, a number of solutions are constructed by the ants; these solutions are then improved through a local search (this step is optional), and finally the pheromone is updated. The following is a more detailed description of the three phases:

ConstructAntSolutions: A set of m artificial ants constructs solutions from elements of a finite set of available solution components $\mathbf{C} = \{c_{ij}\},\ i = 1, \ldots, n,\ j = 1, \ldots, |\mathbf{D}_i|$. A solution construction starts from an empty partial solution $s^p = \emptyset$. At each construction step, the partial solution s^p is extended by adding a feasible solution component from the set $\mathbf{N}(s^p) \subseteq \mathbf{C}$, which is defined as the set of components that can be added to the current partial solution s^p without violating any of the constraints in Ω. The process of constructing solutions can be regarded as a walk on the construction graph $G_C = (\mathbf{V}, \mathbf{E})$.

Algorithm 1 The Ant Colony Optimization Metaheuristic

Set parameters, initialize pheromone trails
while termination condition not met **do**
  ConstructAntSolutions
  ApplyLocalSearch (optional)
  UpdatePheromones
endwhile

The choice of a solution component from $\mathbf{N}(s^p)$ is guided by a stochastic mechanism, which is biased by the pheromone associated with each of the elements of $\mathbf{N}(s^p)$. The rule for the stochastic choice of solution components vary across different ACO algorithms but, in all of them, it is inspired by the model of the behavior of real ants given in Equation 1.

ApplyLocalSearch: Once solutions have been constructed, and before updating the pheromone, it is common to improve the solutions obtained by the ants through a local search. This phase, which is highly problem-specific, is optional although it is usually included in state-of-the-art ACO algorithms.

UpdatePheromones: The aim of the pheromone update is to increase the pheromone values associated with good or promising solutions, and to decrease those that are associated with bad ones. Usually, this is achieved (i) by decreasing all the pheromone values through *pheromone evaporation*, and (ii) by increasing the pheromone levels associated with a chosen set of good solutions.

The first ant colony optimization algorithm is known as Ant System and was proposed in the early nineties. Since then, several other ACO algorithms have been proposed.

C. Main ACO Algorithms

Several ACO algorithms have been proposed in the literature. Here we present the original Ant System, and the two most successful variants: $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System and Ant Colony System. In order to illustrate the differences between these three algorithms, we use the traveling salesman problem as a concrete example.

1. Ant System (AS)

Ant System is the first ACO algorithm proposed in the literature [6]–[8]. Its main characteristic is that, at each iteration, the pheromone values are updated by *all* the m ants that have built a solution in the iteration itself. The pheromone τ_{ij}, associated with the edge joining cities i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}, \tag{2}$$

where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone laid on edge (i, j) by ant k:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ used edge } (i, j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where Q is a constant, and L_k is the length of the tour constructed by ant k.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant k is in city i and has so far constructed the partial solution s^p, the probability of going to city j is given by:

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}{\sum_{c_{il}\in\mathbf{N}(s^p)}\tau_{il}^{\alpha}\cdot\eta_{il}^{\beta}} & \text{if } c_{ij}\in\mathbf{N}(s^p), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathbf{N}(s^p)$ is the set of feasible components; that is, edges (i, l) where l is a city not yet visited by the ant k. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}}, \tag{5}$$

where d_{ij} is the distance between cities i and j.

2. $\mathcal{MAX}-\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS)

This algorithm [15] is an improvement over the original Ant System. Its characterizing elements are that only the best ant updates the pheromone trails and that the value of the pheromone is bound. The pheromone update is implemented as follows:

$$\tau_{ij} \leftarrow \left[(1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij}^{\text{best}}\right]_{\tau_{\min}}^{\tau_{\max}}, \tag{6}$$

where $\tau_{\max}$ and $\tau_{\min}$ are respectively the upper and lower bounds imposed on the pheromone; the operator $[x]_b^a$ is defined as:

$$[x]_b^a = \begin{cases} a & \text{if } x > a, \\ b & \text{if } x < b, \\ x & \text{otherwise;} \end{cases} \tag{7}$$

and $\Delta\tau_{ij}^{\text{best}}$ is:

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} 1/L_{\text{best}} & \text{if } (i, j) \text{ belongs to the best tour,} \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where L_{best} is the length of the tour of the best ant. This may be (subject to the algorithm designer decision) either the best tour found in the current iteration—*iteration-best*, L_{ib}—or the best solution found since the start of the algorithm—*best-so-far*, L_{bs}—or a combination of both.

Concerning the lower and upper bounds on the pheromone values, $\tau_{\min}$ and $\tau_{\max}$, they are typically obtained empirically and tuned on the specific problem considered [35]. Nonetheless, some guidelines have been provided for defining $\tau_{\min}$ and $\tau_{\max}$ on the basis of analytical considerations [15].

3. Ant Colony System (ACS)

The most interesting contribution of ACS [10]–[12] is the introduction of a *local pheromone update* in addition to the pheromone update performed at the end of the construction process (called *offline* pheromone update).

The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1-\varphi)\cdot\tau_{ij} + \varphi\cdot\tau_0, \tag{9}$$

where $\varphi \in (0, 1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during an iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration.

The offline pheromone update, similarly to $\mathcal{MM}$AS, is applied at the end of each iteration by only one ant, which can be either the *iteration-best* or the *best-so-far*. However, the update formula is slightly different:

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} & \text{if } (i, j) \text{ belongs to best tour,} \\ \tau_{ij} & \text{otherwise.} \end{cases} \tag{10}$$

As in $\mathcal{MM}$AS, $\Delta\tau_{ij} = 1/L_{\text{best}}$, where L_{best} can be either L_{ib} or L_{bs}.

Another important difference between ACS and AS is in the decision rule used by the ants during the construction process. In ACS, the so-called *pseudorandom proportional* rule is used: the probability for an ant to move from city i to city j depends on a random variable q uniformly distributed over [0, 1], and a parameter q_0; if $q \leq q_0$, then $j = \arg\max_{c_{il} \in N(s^p)}\{\tau_{il}\eta_{il}^{\beta}\}$, otherwise Equation 4 is used.[3]

III. Theoretical Results

The initial work on ACO has been driven by experimental work, with the aim of showing that the ideas underlying this technique can lead to successful algorithms. After this initial phase, researchers tried to deepen their understanding of the technique by building theoretical foundations.

Typically, the first question considered when dealing with metaheuristics concerns convergence: will a given ACO algorithm ever find an optimal solution? The first convergence proofs were presented by Gutjahr for an ACO algorithm called graph-based ant system (GBAS). Gutjahr proved convergence with probability $1 - \epsilon$ to the optimal solution [36], and more in general to any optimal solution [37]. GBAS is a rather peculiar ACO algorithm and the above mentioned results do not directly extend to other ACO algorithms. In particular, they do not extend to ACO algorithms that are commonly adopted in applications. Nonetheless, for two of the top performing ACO algorithms, ACS and $\mathcal{MM}$AS, convergence has been proved [34], [38]. Unfortunately, all these convergence results do not allow one to predict how quickly optimal solutions can be found. Only recently, Gutjahr presented an analytical framework that allows theoretical predictions about the speed of convergence of specific ACO algorithms to be derived [39].

Other research in ACO theory has focused on establishing formal links of ACO to other techniques for learning and optimization. One research direction focused on the connection between ACO and the fields of optimal control and reinforcement learning [40], while another aimed at examining the connections between ACO and probabilistic learning algorithms such as stochastic gradient ascent (SGA) [41], and the crossentropy (CE) method [42]. In particular, Zlochin et al. [42] have proposed a unifying framework for so-called *model-based search* (MBS) algorithms. Among other advantages, this framework allows a better understanding of ACO and will possibly lead to a cross-fertilization among MBS algorithms.

While convergence proofs give insight into some mathematically relevant properties of algorithms, they usually do not provide guidance to practitioners for the implementation of efficient algorithms. More relevant for practical applications are research efforts that aim at a better understanding of the behavior of ACO algorithms. Blum and Dorigo [43], [44] have shown that ACO algorithms in general suffer from *first order deception* in the same way as genetic algorithms suffer from deception. They further introduced the concept of *second order deception*, which occurs, for example, in situations where some solution components on average receive updates from more solutions than others with which they compete [45]. The first to study the behavior of ACO algorithms by analyzing the dynamics of the pheromone model were Merkle and Middendorf [46]. They showed that, in idealized permutation problems, constraints on the feasibility of solutions introduce what they called *selection bias* in the solution construction process.

FIGURE 4 The front page of the official Web site of the ant colony metaheuristic: www.aco-metaheuristic.org

[3]The notation $\arg\max_x f(x)$ stands for the value of x for which $f(\cdot)$ is maximized. If the maximum is attained for more than one value of x, it is a matter of indifference which one is considered.

TABLE 2 A non-exhaustive list of applications of ACO algorithms grouped by problem type.

PROBLEM TYPE	PROBLEM NAME	AUTHORS	YEAR	REFERENCES
ROUTING	TRAVELING SALESMAN	DORIGO ET AL.	1991, 1996	[6], [8]
		DORIGO & GAMBARDELLA	1997	[11]
		STÜTZLE & HOOS	1997, 2000	[15], [47]
	VEHICLE ROUTING	GAMBARDELLA ET AL.	1999	[48]
		REIMANN ET AL.	2004	[49]
	SEQUENTIAL ORDERING	GAMBARDELLA & DORIGO	2000	[50]
ASSIGNMENT	QUADRATIC ASSIGNMENT	STÜTZLE & HOOS	2000	[15]
		MANIEZZO	1999	[18]
	COURSE TIMETABLING	SOCHA ET AL.	2002, 2003	[35], [51]
	GRAPH COLORING	COSTA & HERTZ	1997	[52]
SCHEDULING	PROJECT SCHEDULING	MERKLE ET AL.	2002	[53]
	TOTAL WEIGHTED TARDINESS	DEN BESTEN ET AL.	2000	[54]
		MERKLE & MIDDENDORF	2000	[55]
	OPEN SHOP	BLUM	2005	[56]
SUBSET	SET COVERING	LESSING ET AL.	2004	[57]
	l-CARDINALITY TREES	BLUM & BLESA	2005	[58]
	MULTIPLE KNAPSACK	LEGUIZAMÓN & MICHALEWICZ	1999	[59]
	MAXIMUM CLIQUE	FENET & SOLNON	2003	[60]
OTHER	CONSTRAINT SATISFACTION	SOLNON	2000, 2002	[61], [62]
	CLASSIFICATION RULES	PARPINELLI ET AL.	2002	[63]
		MARTENS ET AL.	2006	[64]
	BAYESIAN NETWORKS	CAMPOS, ET AL.	2002	[65], [66]
	PROTEIN FOLDING	SHMYGELSKA & HOOS	2005	[67]
	PROTEIN-LIGAND DOCKING	KORB ET AL.	2006	[68]

IV. Applications of Ant Colony Optimization

In recent years, the interest of the scientific community in ACO has risen sharply. In fact, several successful applications of ACO to a wide range of different discrete optimization problems are now available. The large majority of these applications are to $\mathcal{NP}$-hard problems; that is, to problems for which the best known algorithms that guarantee to identify an optimal solution have exponential time worst case complexity. The use of such algorithms is often infeasible in practice, and ACO algorithms can be useful for quickly finding high-quality solutions. Other popular applications are to dynamic shortest path problems arising in telecommunication networks problems. The number of successful applications to academic problems has motivated people to adopt ACO for the solution of industrial problems, proving that this computational intelligence technique is also useful in real-world applications.

A. Applications to $\mathcal{NP}$-Hard Problems

The usual approach to show the usefulness of a new metaheuristic technique is to apply it to a number of different problems and to compare its performance with that of already available techniques. In the case of ACO, this type of research initially consisted of testing the algorithms on the TSP. Subsequently, other $\mathcal{NP}$-hard problems were also considered. So far, ACO has been tested on probably more than one hundred different $\mathcal{NP}$-hard problems. Many of the tackled problems can be considered as falling into one of the following categories: *routing problems* as they arise, for example, in the distribution of goods; *assignment problems*, where a set of items (objects, activities, etc.) has to be assigned to a given number of resources (locations, agents, etc.) subject to some constraints; *scheduling problems*, which–in the widest sense–are concerned with the allocation of scarce resources to tasks over time; and *subset problems*, where a solution to a problem is considered to be a selection of a subset of available items. In addition, ACO has been successfully applied to other problems emerging in fields such as machine learning and bioinformatics.

The good results of ACO algorithms on academic problems has made them appealing for applications in industrial settings.

Common to many of these applications is that the best-performing ACO algorithms make intensive use of the optional local search phase of the ACO metaheuristic (see Algorithm 1). This is typically very effective since, on the one hand, the solutions constructed by the ants can often be improved by an adequate local search algorithm; on the other hand, generating proper initial solutions for local search algorithms is a difficult task and many experimental results show that the probabilistic, adaptive solution generation process of ant colony optimization is particularly suited to this task.

In Table 2, we report some of the most noteworthy applications of ACO algorithms; for a detailed description of these and several other applications, we refer the reader to [34]. The overall result that emerges from these applications is that, for many problems, ACO algorithms produce results that are very close to those of the best-performing algorithms, while on some problems they are the state-of-the-art. These latter problems include the sequential ordering problem, open-shop scheduling problems, some variants of vehicle routing problems, classification problems, and protein-ligand docking.

B. Applications to Telecommunication Networks

ACO algorithms have shown to be a very effective approach for routing problems in telecommunication networks where the properties of the system, such as the cost of using links or the availability of nodes, vary over time. ACO algorithms were first applied to routing problems in circuit switched networks (such as telephone networks) [69] and then in packet-switched networks (such as local area networks or the Internet) [70]. Following the proof of concept provided by Schoonderwoerd et al., ant-inspired routing algorithms for telecommunication networks improved to the point of being state-of-the-art in wired networks. A well-known example is AntNet [70]. AntNet has been extensively tested, in simulation, on different networks and under different traffic patterns, proving to be highly adaptive and robust. A comparison with state-of-the-art routing algorithms has shown that, in most of the considered situations, AntNet outperforms its competitors.

Ant-based algorithms have given rise to several other routing algorithms, enhancing performance in a variety of wired network scenarios; see [71], [72] for a survey. More recently, an ACO algorithm designed for the challenging class of mobile ad hoc networks was shown to be competitive with state-of-the-art routing algorithms [73], [74], while at the same time offering better scalability.

C. Applications to Industrial Problems

The success on academic problems has raised the attention of a number of companies that have started to use ACO algorithms for real-world applications. Among the first to exploit algorithms based on the ACO metaheuristic is EuroBios (www.eurobios.com). They have applied ACO to a number of different scheduling problems such as a continuous two-stage flow shop problem with finite reservoirs. The problems modeled included various real-world constraints such as setup times, capacity restrictions, resource compatibilities and maintenance calendars. Another company that has played, and still plays, a very important role in promoting the real-world application of ACO is AntOptima (www.antoptima.com). AntOptima's researchers have developed a set of tools for the solution of vehicle routing problems whose optimization algorithms are based on ACO. Particularly successful products based on these tools are (i) DYVOIL, for the management and optimization of heating oil distribution with a nonhomogeneous fleet of trucks, used for the first time by Pina Petroli in Switzerland, and (ii) AntRoute, for the routing of hundreds of vehicles of companies such as Migros, the main Swiss supermarket chain, or Barilla, the main Italian pasta maker. Still another vehicle routing application was developed by BiosGroup for the French company Air Liquide. Other interesting real-world applications are those by Gravel, Price and Gagné [75], who have applied ACO to an industrial scheduling problem in an aluminum casting center, and by Bautista and Pereira [76], who successfully applied ACO to solve an assembly line balancing problem with multiple objectives and constraints between tasks.

For the best-performing ACO algorithms, convergence to optimal solutions has been proved.

V. Current Hot Topics in ACO

A significant part of research on ACO is still concerned with applications as they have been presented in the previous section. However, increasing attention is and will be given to even more challenging problems that, for example, involve *multiple objectives*, *dynamic modifications* of the data, and the *stochastic nature* of the objective function and of the constraints. Other developments focus on the extension of the applicability of ACO algorithms from discrete to continuous optimization problems and to the study of parallel implementations of ACO algorithms.

A. Dynamic Optimization Problems

Dynamic problems are characterized by the fact that the search space changes during time. Hence, while searching, the conditions of the search, the definition of the problem instance and, thus, the quality of the solutions already found may change. In such a situation, it is crucial that the algorithm be able to adjust the search direction, following the changes of the problem being solved.

A paradigmatic example is routing in telecommunication networks, an application problem already discussed in the previous section. For this problem, ACO algorithms belong to the state-of-the-art techniques [70], [74]. ACO algorithms have also been applied to dynamic versions of the TSP, where either the distance between some pairs of cities changes [77]–[79], or cities are dynamically added or removed from the set of cities to be visited. More recently, an ACS algorithm has also been applied to dynamic vehicle routing problems [80], showing good behavior on randomly generated as well as real-world instances.

B. Stochastic Optimization Problems

In stochastic optimization problems, some variables have a stochastic nature. Apart from the network routing problems, for which the main focus was put on their dynamic character, the stochastic traveling salesman problem (PTSP) was the first stochastic problem tackled by ACO algorithms. In the PTSP,

Resources on Ant Colony Optimization

- **Web pages:**
  - www.aco-metaheuristic.org: The official Web site of the ant colony metaheuristic (see Figure 4).
  - www.metaheuristics.org: Web site of the "Metaheuristics Network" project. This European Union funded project was dedicated to the theoretical analysis and experimental comparison of metaheuristics.
- **Books:**
  - M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
  - E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- **Scientific Journals:** Scientific articles on ACO are published in many journals, including "IEEE Transactions on Systems, Man, and Cybernetics", "IEEE Transactions on Evolutionary Computation", "Artificial Life", "INFORMS Journal on Computing", "Journal of Heuristics", "Computers and Operations Research", "Computational Optimization and Applications", and "European Journal of Operational Research". The new journal "Swarm Intelligence," whose first issue is forecast for June 2007, will certainly become the main scientific periodical for disemination of ACO and related research.
- **Conferences:**
  - The biannual series of workshops "ANTS – The International Workshop on Ant Colony Optimization and Swarm Intelligence" (iridia.ulb.ac.be/∼ants), held for the first time in 1998, is the oldest conference in the ACO and swarm intelligence fields.
  - The series of conferences "IEEE Swarm Intelligence" (www.computelligence.org/sis) focuses on swarm intelligence techniques and also ant colony optimization.
  - Articles on ACO are regularly presented at other conferences such as "IEEE Congress on Evolutionary Computation (CEC)", "Genetic and Evolutionary Computation Conference (GECCO)", "Parallel Problem Solving from Nature (PPSN)", "INFORMS" meetings, "European Chapter on Combinatorial Optimization (ECCO)" meetings, the "Metaheuristics International Conference (MIC)" and many others.
- **Software:** Software, distributed under the GNU license, is available at: www.aco-metaheuristic.org/aco-code/
- **Popular press:** ACO is often covered by the popular press. Pointers to popularization articles can be found at: www.aco-metaheuristic. org/aco-in-the-press.html
- **Mailing list:** A moderated mailing list dedicated to the exchange of information related to ACO is accessible at: www.aco-metaheuristic. org/mailing-list.html

It is foreseeable that future research on ACO will focus more strongly on rich optimization problems that include stochasticity, dynamic data modifications, and multiple objectives.

each city has a given probability of requiring a visit and the goal is to find an *a priori* tour of minimal expected length over all the cities, with the strategy of visiting a random subset of cities in the same order as they appear in the a priori tour. The first ACO algorithm for this problem was proposed by Bianchi et al. [81]. Further ACO algorithms for the PTSP have been proposed by Branke and Guntsch [82], Gutjahr [83], [84], and Birattari et al. [85].

C. Multi-Objective Optimization

Multiple objectives can often be handled by ordering or weighting them according to their relative importance. In the two-colony ACS algorithm for the vehicle routing problem with time window constraints [48] and in the $\mathcal{MM}$AS for the bi-objective two-machine permutation flow shop problem [86], the multi-objective optimization problem is handled by ordering the objectives; differently, Doerner et al. [87] apply ACO to a bi-objective transportation problem and combine the objectives in a weighted sum. On the other hand, if preferences or weights cannot be given *a priori*, the goal is to find a set of *non-dominated* solutions that are optimal in the Pareto sense. The first ACO algorithm for finding non-dominated solutions was proposed by Iredi et al. [88] for the bi-objective scheduling problem. Other applications include portfolio optimization [89] and the quadratic assignment problem [90].

D. Parallel Implementations

ACO algorithms lend themselves to be parallelized in the data or population domains. In particular, any parallel models used in other population-based algorithms can be easily adapted to ACO. Two main strategies have been followed. In *fine-grained* parallelization, very few individuals are assigned to single processors and information exchange among the processors is frequent. In *coarse-grained* approaches, on the contrary, larger subpopulations are assigned to single processors and information exchange is rather rare. Research on parallel ACO algorithms has quickly shown that fine-grained parallelization results in a very significant communication overhead. Therefore, the focus has mostly turned to coarse-grained parallelization schemes, where p colonies run parallel on p processors [91]–[95].

E. Continuous Optimization

Recently, ACO algorithms have been applied to continuous optimization. When an algorithm designed for combinatorial optimization is used to tackle a continuous problem, the simplest approach would be to divide the domain of each variable into a set of intervals. However, when the domain of the variables is large and the required accuracy is high, this approach is not viable. For this reason, ACO algorithms have been developed, which are specifically designed for continuous and mixed continuous-discrete variables [96], [97]. Research in this direction is currently ongoing.

VI. Other Ant-Inspired Algorithms

The source of inspiration of ACO is the path marking behavior that some ant species exhibit when foraging. Nonetheless, this behavior is not the only behavior of ants that has inspired computer scientists. We present here, in a very concise way, some other examples of algorithms that are inspired by ants. The common trait of all these techniques is that they make use of *stigmergic variables*; that is, variables associated with the environment that hold the information that artificial ants share and exploit. (A more comprehensive discussion of ant algorithms and stigmergy can be found in [98].)

A. Other Algorithms Inspired by Foraging and Path Marking

Apart from ACO, a few other approaches take inspiration from the path marking behavior of ants. Two algorithms have been proposed for graph exploration: *Edge Ant Walk* [99] and *Vertex Ant Walk* [100]. In these algorithms, ants mark with pheromone the edges they visit to coordinate graph exploration. Contrary to ACO, in these algorithms the pheromone directs the ants toward unexplored areas of the search space. In fact, the goal is to cover the graph; that is to visit all the nodes, without knowing the graph topology. Another example of algorithm inspired by ants' path marking is a search algorithm for continuous optimization problems that was inspired by the foraging behavior of the *Pachycondyla apicalis* ants [101].

B. Algorithms Inspired by Brood Sorting

Brood sorting is an activity that can be observed in many ant species (e.g., in *Pheidole pallidula* ants [102]). These ants compactly cluster their eggs and smaller larvae at the center of the nest brood area and the larger larvae at the periphery of the brood cluster. Deneubourg et al. [102] have proposed a model of this phenomenon in which an ant picks up and drops an item according to the number of similar surrounding items. Lumer and Faieta [103] and Kuntz et al. [104] have applied this model to a specific clustering problem, obtaining results that were qualitatively equivalent to those obtained by classical techniques but at a lower computational cost. Recently, Handl et al. [105] described an improved version of Lumer and Faieta's algorithm, and compared its performance to other standard clustering techniques, such as k-means. One of the salient features of this ant-based algorithm is its ability to propose a "natural" number of clusters. For an overview of other developments, we refer to [105].

Various algorithmic techniques have been inspired by behaviors of ants. Ant colony optimization is the most successful and best-known among them.

C. Algorithms Inspired by Division of Labor

In ant colonies, individual workers tend to specialize on specific tasks in their lifetime [106]. However, ants can adapt their behavior to the circumstances: a soldier ant can become a forager, a nurse ant a guard, and so on. This combination of specialization and flexibility is a desirable feature for multi-agent optimization and control, especially in task or resource allocation problems that require continuous adaptation to changing conditions. Many approaches inspired by division of labor in real ant colonies are based on a threshold model developed by Robinson [106], in which workers with low response thresholds respond to lower levels of stimuli than do workers with high response thresholds. Such a response-threshold model has been applied to the problem of choosing a paint booth for trucks coming out of an assembly line in a truck factory [98], [107]–[110].

D. Algorithms Inspired by Cooperative Transport

The behavior of ant colonies has also inspired research in robotics, in particular for the design of distributed control algorithms for groups of robots [111]. An example of a task that has been used as a benchmark for ant algorithms applied to distributed robotics problems is cooperative box pushing [112]. Another example of application of ant algorithms is the one to the related problem of pulling an object. This has been achieved [113] within the Swarm-bots project (www.swarm-bots.org), a project dedicated to the study of ant algorithms for autonomous robotics applications.

VII. Outlook and Conclusions

As we have discussed, nowadays hundreds of researchers worldwide are applying ACO to classic $\mathcal{NP}$-hard optimization problems, while only a few works concern variations that include dynamic and stochastic aspects as well as multiple objectives. The study of how best to apply ACO to such variations will certainly be one of the major research directions in the near future. A better understanding of the theoretical properties of ACO algorithm is certainly another research direction that will be pursued in the future.

Fifteen years ago, when the first ACO algorithm was introduced, taking inspiration from ants for designing optimization algorithms seemed a crazy idea. The many successful applications presented in this article have changed our perspective: what seemed a far out idea is now considered one of the most promising approaches to the approximate solution of difficult optimization problems.

References

[1] P.-P. Grassé, *Les Insectes Dans Leur Univers*, Paris, France: Ed. du Palais de la découverte, 1946.

[2] P.-P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes Natalensis et Cubitermes sp*. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41–81, 1959.

[3] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant," *Journal of Insect Behavior*, vol. 3, p. 159, 1990.

[4] S. Goss, S. Aron, J.-L. Deneubourg, and J.-M. Pasteels, "Self-organized shortcuts in the Argentine ant," *Naturwissenschaften*, vol. 76, pp. 579–581, 1989.

[5] J.-M. Pasteels, J.-L. Deneubourg, and S. Goss, "Self-organization mechanisms in ant societies (i): Trail recruitment to newly discovered food sources," *Experientia Supplementum*, vol. 54, p. 155, 1987.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Tech. Rep. 91-016, 1991.

[7] M. Dorigo, "Optimization, learning and natural algorithms (*in italian*)," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

[8] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[9] L.M. Gambardella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem," in *Proc. Twelfth International Conference on Machine Learning (ML-95)*, A. Prieditis and S. Russell, Eds., Morgan Kaufmann Publishers, pp. 252–260, 1995.

[10] M. Dorigo and L.M. Gambardella, "Ant colonies for the traveling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.

[11] ——, "Ant Colony System: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[12] L.M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in *Proc. 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, T. Baeck et al., Eds. IEEE Press, Piscataway, NJ, pp. 622–627, 1996.

[13] T. Stützle and H.H. Hoos, "Improving the Ant System: A detailed report on the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System," FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12, Aug. 1996.

[14] T. Stützle, *Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications*, ser. DISKI. Infix, Sankt Augustin, Germany, vol. 220, 1999.

[15] T. Stützle and H.H. Hoos, "$\mathcal{MAX}$–$\mathcal{MIN}$ Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[16] B. Bullnheimer, R.F. Hartl, and C. Strauss, "A new rank based version of the Ant System—a computational study," *Institute of Management Science*, University of Vienna, Tech. Rep., 1997.

[17] ——, "A new rank-based version of the Ant System: A computational study," *Central European Journal for Operations Research and Economics*, vol. 7, no. 1, pp. 25–38, 1999.

[18] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," *INFORMS Journal on Computing*, vol. 11, no. 4, pp. 358–369, 1999.

[19] O. Cordón, I.F. de Viana, F. Herrera, and L. Moreno, "A new ACO model integrating evolutionary computation concepts: The best-worst Ant System," in *Proc. ANTS 2000*, M. Dorigo et al., Eds., IRIDIA, Université Libre de Bruxelles, Belgium, pp. 22–29, 2000.

[20] C. Blum, A. Roli, and M. Dorigo, "HC–ACO: The hyper-cube framework for Ant Colony Optimization," in *Proc. MIC'2001—Metaheuristics International Conference*, vol. 2, Porto, Portugal, pp. 399–403, 2001.

[21] C. Blum and M. Dorigo, "The hyper-cube framework for ant colony optimization," *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 34, no. 2, pp. 1161–1172, 2004.

[22] M. Dorigo and G. Di Caro, "The Ant Colony Optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11–32, 1999.

[23] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.

[24] V. Cerný, "A thermodynamical approach to the traveling salesman problem," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[25] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[26] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[27] ——, "Tabu search—part II," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[28] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[29] H.R. Lourenço, O. Martin, and T. Stützle, "Iterated local search," in *Handbook of Metaheuristics, ser. International Series in Operations Research & Management Science*, F. Glover and G. Kochenberger, Eds., Kluwer Academic Publishers, vol. 57, pp. 321–353, 2002.

[30] L.J. Fogel, A.J. Owens, and M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, 1966.

[31] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

[32] I. Rechenberg, *Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Information*, Fromman Verlag, Freiburg, Germany, 1973.

[33] H.-P. Schwefel, *Numerical Optimization of Computer Models*. John Wiley & Sons, 1981.

[34] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[35] K. Socha, J. Knowles, and M. Sampels, "A $\mathcal{MAX}$–$\mathcal{MIN}$ ant system for the university timetabling problem," in *Proc. ANTS 2002*, ser. LNCS, M. Dorigo et al., Eds., vol. 2463, p. 1, Berlin, Germany: Springer Verlag, 2002.

[36] W.J. Gutjahr, "A graph-based ant system and its convergence," *Future Generation Computer Systems*, vol. 16, no. 9, pp. 873–888, 2000.

[37] ——, "ACO algorithms with guaranteed convergence to the optimal solution," *Information Processing Letters*, vol. 82, no. 3, pp. 145–153, 2002.

[38] T. Stützle and M. Dorigo, "A short convergence proof for a class of ACO algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 358–365, 2002.

[39] W.J. Gutjahr, "On the finite-time dynamics of ant colony optimization," *Methodology and Computing in Applied Probability*, vol. 8, no. 1, pp. 105–133, 2006.

[40] M. Birattari, G. Di Caro, and M. Dorigo, "Toward the formal foundation of ant programming," in *Proc. ANTS 2002*, ser. LNCS, M. Dorigo et al., Eds., vol. 2463, Springer Verlag, pp. 188–201, 2002.

[41] N. Meuleau and M. Dorigo, "Ant colony optimization and stochastic gradient descent," *Artificial Life*, vol. 8, no. 2, pp. 103–121, 2002.

[42] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 373–395, 2004.

[43] C. Blum, *Theoretical and Practical Aspects of Ant Colony Optimization*, ser. Dissertations in Artificial Intelligence, Akademische Verlagsgesellschaft Aka GmbH, Berlin, Germany, vol. 282, 2004.

[44] C. Blum and M. Dorigo, "Search bias in ant colony optimization: On the role of competition balanced systems," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 159–174, 2005.

[45] C. Blum, M. Sampels, and M. Zlochin, "On a particularity in model-based search," in *Proc. Genetic and Evolutionary Computation Conference (GECCO-2002)*, W. B. Langdon et al., Eds. Morgan Kaufmann Publishers, pp. 35–42, 2002.

[46] D. Merkle and M. Middendorf, "Modelling the dynamics of ant colony optimization algorithms," *Evolutionary Computation*, vol. 10, no. 3, pp. 235–262, 2002.

[47] T. Stützle and H.H. Hoos, "The $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System and local search for the traveling salesman problem," in *Proc. 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, T. Bäck et al., Eds. IEEE Press, Piscataway, NJ, pp. 309–314, 1997.

[48] L.M. Gambardella, E.D. Taillard, and G. Agazzi, "MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 63–76, 1999.

[49] M. Reimann, K. Doerner, and R.F. Hartl, "D-ants: Savings based ants divide and conquer the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 4, pp. 563–591, 2004.

[50] L.M. Gambardella and M. Dorigo, "Ant Colony System hybridized with a new local search for the sequential ordering problem," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 237–255, 2000.

[51] K. Socha, M. Sampels, and M. Manfrin, "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," in *Applications of Evolutionary Computing, Proc. EvoWorkshops 2003*, ser. LNCS, G. R. Raidl et al., Eds., vol. 2611, Springer Verlag, pp. 334–345, 2003.

[52] D. Costa and A. Hertz, "Ants can colour graphs," *Journal of the Operational Research Society*, vol. 48, pp. 295–305, 1997.

[53] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 333–346, 2002.

[54] M.L. den Besten, T. Stützle, and M. Dorigo, "Ant colony optimization for the total weighted tardiness problem," in *Proc. PPSN-VI*, ser. LNCS, M. Schoenauer et al., Eds., vol. 1917, Springer Verlag, pp. 611–620, 2000.

[55] D. Merkle and M. Middendorf, "Ant colony optimization with global pheromone evaluation for scheduling a single machine," *Applied Intelligence*, vol. 18, no. 1, pp. 105–111, 2003.

[56] C. Blum, "Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling," *Computers & Operations Research*, vol. 32, no. 6, pp. 1565–1591, 2005.

[57] L. Lessing, I. Dumitrescu, and T. Stützle, "A comparison between ACO algorithms for the set covering problem," in *Proc. ANTS'2004*, ser. LNCS, M. Dorigo et al., Eds., vol. 3172, Springer Verlag, pp. 1–12, 2004.

[58] C. Blum and M.J. Blesa, "New metaheuristic approaches for the edge-weighted k-cardinality tree problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1355–1377, 2005.

[59] G. Leguizamón and Z. Michalewicz, "A new version of Ant System for subset problems," in *Proc. CEC'99*, IEEE Press, Piscataway, NJ, pp. 1459–1464, 1999.

[60] S. Fenet and C. Solnon, "Searching for maximum cliques with ant colony optimization," in *Applications of Evolutionary Computing, Proc. EvoWorkshops 2003*, ser. LNCS, G. R. Raidl et al., Eds., Springer Verlag, vol. 2611, pp. 236–245, 2003.

[61] C. Solnon, "Solving permutation constraint satisfaction problems with artificial ants," in *Proc. ECAI'2000*, Amsterdam, The Netherlands: IOS Press, pp. 118–122, 2000.

[62] ——, "Ants can solve constraint satisfaction problems," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 347–357, 2002.

[63] R.S. Parpinelli, H.S. Lopes, and A.A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, 2002.

[64] D. Martens, M.D. Backer, R. Haesen, B. Baesens, C. Mues, and J. Vanthienen, "Ant-based approach to the knowledge fusion problem," in *Proc. ANTS 2006*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 4150, pp. 84–95, 2006.

[65] L.M. de Campos, J.M. Fernández-Luna, J.A. Gámez, and J.M. Puerta, "Ant colony optimization for learning Bayesian networks," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 291–311, 2002.

[66] L.M. de Campos, J.A. Gamez, and J.M. Puerta, "Learning Bayesian networks by ant colony optimisation: Searching in the space of orderings," *Mathware and Soft Computing*, vol. 9, no. 2–3, pp. 251–268, 2002.

[67] A. Shmygelska and H.H. Hoos, "An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem," *BMC Bioinformatics*, vol. 6, no. 30, 2005.

[68] O. Korb, T. Stützle, and T.E. Exner, "Application of ant colony optimization to structure-based drug design," in *Proc. ANTS 2006*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 4150, pp. 247–258, 2006.

[69] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1996.

[70] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[71] K.M. Sim and W.H. Sun, "Ant colony optimization for routing and load-balancing: Survey and new directions," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2003.

[72] G. Di Caro, "Ant colony optimization and its application to adaptive routing in telecommunication networks," Ph.D. dissertation, Université Libre de Bruxelles, Brussels, Belgium, 2004.

[73] F. Ducatelle, G. Di Caro, and L.M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks," *International Journal of Computational Intelligence and Applications*, vol. 5, no. 2, pp. 169–184, 2005.

[74] G. Di Caro, F. Ducatelle, and L.M. Gambardella, "AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[75] M. Gravel, W.L. Price, and C. Gagné, "Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic," *European Journal of Operational Research*, vol. 143, pp. 218–229, 2002.

[76] J. Bautista and J. Pereira, "Ant algorithms for assembly line balancing," in *Proc. ANTS 2002*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 2463, pp. 65–75, 2002.

[77] M. Guntsch and M. Middendorf, "Pheromone modification strategies for ant algorithms applied to dynamic TSP," in *Applications of Evolutionary Computing: Proc. EvoWorkshops 2001*, ser. LNCS, E. J. W. Boers et al., Eds., Springer Verlag, vol. 2037, pp. 213–222, 2001.

[78] ——, "A population based approach for ACO," in *Applications of Evolutionary Computing, Proc. EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTim*, ser. LNCS, S. Cagnoni et al., Eds., Springer Verlag, vol. 2279, pp. 71–80, 2002.

[79] C.J. Eyckelhof and M. Snoek, "Ant systems for a dynamic TSP: Ants caught in a traffic jam," in *Proc. ANTS 2002*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 2463, pp. 88–99, 2002.

[80] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, pp. 327–343, 2005.

[81] L. Bianchi, L.M. Gambardella, and M. Dorigo, "An ant colony optimization approach to the probabilistic traveling salesman problem," in *Proc. PPSN-VII*, ser. LNCS, J. J. Merelo et al., Eds., Springer Verlag, vol. 2439, pp. 883–892, 2002.

[82] J. Branke and M. Guntsch, "New ideas for applying ant colony optimization to the probabilistic TSP," in *Applications of Evolutionary Computing, Proc. EvoWorkshops 2003*, ser. LNCS, G. R. Raidl et al., Eds., Springer Verlag, vol. 2611, pp. 165–175, 2003.

[83] W.J. Gutjahr, "A converging ACO algorithm for stochastic combinatorial optimization," in *Proc. SAGA 2003*, ser. LNCS, A. Albrecht and T. Steinhöfl, Eds., Berlin, Germany: Springer Verlag, vol. 2827, pp. 10–25, 2003.

[84] ——, "S-ACO: An ant based approach to combinatorial optimization under uncertainity," in *Proc. ANTS 2004*, ser. LNCS, Dorigo et al., Eds., Berlin, Germany: Springer Verlag, vol. 3172, pp. 1–12, 2004.

[85] M. Birattari, P. Balaprakash, and M. Dorigo, "ACO/F-Race: Ant colony optimization and racing techniques for combinatorial optimization under uncertainty," in *MIC 2005: The 6th Metaheuristics International Conference*, K. F. Doerner et al., Eds., Vienna, Austria: University of Vienna, Department of Business Administration, pp. 107–112, 2005.

[86] V. T'kindt, N. Monmarché, F. Tercinet, and D. Laügt, "An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem," *European Journal of Operational Research*, vol. 142, no. 2, pp. 250–257, 2002.

[87] K. Doerner, R.F. Hartl, and M. Reimann, "CompetAnts for problem solving—the case of full truckload transportation," *Central European Journal for Operations Research and Economics*, vol. 11, no. 2, pp. 115–141, 2003.

[88] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *First International Conference on Evolutionary Multi-Criterion Optimization, (EMO'01)*, ser. LNCS, E. Zitzler et al., Eds., Springer Verlag, vol. 1993, pp. 359–372, 2001.

[89] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization in multiobjective project portfolio selection with ILP preprocessing," *European Journal of Operational Research*, vol. 171, no. 3, pp. 830–841, 2006.

[90] M.L. Ibáñez, L. Paquete, and T. Stützle, "On the design of ACO for the biobjective quadratic assignment problem," in *Proc. ANTS'2004*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 3172, pp. 214–225, 2004.

[91] T. Stützle, "Parallelization strategies for ant colony optimization," in *Proc. PPSN-V*, ser. LNCS, A. E. Eiben et al., Eds., Springer Verlag, vol. 1498, pp. 722–731, 1998.

[92] E.-G. Talbi, O. Roux, C. Fonlupt, and D. Robillard, "Parallel ant colonies for combinatorial optimization problems," in *Parallel and Distributed Processing, 11 IPPS/SPDP'99 Workshops*, ser. LNCS, J. Rolim et al., Eds., vol. 1586, pp. 239–247, 1999.

[93] M. Middendorf, F. Reischle, and H. Schmeck, "Multi colony ant algorithms," *Journal of Heuristics*, vol. 8, no. 3, pp. 305–320, 2002.

[94] M. Manfrin, M. Birattari, T. Stützle, and M. Dorigo, "Parallel ant colony optimization for the traveling salesman problem," in *Proc. ANTS 2006*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 4150, pp. 224–234, 2006.

[95] S. Benker, K.F. Doerner, R.F. Hartl, G. Kiechle, and M. Lucka, "Communication strategies for parallel cooperative ant colony optimization on clusters and grids," in *Complimentary Proc. PARA'04 Workshop on State-of-the-Art in Scientific Computing*, pp. 3–12, 2005.

[96] K. Socha, "ACO for continuous and mixed-variable optimization," in *Proc. ANTS 2004*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 3172, pp. 25–36, 2004.

[97] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, in press.

[98] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.

[99] I.A. Wagner, M. Lindenbaum, and A.M. Bruckstein, "Smell as a computational resource—a lesson we can learn from the ant," in *Proc. Fourth Israeli Symposium on Theory of Computing and Systems (ISTCS-99)*, pp. 219–230, 1996.

[100] ——, "Efficient graph search by a smell-oriented vertex process," *Annals of Mathematics and Artificial Intelligence*, vol. 24, pp. 211–223, 1998.

[101] N. Monmarché, G. Venturini, and M. Slimane, "On how Pachycondyla apicalis ants suggest a new search algorithm," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 937–946, 2000.

[102] 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 *Proc. First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, J.-A. Meyer and S. W. Wilson, Eds., MIT Press, Cambridge, MA, pp. 356–363, 1991.

[103] E. Lumer and B. Faieta, "Diversity and adaptation in populations of clustering ants," in *Proc. Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, J.-A. Meyer and S. W. Wilson, Eds., MIT Press, Cambridge, MA, pp. 501–508, 1994.

[104] 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*, vol. 5, pp. 327–351, 1999.

[105] J. Handl, J. Knowles, and M. Dorigo, "Ant-based clustering and topographic mapping," *Artificial Life*, vol. 12, no. 1, pp. 35–62, 2006.

[106] G.E. Robinson, "Regulation of division of labor in insect societies," *Ann. Rev. Entomol.*, vol. 37, pp. 637–665, 1992.

[107] M. Campos, E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Dynamic scheduling and division of labor in social insects," *Adaptive Behavior*, vol. 8, no. 3, pp. 83–96, 2000.

[108] V.A. Cicirello and S.F. Smith, "Ant colony control for autonomous decentralized shop floor routing," in *Proc. International Symposium on Autonomous Decentralized Systems*, IEEE Computer Society Press, pp. 383–390, 2001.

[109] S. Nouyan, "Agent-based approach to dynamic task allocation," in *Proc. ANTS 2002*, ser. LNCS, M. Dorigo et al., Eds., Springer Verlag, vol. 2463, pp. 28–39, 2002.

[110] S. Nouyan, R. Ghizzioli, M. Birattari, and M. Dorigo, "An insect-based algorithm for the dynamic task allocation problem," *Künstliche Intelligenz*, vol. 4/05, pp. 25–31, 2005.

[111] A. Martinoli, M. Yamamoto, and F. Mondada, "On the modelling of bio-inspired collective experiments with real robots," in *Proc. Fourth European Conference on Artificial Life*, P. Husbands and I. Harvey, Eds., MIT Press, Cambridge, MA, 1997.

[112] C.R. Kube and H. Zhang, "Collective robotics: From social insects to robots," *Adaptive Behavior*, vol. 2, pp. 189–218, 1994.

[113] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T.H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L.M. Gambardella, "Evolving self-organizing behaviors for a Swarm-bot," *Autonomous Robots*, vol. 17, no. 2–3, pp. 223–245, 2004.