1 From Real to Artificial Ants

I am lost! Where is the line?!
—*A Bug's Life*, Walt Disney, 1998

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 capabilities of a single ant.

The field of "ant algorithms" studies models derived from the observation of real ants' behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems.

The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems. Several different aspects of the behavior of ant colonies have inspired different kinds of ant algorithms. Examples are foraging, division of labor, brood sorting, and cooperative transport. In all these examples, ants coordinate their activities via *stigmergy*, a form of indirect communication mediated by modifications of the environment. For example, a foraging ant deposits a chemical on the ground which increases the probability that other ants will follow the same path. Biologists have shown that many colony-level behaviors observed in social insects can be explained via rather simple models in which only stigmergic communication is present. In other words, biologists have shown that it is often sufficient to consider stigmergic, indirect communication to explain how social insects can achieve self-organization. The idea behind ant algorithms is then to use a form of *artificial stigmergy* to coordinate societies of artificial agents.

One of the most successful examples of ant algorithms is known as "ant colony optimization," or ACO, and is the subject of this book. ACO is inspired by the foraging behavior of ant colonies, and targets discrete optimization problems. This introductory chapter describes how real ants have inspired the definition of artificial ants that can solve discrete optimization problems.

1.1 Ants' Foraging Behavior and Optimization

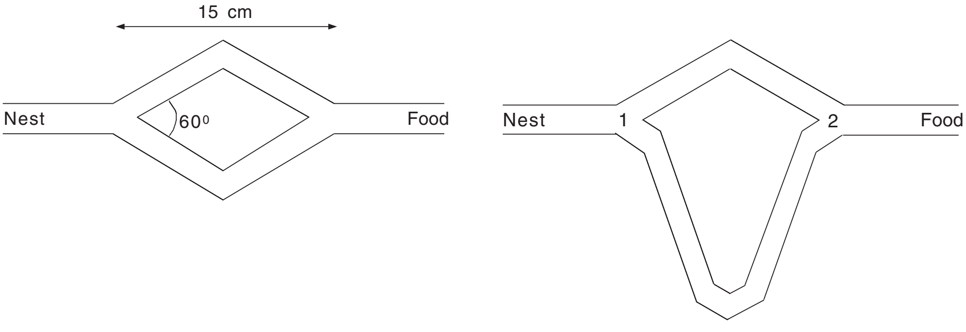
The visual perceptive faculty of many ant species is only rudimentarily developed and there are ant species that are completely blind. In fact, an important insight of early research on ants' behavior was that most of the communication among individuals, or between individuals and the environment, is based on the use of chemicals produced by the ants. These chemicals are called *pheromones*. This is different from, for example, what happens in humans and in other higher species, whose most important senses are visual or acoustic. Particularly important for the social life of some ant species is the *trail pheromone*. Trail pheromone is a specific type of pheromone that some ant species, such as *Lasius niger* or the Argentine ant *Iridomyrmex humilis* (Goss, Aron, Deneubourg, & Pasteels, 1989), use for marking paths on the ground, for example, paths from food sources to the nest. By sensing pheromone trails foragers can follow the path to food discovered by other ants. This collective trail-laying and trail-following behavior whereby an ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

1.1.1 Double Bridge Experiments

The foraging behavior of many ant species, as, for example, *I. humilis* (Goss et al., 1989), *Linepithema humile*, and *Lasius niger* (Bonabeau et al., 1997), is based on indirect communication mediated by pheromones. While walking from food sources to the nest and vice versa, ants deposit pheromones on the ground, forming in this way a pheromone trail. Ants can smell the pheromone and they tend to choose, probabilistically, paths marked by strong pheromone concentrations.

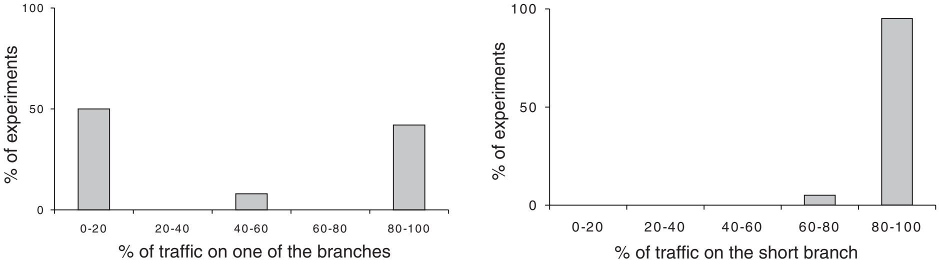
The pheromone trail-laying and -following behavior of some ant species has been investigated in controlled experiments by several researchers. One particularly brilliant experiment was designed and run by Deneubourg and colleagues (Deneubourg, Aron, Goss, & Pasteels, 1990; Goss et al., 1989), who used a double bridge connecting a nest of ants of the Argentine ant species *I. humilis* and a food source. They ran experiments varying the ratio $r = l_l/l_s$ between the length of the two branches of the double bridge, where l_l was the length of the longer branch and l_s the length of the shorter one.

In the first experiment the bridge had two branches of equal length ($r = 1$; see figure 1.1a). At the start, ants were left free to move between the nest and the food source and the percentage of ants that chose one or the other of the two branches were observed over time. The outcome was that (see also figure 1.2a), although in the initial phase random choices occurred, eventually all the ants used the same branch. This result can be explained as follows. When a trial starts there is no pheromone on the two branches. Hence, the ants do not have a preference and they select with the same probability any of the branches. Yet, because of random fluctuations, a few more ants will select one branch over the other. Because ants deposit pheromone while walking, a larger number of ants on a branch results in a larger amount of pheromone on that branch; this larger amount of pheromone in turn stimulates more ants to choose that branch again, and so on until finally the ants converge to one



(a) (b)

Figure 1.1
Experimental setup for the double bridge experiment. (a) Branches have equal length. (b) Branches have different length. Modified from Goss et al. (1989).



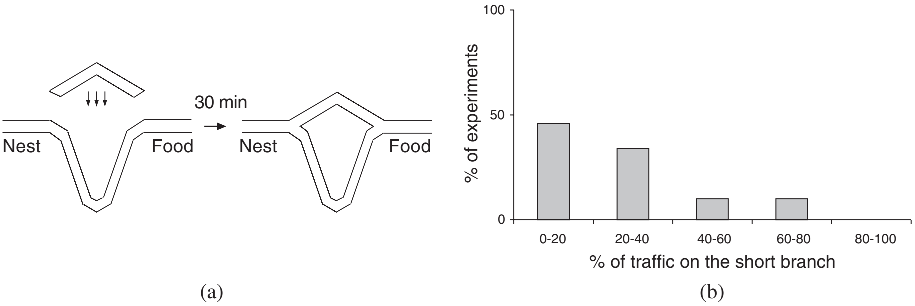
(a) (b)

Figure 1.2
Results obtained with *Iridomyrmex humilis* ants in the double bridge experiment. (a) Results for the case in which the two branches have the same length ($r = 1$); in this case the ants use one branch or the other in approximately the same number of trials. (b) Results for the case in which one branch is twice as long as the other ($r = 2$); here in all the trials the great majority of ants chose the short branch. Modified from Goss et al. (1989).

single path. This *autocatalytic* or *positive feedback* process is, in fact, an example of a self-organizing behavior of the ants: a macroscopic pattern (corresponding to the convergence toward one branch) emerges out of processes and interactions taking place at a "microscopic" level (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001; Haken, 1983; Nicolis & Prigogine, 1977). In our case the convergence of the ants' paths to one branch represents the macroscopic collective behavior, which can be explained by the microscopic activity of the ants, that is, by the local interactions among the individuals of the colony. It is also an example of stigmergic communication (for a definition of *stigmergy*, see section 1.4): ants coordinate their activities, exploiting indirect communication mediated by modifications of the environment in which they move.

In the second experiment, the length ratio between the two branches was set to $r = 2$ (Goss et al., 1989), so that the long branch was twice as long as the short one (figure 1.1b shows the experimental setup). In this case, in most of the trials, after some time all the ants chose to use only the short branch (see figure 1.2b). As in the first experiment, ants leave the nest to explore the environment and arrive at a decision point where they have to choose one of the two branches. Because the two branches initially appear identical to the ants, they choose randomly. Therefore, it can be expected that, on average, half of the ants choose the short branch and the other half the long branch, although stochastic oscillations may occasionally favor one branch over the other. However, this experimental setup presents a remarkable difference with respect to the previous one: because one branch is shorter than the other (see figure 1.1b), the ants choosing the short branch are the first to reach the food and to start their return to the nest. But then, when they must make a decision between the short and the long branch, the higher level of pheromone on the short branch will bias their decision in its favor. Therefore, pheromone starts to accumulate faster on the short branch, which will eventually be used by all the ants because of the autocatalytic process described previously. When compared to the experiment with the two branches of equal length, the influence of initial random fluctuations is much reduced, and stigmergy, autocatalysis, and *differential path length* are the main mechanisms at work. Interestingly, it can be observed that, even when the long branch is twice as long as the short one, not all the ants use the short branch, but a small percentage may take the longer one. This may be interpreted as a type of "path exploration."

It is also interesting to see what happens when the ant colony is offered, after convergence, a new shorter connection between the nest and the food. This situation was studied in an additional experiment in which initially only the long branch was



(a) (b)

Figure 1.3
In this experiment initially only the long branch was offered to the colony. After 30 minutes, when a stable pheromone trail has formed on the only available branch, a new shorter branch is added. (a) The initial experimental setup and the new situation after 30 minutes, when the short branch was added. (b) In the great majority of the experiments, once the short branch is added the ants continue to use the long branch.

offered to the colony and after 30 minutes the short branch was added (see figure 1.3). In this case, the short branch was only selected sporadically and the colony was trapped on the long branch. This can be explained by the high pheromone concentration on the long branch and by the slow evaporation of pheromone. In fact, the great majority of ants choose the long branch because of its high pheromone concentration, and this autocatalytic behavior continues to reinforce the long branch, even if a shorter one appears. Pheromone evaporation, which could favor exploration of new paths, is too slow: the lifetime of the pheromone is comparable to the duration of a trial (Goss et al., 1989), which means that the pheromone evaporates too slowly to allow the ant colony to "forget" the suboptimal path to which they converged so that the new and shorter one can be discovered and "learned."

1.1.2 A Stochastic Model

Deneubourg and colleagues (Deneubourg et al., 1990; Goss et al., 1989) proposed a simple stochastic model that adequately describes the dynamics of the ant colony as observed in the double bridge experiment. In this model, ψ ants per second cross the bridge in each direction at a constant speed of v cm/s, depositing one unit of pheromone on the branch. Given the lengths l_s and l_l (in cm) of the short and of the long branch, an ant choosing the short branch will traverse it in $t_s = l_s/v$ seconds, while an ant choosing the long branch will use $r \cdot t_s$ seconds, where $r = l_l/l_s$.

The probability $p_{ia}(t)$ that an ant arriving at decision point $i \in \{1,2\}$ (see figure 1.1b) selects branch $a \in \{s,l\}$, where s and l denote the short and long branch respectively, at instant t is set to be a function of the total amount of pheromone $\varphi_{ia}(t)$

on the branch, which is proportional to the number of ants that used the branch until time t. For example, the probability $p_{is}(t)$ of choosing the short branch is given by

$$p_{is}(t) = \frac{(t_s + \varphi_{is}(t))^{\alpha}}{(t_s + \varphi_{is}(t))^{\alpha} + (t_s + \varphi_{il}(t))^{\alpha}}, \tag{1.1}$$

where the functional form of equation (1.1), as well as the value $\alpha = 2$, was derived from experiments on trail-following (Deneubourg et al., 1990); $p_{il}(t)$ is computed similarly, with $p_{is}(t) + p_{il}(t) = 1$.

This model assumes that the amount of pheromone on a branch is proportional to the number of ants that used the branch in the past. In other words, no pheromone evaporation is considered by the model (this is in accordance with the experimental observation that the time necessary for the ants to converge to the shortest path has the same order of magnitude as the mean lifetime of the pheromone (Goss et al., 1989; Beckers, Deneubourg, & Goss, 1993)). The differential equations that describe the evolution of the stochastic system are

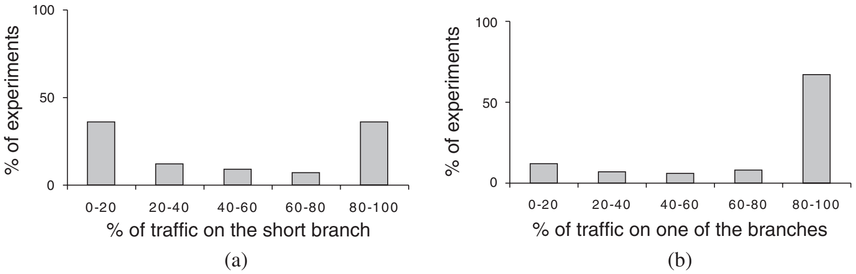
$$\mathrm{d}\varphi_{is}/\mathrm{d}t = \psi p_{js}(t - t_s) + \psi p_{is}(t), \qquad (i = 1, j = 2; i = 2, j = 1), \tag{1.2}$$

$$\mathrm{d}\varphi_{il}/\mathrm{d}t = \psi p_{jl}(t - r \cdot t_s) + \psi p_{il}(t), \qquad (i = 1, j = 2; i = 2, j = 1). \tag{1.3}$$

Equation (1.2) can be read as follows: the instantaneous variation, at time t, of pheromone on branch s and at decision point i is given by the ants' flow ψ, assumed constant, multiplied by the probability of choosing the short branch at decision point j at time $t - t_s$ plus the ants' flow multiplied by the probability of choosing the short branch at decision point i at time t. The constant t_s represents a time delay, that is, the time necessary for the ants to traverse the short branch. Equation (1.3) expresses the same for the long branch, except that in this case the time delay is given by $r \cdot t_s$.

The dynamic system defined by these equations was simulated using the Monte Carlo method (Liu, 2001). In figure 1.4, we show the results of two experiments consisting of 1000 simulations each and in which the branch length ratio was set to $r = 1$ and to $r = 2$. It can be observed that when the two branches have the same length ($r = 1$) the ants converge toward the use of one or the other of the branches with equal probability over the 1000 simulations. Conversely, when one branch is twice as long as the other ($r = 2$), then in the great majority of experiments most of the ants choose the short branch (Goss et al., 1989).

In this model the ants deposit pheromone both on their forward and their backward paths. It turns out that this is a necessary behavior to obtain convergence of the ant colony toward the shortest branch. In fact, if we consider a model in which ants deposit pheromone only during the forward or only during the backward trip, then



(a) (b)

Figure 1.4
Results of 1000 Monte Carlo simulations of the model given by equations (1.1), (1.2), and (1.3), with $\psi = 0.5$ ant per second. Ants were counted between the 501st and 1000th ant crossing the bridge. (a) The ratio between the long and the short branch was set to $r = 1$. (b) The ratio between the long and the short branch was set to $r = 2$. Modified from Goss et al. (1989).

the result is that the ant colony is unable to choose the shortest branch. The observation of real ant colonies has confirmed that ants that deposit pheromone only when returning to the nest are unable to find the shortest path between their nest and the food source (Deneubourg, 2002).

1.2 Toward Artificial Ants

The double bridge experiments show clearly that ant colonies have a built-in optimization capability: by the use of probabilistic rules based on local information they can find the shortest path between two points in their environment. Interestingly, by taking inspiration from the double bridge experiments, it is possible to design artificial ants that, by moving on a graph modeling the double bridge, find the shortest path between the two nodes corresponding to the nest and to the food source.

As a first step toward the definition of artificial ants, consider the graph of figure 1.5a, which is a model of the experimental setup shown in figure 1.1b. The graph consists of two nodes (1 and 2, representing the nest and the food respectively) that are connected by a short and a long arc (in the example the long arc is r times longer than the short arc, where r is an integer number). Additionally, we assume the time to be discrete $(t = 1, 2, \ldots)$ and that at each time step each ant moves toward a neighbor node at constant speed of one unit of length per time unit. By doing so, ants add one unit of pheromone to the arcs they use. Ants move on the graph by choosing the path probabilistically: $p_{is}(t)$ is the probability for an ant located in node i at time t to choose the short path, and $p_{il}(t)$ the probability to choose the long path. These probabilities are a function of the pheromone trails φ_{ia} that ants in node i $(i \in \{1, 2\})$

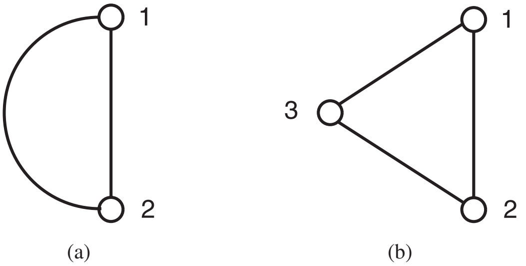


Figure 1.5
The graphs are two equivalent models of the experimental setup shown in figure 1.1b. In both cases, ants move from the nest to the food and back either via a short or via a long branch. (a) In this model the long branch is *r* times longer than the shorter one. An ant entering the long branch updates the pheromone on it *r* time units later. (b) In this model, each arc of the graph has the same length, and a longer branch is represented by a sequence of arcs. Here, for example, the long branch is twice as long as the short branch. Pheromone updates are done with one time unit delay on each arc.

encounter on the branch a, ($a \in \{s, l\}$):

$$p_{is}(t) = \frac{[\varphi_{is}(t)]^{\alpha}}{[\varphi_{is}(t)]^{\alpha} + [\varphi_{il}(t)]^{\alpha}}, \qquad p_{il}(t) = \frac{[\varphi_{il}(t)]^{\alpha}}{[\varphi_{is}(t)]^{\alpha} + [\varphi_{il}(t)]^{\alpha}}. \tag{1.4}$$

Trail update on the two branches is performed as follows:

$$\varphi_{is}(t) = \varphi_{is}(t-1) + p_{is}(t-1)m_i(t-1) + p_{js}(t-1)m_j(t-1),$$
$$(i = 1, j = 2; i = 2, j = 1), \tag{1.5}$$

$$\varphi_{il}(t) = \varphi_{il}(t-1) + p_{il}(t-1)m_i(t-1) + p_{jl}(t-r)m_j(t-r),$$
$$(i = 1, j = 2; i = 2, j = 1), \tag{1.6}$$

where $m_i(t)$, the number of ants on node i at time t, is given by

$$m_i(t) = p_{js}(t-1)m_j(t-1) + p_{jl}(t-r)m_j(t-r),$$
$$(i = 1, j = 2; i = 2, j = 1). \tag{1.7}$$

This model differs from the one presented in section 1.1.2 in two important aspects:

- It considers the average behavior of the system, and not the stochastic behavior of the single ants.
- It is a discrete time model, whereas the previous one was a continuous time model; accordingly, it uses difference instead of differential equations.

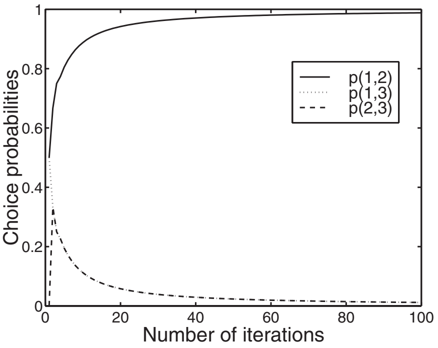


Figure 1.6
Result of the simulation of the model described by equations (1.4) through (1.7). The figure shows the probability of choosing the three branches of the graph in figure 1.5b. After a short transitory period the probabilities of choosing the long branch (($p(1,3) \equiv p_{1l}$ and ($p(2,3) \equiv p_{2l}$) become vanishingly small (in the graph they are superimposed after a few iterations from the start), while the probability of choosing the short branch ($p(1,2) \equiv p_{1s} \equiv p_{2s}$) tends to 1. Note that probabilities are symmetric: $p(i,j) = p(j,i)$. Parameter settings: $\alpha = 2$, $r = 2$, $t = 100$.

Another way of modeling the experimental apparatus of figure 1.1b with a graph is shown in figure 1.5b. In this model each arc of the graph has the same length, and a longer branch is represented by a sequence of arcs. In the figure, for example, the long branch is twice as long as the short branch. Pheromone updates are done with one time unit delay on each arc. The two models are equivalent from a computational point of view, yet the second model permits an easier algorithmic implementation when considering graphs with many nodes.

Simulations run with this discrete time model give results very similar to those obtained with the continuous time model of equations (1.1) to (1.3). For example, by setting the number of ants to twenty, the branch length ratio to $r = 2$, and the parameter α to 2, the system converges rather rapidly toward the use of the short branch (see figure 1.6).

1.3 Artificial Ants and Minimum Cost Paths

In the previous section we have shown that a set of difference equations can reproduce rather accurately the mean behavior of the continuous model of Deneubourg et al. Yet, our goal is to define algorithms that can be used to solve minimum cost problems on more complicated graphs than those representing the double bridge experiment (see, e.g., the graph in figure 1.7).

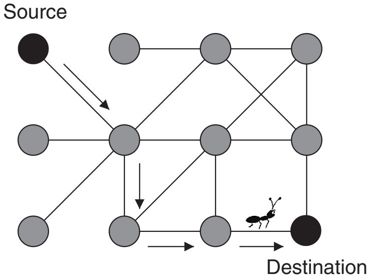


Figure 1.7
Ants build solutions, that is, paths from a source to a destination node.

With this goal in mind, let us consider a static, connected graph $G = (N, A)$, where N is the set of $n = |N|$ nodes and A is the set of undirected arcs connecting them. The two points between which we want to establish a minimum cost path are called source and destination nodes, as typically done in the literature on minimum cost path problems (when the cost of arcs is given by their length, the minimum cost path problem is the same as the shortest-path problem); sometimes, in analogy to the shortest-path–finding behavior of real ants, we will also call them *nest* and *food source*.

Unfortunately, if we try to solve the minimum cost path problem on the graph G using artificial ants whose behavior is a straightforward extension of the behavior of the ants described in the previous section, the following problem arises: the ants, while building a solution, may generate loops. As a consequence of the forward pheromone trail updating mechanism, loops tend to become more and more attractive and ants can get trapped in them. But even if an ant can escape such loops, the overall pheromone trail distribution becomes such that short paths are no longer favored and the mechanism that in the simpler double bridge situation made the ant choose the shortest path with higher probability does not work anymore. Because this problem is due to forward pheromone trail updating, it might seem that the simplest solution to this problem would be the removal of the forward updating mechanism: in this way ants would rely only on backward updating. Still, this is not a solution: as was said before (see section 1.1.2, but see also exercise 1.1 at the end of this chapter), if the forward update is removed the system does not work anymore, not even in the simple case of the double bridge experiment.

We therefore need to extend the capabilities of the artificial ants in a way that, while retaining the most important characteristics of real ants, allows them to solve minimum cost path problems on generic graphs. In particular, artificial ants are given a limited form of memory in which they can store the partial paths they have followed so far, as well as the cost of the links they have traversed. Via the use of memory, the ants can implement a number of useful behaviors that allow them to efficiently build solutions to the minimum cost path problem. These behaviors are (1) probabilistic solution construction biased by pheromone trails, without forward pheromone updating; (2) deterministic backward path with loop elimination and with pheromone updating; and (3) evaluation of the quality of the solutions generated and use of the solution quality in determining the quantity of pheromone to deposit (note that while in the simple case of minimum cost path search an estimate of the solution quality can be made by the ant also during the solution construction, this is not necessarily true in other problems, in which there may not exist an easy way to evaluate partial solutions).

Additionally, we show that by taking into account pheromone evaporation, which was not necessary to explain real ants' behavior, performance can be greatly improved.

In the following we briefly explain how the above-mentioned ants' behavior, as well as pheromone evaporation, is implemented in an algorithm that we call Simple-ACO (S-ACO for short). It should be noted that, although it represents a significant step toward the definition of an efficient algorithm for the solution of minimum cost problems on graphs, S-ACO should be taken for what it is: a didactic tool to explain the basic mechanisms underlying ACO algorithms.

Probabilistic forward ants and solution construction. S-ACO ants can be thought of as having two working modes: *forward* and *backward*. They are in forward mode when they are moving from the nest toward the food, and they are in backward mode when they are moving from the food back to their nest. Once an ant in forward mode reaches its destination, it switches to backward mode and starts its travel back to the source. In S-ACO, forward ants build a solution by choosing probabilistically the next node to move to among those in the neighborhood of the graph node on which they are located. (Given a graph $G = (N, A)$, two nodes $i, j \in N$ are neighbors if there exists an arc $(i, j) \in A$.) The probabilistic choice is biased by pheromone trails previously deposited on the graph by other ants. Forward ants do not deposit any pheromone while moving. This, together with deterministic backward moves, helps in avoiding the formation of loops.

Deterministic backward ants and pheromone update. The use of an explicit memory allows an ant to retrace the path it has followed while searching the destination node. Moreover, S-ACO ants improve the system performance by implementing loop elimination. In practice, before starting to move backward on the path they memorized while searching the destination node (i.e., the forward path), S-ACO ants eliminate any loops from it. While moving backward, S-ACO ants leave pheromone on the arcs they traverse.

Pheromone updates based on solution quality. In S-ACO the ants memorize the nodes they visited during the forward path, as well as the cost of the arcs traversed if the graph is weighted. They can therefore evaluate the cost of the solutions they generate and use this evaluation to modulate the amount of pheromone they deposit while in backward mode. Making pheromone update a function of the generated solution quality can help in directing future ants more strongly toward better solutions. In fact, by letting ants deposit a higher amount of pheromone on short paths, the ants' path searching is more quickly biased toward the best solutions. Interestingly, the dependence of the amount of pheromone trail deposit on the solution quality is also present in some ant species: Beckers et al. (1993) found that in the ant species *Lasius niger* the ants returning from rich food sources tend to drop more pheromone than those returning from poorer food sources.

Pheromone evaporation. In real ant colonies, pheromone intensity decreases over time because of evaporation. In S-ACO evaporation is simulated by applying an appropriately defined pheromone evaporation rule. For example, artificial pheromone decay can be set to a constant rate. Pheromone evaporation reduces the influence of the pheromones deposited in the early stages of the search, when artificial ants can build poor-quality solutions. Although in the experiments run by Deneubourg and colleagues (Deneubourg et al., 1990; Goss et al., 1989) pheromone evaporation did not play any noticeable role, it can be very useful for artificial ant colonies, as we will show in the following sections.

1.3.1 S-ACO

We now present the details of the S-ACO algorithm which adapts the real ants' behavior to the solution of minimum cost path problems on graphs. To each arc (i, j) of the graph $G = (N, A)$ we associate a variable τ_{ij} called *artificial pheromone trail*, shortened to pheromone trail in the following. Pheromone trails are read and written by the ants. The amount (intensity) of a pheromone trail is proportional to the utility, as estimated by the ants, of using that arc to build good solutions.

Ants' Path-Searching Behavior

Each ant builds, starting from the source node, a solution to the problem by applying a step-by-step decision policy. At each node, local information stored on the node itself or on its outgoing arcs is read (sensed) by the ant and used in a stochastic way to decide which node to move to next. At the beginning of the search process, a constant amount of pheromone (e.g., $\tau_{ij} = 1, \forall(i,j) \in A$) is assigned to all the arcs. When located at a node i an ant k uses the pheromone trails τ_{ij} to compute the probability of choosing j as next node:

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^\alpha}{\sum_{l \in \mathcal{N}_i^k} \tau_{il}^\alpha}, & \text{if } j \in \mathcal{N}_i^k; \\ 0, & \text{if } j \notin \mathcal{N}_i^k; \end{cases} \tag{1.8}$$

where $\mathcal{N}_i^k$ is the neighborhood of ant k when in node i. In S-ACO the neighborhood of a node i contains all the nodes directly connected to node i in the graph $G = (N, A)$, except for the predecessor of node i (i.e., the last node the ant visited before moving to i). In this way the ants avoid returning to the same node they visited immediately before node i. Only in case $\mathcal{N}_i^k$ is empty, which corresponds to a dead end in the graph, node i's predecessor is included into $\mathcal{N}_i^k$. Note that this decision policy can easily lead to loops in the generated paths (recall the graph of figure 1.7).

An ant repeatedly hops from node to node using this decision policy until it eventually reaches the destination node. Due to differences among the ants' paths, the time step at which ants reach the destination node may differ from ant to ant (ants traveling on shorter paths will reach their destinations faster).

Path Retracing and Pheromone Update

When reaching the destination node, the ant switches from the forward mode to the backward mode and then retraces step by step the same path backward to the source node. An additional feature is that, before starting the return trip, an ant eliminates the loops it has built while searching for its destination node. The problem of loops is that they may receive pheromone several times when an ant retraces its path backward to deposit pheromone trail, leading to the problem of self-reinforcing loops. Loop elimination can be done by iteratively scanning the node identifiers position by position starting from the source node: for the node at the i-th position, the path is scanned starting from the destination node until the first occurrence of the node is encountered, say, at position j (it always holds that $i \leq j$ because the scanning process stops at position i at the latest). If we have $j > i$, the subpath from position $i+1$ to position j corresponds to a loop and can be eliminated. The scanning process is visualized in figure 1.8. The example also shows that our loop elimination procedure does not necessarily eliminate the largest loop. In the example, the loop 3-4-5-3 of length 3 is eliminated. Yet, the longest loop in this example, the loop 5-3-2-8-5 of length 4, is not eliminated because it is no longer present after eliminating the first loop. In general, if the path contains nested loops, the final loop-free path will

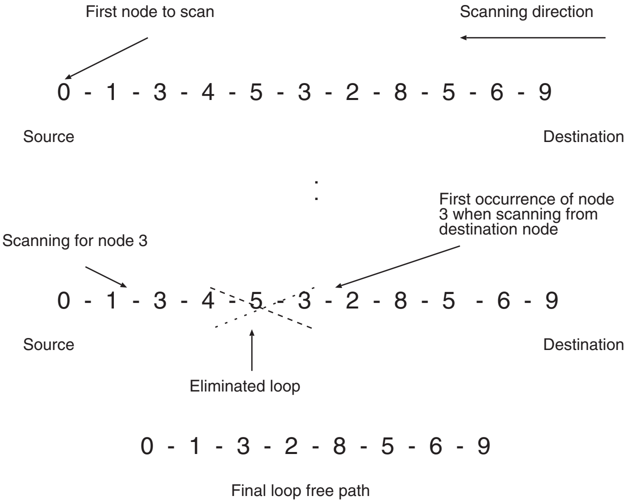


Figure 1.8
Illustration of the scanning process for loop elimination.

depend on the sequence in which the loops are eliminated. In S-ACO, loop elimination is implemented so that loops are eliminated in the same order as they are created.

During its return travel to the source the k-th ant deposits an amount $\Delta\tau^k$ of pheromone on arcs it has visited. In particular, if ant k is in the backward mode and it traverses the arc (i, j), it changes the pheromone value τ_{ij} as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k. \tag{1.9}$$

By this rule an ant using the arc connecting node i to node j increases the probability that forthcoming ants will use the same arc in the future.

An important aspect is the choice of $\Delta\tau^k$. In the simplest case, this can be the same constant value for all the ants. In this case, only the *differential path length* works in favor of the detection of short paths: ants which have detected a shorter path can deposit pheromone earlier than ants traveling on a longer path. In addition to the deterministic backward pheromone trail update, the ants may also deposit an amount of pheromone trail which is a function of the path length—the shorter the path the more pheromone is deposited by an ant. Generally, we require the amount of pheromone deposited by an ant to be a nonincreasing function of the path length.

Pheromone Trail Evaporation

Pheromone trail evaporation can be seen as an exploration mechanism that avoids quick convergence of all the ants toward a suboptimal path. In fact, the decrease in pheromone intensity favors the exploration of different paths during the whole search process. In real ant colonies, pheromone trails also evaporate, but, as we have seen, evaporation does not play an important role in real ants' shortest-path finding. The fact that, on the contrary, pheromone evaporation seems to be important in artificial ants is probably due to the fact that the optimization problems tackled by artificial ants are much more complex than those real ants can solve. A mechanism like evaporation that, by favoring the forgetting of errors or of poor choices done in the past, allows a continuous improvement of the "learned" problem structure seems therefore to be necessary for artificial ants. Additionally, artificial pheromone evaporation also plays the important function of bounding the maximum value achievable by pheromone trails.

Evaporation decreases the pheromone trails with exponential speed. In S-ACO, the pheromone evaporation is interleaved with the pheromone deposit of the ants. After each ant k has moved to a next node according to the ants' search behavior described earlier, pheromone trails are evaporated by applying the following equation to all the arcs:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in A, \tag{1.10}$$

where $\rho \in (0,1]$ is a parameter. After pheromone evaporation has been applied to all arcs, the amount of pheromone $\Delta\tau^k$ is added to the arcs. We call an iteration of S-ACO a complete cycle involving ants' movement, pheromone evaporation, and pheromone deposit.

1.3.2 Experiments with S-ACO

We have run experiments to evaluate the importance of some aspects of S-ACO: evaporation, number of ants, and type of pheromone update (function of the solution quality or not).

In the experiments presented in the following the behavior of S-ACO is judged with respect to convergence toward the minimum cost (shortest) path, in a way similar to what was done for the outcome of the simulation experiments of Deneubourg et al. and for the experiments with the discrete model introduced in section 1.2. Informally, by convergence we mean that, as the algorithm runs for an increasing number of iterations, the ants' probability of following the arcs of a particular path increases—in the limit to a point where the probability of selecting the arcs of this path becomes arbitrarily close to 1 while for all the others, it becomes arbitrarily close to 0.

The experiments have been run using two simple graphs: the double bridge of figure 1.5b and the more complex graph called *extended double bridge* given in figure

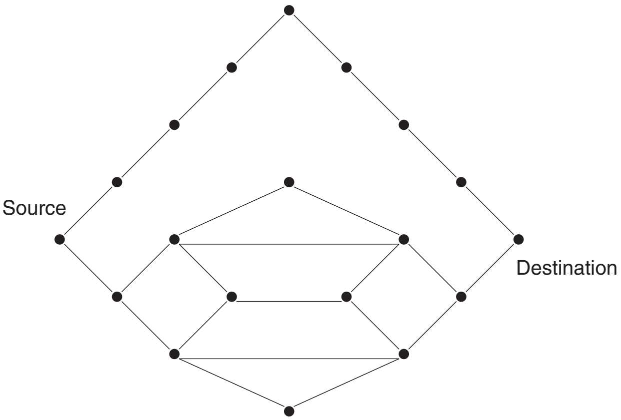


Figure 1.9
Extended double bridge. An ant starting in the source node can choose between the upper and the lower parts of the graph. The upper part consists of a single path of length 8 leading directly to the destination node. Differently, the lower part of the graph consists of a set of paths (which includes many paths shorter than eight steps) and the ant has many decisions to do before reaching the destination node. Therefore, ants choosing the upper part will always find a path of length 8, while ants choosing the lower part may find paths shorter than 8, but they may also enter loops and generate very long paths.

1.9. This second graph is designed in such a way that converging to the minimum cost path is not a trivial task for S-ACO. The difficulty of the graph is given by the fact that, in order to find the minimum cost path, an ant has to make a number of "correct" choices and if some of these choices are wrong, the ant generates suboptimal paths. To understand why, consider the graph of figure 1.9: ants exiting the source node have to choose between the loop-free, but worse than optimal, upper path of the graph, and the set of paths in the lower part of the same graph that contains two optimal paths of length 5, as well as many longer loop-free paths and infinitely many, much longer "loopy" paths. There is a trade-off between converging toward the use of an "easy" but suboptimal path, and searching for the optimal path in a region of the search space where suboptimal paths can easily be generated. In other words, to obtain convergence to the optimal solutions the ants need to choose the lower part of the graph, but then the greater number of decisions to be taken makes converging to the minimum cost path a difficult task.

Note that the choice of judging the algorithm using convergence as defined above instead of using more standard performance indices, such as the time or the number of iterations necessary to find the optimal solution, is consistent with our goals, that is, studying and understanding the relationship between design choices and the algorithm's behavior. In fact, such a study requires working on simple graphs like those discussed above so that simulation times remain reasonably short and the behavior of ants can be easily observed. But in simple graphs the shortest path is always found very quickly because of the large number of ants compared to the relatively small search space. Therefore, a performance index based on the time (or number of iterations) necessary to find the optimal solution would not be very meaningful. In fact, convergence as defined above, by requiring that all the ants do use the same path, is a more reasonable index for our purposes.

On the contrary, as we will see in the forthcoming chapters, when attacking more complex problems like $\mathcal{NP}$-hard optimization problems or routing in dynamic networks, the way experimental results are judged is different. In $\mathcal{NP}$-hard optimization problems the main goal is to find quickly very high-quality solutions and therefore we are interested mainly in the solution quality of the best solution(s) found by the ACO algorithm. In dynamic networks routing the algorithm has to be able to react rapidly to changing conditions and to maintain exploration capabilities so that it can effectively evaluate alternative paths which, due to the dynamics of the problem, may become more desirable; in both cases we will need a different definition of algorithm convergence.

Number of Ants and Types of Pheromone Update: Experiments with the Double Bridge
We ran a first set of experiments in which we studied the influence that the number of ants used and the way the amount of pheromone to be deposited is determined by ants have on the behavior of S-ACO. The experiments were run using the double bridge (see figure 1.5b). The choice of the double bridge was due to the desire of comparing the results obtained with S-ACO to those obtained with the model of real ants' behavior described in section 1.2. Note that a major difference between that model and S-ACO is that equations (1.4) through (1.7) describe the average behavior of the system, whereas in S-ACO a fixed number of ants move autonomously on the graph. Intuitively, an increasing number of ants in S-ACO should approximate better and better the average behavior given by equations (1.4) through (1.7).

In the following we report the results of two experiments:

1. Run S-ACO with different values for the number m of ants and with ants depositing a constant amount of pheromone on the visited arcs [i.e., $\Delta\tau^k = constant$ in equation (1.9)].
2. Same as in 1. above, except that the ants deposit an amount of pheromone which is inversely proportional to the length of the path they have found (i.e., $\Delta\tau^k = 1/L^k$, where L^k is the length of ant k's path).

Table 1.1
Percentage of trials in which S-ACO converged to the long path (100 independent trials for varying values of m, with $\alpha = 2$ and $\rho = 0$)

m	1	2	4	8	16	32	64	128	256	512
`without path length`	50	42	26	29	24	18	3	2	1	0
`with path length`	18	14	8	0	0	0	0	0	0	0

Column headings give the number m of ants in the colony. The first row shows results obtained performing pheromone updates without considering path length; the second row reports results obtained performing pheromone updates proportional to path length.

For each experiment we ran 100 trials and each trial was stopped after each ant had moved 1000 steps. Evaporation [see equation (1.10)] was set to $\rho = 0$, and the parameter α [see equation (1.8)] was set to 2, as in equation (1.1) of Deneubourg et al. approximating real ants' behavior. At the end of the trial we checked whether the pheromone trail was higher on the short or on the long path. In table 1.1, which gives the results of the two experiments, we report the percentage of trials in which the pheromone trail was higher on the long path. We found that, for the given parameter settings, S-ACO showed convergence behavior after 1000 ant steps so that the reported percentage is significant for understanding the algorithm behavior.

Let us focus first on the results of experiment 1. For a small number of ants (up to 32), S-ACO converged relatively often to the longer path. This is certainly due to fluctuations in the path choice in the initial iterations of the algorithm which can lead to a strong reinforcement of the long path. Yet, with an increasing number of ants, the number of times we observed this behavior decreased drastically, and for a large number of ants (here 512) we never observed convergence to the long path in any of the 100 trials. The experiments also indicate that, as could be expected, S-ACO performs poorly when only one ant is used: the number of ants has to be significantly larger than one to obtain convergence to the short path.

The results obtained in experiment 2 with pheromone updates based on solution quality are much better. As can be observed in table 1.1, S-ACO converged to the long path far less frequently than when pheromone updates were independent of the solution quality. With only one ant, S-ACO converged to the long path in only 18 out of 100 trials, which is significantly less than in experiment 1, and with eight ants or more it always converged to the short path.

In additional experiments, we examined the influence of the parameter α of equation (1.8) on the convergence behavior of S-ACO, in particular investigating the cases where α was changed in step sizes of 0.25 from 1 to 2. Again, the behavior was dependent on whether pheromone updates based on solution quality were used or not. In the first case we found that increasing α had a negative effect on the convergence behavior, while in the second case the results were rather independent of the particular value of α. In general, we found that, for a fixed number of ants, the algorithm tended to converge to the shortest path more often when α was close to 1. This is intuitively clear because large values of α tend to amplify the influence of initial random fluctuations. If, by chance, the long path is initially selected by the majority of ants, then the search of the whole colony is quickly biased toward it. This happens to a lower extent when the value of α is close to 1.

These results show that, as in the case of real ants, in S-ACO both *autocatalysis* and *differential path length* are at work to favor the emergence of short paths. While the results with S-ACO indicate that differential path length alone can be enough to let S-ACO converge to the optimal solution on small graphs, they also show that relying on this effect as the main driving force of the algorithm comes at the price of having to use large colony sizes, which results in long simulation times. In addition, the effectiveness of the differential path length effect strongly decreases with increasing problem complexity. This is what is shown by the experiments reported in the next subsection.

Pheromone Evaporation: Experiments with the Extended Double Bridge

In a second set of experiments, we studied the influence that pheromone trail evaporation has on the convergence behavior of S-ACO. Experiments were run using the extended double bridge graph (see figure 1.9).

In these experiments the ants deposit an amount of pheromone that is the inverse of their path length (i.e., $\Delta\tau^k = 1/L^k$); also, before depositing it, they eliminate loops using the procedure described in figure 1.8.

To evaluate the behavior of the algorithm we observe the development of the path lengths found by the ants. In particular, we plot the moving averages of the path lengths after loop elimination (moving averages are calculated using the $4 \cdot m$ most recent paths found by the ants, where m is the number of ants). In other words, in the graph of figure 1.10 a point is plotted each time an ant has completed a journey from the source to the destination and back (the number of journeys is on the x-axis), and the corresponding value on the y-axis is given by the length of the above-mentioned moving average after loop elimination.

We ran experiments with S-ACO and different settings for the evaporation rate $\rho \in \{0, 0.01, 0.1\}$ ($\alpha = 1$ and $m = 128$ in all experiments). If $\rho = 0$, no pheromone evaporation takes place. Note that an evaporation rate of $\rho = 0.1$ is rather large, because evaporation takes place at each iteration of the S-ACO algorithm: after ten iterations, which corresponds to the smallest number of steps that an ant needs to

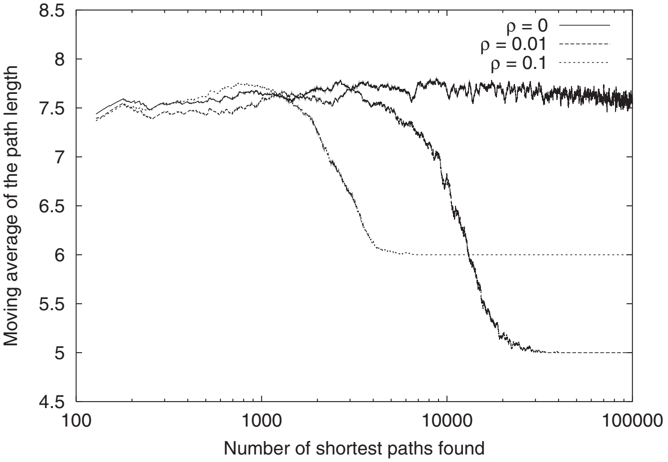


Figure 1.10
The graph plots the moving averages (given on the y-axis) of the ants' path length for the graph of figure 1.9 as a function of the number of completed paths (given on the x-axis). We give plots for not using evaporation ($\rho = 0$), low evaporation ($\rho = 0.01$), and high evaporation ($\rho = 0.1$). The trials were stopped after 5000 iterations; $\alpha = 1$ and $m = 128$.

build the shortest path and to come back to the source, roughly 65% of the pheromone on each arc evaporates, while with $\rho = 0.01$ this evaporation is reduced to around 10%.

Figure 1.10 gives the observed moving averages. Although the graphs only show results of a single run of the algorithm, they are representative of the typical algorithm behavior. If no evaporation is used, the algorithm does not converge, which can be seen by the fact that the moving average has approximately the value 7.5, which does not correspond to the length of any path (with these parameter settings, this result typically does not change if the run lasts a much higher number of iterations). With pheromone evaporation, the behavior of S-ACO is significantly different. After a short transitory phase, S-ACO converges to a single path: either the shortest one (the moving average takes the value 5 for $\rho = 0.01$) or the path of length 6 for $\rho = 0.1$. A closer examination of the results revealed that in both cases at convergence all the ants had built loop-free paths of the indicated length.

In further experiments with S-ACO on this graph we made the following general observations:

- Without pheromone updates based on solution quality, S-ACO performance is much worse. In particular, the algorithm converges very often to the suboptimal solution of length 8; the larger the parameters α or ρ, the faster S-ACO converges to this suboptimal solution.

- The pheromone evaporation rate ρ can be critical. In particular, we observed that S-ACO often converged to suboptimal paths when evaporation was set to a value that was too high. For example, in fifteen trials with ρ set to 0.2, S-ACO converged once to a path of length 8, once to a path of length 7, and twice to a path of length 6. Setting ρ to 0.01 S-ACO converged to the shortest path in all trials.
- Large values of α generally result in a worse behavior of S-ACO because they excessively emphasize the initial random fluctuations.

Discussion
We have seen that in real ant colonies the emergence of high-level patterns like shortest paths is only possible through the interaction of a large number of individuals. It is interesting that experimental results show that the same is true to a large extent for S-ACO: the use of a colony of ants is important to exploit the differential path length effect and to increase the robustness of the algorithm and reduce its dependence on parameter settings. As we have seen, a colony size larger than one is necessary to solve even simple problems like the double bridge.

In general, we noticed that as problems become more complex, the parameter settings of S-ACO become increasingly important to obtain convergence to the optimal solution. In particular, the experimental results presented above support the following conclusions: (1) the differential path length effect, although important, is not enough to allow the effective solution of large optimization problems; (2) pheromone updates based on solution quality are important for fast convergence; (3) large values for parameter α lead to a strong emphasis of initial, random fluctuations and to bad algorithm behavior; (4) the larger the number of ants, the better the convergence behavior of the algorithm, although this comes at the cost of longer simulation times; and (5) pheromone evaporation is important when trying to solve more complex problems. These observations will be of importance in the following chapters, where design decisions will be made both to define the ACO metaheuristic and to apply it to a multitude of different discrete optimization problems.

1.4 Bibliographical Remarks

The term *stigmergy* was introduced by Grassé to describe a form of indirect communication mediated by modifications of the environment that he observed in the workers caste of two species of termites, *Bellicositermes natalensis* and *Cubitermes* sp.

The original definition of stigmergy (see Grassé, 1959, p. 79), was: "Stimulation of workers by the performance they have achieved."

Termite nest building is the typical example of stigmergy, and is also the original example used by Grassé to introduce the concept. Termite workers build their nest using soil pellets, which they impregnate with a diffusing chemical substance called pheromone. They start nest construction (Grassé, 1959) by randomly depositing pellets on the ground. The deposits of soil pellets stimulate workers to accumulate more material on top of them through a positive feedback mechanism, since the accumulation of material reinforces the attraction of deposits by means of the diffusing pheromone emitted by the pellets (Bruinsma, 1979). This process works only if the density of the termites is above a given threshold. In fact, if the density is too low, pheromones are not added quickly enough and the positive feedback mechanism is inhibited by pheromone evaporation.

Although Grassé introduced the term stigmergy to explain the behavior of termite 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. As we have seen, the foraging behavior of ant colonies described in this chapter is an example of stigmergy: ants stimulate other ants by modifying the environment via pheromone trail updating. A brief history of the notion of stigmergy can be found in Theraulaz & Bonabeau (1999).

1.5 Things to Remember

- Deneubourg and colleagues (Deneubourg et al., 1990; Goss et al., 1989) have shown in controlled experimental conditions that foraging ants can find the shortest path between their nest and a food source by marking the path they follow with a chemical called pheromone.

- The foraging behavior of ant colonies can be replicated in simulation and inspires a class of ant algorithms known as "ant colony optimization" (ACO). ACO, the subject of this book, is currently one of the most successful examples of ant algorithms.

- In experiments with foraging ants, it was shown that the pheromone evaporation rate is so slow compared to the time necessary for the ant colony to converge to the short path that, for modeling purposes, it can be neglected. When considering artificial ants things are different. Experimental results show that on very simple graphs, like the ones modeling the double bridge or the extended double bridge setups, pheromone evaporation is also not necessary. On the contrary, it improves the algorithm's performance in finding good solutions to the minimum cost path problem on more complex graphs.

- Biologists have found that stigmergy is a useful concept to help explain the self-organizing capabilities of social insects (Theraulaz & Bonabeau, 1999; Dorigo, Bonabeau, & Theraulaz, 2000a).

1.6 Thought and Computer Exercises

Exercise 1.1 Prove by hand calculation that artificial ants using only forward (or only backward) pheromone update do not converge toward the common use of the minimum cost path in the double bridge experiment.

Exercise 1.2 Prove by hand calculation that, if artificial ants are given the capability (through the use of memory) to retrace their path to the destination node (recall section 1.3), then they are able to find the minimum cost path in the double bridge experiment even when they use only backward pheromone update.

Exercise 1.3 Implement a computer program that simulates the artificial ants in the double bridge experiment. You can do this in two ways: either by numerically solving equations (1.4) through (1.7), in this way obtaining the expected behavior of the system, or by running simulations. Is there any difference in the results? What happens if you only use a few ants in the simulation?

Exercise 1.4 Using the program above, study what happens when you change the α and r parameters. In particular, if you set $\alpha = 1$, does the probability of choosing the short branch still converge to 1? And how do the convergence properties of the algorithm change when increasing the branch length ratio r?

Exercise 1.5 An alternative model of the double bridge experiment to the one presented in section 1.2 [equations (1.4)–(1.7)] is the following. Let the amount of pheromone on a branch be proportional to the number of ants that used the branch in the past and let $m_s(t)$ and $m_l(t)$ be the numbers of ants that have used the short and the long branches after a total of t ants have crossed the bridge, with $m_s(t) + m_l(t) = t$. The probability $p_s(t)$ with which the $(t+1)$-th ant chooses the short branch can then be written as

$$p_s(t) = \frac{m_s(t)^\alpha}{m_s(t)^\alpha + m_l(t)^\alpha} = 1 - p_l(t). \tag{1.11}$$

The number of ants choosing the short branch is given by

$$m_s(t+1) = \begin{cases} m_s(t) + 1, & \text{if } q \le p_s(t); \\ m_s(t), & \text{otherwise}; \end{cases} \tag{1.12}$$

and the number of ants choosing the long branch by

$$m_l(t+1) = \begin{cases} m_l(t)+1, & \text{if } q > p_l(t); \\ m_l(t), & \text{otherwise;} \end{cases} \tag{1.13}$$

where q is a uniform random number drawn from the interval $[0,1]$.

Run Monte Carlo simulations of the dynamic system defined by the above equations and compare the results with those obtained in the first and second computer exercise.

Exercise 1.6 The ants' path-marking and foraging behavior can also be studied in unconstrained settings. Consider the following experimental setup: a squared environment contains three food sources and one nest. Ants leave the nest to search for food and, once food has been found, they go back to the nest depositing a pheromone trail on the ground. When they are looking for food, ants move stochastically using a probabilistic rule biased by pheromones they sense in the environment (see also Resnick, 1994). Implement a program which simulates the system described above and study how the ants' performance changes for different implementation choices. For example, you can study different forms of the probabilistic rules used by the ants, different ways of depositing pheromone on the ground (only while searching for food, only when going back to the nest, in both cases), different pheromone evaporation rates, and so on. (*Hint*: You may want to use Mitchel Resnick's StarLogo programming language, available at education.mit.edu/starlogo/).

Exercise 1.7 Develop an outline for the implementation of S-ACO (section 1.3.1). Consider the following issues:

- How do you build a structure which represents the individual ants?
- How do you represent the graph, the pheromone trails, and the heuristic information?
- How do you implement the solution construction policy?
- How do you implement loop elimination?
- How do you implement pheromone update?

Once you have implemented the algorithm, run it on a number of graphs. What are your experiences with the algorithm? How do you judge the quality and the convergence of the algorithm? Would you use this algorithm for attacking large minimum cost path problems? (Consider that there exist algorithms, such as the one proposed by Dijkstra [1959], that solve the minimum cost (shortest) path problem in $\mathcal{O}(n^2)$).