Ant Algorithms and Stigmergy

Marco Dorigo [a,1], Eric Bonabeau [b], Guy Theraulaz [c,2]

[a] *IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, mdorigo@ulb.ac.be*

[b] *EuroBios, Paris, France, bonabeau@aol.com*

[c] *Université Paul Sabatier, Toulouse, France, theraula@cict.fr*

Abstract

Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks that in some cases far exceed the individual capacities of a single ant. The study of ant colonies behavior and of their self-organizing capacities is interesting for computer scientists because it provides models of distributed organization which are useful to solve difficult optimization and distributed control problems. In this paper we overview some models derived from the observation of real ants, emphasizing the role played by *stigmergy* as distributed communication paradigm, and we show how these models have inspired a number of novel algorithms for the solution of distributed optimization and distributed control problems.

Key words: Ant algorithms, ant colony optimization, swarm intelligence, social insects, self-organization, metaheuristics.

1 Introduction

Ant colonies have always fascinated human beings. Books on ants, ranging from pure literature [52,86] to detailed scientific accounts of all the aspects of their life [31,42,43], have often met extraordinary public success. What particularly strikes the occasional observer as well as the scientist is the high

[1] Dr. Dorigo acknowledges support by the Belgian Fund for Scientific Research (FNRS) of which he is a Research Associate.

[2] Dr. Theraulaz was partially supported by a grant from the GIS (Groupement d'Intérêt Scientifique) Sciences de la Cognition and a grant form the Conseil Régional Midi-Pyrénées.

degree of societal organization that these insects can achieve, in spite of very limited individual capabilities. Ants appeared on the earth some 100 millions of years ago, and their current population is estimated to be around 10^{16} individuals [43]. An approximate computation tells us that their total weight is the same order of magnitude as the total weight of human beings; like human beings, they can be found virtually everywhere on the earth. Ants are undoubtedly one of the most successful species on the earth today, and they have been so for the last 100 million years. It is therefore not surprising that computer scientists have taken inspiration from studies of the behavior of ant colonies, and more generally of social insects, to design algorithms for the control of multi-agent systems.

A particularly interesting body of work is the one that focuses on the concept of *stigmergy*, a particular form of indirect communication used by social insects to coordinate their activities. By exploiting the stigmergic approach to coordination, researchers have been able to design a number of successful algorithms in such diverse application fields as combinatorial optimization, routing in communication networks, task allocation in a multi-robot system, exploratory data analysis, graph drawing and partitioning, and so on [2,26].

The term *stigmergy* was introduced by Grassé [39] to describe a form of indirect communication mediated by modifications of the environment that he observed in two species of termites: *Bellicositermes Natalensis* and *Cubitermes*. Grassé's original definition of stigmergy was:

> *"Stimulation of workers*[3] *by the performance they have achieved".*

Although Grassé first introduced the term stigmergy to explain the behavior of termites societies, the same term has later been used to indicate indirect communication mediated by modifications of the environment that can be observed also in other social insects [80].

Nest building in termites is the typical example of stigmergy, and is also the original example used by Grassé to introduce the concept. Termite workers use soil pellets, which they impregnate with pheromone (that is, a diffusing chemical substance) to build pillars. Two successive phases take place during nest reconstruction [39]. First, a non-coordinated phase occurs which is characterized by a random deposition of pellets. This phase lasts until one of the deposits reaches a critical size (Figure 1). Then, a coordination phase starts if the group of builders is sufficiently large and pillars emerge. The existence of an initial deposit of soil pellets stimulates workers to accumulate more material through a positive feedback mechanism, since the accumulation of material reinforces the attractivity of deposits through the diffusing

[3] Workers are one of the castes in termite colonies.

pheromone emitted by the pellets [6]. This *autocatalytic*, snowball effect leads to the coordinated phase. If the density of builders is too small, the pheromone disappears between two successive passages by the workers, and the amplification mechanism cannot work, which leads to a non-coordinated behavior. The system undergoes a bifurcation at this critical density: no pillar emerges below it, but pillars can emerge above it. This example therefore illustrates positive feedback (the snowball effect), negative feedback (pheromone decay), the amplification of fluctuations (pillars could emerge anywhere), multiple interactions (through the environment), the emergence of structure (that is, pillars) out of an initially homogenous medium (that is, a random spatial distribution of soil pellets), multistability (again, pillars may emerge anywhere) and bifurcation which make up the signatures of self-organized phenomena. From the experimental observations, Deneubourg [15] designed a chemotaxis-based reaction-diffusion model, that exhibits the desired properties for appropriate parameter values. Figure 2 shows the two-dimensional spatial distribution of pillars obtained with his model. In this model, coordination emerges out of indirect (stigmergic) interactions among workers.

In this paper an *ant algorithm* is informally defined as a multi-agent system inspired by the observation of some real ant colony behavior exploiting stigmergy. In ant algorithms the agents are called *artificial ants*, or often simply *ants*, and coordination among ants is achieved by exploiting the stigmergic communication mechanism.

The implementation of ant algorithms is made possible by the use of so-called *stigmergic variables*, that is, variables that contain the information used by artificial ants to indirectly communicate. In some cases, as discussed, for example, in Section 2, the stigmergic variable is a specifically defined variable used by ants to adaptively change the way they build solutions to the considered problem. In other cases, as discussed in Sections 3 and 4, the stigmergic variable is one of the problem variables: in this case a change in its value determines not only a change in the way a solution to the problem is built, but also a direct change in the solution of the problem itself.

In the rest of this article we provide a number of examples of ant algorithms and for each example we highlight the role played by stigmergy. Each section deals with a particular behavior observed in ant colonies: foraging (section 2), division of labor (section 3), and clustering (section 4). The first part of each section provides a brief description of the observed phenomenon followed by a description of models developed by ethologists to understand the phenomenon; engineering-oriented applications, that make use of the emergent behavior of ant colonies, are then presented. It is worth noting that not all kinds of ant algorithms are equally advanced: some of them are among the best available approaches for selected problems, while others are just proofs of concept and further work needs to be done to fully evaluate their potentialities. Therefore,

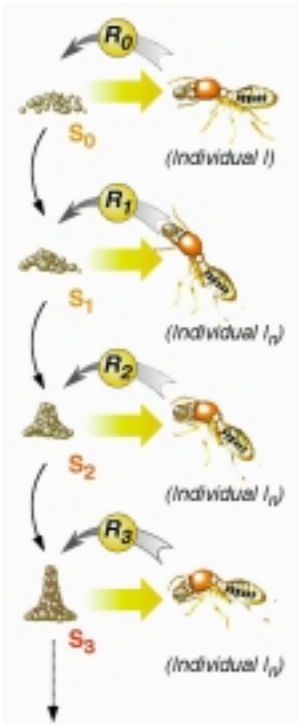


Fig. 1. An example of stigmergic process as it appears in the construction of pillars in termites. Assume that the architecture reaches state S_0, which triggers response R_0 from worker I. S_0 is modified by the action of I (for example, I may drop a soil pellet), and transformed into a new stimulating configuration S_1, that may in turn trigger a new response R_1 from I or any other worker I_n and so forth. The successive responses R_1, R_2, R_n may be produced by any worker carrying a soil pellet. Each worker creates new stimuli in response to existing stimulating configurations. These new stimuli then act on the same termite or on any other worker in the colony. Such a process, where the only relevant interactions taking place among the agents are indirect, through the environment which is modified by the other agents, is also called sematectonic communication [88].

the various sections of this review article may emphasize different aspects, the biology or the engineering side. Sections that emphasize applications are useful because they show very clearly the way our understanding of how ants collectively solve problems can be applied to design algorithms and distributed problem-solving devices; those emphasizing the biology are useful because, we believe, they provide new ideas to design new types of algorithms and distributed artificial devices.

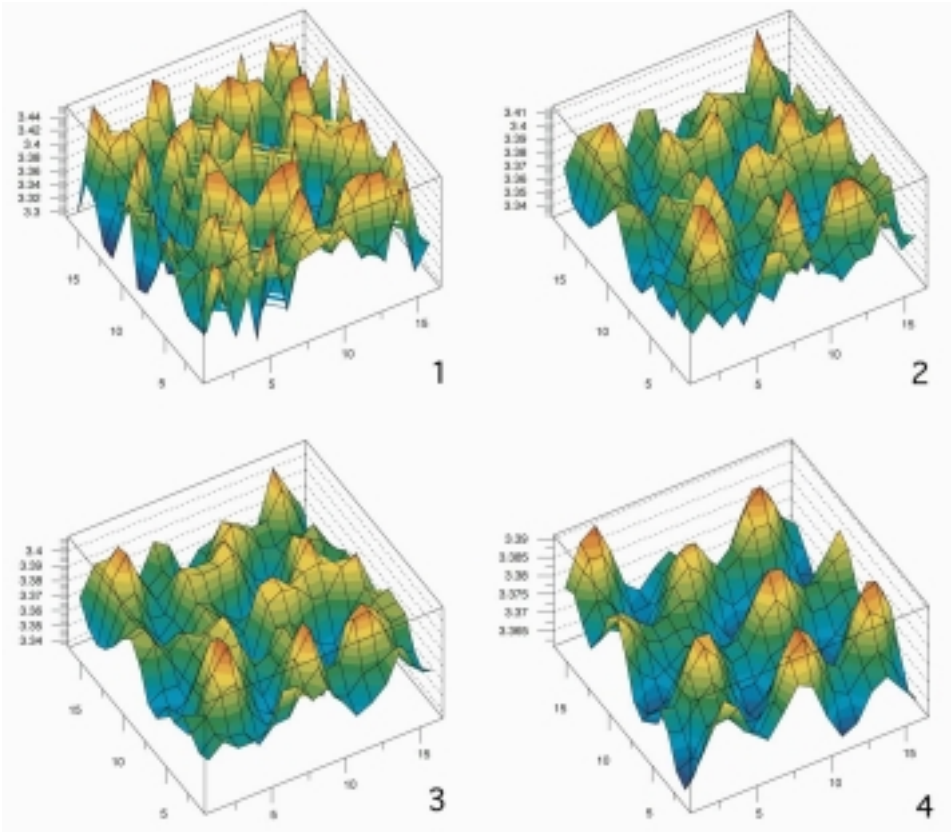


Fig. 2. Four simulation steps showing the temporal evolution of the structure built by termites in a 2D system. This simulation shows the evolution of the density of building material (on the z-axis) used by termites to build their nest obtained from Deneubourg's model [15]. The simulation begins with a random distribution of building material in space (step 1) and the regularity of the inter-pillar spacing emerges progressively over time (step 2, 3 and 4). After [15].

2 Pheromone trail following and discrete optimization

2.1 Foraging in ants

A lot of ant species have a trail-laying/trail-following behavior when foraging [42]: individual ants deposit pheromone while walking and foragers follow pheromone trails with some probability. Deneubourg et al. [16] have shown with an ingenious experiment, run with ants of the species *Linepithema humile*, that this behavior can explain how ants find the shortest path between their nest and a food source. The experimental setup is the following.

A food source is connected to an ant nest by a bridge with two equally long branches (Figure 3). When the experiment starts the ants select randomly, with equal probability, one of the branches. Because of statistical fluctuations

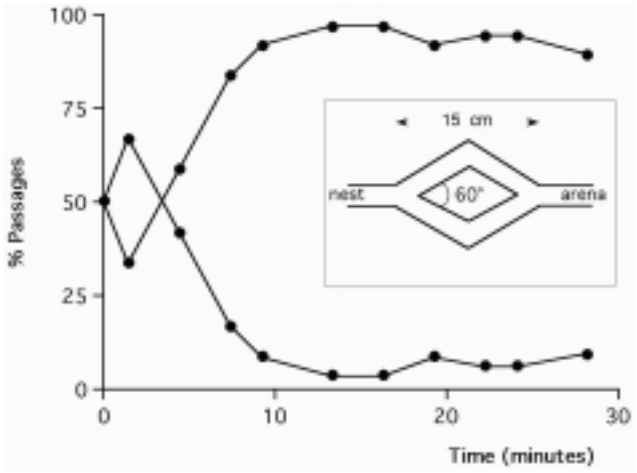


Fig. 3. Experimental setup (insert) and percentage of ants that used lower and upper branches as a function of time. Modified from Deneubourg et al., 1990 [16].

one of the two branches is chosen by a few more ants than the other and therefore is marked by a slightly higher amount of pheromone. The greater amount of pheromone on this branch stimulates more ants to choose it, and so on [17]. This autocatalytic process leads very soon the ant colony to converge towards the use of only one of the two branches.

The experiment can also be run using a bridge with two branches of different length. In this case, the first ants coming back to the nest are those that took the shortest path twice (to go from the nest to the source and to return to the nest), so that more pheromone is present on the short branch than on the long branch immediately after these ants have returned, stimulating nestmates to choose the short branch (Figure 4). This has been called *differential length effect* [25] and explains how ants in the long run end up choosing the shortest of the two paths without using any global knowledge about their environment. Differential length effect and pheromone based autocatalysis are at the earth of some successful ant algorithms for discrete optimization, in which an artificial pheromone plays the role of stigmergic variable, as explained in the following of this section. It is also interesting to note that in some ant species the amount of pheromone deposited is proportional to the quality of the food source found: paths that lead to better food sources receive a higher amount of pheromone. Similarly, in the ant algorithms presented in this section artificial ants deposit a quantity of pheromone proportional to the quality of the solution they found.

2.2 Ant System (AS) and the Traveling Salesman Problem

Ant System (AS) was the first algorithm [29,24] inspired by the trail following behavior of ants to be applied to a discrete optimization problem. The problem

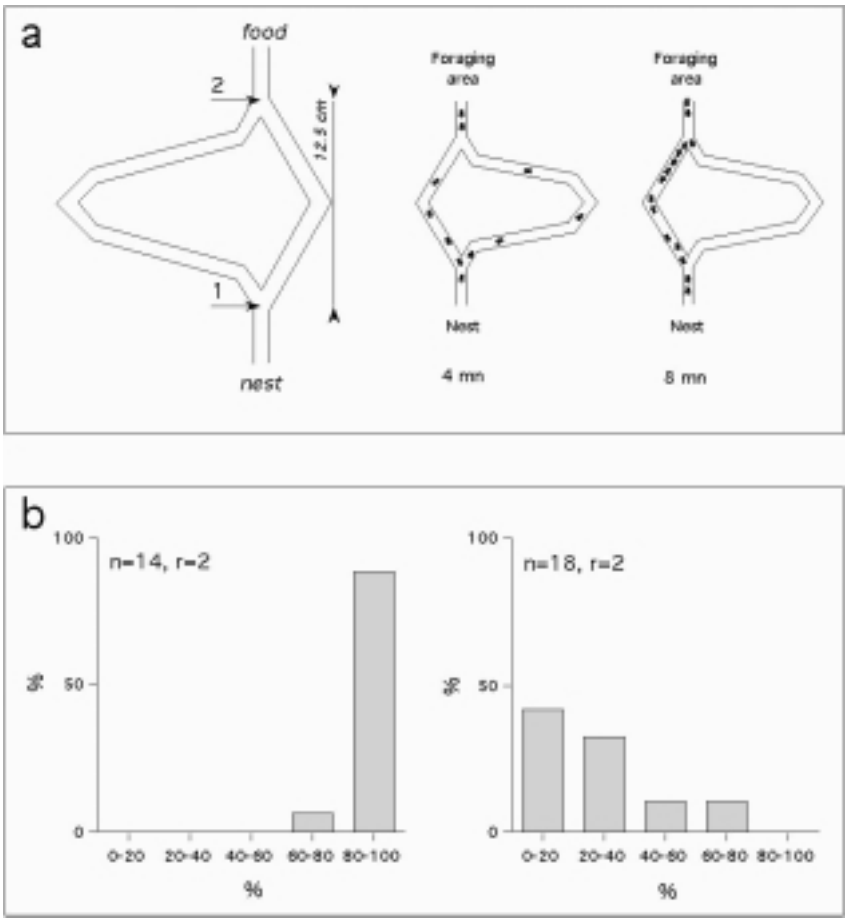


Fig. 4. (a) Experimental setup and drawings of the selection of the short branches by a colony of *Linephitema humile* respectively 4 and 8 minutes after the bridge was placed. (b) Distribution of the percentage of ants that selected the shorter branch over n experiments. The longer branch is r times longer than the short branch. The second graph ($n = 18$, $r = 2$) corresponds to an experiment in which the short branch is presented to the colony 30 minutes after the long branch: the short branch is not selected, and the colony remains trapped on the long branch. Modified from Goss et al., 1989 [38].

chosen for the first experiments was the Traveling Salesman Problem (TSP). In the TSP, one has to find a closed tour of minimal length connecting n given cities. Each city must be visited once and only once. Let d_{ij} be the distance between cities c_i and c_j. The problem can either be defined in Euclidean space (in which case d_{ij} is simply the Euclidean distance between cities i and j), or can be more generally defined on a graph $G = (V, E)$ where the cities are the vertices (V) and the connections between the cities are the edges of the graph (E). Note that the graph need not be fully connected and the distance matrix need not be symmetric: if it is asymmetric the corresponding problem is called the asymmetric TSP.

In AS the ants build solutions in parallel by visiting sequentially the cities of the graph. On each edge (i, j) of the TSP graph an *artificial pheromone trail* $\tau_{ij}(t)$ is maintained. The values $\tau_{ij}(t)$ are used by ants to direct the way they build tours. They are updated by means of a reinforcement procedure: once an ant has completed a tour it updates the edges it has crossed by adding a quantity of pheromone proportional to the goodness of the tour.

More formally, at iteration[4] t, after completing its tour $T_k(t)$, the k-th ant lays a quantity of pheromone $\Delta\tau_{ij}^k(t)$ on each edge (i, j) belonging to $T_k(t)$; $\Delta\tau_{ij}^k(t)$ is a function of the length L_k of tour $T_k(t)$:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L_k} & \text{if edge } (i,j) \in T_k(t) \\ 0 & \text{if edge } (i,j) \notin T_k(t) \end{cases} \tag{1}$$

where Q is an adjustable parameter.

Ants build solutions using a probabilistic transition rule. The probability $p_{ij}^k(t)$ with which an ant k in city i at iteration t chooses the next city j to move to is a function of:

- Whether or not city j has already been visited. For each ant, a list is maintained that contains all the cities that the ant has already visited in order to prevent cities from being visited more than once; the list grows within one tour until it is full, and is then emptied at the end of the iteration; we call $J_k(i)$ the set of cities that remain to be visited by ant k when ant k is in city i.
- An heuristic measure η_{ij} of the *desirability* of adding edge (i, j) to the solution under construction. In the TSP a reasonable heuristic is $\eta_{ij} = 1/d_{ij}$, that is, the inverse of the distance between cities i and j.
- The amount $\tau_{ij}(t)$ of *artificial pheromone* on the edge connecting i to j.

Formally $p_{ij}^k(t)$ is given by:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\displaystyle\sum_{l \in J_k(i)} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{if } j \in J_k(i) \\ 0 & \text{if } j \notin J_k(i) \end{cases} \tag{2}$$

where α and β are two adjustable parameters that control the relative influences of pheromone trail $\tau_{ij}(t)$ and heuristic desirability η_{ij}. If $\alpha = 0$, the closest cities are more likely to be selected: this corresponds to a classical

[4] The iteration counter is incremented by 1 when all ants have completed a tour.

stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed on the cities). If on the contrary $\beta = 0$, only pheromone amplification is at work: this method will lead the system to a *stagnation* situation, that is, to a situation in which all the ants generate a same, suboptimal tour [29,24,30]. The tradeoff between edge length and trail intensity therefore appears to be necessary.

Finally, AS could not perform well without pheromone evaporation. In fact, because the initial exploration of the search space is mostly random, the values of the pheromone trails in the initial phases are not very informative and it is therefore necessary that the systems slowly forgets these initial values to allow the ants to move towards better solutions. Pheromone decay is implemented by introducing a coefficient of evaporation $\rho, 0 \leq \rho < 1$, such that

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{3}$$

where $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$, and m is the number of ants.

The initial amount of pheromone on edges is assumed to be a small positive constant c (that is, there is an homogeneous distribution of pheromone at $t = 0$). The total number m of ants (assumed constant over time) is an important parameter. Too few ants will not produce the expected synergistic effects of cooperation [5] because of the (otherwise necessary) process of pheromone evaporation. On the contrary, too many ants result in a less efficient computational system: the quality of the results produced after a given number of iterations does not improve significantly, but, due to the higher number of ants, it takes longer to perform an algorithm iteration. Dorigo [24] suggests that $m = n$, that is, as many ants as there are cities in the problem, provides a good tradeoff.

Ant System has been tested on several relatively small problems. The experimentally optimized value of the parameters has been set to $\alpha = 1$, $\beta = 5$, $\rho = 0.5$ and $Q = 100$. Although the results obtained were not state-of-the-art on the TSP [29,24,30], AS compared well with other general purpose metaheuristic methods, like simulated annealing, evolutionary computation, and tabu search. But, most important, AS gave raise to a whole set of successful applications and extensions which have recently been unified in a novel metaheuristic called *Ant Colony Optimization* (ACO).

[5] Remember the termites nest building behavior of Section 1: as in that case, too few ants cannot overcome the evaporation of pheromone and stigmergic coordination cannot take place.

The ACO metaheuristic [25] is a novel metaheuristic obtained a posteriori after a careful analysis of the characteristics of a number of ant algorithms inspired by the foraging behavior of ants (most of these algorithms were strongly inspired by AS). ACO algorithms, that is, heuristic algorithms obtained as instances of the ACO metaheuristic, can be used to find feasible minimum cost paths over a graph $G = (C, L, W)$, where feasibility is defined with respect to a set Ω of constraints. The graph $G = (C, L, W)$ and the constraints Ω are defined as follows: $C = \{c_1, c_2, \ldots, c_n\}$ is a finite set of problem *components*, $L = \{l_{c_i c_j} \mid c_i, c_j \in C\}$ is a finite set of possible *connections* among the elements of C, W is a set of weights associated either to the components C or to the connections L or to both, and $\Omega(C, L, \theta)$ is a finite set of *constraints* assigned over the elements of C and L (θ indicates that the set of constraints can change over time). For example, in the traveling salesman problem defined in Section 2.2, C is the set of cities, L is the set of edges connecting cities, W is the length of the edges in L, and the constraints Ω impose that in any feasible solution each city appears once and only once. A feasible path over G is called a solution ψ and a minimum cost path is an optimal solution and is indicated by ψ^*; $f(\psi)$ is the cost of solution ψ, and $f(\psi^*)$ the cost of the optimal solution. In the TSP a solution ψ is an Hamiltonian circuit and ψ^* is the shortest feasible Hamiltonian circuit.

In ACO algorithms a colony of ants concurrently, asynchronously and incrementally build solutions of the problem defined by G and Ω. Each ant k starts with a partial solution $\psi_k(1)$ consisting of one element (one of the components in C) and adds components to $\psi_k(h)$ till a complete feasible solution ψ is built, where h is the step counter. Components to be added to $\psi_k(h)$ are stochastically chosen in an appropriately defined neighborhood of the last component added to $\psi_k(h)$. The ants stochastic choice is made by applying a stochastic local decision policy that makes use of local information available at the visited vertices/components. Once an ant has built a solution, or while the solution is being built, the ant evaluates the (partial) solution and adds pheromone, that is, information about the quality of the (partial) solution, on the components and/or the connections it used. This pheromone information will direct the search of the ants in the following iterations.

Besides ants' activity, an ACO algorithm includes a *pheromone trail evaporation* procedure and an optional *daemon actions* procedure. Pheromone evaporation, as it was the case in Ant System, is the process by which the pheromone trail automatically decreases over time. 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 perspective. As a practical example, the daemon can observe the path found by each ant in the colony and choose to deposit extra pheromone on the edges used by the ant that made the shortest path. In most applications to combinatorial optimization problems the *ants_activity()*, *pheromone_evaporation()* and *daemon_actions()* procedures (see Figure 5) are scheduled sequentially. Nevertheless, the **schedule_activities** construct of the ACO metaheuristic (Figure 5) leaves the decision on how these three procedures must be synchronized to the user, that is left free to match synchronization policies to the considered problems (for example, in the applications to routing in telecommunications networks the execution of the three procedures is often interleaved).

The ACO metaheuristic, which has been introduced in [25] where the interested reader can find a more detailed formal definition, has been successfully applied to many discrete optimization problems, as listed in Table 1. Among the most studied problems there are the traveling salesman, the quadratic assignment and routing in telecommunication networks. When applied to these problems ACO algorithms result to be competitive with the best available heuristic approaches. In particular:

- Results obtained by the application of ACO algorithms to the TSP are very encouraging (ACO algorithms for the TSP are overviewed in [73]): they are often better than those obtained using other general purpose heuristics like evolutionary computation or simulated annealing. Also, when adding to ACO algorithms local search procedures based on 3-opt [50], the quality of the results obtained [28,72] is close to that obtainable by other state-of-the-art methods.
- ACO algorithms are currently one of the best performing heuristics available for the particularly important class of quadratic assignment problems which model real world problems [57,56,53,37].
- AntNet [21,23], an ACO algorithm for routing in packet switched networks, outperformed a number of state-of-the-art routing algorithms for a set of benchmark problems. AntNet-FA, an extension of AntNet for connection oriented network routing problems, also shows competitive performance [23].
- HAS-SOP, an ACO algorithm coupled to a local search routine, has improved many of the best known results on a wide set of benchmark instances of the sequential ordering problem (SOP) [34], that is, the problem of finding the shortest Hamiltonian path on a graph which satisfies a set of precedence constraints on the order in which cities are visited.

ACO algorithms have also been applied to a number of other discrete optimization problems like the shortest common supersequence problem, the vehicle routing problem, the multiple knapsack, single machine total tardiness, and others (see Table 1), with very promising results.

procedure *ACO metaheuristic*
  while end condition not satisfied **do**
    schedule_activities
      ants_activity()
      pheromone_evaporation()
      daemon_actions()
    end schedule_activities
  end while
end *ACO metaheuristic*

Fig. 5. Outline of the Ant Colony Optimization metaheuristic.

Table 1. 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	[24,29,30]
	Gambardella & Dorigo	Ant-Q	1995	[32]
	Dorigo & Gambardella	ACS & ACS-3-opt	1996	[27,28,33]
	Stützle & Hoos	$\mathcal{MM}$AS	1997	[75,76,78]
	Bullnheimer, Hartl & Strauss	AS_{rank}	1997	[8,10]
Quadratic assignment	Maniezzo, Colorni & Dorigo	AS-QAP	1994	[57]
	Gambardella, Taillard & Dorigo	HAS-QAP[a]	1997	[36,37]
	Stützle & Hoos	$\mathcal{MM}$AS-QAP	1997	[74,77]
	Maniezzo	ANTS-QAP	1998	[53]
	Maniezzo & Colorni	AS-QAP[b]	1999	[56]
Scheduling problems	Colorni, Dorigo & Maniezzo	AS-JSP	1994	[12]
	Stützle	AS-FSP	1997	[71]
	Bauer *et al.*	ACS-SMTTP	1999	[1]
	den Besten, Stützle & Dorigo	ACS-SMP	1999	[14]
Vehicle routing	Bullnheimer, Hartl & Strauss	AS-VRP	1997	[7,9]
	Gambardella, Taillard & Agazzi	HAS-VRP	1999	[35]
Connection-oriented network routing	Schoonderwoerd *et al.*	ABC	1996	[69,68]
	White, Pagurek & Oppacher	ASGA	1998	[87]
	Di Caro & Dorigo	AntNet-FS	1998	[22]
	Bonabeau *et al.*	ABC-smart ants	1998	[3]
Connection-less network routing	Di Caro & Dorigo	AntNet & AntNet-FA	1997	[20,21,23]
	Subramanian, Druschel & Chen	Regular ants	1997	[79]
	Heusse *et al.*	CAF	1998	[41]
	van der Put & Rothkrantz	ABC-backward	1998	[83,84]
Sequential ordering	Gambardella & Dorigo	HAS-SOP	1997	[34]
Graph coloring	Costa & Hertz	ANTCOL	1997	[13]
Shortest common supersequence	Michel & Middendorf	AS-SCS	1998	[58,59]
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998	[54,55]
Generalized assignment	Ramalhinho Lourenço & Serra	MMAS-GAP	1998	[65]
Multiple knapsack	Leguizamón & Michalewicz	AS-MKP	1999	[48]
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999	[61]
Redundancy allocation	Liang & Smith	ACO-RAP	1999	[49]

[a] HAS-QAP is an ant algorithm which does not follow all the aspects of the ACO meta-heuristic.
[b] This is a variant of the original AS-QAP.

3 Labor division and task allocation

3.1 Division of labor in ant colonies

Division of labor is an important and widespread feature of life in ant colonies, and in social insects in general (for a review, see, e.g., [67,70,63]). Social insects are all characterized by one fundamental type of division of labor, reproductive division of labor, a main ingredient in the definition of eusociality [6]. Beyond this primary form of division of labor between reproductive and worker castes, there most often exists a further division of labor among workers, who tend to perform specific tasks for some amount of time, rather than to be generalists who perform various tasks all the time. Workers are divided into age or morphological subcastes. Age subcastes correspond to individuals of the same age that tend to perform identical tasks: this phenomenon is called temporal polyethism. In some species, workers can have different morphologies: workers that belong to different morphological castes tend to perform different tasks. But even within an age or morphological caste, there may be differences among individuals in the frequency and sequence of task performance: one may therefore speak of behavioral castes, to describe groups of individuals that perform the same set of tasks in a given period.

One of the most striking aspects of division of labor is plasticity, a property achieved through the workers' behavioral flexibility: the ratios of workers performing the different tasks that maintain the colony's viability and reproductive success can vary (i.e., workers switch tasks) in response to internal perturbations or external challenges. An important question is to understand how this flexibility is implemented at the level of individual workers, which do not possess any global representation of the colony's needs.

3.2 A simple model of task allocation in ant colonies

Bonabeau et al. [5] have developed a simple model for task allocation in ants, based on the notion of response threshold [66,67]: individuals start to become engaged in task performance when the level of the task-associated stimuli, which plays the role of stigmergic variable, exceeds their threshold. Differences in response thresholds may either reflect actual differences in behavioral

[6] Eusociality characterizes the highest level of sociality in the animal kingdom; an animal group is said to be eusocial when the three following traits are present: (i) a cooperation in caring for the young, (ii) a reproductive division of labour with more or less sterile individuals working on behalf of individuals engaged in reproduction, and (iii) an overlap of generations.

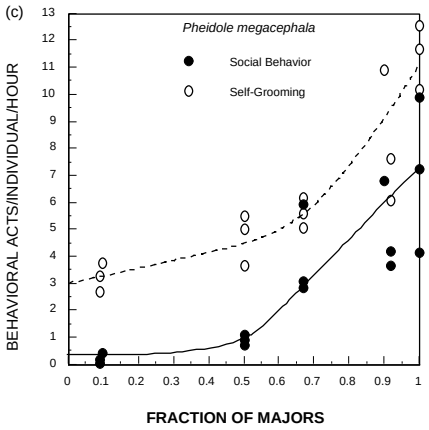


Fig. 6. Number of behavioral acts (social behavior and self-grooming) per major per hour as a function of the fraction of majors in the colony for the species *Pheidole megacephala*. Fitting lines are only visual aids. After Wilson [89]. Reprinted by permission. © Springer-Verlag.

responses, or differences in the way task-related stimuli are perceived. When specialized individuals performing a given task are withdrawn (they have low response thresholds with respect to stimuli related to this task), the associated task demand increases and so does the intensity of the stimulus, until it eventually reaches the higher characteristic response thresholds of the remaining individuals that are not initially specialized into that task; the increase of stimulus intensity beyond threshold has the effect of stimulating these individuals into performing the task. This is exactly what was observed by Wilson [89] in experiments where he artificially reduced the minor/major (minor and major are two ant castes) ratio to below 1 and observed a change in the rate of activity within 1 hour of the ratio change: for small ratios, majors engage in tasks usually performed by minors and efficiently replace the missing minors (the results of one of these experiments are shown in Figure 6).

What is a response threshold? Let s be the intensity of a stimulus associated with a particular task: s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. A response threshold θ, expressed in units of stimulus intensity, is an internal variable that determines the tendency of an individual to respond to the stimulus s and perform the associated task. More precisely, θ is such that the probability of response is low for $s \ll \theta$ and high for $s \gg \theta$. One family of response functions $T_\theta(s)$ that can be parametrized with thresholds that satisfy this requirement is given by

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n} \tag{4}$$

where $n > 1$ determines the steepness of the threshold. In the rest of the section, we use $n = 2$, but similar results can be obtained with other values of $n > 1$. The meaning of θ is clear: for $s \ll \theta$, the probability of engaging

in task performance is close to 0, and for $s \gg \theta$, this probability is close to 1; at $s = \theta$, this probability is exactly 1/2. Therefore, individuals with a lower value of θ are likely to respond to a lower level of stimulus.

Assume that there are two castes and that only one task needs to be performed. This task is associated with a stimulus or demand, the level of which increases if it is not satisfied (because the task is not performed by enough individuals, or not performed with enough efficiency). Let $\mathcal{S}_i$ be the state of an individual i ($\mathcal{S}_i = 0$ corresponds to inactivity, $\mathcal{S}_i = 1$ corresponds to performing the task), and θ_i the response threshold of i, $i = 1, 2$.

An inactive individual starts performing the task with a probability P per unit time:

$$P(\mathcal{S}_i = 0 \to \mathcal{S}_i = 1) = \frac{s^n}{s^n + \theta_i^n} \tag{5}$$

The probability that an individual will perform a task depends on s, the magnitude of the task-associated stimulus, that affects the probability of being exposed to it, and on θ_i, the probability of responding to task-related stimuli [67].

An active individual gives up task performance and becomes inactive with probability p per time unit (that we take identical for the two castes, i.e., $p_1 = p_2 = p$):

$$P(\mathcal{S}_i = 1 \to \mathcal{S}_i = 0) = p \tag{6}$$

$1/p$ is the average time spent by an individual in task performance before giving up the task. It is assumed that p is fixed, and independent of the stimulus. Individuals give-up task performance after $1/p$, but may become engaged again immediately if the stimulus is still large. Variations in stimulus intensity are due to task performance, which reduces stimulus intensity, and to the autonomous increase of demand, that is, irrespective of whether or not the task is performed. The resulting equation for the evolution of stimulus intensity s is therefore (in discrete time):

$$s(t + 1) = s(t) + \delta - \alpha \cdot n_{act} \tag{7}$$

where δ is the increase, supposed constant, in stimulus intensity per unit time, n_{act} is the number of active individuals, and α is a scale factor measuring the decrease in stimulus intensity due to the activity of one individual, that is, the efficiency of individual task performance. In Monte Carlo simulations [5], this simple fixed threshold model shows remarkable agreement with experimental

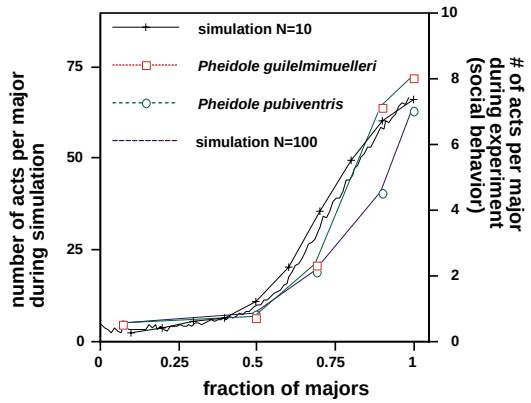


Fig. 7. Comparison between simulation results and real ants data. On the left vertical axis: number of acts per majors during Monte Carlo simulations as a function of the fraction f of majors in the colony (parameters: $\theta_1 = 8, \theta_2 = 1, \alpha = 3, \delta = 1, p = 0.2$). On the right vertical axis: Wilson's [89] results (scaled so that curves of model and experiments lie within the same range): number of social behavior acts per major within time of experiments in *Pheidola guilelmimuelleri* and *Pheidole pubiventris* as a function of the fraction f of majors in the colony. After [5]. Reprinted by permission. © Elsevier.

results in the case where there are two castes characterized by two different values of the response threshold: when "minors", with low response thresholds, are removed from the simulated colony, "majors", with higher response thresholds, start to perform tasks usually performed by minors. Figure 7 shows the fraction of majors engaged in task performance as a function of the fraction of majors in the colony. This curve is very similar to the one observed by Wilson [89]. This simple model with one task can be easily extended to the case where there are two or more tasks to perform. In this case, each individual has a set of thresholds, each threshold being associated to the stimulus of a specific task or group of tasks.

The fixed-threshold model described above has been used to organize a group of robots by Krieger and Billeter [44]. They designed a group of Khepera robots (miniature mobile robots aimed at "desktop" experiments [60]) to collectively perform a puck-foraging task. In one of the experiments they performed, pucks spread in the environment are taken back by the robots to the "nest" where they are dropped in a basket. The available "energy" of the group, which plays the role of stigmergic variable, decreases regularly with time, but increases when pucks are dropped into the basket. More energy is consumed during foraging trips than when robots are immobile in the nest. Each robot has a foraging threshold: when the energy of the colony goes below the foraging threshold of a robot, the robot leaves the nest to look for pucks in the environment. Krieger and Billeter's experiment has shown the viability of the threshold-based stigmergic approach to self-organization in a rather simple environment. Further experimentation is necessary to test the

methodology on more complex tasks.

3.3 Adaptive task allocation: the example of mail retrieval

The simple response threshold model, which assumes that each worker responds to a given stimulus when stimulus intensity exceeds the worker's threshold, can explain how flexibility at the colony level results from the workers' behavioral flexibility [5]. But it has several limitations, because it assumes that workers' thresholds are fixed over the studied time scale. In fact, it cannot account for the genesis of task allocation for it assumes that individuals are differentiated and roles preassigned. Neither can it account for robust task specialization within (physical or temporal) castes. Finally, as a model of real ants behavior it is valid only over sufficiently short time scales, where thresholds can be considered constant.

In order to overcome these limitations Theraulaz et al. [82,81] have extended the fixed threshold model by allowing thresholds to vary in time, following a simple reinforcement process: a threshold decreases when the corresponding task is performed, and increases when the corresponding task is not performed. This idea had been previously introduced by Oster [62], Deneubourg et al. [19], and Plowright and Plowright [64], who did not attempt to explore its consequences in detail, especially when several tasks need to be performed. It is this model with threshold reinforcement that has been applied by Bonabeau et al. [4] to a problem of adaptive mail retrieval.

Imagine that a group of mailmen belonging to an express mail company have to pick-up letters in a city. Customers should not have to wait more than a given amount of time: the aim of the mail company is therefore to allocate the mailmen to the various demands that appear in the course of the day so as to keep the global demand as low as possible. The probability that mailman i, located in zone z_i, responds to a demand intensity s_j in zone j is given by

$$p_{ij} = \frac{s_j^2}{s_j^2 + \alpha\theta_{ij}^2 + \beta d_{z_i j}^2} \tag{8}$$

where $\theta_{ij} \in [\theta_{min}, \theta_{max}]$ is a response threshold of mailman i to a demand from zone j, $d_{z_i j}$ is the distance between z_i and j (this distance can either be Euclidean or include factors such as one-ways, lights, traffic jams, etc.), and α and β are two parameters that modulate the respective influences of θ and d. Each time a mailman allocates himself to zone j to retrieve mail, his response thresholds are updated in the following way:

$$\theta_{ij} \leftarrow \theta_{ij} - \xi_0 \tag{9}$$

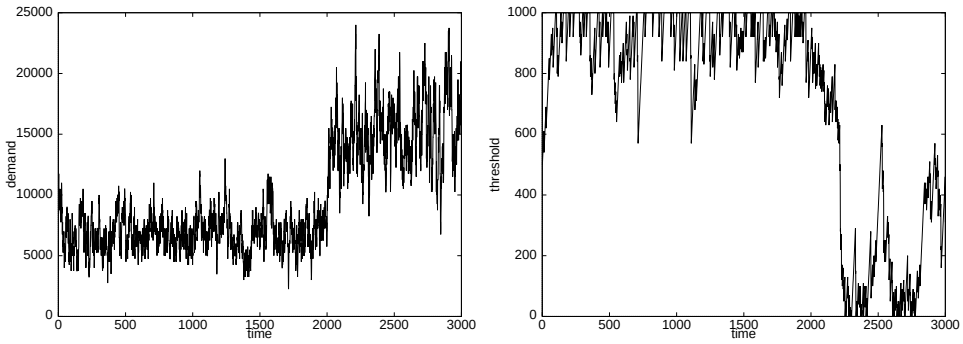


Fig. 8. Adaptive task allocation: simulation results. (a, left side) Demand as a function of time (one iteration at each time step): one mailman is removed at time $t = 2,000$. (b, right side) Threshold dynamics of a particular mailman with respect to the zone for which a specialist is removed at time $t = 2,000$. After [4]. Reprinted by permission. © World Scientific.

$$\theta_{il} \leftarrow \theta_{il} - \xi_1, l \in n_j \tag{10}$$

$$\theta_{ik} \leftarrow \theta_{ik} + \phi, \text{ for } k \neq j \text{ and } k \notin n_j \tag{11}$$

where n_j is the set of zones surrounding j, ξ_0 and ξ_1 are two learning coefficients corresponding to the new zone where that agent moved, and ϕ is the forgetting coefficient applied to response thresholds associated with other zones.

Simulations have been performed with a grid of 5×5 zones (we consider four neighbors for the update of equation (11), with periodic boundary conditions), and 5 mailmen; at every iteration, the demand increases in 5 randomly selected zones by an amount of 50, $\alpha = 0.5$, $\beta = 500$, $\theta_{min} = 0$, $\theta_{max} = 1,000$, $\xi_0 = 150$, $\xi_1 = 70$, $\phi = 10$. Mailmen are swept in random order, and they decide to respond to the demand from a particular zone according to equation (8). If no mailman responds after 5 sweepings, the next iteration starts. If a mailman responds, this mailman will be unavailable for an amount of time that we take to be equal to the distance separating his current location from the zone where the demand comes from. Once the mailman decides to allocate himself to that zone, the associated demand in that zone is maintained at 0 (since any demand emerging between the time of the mailman's response and his arrival in the zone will be satisfied by the same mailman). Figure 8a shows how the demand increases but is still kept under control when one mailman fails to perform his task. Figure 8b shows how the threshold of a mailman with respect to a single zone can vary as a function of time. A special behavior can be observed after the removal of a mailman specialist of a given zone: another mailman lowers his threshold with respect to that zone and becomes in turn a new specialist of that zone. This is what is observed in Figure 8b. However, because the workload may be too high to allow mailmen to settle into a given specialization, response thresholds may oscillate in time. All these features point to the flexibility and robustness of this algorithm.

Although we have presented the performance of the algorithm on one specific example, it can certainly be modified to apply to virtually any kind of task allocation problem: the demand s_j can be the abstract demand associated to some task j, θ_{ij} is a response threshold of actor i with respect to the task-associated stimulus s_j. Finally, $d_{z_i j}$ is an abstract distance between i and task j which can, for example, represent the ability or lack of ability of i to deal with task j: if i is not the most efficient actor to perform task j, it will not respond preferentially to s_j, but if no other actor is in a position to respond, it will eventually perform the task. It is certainly possible to design a scheme in which d can vary depending on the efficiency of i in performing task j.

4 Cemetery formation and exploratory data analysis

4.1 Cemetery organization

Chrétien [11] has performed intensive experiments on the ant *Lasius niger* to study the organization of cemeteries. Other experiments on the ant *Pheidole pallidula* are also reported in [18], and it is now known that many species actually organize a cemetery. The phenomenon that is observed in these experiments is the aggregation of dead bodies by workers. If dead bodies, or more precisely items belonging to dead bodies, are randomly distributed in space at the beginning of the experiment, the workers will form clusters within a few hours (see Figure 9). If the experimental arena is not sufficiently large, or if it contains spatial heterogeneities, the clusters will be formed along the borders of the arena or more generally along the heterogeneities. The basic mechanism underlying this type of aggregation phenomenon is an attraction between dead items mediated by the ant workers: small clusters of items grow by attracting workers to deposit more items. It is this positive feedback that leads to the formation of larger and larger clusters. In this case it is therefore the distribution of the clusters in the environment that plays the role of stigmergic variable.

Deneubourg et al. [18] have proposed a model relying on biologically plausible assumptions to account for the above-mentioned phenomenon of dead body clustering in ants. The model, called in the following *basic model* (BM), relies on the general idea that isolated items should be picked-up and dropped at some other location where more items of that type are present. Let us assume that there is only one type of item in the environment. The probability for a randomly moving ant that is currently not carrying an item to pick-up an

item is given by

$$p_p = \left(\frac{k_1}{k_1 + f} \right)^2 \tag{12}$$

where f is the perceived fraction of items in the neighborhood of the ant and k_1 is a threshold constant: for $f \ll k_1$, p_p is close to 1 (that is, the probability of picking-up an item is high when there are not many items in the neighborhood), and p_p is close to 0 if $f \gg k_1$ (that is, items are unlikely to be removed from dense clusters). The probability p_d for a randomly moving loaded ant to deposit an item is given by:

$$p_d = \left(\frac{f}{k_2 + f} \right)^2 \tag{13}$$

where k_2 is another threshold constant: for $f \ll k_2$, p_d is close to 0, whereas for $f \gg k_2$, p_d is close to 1. As expected, the pick-up and deposit behaviors obey roughly opposite rules. The question is now to define how f is evaluated. Deneubourg et al. [18], having in mind a robotic implementation, moved away from biological plausibility and assumed that f is computed thanks to a short-term memory that each ant possesses: an ant keeps track of the last T time units, and f is simply the number N of items encountered during these last T time units divided by the largest possible number of items that can be encountered during T time units. If one assumes that only 0 or 1 object can be found within a time unit, then $f = N/T$. Figure 10 shows a simulation of this model: small evenly spaced clusters emerge within a relatively short time and then merge into fewer larger clusters. BM can be easily extended to the case in which there are more than one type of items. Consider, for example, the case with two types a and b of items in the environment. The principle is the same as before, but now f is replaced by f_a and f_b, the respective fractions of items of types a and b encountered during the last T time units. Figure 11 shows a simulation of this sorting model with two items.

4.2 Exploratory data analysis

Lumer and Faieta [51] have generalized Deneubourg et al.'s BM [18] to apply it to exploratory data analysis. The idea here is to define a "dissimilarity" d (or distance) between objects in the space of object attributes: for instance, in BM, two objects o_i and o_j can only be either similar or different, so that a binary metric can be defined, where $d(o_i, o_j) = 0$ if o_i and o_j are identical objects, and $d(o_i, o_j) = 1$ otherwise. Obviously, the very same idea can be extended to include more complicated objects, that is, objects with more attributes, and/or

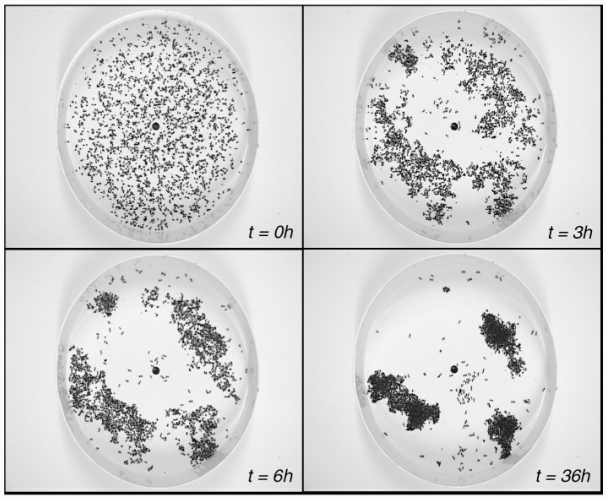


Fig. 9. Real ants clustering behavior. The figures show four successive pictures of the circular arena. From left to right and from up to down: the initial state, after 3, 6 and 36 hours respectively.

more complicated distances. It is classical in data analysis to have to deal with objects that can be described by a finite number n of real-valued attributes, so that objects can be seen as points in $\mathbb{R}^n$ and $d(o_i, o_j)$ is the Euclidean norm (or any other usual metric, such as $\| \dots \|_\infty$). The algorithm introduced by Lumer and Faieta [51] (hereafter LF) consists in projecting the space of attributes onto some lower dimensional space, so as to make clusters appear with the following property: intra-cluster distances (that is, attribute distances between objects within clusters) should be small with respect to inter-cluster distances (that is, attribute distances between objects that belong to different clusters). Such a mapping should therefore keep some of the neighborhood relationships present in the higher-dimensional space (which is relatively easy since, for instance, any continuous mapping can do the job) without creating too many new neighbors in m-dimensions, $m < n$, that would be false neighbors in n-dimensions (which is much less trivial since projections tend to compress

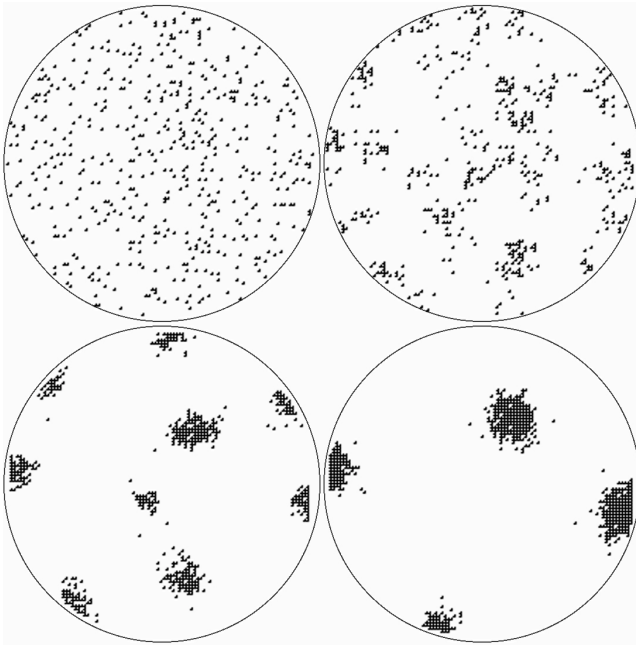


Fig. 10. Computer simulation of the clustering model. The figures show four successive pictures of the simulated circular arena (diameter = 200 grid sites; total area: 31,416 sites). From left to right and from up to down: the initial state, with 5,000 items placed randomly in the arena, the arena at $t = 50,000$, at $t = 1,000,000$, and at $t = 5,000,000$. Parameters: $T = 50, k_1 = 0.1, k_2 = 0.3$, 10 ants.

information and may map several well-separated points in the n-dimensional space onto one single point in the m-dimensional subspace).

The LF algorithm works as follows. Let us assume that $m = 2$; instead of embedding the set of objects into $\mathbb{R}^2$, the LF algorithm approximates this embedding by considering a grid, that is, a subspace of $\mathbb{Z}^2$. Ants can directly perceive a surrounding region of area $s^2 - 1$ (a square $Neigh_{s\times s}$ of $s \times s$ sites surrounding site r). Obviously, direct perception allows a more efficient evaluation of the state of the neighborhood than the memory-based procedure used in the BM: while the BM was aimed to a robotic implementation, the LF algorithm is to be implemented in a computer, with many less practical constraints. Let $d(o_i, o_j)$ be the distance between two objects o_i and o_j in the space of attributes. Let us also assume that an ant is located at site r at time t, and finds an object o_i at that site. The local density of objects similar to type o_i at site r is given by

$$f(o_i) = \max\left\{0, \frac{1}{s^2} \sum_{o_j \in Neigh_{s\times s}(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha}\right]\right\} \tag{14}$$

$f(o_i)$ is a measure of the average similarity of object o_i with the other objects o_j present in its neighborhood: this expression replaces the fraction f of similar

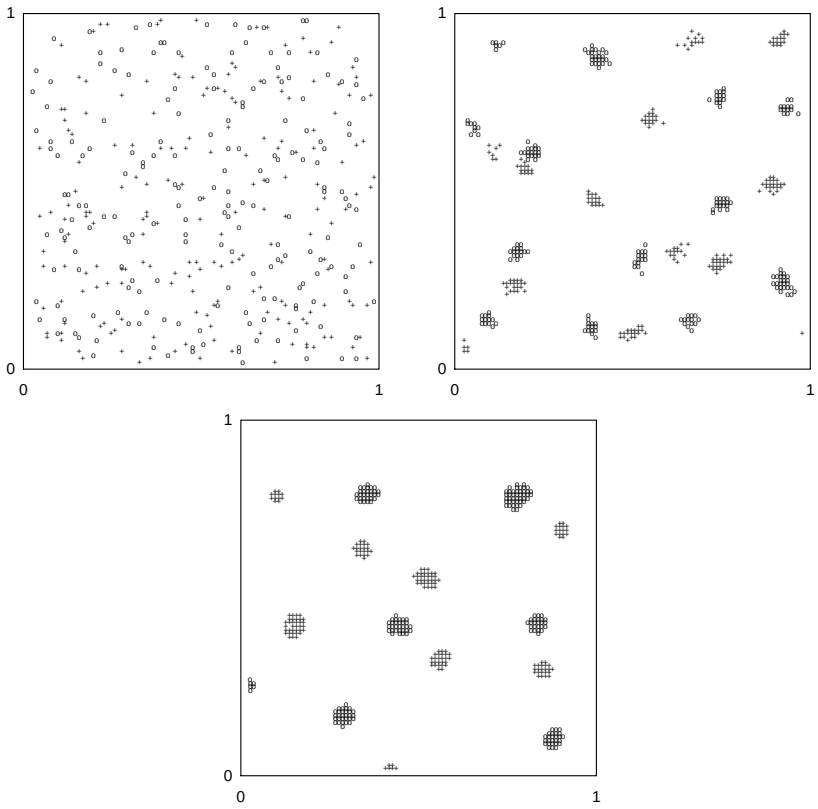


Fig. 11. Simulation of the sorting model. From left to right and from up to down: (a) initial spatial distribution of 400 items of two types, denoted by o and +, on a 100×100 grid; (b) spatial distribution of items at $t = 500,000$, and (c) at $t = 5,000,000$. Parameters: $T = 50, k_1 = 0.1, k_2 = 0.3$, 10 ants. After [2]. Reprinted by permission. © Oxford University Press.

objects of the BM. The parameter α defines the scale for dissimilarity: its value is important for it determines when two items should or should not be located next to each other. For example, if α is too large, there is not enough discrimination between different items, leading to the formation of clusters composed of items which should not belong to the same cluster. If, on the other hand, α is too small, distances between items in attribute space are amplified to the point where items which are relatively close in attribute space cannot be clustered together because discrimination is too high. Lumer and Faieta [51] define picking-up and dropping probabilities as follows:

$$p_p(o_i) = \left(\frac{k_1}{k_1 + f(o_i)}\right)^2, \qquad p_d(o_i) = \begin{cases} 2 \cdot f(o_i) & \text{if } f(o_i) < k_2 \\ \\ 1 & \text{if } f(o_i) \geq k_2 \end{cases} \tag{15}$$

where k_1 and k_2 are two constants that play a role similar to k_1 and k_2 in the BM.

As an illustration, Lumer and Faieta [51] have used a simple example where

the attribute space is $\mathbb{R}^2$, and the values of the two attributes for each object correspond to its coordinates (x, y) in $\mathbb{R}^2$. Four clusters of 200 points each are generated in attribute space, with x and y distributed according to normal (or Gaussian) distributions (see Figure 12a for the scatter of points in attribute space). The data points were then assigned random locations on a 100×100 grid, and the clustering algorithm was run with 10 ants. Figures 12b, 12c and 12d show the system at $t = 0$, $t = 500,000$ and $t = 1,000,000$ (at each iteration, indexed by the counter t, all ants have made a random move and possibly performed an action). Objects that are clustered together belong to the same initial distribution, and objects that do not belong to the same initial distribution are found in different clusters.

Because there are generally more clusters in the projected system than in the initial distribution, Lumer and Faieta [51] have added three features to their systems that help to solve this problem:

Ants with different moving speeds. Let the speed v of an ant be distributed uniformly in $[1, v_{max}]$ (v is the number of grid units walked per unit time by an ant along a given grid axis; the simulations use $v_{max} = 6$). The speed v influences the tendency of an ant to either pick-up or drop an object through the function $f(o_i)$:

$$f(o_i) = \max\left\{0, \frac{1}{s^2} \sum_{o_j \in Neigh_{s\times s}(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha \cdot \left(1 + \frac{v-1}{v_{max}}\right)}\right]\right\} \qquad (16)$$

Fast moving ants are not as selective as slow ants in their estimation of the average similarity of an object to its neighbors. The diversity of ants allows to form the clusters over various scales simultaneously: fast ants form coarse clusters on large scales, that is, drop items approximately in the right coarse-grained region, while slow ants take over at smaller scales by placing objects with more accuracy.

A short-term memory. Ants can remember the last m items they have dropped along with their locations. Each time an item is picked up, the ant compares the properties of the item with those of the m memorized items and goes toward the location of the most similar instead of moving randomly. This behavior leads to a reduction in the number of statistically equivalent clusters, since similar items have a lower probability of initiating independent clusters.

Behavioral switches. Ants can start to destroy clusters if they haven't performed any pick up or deposit actions for a given number of time steps. This procedure allows to "heat-up" the system to escape local non-optimal configurations.

Figure 13, which should be compared with Figure 12d, shows the system at

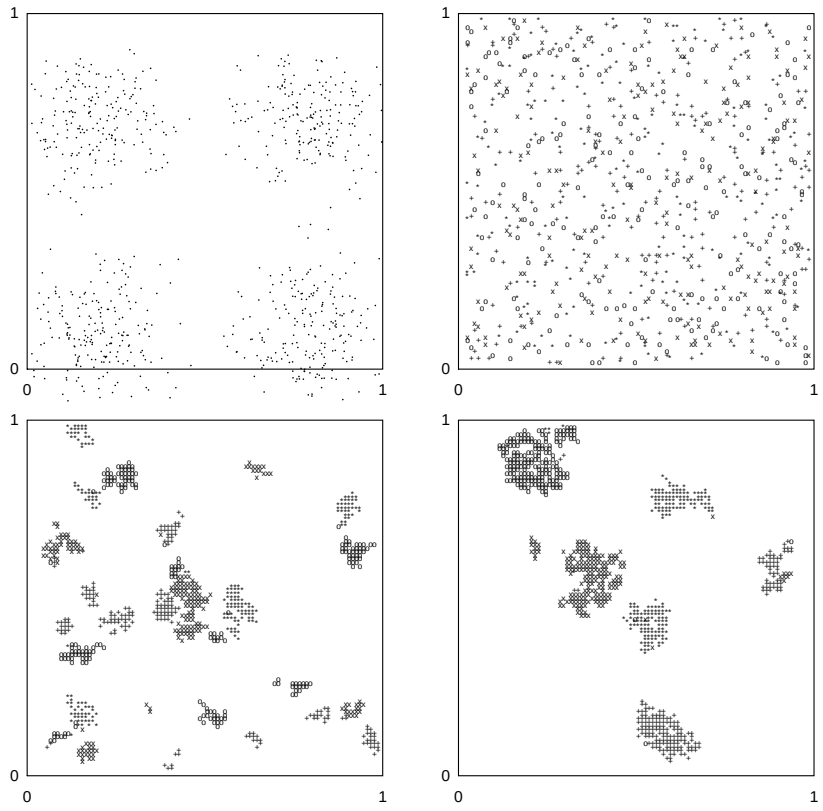


Fig. 12. Simulation of the clustering algorithm with 10 ants. From left to right and from up to down: (a) distribution of points in "attribute space": 4 clusters of 200 points each are generated in attribute space, with x and y distributed according to normal (or Gaussian) distributions $N(\mu, \sigma)$: $[x \propto N(0.2, 0.1), y \propto N(0.2, 0.1)]$, $[x \propto N(0.8, 0.1), y \propto N(0.2, 0.1)]$, $[x \propto N(0.8, 0.1), y \propto N(0.8, 0.1)]$, and $[x \propto N(0.2, 0.1), y \propto N(0.8, 0.1)]$, for clusters 1, 2, 3 and 4 respectively; (b) initial spatial distribution of the 800 items on a 100×100 grid (grid coordinates are scaled in the unit square); (c) distribution of the items at $t = 500,000$; (d) distribution of the items at $t = 1,000,000$. Items that belong to different clusters are represented by different symbols: $o, +, *, \times$. Parameters: $k_1 = 0.1, k_2 = 0.15, \alpha = 0.5, s^2 = 9$. After [2]. Reprinted by permission. © Oxford University Press.

$t = 1,000,000$ in the case of ants with different speeds and short term memory. The effects of behavioral switches, not included here, can be found in [51].

Lumer and Faieta [51] suggest that their algorithm is halfway between a cluster analysis – insofar as elements belonging to different concentration areas in their n-dimensional space end up in different clusters – and a multidimensional scaling, in which an intracluster structure is constructed. Note that in the present example, the exact locations of the various clusters on the 2-dimensional space are arbitrary, whereas they usually have a meaning in classical factorial analysis. In a lot of cases, information about the locations of the clusters is not necessary or useful (especially in the context of textual databases), and relaxing the global positioning constraints allows to speed-up

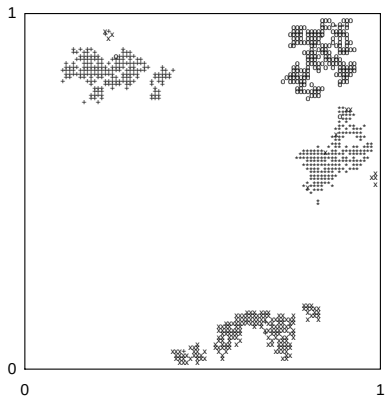


Fig. 13. Extended clustering algorithm at $t = 1,000,000$. There are 10 ants and 800 items on a 100×100 grid (grid coordinates are scaled in the unit square). Items that belong to different clusters are represented by different symbols: $o, +, *, \times$. Parameters: $k_1 = 0.1, k_2 = 0.15, \alpha = 0.5, s^2 = 9, m = 8, v_{max} = 6$. After [2]. Reprinted by permission. © Oxford University Press.

the clustering process significantly.

Finally, we mention that the LF algorithm has been successfully extended by Kuntz, Layzell and Snyers [45–47] so that it can be applied to a variety of graph drawing and graph partitioning problems. In this case the objects moved around by the artificial ants are projections on a space $\mathbb{R}^n$ of the vertices of the graph, and the ants goal is to find configurations of these objects that either minimize some objective function (in the graph partitioning applications) or please the observer eye (in the graph drawing applications).

5 Conclusions

In this article we informally defined an *ant algorithm* to be a multi-agent system inspired by the observation of some real ant colony behavior exploiting the stigmergic communication paradigm. In ant algorithms stigmergic communication is implemented by means of a stigmergic variable which takes different forms in the different applications: artificial pheromone trail in shortest path problems, level of nest energy in puck-foraging, level of customer demand in the mailmen example, puck distribution in robotic clustering, and the distribution of objects in the lower-dimensional space in exploratory data analysis. Ant algorithms exhibit a number of interesting properties like flexibility (a colony responds to internal perturbations and external challenges), robustness (tasks are completed even if some individuals fail), decentralization (there exists no central control) and self-organization (solutions to problems faced by a colony are emergent rather than predefined), which make them well-suited for the so-

lution of problems that are distributed in nature, dynamically changing, and require built-in fault-tolerance.

Notwithstanding the number of interesting applications presented, a number of open problems need to be addressed and solved before that of ant algorithms becomes a mature field. For example, it would be interesting to give an answer to the following questions: How can we define "methodologies" to program ant algorithms? How do we define "artificial ants"? How complex should they be? Should they all be identical? What basic capabilities should they be given? Should they be able to learn? Should they be purely reactive? How local should their environment knowledge be? Should they be able to communicate directly? If yes, what type of information should they communicate?

What is also missing, similarly to what happens with many other adaptive systems, is a theory that allows to predict the system behavior as a function of its parameters and of the characteristics of the application domain. On this aspect, let us mention a couple of recent and intriguing results: Gutjahr has recently proved (see this special issue [40]) convergence to the optimal solution for a particular version of Ant System, while Wagner, Lindenbaum and Bruckstein (see this special issue [85]) have proved an upper bound to the time necessary to an ant-like agent to cover static and dynamic graphs.

6 Acknowledgements

We are grateful to Gianni Di Caro and to Thomas Stützle for critical reading of a draft version of this article.

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*, pages 1445–1450. Piscataway, NJ:IEEE Press, 1999.

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

[3] E. Bonabeau, F. Henaux, S. Guérin, D. Snyers, P. Kuntz, and G. Theraulaz. Routing in telecommunication networks with "Smart" ant-like agents. In *Proceedings of IATA'98, Second Int. Workshop on Intelligent Agents for Telecommunication Applications*. Lectures Notes in AI vol. 1437, Springer Verlag, 1998.

[4] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg. Adaptive task allocation inspired by a model of division of labour in social insects. In D. Lundh, B. Olsson, and A. Narayanan, editors, *Biocomputation and Emergent Computing*, pages 36–45. Singapore: World Scientific, 1997.

[5] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings Roy. Soc. London B*, 263:1565–1569, 1996.

[6] O. H. Bruinsma. *An Analysis of Building Behaviour of the Termite* Macrotemes Subhyalinus. PhD thesis, Lanbouwhogeschool te Wageningen, The Netherlands, 1979.

[7] B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved Ant System algorithm for the vehicle routing problem. Technical Report POM-10/97, Institute of Management Science, University of Vienna, Austria, 1997. Accepted for publication in the *Annals of Operations Research, 89, 1999*.

[8] B. Bullnheimer, R. F. Hartl, and C. Strauss. A new rank-based version of the Ant System: a computational study. Technical Report POM-03/97, Institute of Management Science, University of Vienna, Austria, 1997.

[9] 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. Boston, MA:Kluwer Academics, 1999.

[10] 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.

[11] L. Chrétien. *Organisation spatiale du matériel provenant de l'excavation du nid chez* Messor Barbarus *et des cadavres d'ouvrières chez* Lasius Niger. PhD thesis, Université Libre de Bruxelles, Brussels, 1996.

[12] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. Ant System for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)*, 34:39–53, 1994.

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

[14] M. den Besten, T. Stützle, and M. Dorigo. Scheduling single machines by ants. Technical Report IRIDIA/99-16, IRIDIA, Université Libre de Bruxelles, Belgium, 1999.

[15] J.-L. Deneubourg. Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. *Insectes Sociaux*, 24:117–130, 1977.

[16] 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.

[17] J.-L. Deneubourg and S. Goss. Collective patterns and decision making. *Ethol. Ecol. & Evol.*, 1:295–311, 1989.

[18] 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. Cambridge,MA:MIT Press/Bradford Books, 1991.

[19] J.-L. Deneubourg, S. Goss, J. M. Pasteels, D. Fresneau, and J.-P. Lachaud. Self-organization mechanisms in ant societies (II): Learning in foraging and division of labour. *Experientia Suppl.*, 54:177–196, 1987.

[20] 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.

[21] G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research (JAIR)*, 9:317–365, 1998.

[22] 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.

[23] G. Di Caro and M. Dorigo. Two ant colony algorithms for best-effort routing in datagram networks. In *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pages 541–546. IASTED/ACTA Press, 1998.

[24] M. Dorigo. *Optimization, Learning and Natural Algorithms* (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

[25] 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. London,UK: McGraw-Hill, 1999.

[26] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.

[27] M. Dorigo and L. M. Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*, 43:73–81, 1997.

[28] 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.

[29] 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.

[30] 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.

[31] J.-H. Fabre. *Souvenirs Entomologiques*. Paris:Librairie Delagrave, 1925.

[32] L. M. Gambardella and M. Dorigo. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In *Proceedings of the Twelfth International Conference on Machine Learning, ML-95*, pages 252–260. Palo Alto, California: Morgan Kaufmann, 1995.

[33] L. M. Gambardella and M. Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC96*, pages 622–627. IEEE Press, 1996.

[34] 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.

[35] L. M. Gambardella, È. 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. London,UK: McGraw-Hill, 1999.

[36] L. M. Gambardella, È. D. Taillard, and M. Dorigo. Ant colonies for the QAP. Technical Report IDSIA-4-97, IDSIA, Lugano, Switzerland, 1997.

[37] L. M. Gambardella, È. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society (JORS)*, 50:167–176, 1999.

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

[39] 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*, 6:41–81, 1959.

[40] W. J. Gutjahr. A graph-based Ant System and its convergence. *Future Generation Computer Systems*, 2000. This issue.

[41] M. Heusse, S. Guérin, D. Snyers P., and Kuntz. Adaptive agent-driven routing and load balancing in communication networks. *Advances in Complex Systems*, 1:234–257, 1998.

[42] B. Hölldobler and E. O. Wilson. *The Ants*. Berlin, Germany:Springer-Verlag, 1990.

[43] B. Hölldobler and E. O. Wilson. *Journey to the Ants: A Story of Scientific Exploration*. Cambridge, MA:Harvard University Press, 1994.

[44] M. J. B. Krieger and J.-B. Billeter. The call of duty: Self-organized task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, in press.

[45] P. Kuntz and P. Layzell. A new stochastic approach to find clusters in vertex set of large graphs with applications to partitioning in VLSI technology. Technical Report LIASC, Ecole Nationale Supérieure des Télécommunications de Bretagne, 1995.

[46] P. Kuntz, P. Layzell, and D. Snyers. A colony of ant-like agents for partitioning in VLSI technology. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 417–424. Cambridge,MA:MIT Press, 1997.

[47] P. Kuntz and D. Snyers. New results on an ant-based heuristic for highlighting the organization of large graphs. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1451–1458. Piscataway, NJ:IEEE Press, 1999.

[48] G. Leguizamón and Z. Michalewicz. A new version of Ant System for subset problems. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1459–1464. Piscataway, NJ:IEEE Press, 1999.

[49] 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. Piscataway, NJ:IEEE Press, 1999.

[50] S. Lin. Computer solutions for the traveling salesman problem. *Bell Systems Journal*, 44:2245–2269, 1965.

[51] 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. Cambridge,MA:MIT Press/Bradford Books, 1994.

[52] M. Maeterlinck. *The Life of the White Ant*. London:George Allen and Unwin, 1927.

[53] V. Maniezzo. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. Technical Report CSR 98-1, Scienze dell'Informazione, Universitá di Bologna, Sede di Cesena, Italy, 1998.

[54] V. Maniezzo and A. Carbonaro. An ANTS heuristic for the frequency assignment problem. Technical Report CSR 98-4, Scienze dell'Informazione, Universitá di Bologna, Sede di Cesena, Italy, 1998.

[55] V. Maniezzo and A. Carbonaro. An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, 2000. This issue.

[56] V. Maniezzo and A. Colorni. The Ant System applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 1999.

[57] 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.

[58] R. Michel and M. Middendorf. An island model based Ant System with lookahead for the shortest supersequence problem. In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature*, pages 692–701. Berlin, Germany:Springer-Verlag, 1998.

[59] 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. London,UK: McGraw-Hill, 1999.

[60] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Simulation on Experimental Robotics (ISER-93)*, pages 501–513, 1993.

[61] 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*, pages 1809–1816. Piscataway, NJ:IEEE Press, 1999.

[62] G. F. Oster. Modeling social insect populations. I. Ergonomics of foraging and population growth in bumblebees. *Am.Nat.*, 110:215–245, 1977.

[63] G. F. Oster and E. O. Wilson. *Caste and Ecology in the Social Insects*. Princeton, NJ: Princeton University Press, 1978.

[64] R. C. Plowright and C. M. S. Plowright. Elitism in social insects: A positive feedback model. In R. L. Jeanne, editor, *Interindividual Behavior Variability in Social Insects*. Boulder, CO: Westview Press, 1988.

[65] H. Ramalhinho Lourenço and D. Serra. Adaptive approach heuristics for the generalized assignment problem. Technical Report EWP Series No. 304, Universitat Pompeu Fabra, Dept. of Economics and Management, Barcelona, 1998.

[66] G. E. Robinson. Modulation of alarm pheromone perception in the honey bee: Evidence for division of labour based on hormonally regulated response thresholds. *Jour. Comp. Physiol. A*, 160:613–619, 1987.

[67] G. E. Robinson. Regulation of division of labor in insect societies. *Ann. Rev. Entomol.*, 37:637–665, 1992.

[68] 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.

[69] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2):169–207, 1996.

[70] T. D. Seeley. Adaptive significance of the age polyethism schedule in honey bee colonies. *Behav. Ecol. Sociobiol.*, 11:287–293, 1982.

[71] T. Stützle. An Ant Approach to the Flow Shop Problem. Technical Report AIDA–97–07, FG Intellektik, FB Informatik, TH Darmstadt, September 1997.

[72] T. Stützle. *Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications*. PhD thesis, Fachbereich Informatik, TU Darmstadt, Germany, 1998.

[73] T. Stützle and M. Dorigo. ACO Algorithms for the Traveling Salesman Problem. In P. Neittaanmäki, J. Periaux, K. Miettinen, and M. M. Mäkelä, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 163–183. Chichester,UK:John Wiley & Sons, 1999.

[74] T. Stützle and H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System for the quadratic assignment problem. Technical Report AIDA–97–4, FG Intellektik, TH Darmstadt, jul 1997.

[75] T. Stützle and 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 IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference*, pages 309–314. IEEE Press, 1997.

[76] T. Stützle and H. Hoos. Improvements on the Ant System: Introducing $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 245–249. Springer Verlag, Vienna, 1998.

[77] T. Stützle and H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant system and local search for combinatorial optimization problems. In S. Voß, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 313–329. Boston,MA:Kluwer Academics, 1999.

[78] T. Stützle and H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. *Future Generation Computer Systems*, 2000. This issue.

[79] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 832–838. Morgan Kaufmann, 1997.

[80] G. Theraulaz and E. Bonabeau. A brief history of stigmergy. *Artificial Life*, 5:97–116, 1999.

[81] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg. Threshold reinforcement and the regulation of division of labour in insect societies. *Proceedings Roy. Soc. London B*, 265:327–332, 1998.

[82] G. Theraulaz, S. Goss, J. Gervet, and J.-L. Deneubourg. Task differentiation in *Polistes* wasp colonies: A model for self-organizing groups of 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 346–355. Cambridge,MA:MIT Press/Bradford Books, 1991.

[83] R. van der Put. Routing in the faxfactory using mobile agents. Technical Report R&D-SV-98-276, KPN Research, The Netherlands, 1998.

[84] R. van der Put and L. Rothkrantz. Routing in packet switched networks using agents. *Simulation Practice and Theory*, 1999, in press.

[85] I. Wagner, Lindenbaum, and F. Bruckstein. ANTS: Agents, networks, trees, and subgraphs. *Future Generation Computer Systems*, 2000. This issue.

[86] B. Werber. *Les fourmis*. Editions Albin Michel, 1991. Translated in English as: *Empire of the Ants*, New York:Bantam Books, 1996.

[87] T. White, B. Pagurek, and F. Oppacher. Connection management using adaptive mobile agents. In H.R. Arabnia, editor, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98)*, pages 802–809. CSREA Press, 1998.

[88] E. O. Wilson. *Sociobiology*. Cambridge, MA: Harvard University Press, 1975.

[89] E. O. Wilson. The relation between caste ratios and division of labour in the ant *Genus Pheidole* (hymenoptera: Formicidae). *Behav. Ecol. Sociobiol.*, 16:89–98, 1984.



Marco Dorigo was born in Milan, Italy, in 1961. He received the PhD degree in information and systems electronic engineering in 1992 from Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur in Artificial Intelligence, from the Université Libre de Bruxelles, Belgium, in 1995. From 1992 to 1993 he was a research fellow at the International Computer Science Institute of Berkeley, CA. In 1993 he was a NATO-CNR fellow, and from 1994 to 1996 a Marie Curie fellow at the IRIDIA lab of the Université Libre de Bruxelles. Since 1996 he holds a tenured research position in the same lab as a Research Associate of the FNRS, the Belgian National Fund for Scientific Research. His main current research interest is in ant algorithms, a novel research area initiated by his seminal doctoral work. Other research interests include evolutionary computation, autonomous robotics, and reinforcement learning. He is the author of a book on learning robots and of a book on swarm intelligence, the editor of three books on

evolutionary computation and other modern heuristic techniques, and of more than fifty scientific articles published in international journals and conference proceedings. Dr. Dorigo is an Associate Editor for the IEEE Transactions on Systems, Man, and Cybernetics, the IEEE Transactions on Evolutionary Computation, the Journal of Heuristics, and the Journal of Cognitive Systems Research. He is a member of the Editorial Board of the Evolutionary Computation journal, the Adaptive Behavior journal, and the Journal of Genetic Programming and Evolvable Machines. He was awarded the 1996 Italian Prize for Artificial Intelligence.

Eric Bonabeau is the managing director of EuroBios, a Paris-based start-up company pplying complexity science to business problems. Prior to this appointment, he was Director of Research at France Telecom, R&D engineer with Cadence Design Systems, and Research Fellow at the Santa Fe Institute. He received his PhD in theoretical physics from the University of Orsay, France, and engineering degrees from Ecole Polytechnique and Ecole National Supèrieure des Tèlècommunications (France). The Editor-in-Chief of Advances in Complex Systems, Dr. Bonabeau is the co-author of three books and a hundred scientific articles.

Guy Theraulaz received an MS degree in behavioral neurosciences in 1986, and a PhD in ethology in 1991 from the Provence University, Marseille, France. He is a Research Associate with the French CNRS, Centre National de la Recherche Scientifique, and is currently working at the Ethology and Animal Cognition Laboratory, Paul Sabatier University in Toulouse, where he is the Head of the Collective Intelligence in Social Insects and Artificial Systems group. His research interests are collective behaviors in animal societies, modelling collective phenomena and designing distributed adaptive algorithms inspired by social insects. In 1996 he was awarded the bronze medal of the CNRS for his work on Swarm Intelligence.

Correspondence:

Dr. Marco Dorigo, IRIDIA, Université Libre de Bruxelles, CP 194/6, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium
Phone: +32 2 650 3169, Fax: +32 2 650 2715, Email: mdorigo@ulb.ac.be