

## CHAPTER 10

---

# Ant Colony Optimization

---

Go to the ant, you sluggard; consider its ways, and be wise!

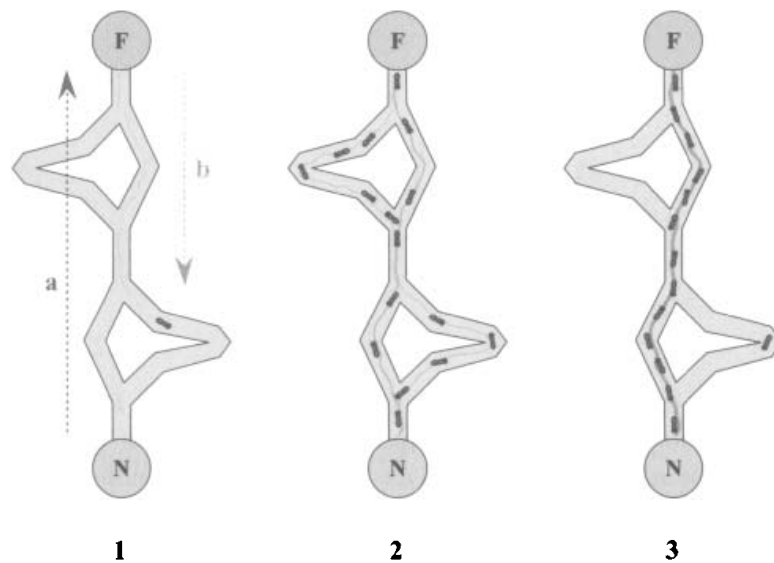
—Proverbs 6:6

Ants are simple creatures but they can accomplish a lot by working together. The quote at the beginning of this chapter presents ants as a paradigm of hard work, but they can also be portrayed as the epitome of selfless cooperation. A single ant does not have much to offer. A solitary ant might wander aimlessly in circles until it dies of exhaustion [Delsuc, 2003]. The average ant has only 10,000 neurons in its brain, which doesn't seem like enough to accomplish much. But ants join together in colonies that can number in the millions. A one-million member ant colony has a collective neuron count of 10 billion, which begins to rival the neuron count of an average human. Ants seem to operate as a single entity and are therefore sometimes referred to as a superorganism [Hölldobler and Wilson, 2008]. An ant colony discovered on the Japanese island of Hokkaido was reported in 1979 to contain over 300 million ants living in 45,000 interconnected nests [Hölldobler and Wilson, 1990, page 1]. Ants thrive in almost every environment on earth, and are estimated to comprise over 15% of the mass of all land animals on earth [Schultz, 2000]. Myrmecologists (that is, those who study ants) tell us that the number of ant species is about 8,800 and have a global population of about one quadrillion, which means that there are about 150,000 ants for every person on earth. Why is it that

ants, such tiny creatures, have been so successful for so long in so many different environments? Scientists attribute their adaptability and dominance to their social organization. Perhaps it is no coincidence that the most dominant mammal on Earth, humans, is also the clear leader in social organization.

Ants communicate mainly by using pheromones, which are chemical substances that they excrete. When ants travel along a path to a food source and bring food back to their colony, they leave a trail of pheromone. Other ants smell the pheromone with their antennae, follow the path, and bring more food back to the colony. In the process, ants continue to lay down pheromone, which other ants continue to smell, and the path to the food source is reinforced. The shortest path to the food thus becomes more attractive over time as it is strengthened by positive feedback.

Sometimes the food source is depleted or an obstacle prevents travel to the food source. When ants travel along a path and do not find food, they wander until they do find food. If they do not return to their colony using the original path, they do not deposit any additional pheromone on that path. As time passes, the pheromone on the original path evaporates, fewer ants take the original path, more ants take the new path to the new food source, and a new optimal path is discovered by the ants. This general process is depicted in Figure 10.1.



**Figure 10.1** Ants depositing and following pheromone. (1) The first ant travels in the direction indicated by *a*, finds a food source *F*, and returns to the nest *N* in the direction indicated by *b*, laying a pheromone trail as it travels. (2) The ants follow one of four possible paths from *N* to *F*, but pheromone reinforcement makes the shortest path more appealing. (3) The ants tend to follow the path with the most pheromone, continuing to reinforce its desirability, while the pheromone on the longer paths evaporates. (This figure was created by Johann Dréo, was copied from [http://en.wikipedia.org/wiki/File:Aco\\_branches.svg](http://en.wikipedia.org/wiki/File:Aco_branches.svg), and is distributed under the provisions of the GNU Free Documentation License.)

Ants can not only find the optimal path to a food source, but they can also perform many other impressive tasks by working together. They can build complex networks of tunnels, either underground, or, in the case of weaver ants, in trees. Their colonies have specialized rooms for storing, mating, and caring for larvae.

They can plant gardens to cultivate their own food source [Schultz, 1999]. They can form chains to cross gaps over the ground or over water (see Figure 10.2). They can form rafts to survive a flood, or to travel across water.



**Figure 10.2** Ants form a bridge between leaves. Ants use bridges not only for transportation, but also to pull leaves together during nest construction. Many other photographs like this can be found in [Hölldobler and Wilson, 1994]. (This photograph was taken by Sean Hoyland, was copied from <http://en.wikipedia.org/wiki/File:SSL11903p.jpg>, and is distributed under the provisions of the GNU Free Documentation License.)

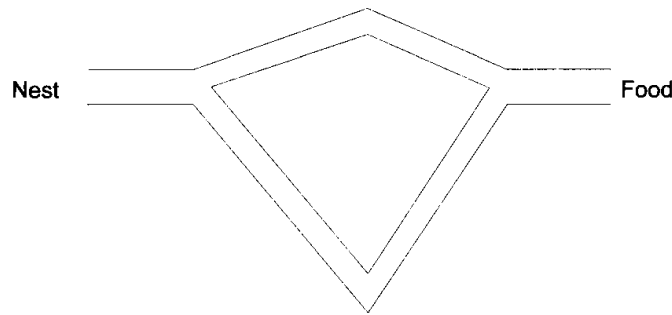
## Overview of the chapter

In this chapter we discuss ant colony optimization (ACO), which is an algorithm that is motivated by the pheromone-depositing behavior of ants. Most ACO researchers emphasize that it is not an EA since candidate solutions do not directly exchange solution information with each other. We include ACO in this book not because we want to engage in the EA/non-EA debate, but simply because ACO is an interesting and effective biologically-motivated, population-based optimization algorithm.

ACO was developed by Martin Dorigo in his doctoral dissertation and was first published in 1991 [Colormi et al., 1991]. In Section 10.1 we discuss the pheromone deposition of biological ants, its evaporation, and mathematical models that describe these processes. In Section 10.2 we discuss the ant system (AS), which was proposed in the mid-1990s, and which was the first ACO algorithm. ACO was originally proposed to find optimal paths, but was quickly modified to deal with optimization problems with continuous domains, and that is what we discuss in Section 10.3. In Section 10.4 we discuss some popular modifications that have been made to the basic ant system algorithm, including the max-min ant system (MMAS) and the ant colony system (ACS). In Section 10.5 we give a brief overview of ACO research in the area of theory and modeling.

## 10.1 PHEROMONE MODELS

Suppose that we observe an ant nest and a food source, and that ants have two possible routes to obtain food. One route is long, and the other is short, as shown in Figure 10.3. Goss and his coworkers ran many experiments of this type with the Argentine ant, and they found that in 95% of their experiments, over 80% of the ant traffic was on the shorter path [Goss et al., 1989]. As ants reached the fork of the path, they made a random decision which path to take. The ants that chose the shorter path were able to return to their nest sooner than the ants that chose the longer path. This resulted in more ants taking the shorter path per unit time. This, in turn, resulted in more pheromone deposition on the shorter path. Finally, the larger amount of pheromone on the shorter path motivated later ants to take that path.



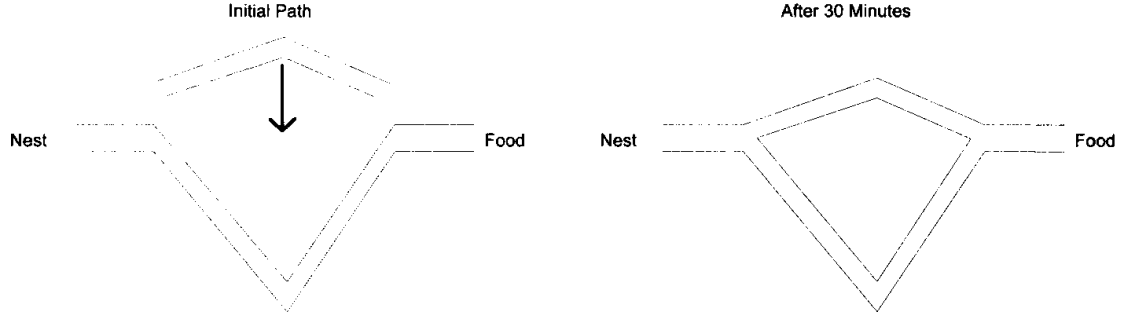
**Figure 10.3** An experimental setup to explore how ants find the shortest distance to food. In 95% of the experiments, over 80% of the ant traffic was on the shorter path. Adapted from [Goss et al., 1989].

We see that ant travel is a positive feedback phenomenon, at least to a certain point. There were always some ants that chose the longer path because their choices are partially governed by random processes. However, in general, as more ants choose the shorter path, the shorter path receives more pheromone; and as the shorter path receives more pheromone, more ants choose it.

This positive feedback phenomenon is also a characteristic of EAs. For example, in GAs, individuals in the first generation with beneficial genetic features are more likely to be selected for recombination. This means that the second generation is more likely to possess those genetic features. The increased prevalence of those beneficial features in the second generation then makes it more likely that they will be passed on to the third generation. This positive feedback phenomenon is seen not only in ACO and GAs, but in all EAs.

Pheromones are not only deposited by ants, but they also evaporate. Goss performed another experiment in which only one path was available to the ant colony. The ants ran back and forth between their nest and the food source on this path because they did not have any other options. After a while, Goss added a short path to the food, as shown in Figure 10.4. The ants now had a choice, but all of the pheromone was on the long path. However, when ants reach a fork in the road, they do not automatically take the pheromone-saturated path. They are more likely to take a path with more pheromones, but there is also a random element to their behavior. Therefore, some of the ants took the newly-presented short path.

As they took this short path, they deposited pheromones on it, which made it more attractive to subsequent ant travelers. In about 20% of the experiments, the majority of ants ended up taking the short path, even though it did not have any initial pheromone deposits. This demonstrates the fact that pheromones evaporate. However, in this experiment, they did not evaporate quickly enough for the ants to take the shorter path more often than the longer path.



**Figure 10.4** An experimental setup to explore how ants react when a shorter path to food is added. In 20% of the experiments, over 50% of the ant traffic converged to the shorter path. Adapted from [Goss et al., 1989].

In view of these experiments, and others like them, Deneubourg and his colleagues proposed a mathematical model for the deposition and evaporation of pheromone [Deneubourg et al., 1990]. Given two paths to choose from, the probability that an ant chooses path 1 is given by

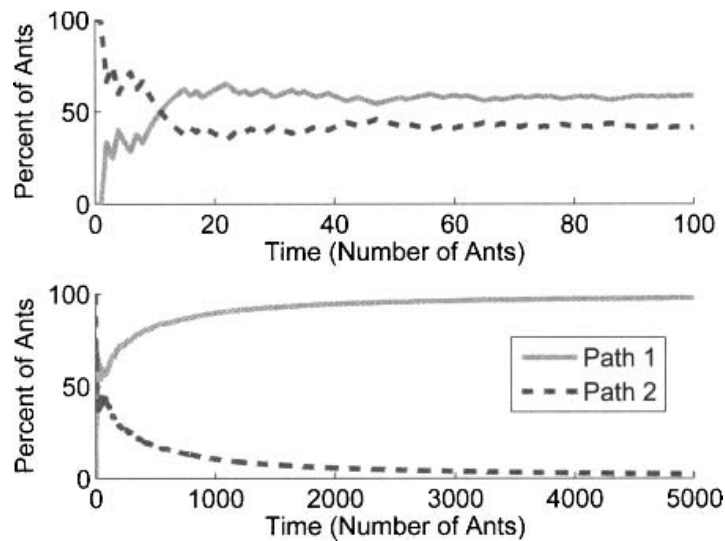
$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \quad (10.1)$$

where  $m_i$  is the number of ants that have previously chosen path  $i$ , and  $h$  and  $k$  are experimentally-determined parameters. This initial model does not take pheromone evaporation into account. Typical values for  $k$  and  $h$  are

$$k \approx 20, \quad h \approx 2. \quad (10.2)$$

Figure 10.5 shows the results of a simulation of Equation (10.1), which applies to two paths of equal length. The top plot shows that the behavior of the first 100 ants is not predictable. The ants' behavior is mostly random, and there is about a 50% chance that an ant will choose either path 1 or path 2. The bottom plot shows that as one path starts to receive the majority of pheromone deposits, it becomes more attractive, which results in the positive feedback phenomenon that we discussed earlier in this section. Eventually 100% of the ant traffic will be on only one of the two paths.

We see that ants are able to find optimal solutions to the problems that they encounter in their everyday lives. This motivates us to simulate artificial ants to find optimal solutions to engineering problems.



**Figure 10.5** Simulation results of Equation (10.1). Initially the ants have about a 50% chance of choosing either path. After awhile, one path receives the predominance of pheromone, which results in a positive feedback phenomenon, and 100% of the ant traffic settles on one of the two paths.

## 10.2 ANT SYSTEM

The ant system was the first ACO algorithm that was published [Colormi et al., 1991], [Dorigo et al., 1996]. It can be illustrated on the traveling salesman problem (TSP; see Section 2.5 and Chapter 18). Each ant in the ACO simulation travels from one city to another, and the simulation deposits pheromone on the ants' paths after they complete their trip. Pheromones are not only deposited, but they also evaporate. The probability that an ant travels from its current city to some other city is proportional to the amount of pheromone between the cities. Ants are also assumed to have some knowledge about the problem that helps them make decisions during their travels. They know the distance from their current city to other cities, and they are more likely to travel to a close city than to a distant city, since the objective of the algorithm is to find the shortest path. The ant system algorithm is illustrated in Figure 10.6.

Figure 10.6 shows that the probability of each ant traveling from city  $i$  to city  $j$  is proportional to the amount of pheromone on the path between those cities, and is inversely proportional to the distance between those cities. The ratio  $\alpha/\beta$  determines the relative importance of pheromone information to distance information when deciding which city to travel to. When an ant travels from city  $i$  to city  $j$ , the amount of pheromone on that path is increased in an amount proportional to the quality of that ant's solution (that is, inversely proportional to the ant's total travel distance).

Figure 10.6 is a fairly complete algorithm, but there are still some implementation details that are left to the programmer. For example, is  $\tau_{ij} = \tau_{ji}$ ? In a biological ant system, the amount of pheromone between nodes  $i$  and  $j$  is the same as the

---

```

 $n$  = number of cities
 $\alpha, \beta$  = relative importance of pheromones vs. heuristic information
 $Q$  = deposition constant
 $\rho$  = evaporation rate  $\in (0, 1)$ 
 $\tau_{ij} = \tau_0$  (initial pheromone between cities  $i$  and  $j$ ) for  $i \in [1, n]$  and  $j \in [1, n]$ 
 $d_{ij}$  = distance between cities  $i$  and  $j$  for  $i \in [1, n]$  and  $j \in [1, n]$ 
While not(termination criterion)
  For  $q = 1$  to  $n - 1$ 
    For each ant  $k \in [1, N]$ 
      Initialize the starting city  $c_{k1}$  of each ant  $k \in [1, N]$ 
      Initialize the set of cities visited by ant  $k$ :  $C_k \leftarrow \{c_{k1}\}$  for  $k \in [1, N]$ 
      For each city  $j \in [1, n]$ ,  $j \notin C_k$ 
        probability  $p_{ij}^{(k)} \leftarrow (\tau_{ij}^\alpha / d_{ij}^\beta) / (\sum_{m=1, m \notin C_k}^n \tau_{im}^\alpha / d_{im}^\beta)$ 
      Next  $j$ 
      Let ant  $k$  go to city  $j$  with probability  $p_{ij}^{(k)}$ 
      Use  $c_{k,q+1}$  to denote the city selected in the previous line
       $C_k \leftarrow C_k \cup \{c_{k,q+1}\}$ 
    Next ant
  Next  $q$ 
   $L_k \leftarrow$  total path length constructed by ant  $k$ , for  $k \in [1, N]$ 
  For each city  $i \in [1, n]$  and each city  $j \in [1, n]$ 
    For each ant  $k \in [1, N]$ 
      If ant  $k$  went from city  $i$  to city  $j$ 
         $\Delta\tau_{ij}^{(k)} \leftarrow Q / L_k$ 
      else
         $\Delta\tau_{ij}^{(k)} \leftarrow 0$ 
    End if
  Next ant
   $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)}$ 
Next city pair
Next generation

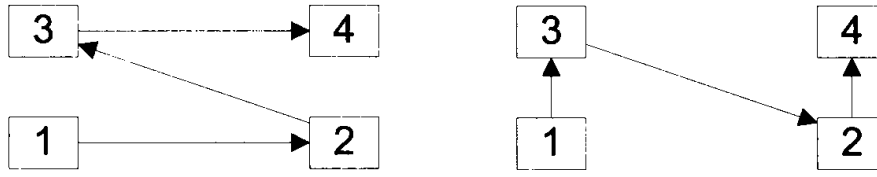
```

---

**Figure 10.6** A simple ant system (AS) for solving a TSP. Each generation, some of the pheromone between cities  $i$  and  $j$  evaporates, but the pheromone also increases due to ants that travel between the two cities.

amount between nodes  $j$  and  $i$ , but this is not necessary in ant system simulations. It can easily be imagined that travel from node  $i$  to  $j$  could lead to a good solution, while travel from node  $j$  to  $i$  could lead to a poor solution. This would result in  $\tau_{ij} \neq \tau_{ji}$ , which corresponds to the asymmetric TSP (see Figure 10.7).

Other implementation details that could be added to Figure 10.6 include intelligent initialization, elitism, and mutation. First, ACO performance, like EA performance, can strongly depend on proper initialization (see Section 8.1). For the TSP we may want to initialize certain individuals using a simple heuristic algorithm. For example, at the first generation we could force one of the ants to



**Figure 10.7** In this example, we assume that the tour begins at node 1. The tour on the left is much worse (that is, it has a longer distance) than the tour on the right, but both tours have paths between nodes 2 and 3. Since the tour on the left is long and the tour on the right is short, we would expect  $\tau_{23}$  to be less than  $\tau_{32}$  for an effective ACO algorithm; that is, it should be less attractive to go from node 2 to 3, than to go from node 3 to 2.

deterministically visit the closest city at each decision point. We discuss TSP initialization in more detail in Section 18.2. Second, elitism can be used in ACO, just as in any EA (see Section 8.4). Elitism could be incorporated by keeping track of the best few ants each generation, and forcing them to repeat the same route at the next generation. This ensures that the best route is not lost from one generation to the next. An ant system algorithm with elitism is sometimes called an elitist ant system [Dorigo et al., 1996], [Blum, 2005a]. Third, mutation could also be used in ACO, just like in any EA (see Section 8.9). Mutation could be incorporated by randomly altering routes with some mutation probability. Researchers have proposed several mechanisms for mutating TSP routes, which we discuss in Section 18.4.

Figure 10.6 shows that there are several parameters that need to be tuned for an ant system. These parameters include:

- The number of ants  $N$ , which is the population size;
- $\alpha$  and  $\beta$ , which are the relative importance of pheromone amounts and heuristic information;
- $Q$ , which is the deposition constant;
- $\rho$ , which is the evaporation rate;
- $\tau_0$ , which is the initial pheromone amount between each city.

The effects of these parameters have been studied by several researchers. As a typical example of recommended values [Dorigo et al., 1996]:

- $N = n$  (that is, number of ants = number of cities);
- $\alpha = 1$  and  $\beta = 5$ ;
- $Q = 100$ , although its effect is not significant;
- $\rho \in [0.5, 0.99]$ ;
- $\tau_0 \approx 10^{-6}$ .



### ■ EXAMPLE 10.1

This example applies the ant system algorithm of Figure 10.6 to the Berlin52 TSP, which consists of 52 locations in Berlin, Germany [Reinelt, 2008]. Berlin52 is a symmetric TSP, which means that we are given a set of nodes and distances between each pair of nodes, and our goal is to find a round trip of minimal total length while visiting each node exactly once. In a symmetric TSP, such as the Berlin52 TSP, the distance from node  $i$  to  $j$  is the same as from node  $j$  to  $i$ . We use the following ant system parameters:

- $N = 53$ ;<sup>1</sup>
- $\alpha = 1$  and  $\beta = 5$ ;
- $Q = 20$ ;
- $\rho = 0.9$ ;
- $\tau_0 = 10^{-6}$ ;
- In general,  $\tau_{ij} \neq \tau_{ji}$ ;<sup>2</sup>
- Random initialization;
- Two elite ants each generation;
- No mutation.

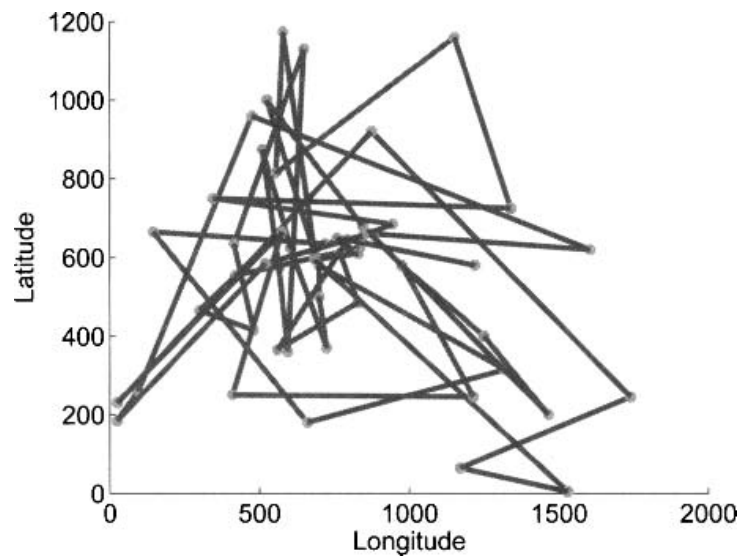
Figure 10.8 shows the best tour of the initial population, with a total distance of 24,780. We see that the best initial tour appears to be quite poor. Figure 10.9 shows the convergence of the AS as it searches for the best tour. We see that the AS converges very quickly, and elitism ensures that the best tour never increases in total distance from one generation to the next. Figure 10.10 shows the best tour found by the AS after 10 generations, with a total distance of 7,796. We see that ACO has found a much better tour than the best one from the initial population, and the total distance has decreased by 69%.

Finally, Figure 10.11 shows the globally optimal tour, which has been proven to be optimal, and which has a total distance of 7,542. Comparing Figures 10.10 and 10.11, we see that ACO has found a tour that is similar to the optimal tour, and that is only 3% worse than the optimal tour. Every ACO simulation will find a different solution since ACO is a stochastic algorithm. But considering the fact that there are  $51! = 1.6 \times 10^{66}$  potential solutions to this problem, ACO does quite well to find a solution that is only 3% worse than optimal.

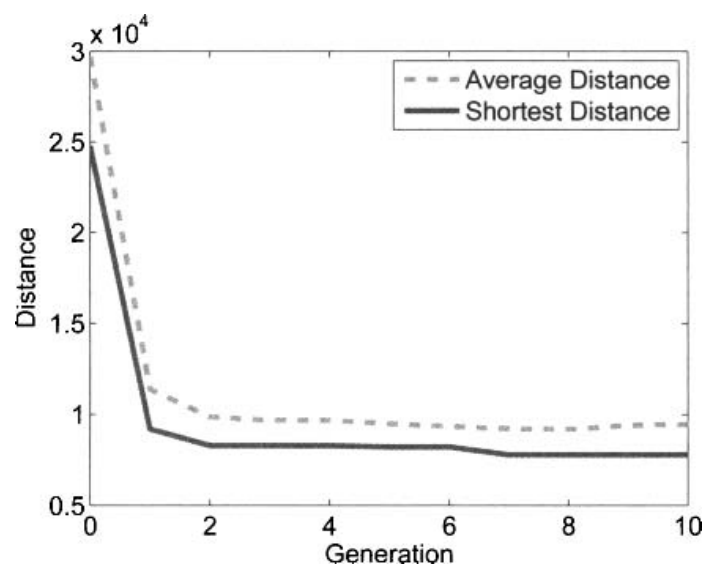
□

<sup>1</sup>The standard population size for ACO varies from one paper to the next. Many ACO and TSP papers use  $N = n$ , while others use  $N = n + 1$ .

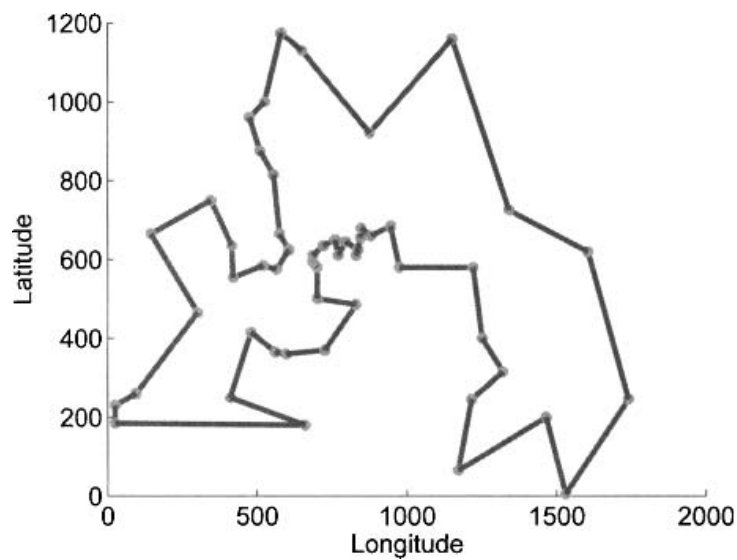
<sup>2</sup>Usually  $\tau_{ij} = \tau_{ji}$  is recommended (but not required) for symmetric TSPs.



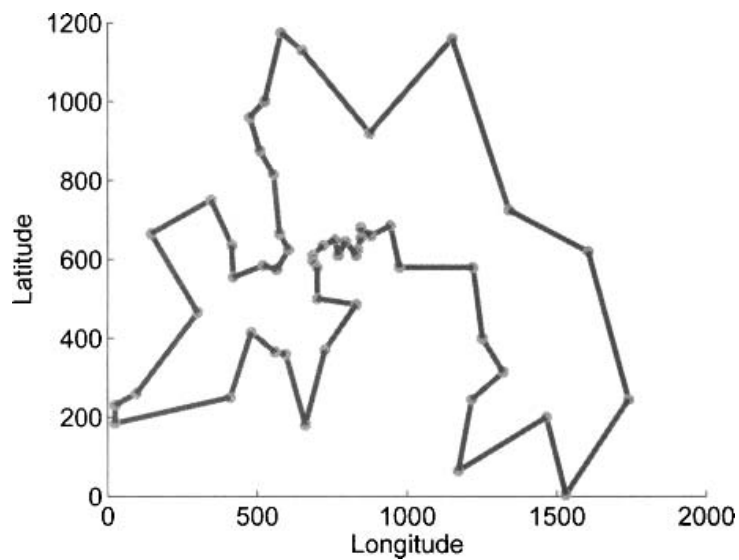
**Figure 10.8** The best initial tour out of 53 random tours for Example 10.1, with a total distance of 24,780.



**Figure 10.9** Ant system convergence of Example 10.1.



**Figure 10.10** The best tour found after 10 ant system generations for Example 10.1, with a total distance of 7,796. This is 69% better than the best initial tour shown in Figure 10.8, and 3% worse than the globally optimal tour shown in Figure 10.11.



**Figure 10.11** The globally optimal tour for Example 10.1, with a total distance of 7,542.

### 10.3 CONTINUOUS OPTIMIZATION

ACO was originally developed for TSP-like problems, but it has since been modified for optimization problems with continuous domains [Socha and Dorigo, 2008], [Tsutsui, 2004], [de Franca et al., 2008]. One simple approach for applying a discrete optimization algorithm like ACO to a continuous-domain problem is to divide each dimension  $i$  of the search space into discretized intervals. That is, we are trying to minimize the  $n$ -dimensional problem  $f(x)$ , where  $x = [x_1, \dots, x_n]$ , and

$$\begin{aligned} x_i &\in [x_{i,\min}, x_{i,\max}] \\ x_{i,\min} &= b_{i1} < b_{i2} < \dots < b_{i,B_i} = x_{i,\max} \end{aligned} \quad (10.3)$$

where  $B_i - 1$  is the number of discrete intervals into which we divide the  $i$ -th domain. Each generation, if the  $i$ -th domain of a candidate solution is between  $b_{ij}$  and  $b_{i,j+1}$ , then we update the pheromone of that interval as in the standard ant system algorithm:

$$\text{if } x_i \in [b_{ij}, b_{i,j+1}] \text{ then } \tau_{ij} \leftarrow \tau_{ij} + Q/f(x) \quad (10.4)$$

where  $Q$  is the standard ant system deposition constant, and we assume that  $f(x) > 0$  for all  $x$ . Equation (10.4) is analogous to the statement  $\Delta\tau_{ij}^{(k)} \leftarrow Q/L_k$  in Figure 10.6. We use pheromone amounts to probabilistically construct new solutions at the beginning of each generation. If the interval  $[b_{ij}, b_{i,j+1}]$  has a lot of pheromone, then there is a large probability that a candidate solution will be constructed such that its  $i$ -th dimension is in that interval. One way of doing this is to set the  $i$ -th dimension of the candidate solution to a random number  $r \in [b_{ij}, b_{i,j+1}]$ .

Figure 10.12 outlines a continuous ant system algorithm. Figure 10.12 assumes that the cost function, which is denoted as  $L_k$ , is positive for all  $k$ . If this property is not satisfied for a given problem, then the cost values of the population should be shifted so that it is satisfied. Figure 10.12 does not show the elitism option, but we can (and should) easily include elitism as described for EAs in Section 8.4. A continuous AS could also be combined with local search so that after an ant is placed in a discrete bin, local search is used to find the optimal solution within that bin.

The use of discretized intervals for each problem dimension is a simple way to extend the AS to continuous problems. A more rigorous implementation of a continuous ant system could use kernels to construct a continuous approximation to the discrete PDF that is represented by the pheromone amounts [Simon, 2006, Chapter 15], [Blum, 2005a].

---

$n$  = number of dimensions  
 Divide the  $i$ -th dimension into  $B_i - 1$  intervals as shown in Equation (10.3),  $i \in [1, n]$   
 $\alpha$  = importance of pheromone amounts  
 $Q$  = deposition constant  
 $\rho$  = evaporation rate  $\in (0, 1)$   
 $\tau_{i,j_i} = \tau_0$  (initial pheromone) for  $i \in [1, n]$  and  $j_i \in [1, B_i - 1]$   
 Randomly initialize a population of ants (candidate solutions)  $a_k$ ,  $k \in [1, N]$   
 While not(termination criterion)  
     For each ant  $a_k$ ,  $k \in [1, N]$   
         For each dimension  $i \in [1, n]$   
             For each discretized interval  $[b_{ij}, b_{i,j+1}]$ ,  $j \in [1, B_i - 1]$   
                 Probability  $p_{ij}^{(k)} \leftarrow \tau_{ij}^\alpha / \sum_{m=1}^{B_i-1} \tau_{im}^\alpha$   
                 Next discretized interval  
                  $a_k(x_i) \leftarrow U[b_{ij}, b_{i,j+1}]$  with probability  $p_{ij}^{(k)}$   
             Next dimension  
         Next ant  
          $L_k \leftarrow$  cost of solution constructed by ant  $a_k$ ,  $k \in [1, N]$   
         For each dimension  $i \in [1, n]$   
             For each discretized interval  $[b_{ij}, b_{i,j+1}]$ ,  $j \in [1, B_i - 1]$   
                 For each ant  $a_k$ ,  $k \in [1, N]$   
                     If  $a_k(x_i) \in [b_{ij}, b_{i,j+1}]$   
                          $\Delta\tau_{ij}^{(k)} \leftarrow Q/L_k$   
                     else  
                          $\Delta\tau_{ij}^{(k)} \leftarrow 0$   
                     End if  
                 Next ant  
                  $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)}$   
             Next discretized interval  
         Next dimension  
     Next generation

---

**Figure 10.12** A simple ant system (AS) for solving a continuous-domain minimization problem.  $a_k(x_i)$  is the  $i$ -th element of the  $k$ -th candidate solution. Each generation, pheromone in each bin evaporates, but the pheromone also increases in an amount proportional to the number of ants that construct a candidate solution in that bin.  $U[b_{ij}, b_{i,j+1}]$  is a random number that is uniformly distributed between  $b_{ij}$  and  $b_{i,j+1}$ .

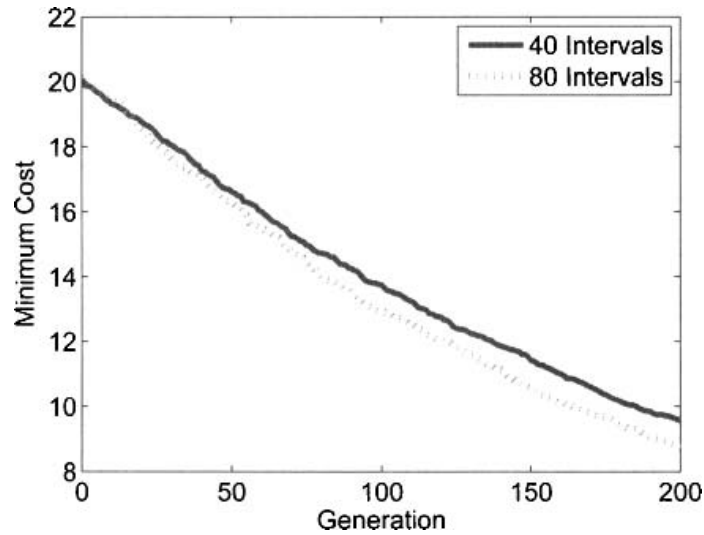
### ■ EXAMPLE 10.2

In this example we optimize the 20-dimensional Ackley function (see Appendix C.1.2). We use the algorithm of Figure 10.12 with the following parameters:

- $N = 50$ ;
- $\alpha = 1$ ;

- $Q = 20$ ;
- $\rho = 0.9$ ;
- $\tau_0 = 10^{-6}$ ;
- Two elite solutions each generation;
- Mutation rate = 1% per dimension per individual per generation;
- Number of intervals  $B_i = 40$  or 80, for  $i \in [1, n]$ .

Figure 10.13 shows the best solution at each generation, averaged over 20 Monte Carlo simulations, for both 40 and 80 intervals per dimension. We see that convergence is better for more intervals per dimension, but computation time increases as the number of intervals increases. There are two reasons for this that can be seen from Figure 10.12. The first reason is the “for each discretized interval” loops. The second reason is that the decision of which interval in which to place  $a_k(x_i)$  is more complicated. However, it should be noted that for most real-world optimization problems, the cost function calculation is the primary computational consideration, so the extra computation required for discretized domain intervals may not be a major problem (see Chapter 21).



**Figure 10.13** Example 10.2: Convergence of the continuous ant system applied to the 20-dimensional Ackley function. The plots show the best solution at each generation averaged over 20 Monte Carlo simulations. We get better performance if we distribute pheromone over more intervals per dimension.

□

## 10.4 OTHER ANT SYSTEMS

Many modifications have been made to the standard ant system algorithm that is described in the previous sections. This section describes two basic modifications: the max-min ant system in Section 10.4.1 and the ant colony system in Section 10.4.2.

### 10.4.1 Max-Min Ant System

The max-min ant system (MMAS) is a simple modification to the standard ant system algorithm [Dorigo et al., 2006], [Stützle and Hoos, 2000]. It is characterized by two main features. First, pheromone is increased only by the best ant each generation. This has the effect of reducing exploration and increasing exploitation of the best known solution. Second, the pheromone amount is bounded from above and below. This has the opposite effect; that is, it increases exploration because even the worst tours retain a nonzero amount of pheromone, and even the best tours cannot get so much pheromone that they completely dominate the ants' decisions.

The first difference between the standard ant system algorithm and MMAS can be seen in the following equations, which are replaced in Figures 10.6 and 10.12:

$$\begin{aligned} \text{Standard AS:} \quad \tau_{ij} &\leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)} \\ \text{MMAS:} \quad \tau_{ij} &\leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{(\text{best})} \end{aligned} \quad (10.5)$$

where *best* is the index of the best candidate solution. In the TSP of Figure 10.6,  $\Delta\tau_{ij}^{(\text{best})}$  is given as

$$\Delta\tau_{ij}^{(\text{best})} \leftarrow \begin{cases} Q/L_{\text{best}} & \text{if city } i \rightarrow \text{city } j \text{ belongs to the best tour} \\ 0 & \text{otherwise} \end{cases} \quad (10.6)$$

and in the continuous AS of Figure 10.12,  $\Delta\tau_{ij}^{(\text{best})}$  is given as

$$\Delta\tau_{ij}^{(\text{best})} \leftarrow \begin{cases} Q/L_{\text{best}} & \text{if the } i\text{-th dimension of the best individual} \in [b_{ij}, b_{i,j+1}] \\ 0 & \text{otherwise.} \end{cases} \quad (10.7)$$

The second difference between the standard ant system algorithm and MMAS is implemented with the following simple equations after  $\tau_{ij}$  has been updated:

$$\begin{aligned} \tau_{ij} &\leftarrow \max(\tau_{ij}, \tau_{\min}) \\ \tau_{ij} &\leftarrow \min(\tau_{ij}, \tau_{\max}) \end{aligned} \quad (10.8)$$

where  $\tau_{\min}$  and  $\tau_{\max}$  are tuned for the specific problem that is being optimized.

With some imagination, we can see that MMAS could be generalized in several different ways. For example, instead of allowing only by the best ant to deposit pheromone, we could allow the best  $M$  ants to deposit pheromone, where  $M$  is a tuning parameter. Or we could allow the  $m$ -th best ant to deposit pheromone with probability  $p_m$ , where  $p_m$  decreases with increasing cost. Assuming that we want more exploration at the beginning of the optimization process and more exploitation

at the end of the process, we could increase  $(\tau_{\max} - \tau_{\min})$  as the generation count increases. With some imagination and experimentation, we could undoubtedly also find other extensions of MMAS that would improve performance on various kinds of problems.

### ■ EXAMPLE 10.3

In this example we repeat the minimization of the Ackley function with  $n = 20$  dimensions as in Example 10.2. We use the following parameters:

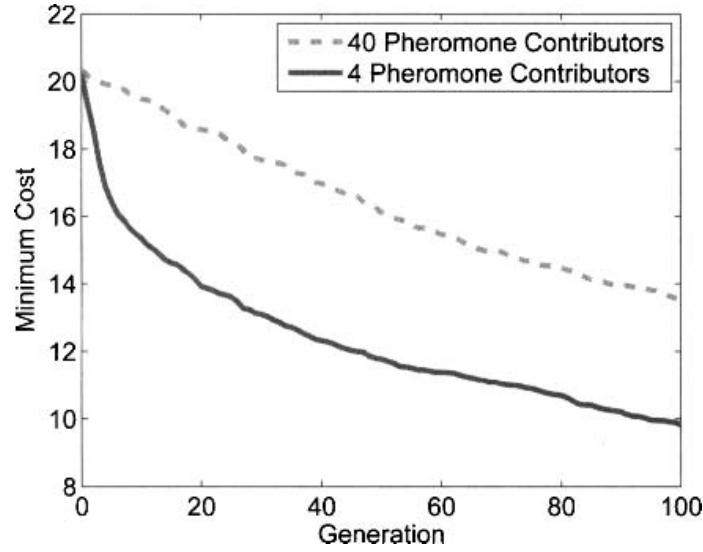
- $N = 40$ ;
- $\alpha = 1$ ;
- $Q = 20$ ;
- $\rho = 0.9$ ;
- $\tau_0 = 10^{-6}$ ;
- Two elite candidate solutions each generation;
- Mutation rate 1% per dimension per individual per generation;
- Number of intervals  $B_i = 20$  for  $i \in [1, n]$ ;
- $\tau_{\min} = 0$  and  $\tau_{\max} = \infty$ .

We only allow  $M$  ants to deposit pheromone:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{(\text{best}_m)} \quad (10.9)$$

for  $m \in [1, M]$ , where  $\text{best}_m$  is the index of the  $m$ -th best individual each generation. That is, only the best  $M$  ants deposit pheromone on the domain that they have explored. Other than this change, the algorithm we use in this example is the same as the ant system algorithm in Example 10.2. Figure 10.14 shows the best solution at each generation, averaged over 20 Monte Carlo simulations, for  $M = 4$  and  $M = 40$ . We see that convergence is much better when fewer ants are allowed to deposit pheromone. This makes sense intuitively. We do not want poor individuals to reinforce their solutions.





**Figure 10.14** Example 10.3: Convergence of the continuous ant system applied to the 20-dimensional Ackley function. The plots show the best solution at each generation averaged over 20 Monte Carlo simulations. We get better performance if we allow only the best ants to deposit pheromone on their solutions.

□

### 10.4.2 Ant Colony System

The ant colony system (ACS) is an extension of AS [Dorigo and Gambardella, 1997a], [Dorigo and Gambardella, 1997b], [Dorigo et al., 2006]. In spite of their common roots, AS and ACS are quite different in their behavior and performance. ACS is characterized by two main extensions to AS. First, a local pheromone update is implemented by each ant as it constructs its solution. As soon as an ant travels from city  $i$  to city  $j$ , the pheromone along that path is updated as follows:

$$\tau_{ij} \leftarrow (1 - \phi)\tau_{ij} + \phi\tau_0 \quad (10.10)$$

where  $\phi \in [0, 1]$  is the local pheromone decay constant, and  $\tau_0$  is the initial pheromone amount. If  $\phi = 0$  then  $\tau_{ij}$  does not change and we are back to the original ant system. Equation (10.10) indicates that pheromone between cities  $i$  and  $j$  decays as ants travel that path. This is not biologically accurate,<sup>3</sup> but it discourages other ants from following the same path and hence encourages exploration and diversity. After all ants have constructed a candidate solution we implement one of the standard global pheromone update rules of Equation (10.5).

The second extension that ACS makes to AS is the use of a pseudo-random proportional rule for candidate solution construction. Denote by  $(a_k \rightarrow j)$  the event

<sup>3</sup>Extensions to ACO and EAs often stray from the algorithms' biological foundations, but our goal is primarily to develop effective optimization algorithms rather than to accurately model biology. The biological roots of ACO and EAs primarily serve as inspiration.

that the  $k$ -th ant goes to city  $j$  while constructing its candidate solution. Denote by  $\Pr(a_k \rightarrow j)$  the probability that  $(a_k \rightarrow j)$ . The difference between standard AS candidate solution construction and ACS candidate solution construction is the following:

$$\begin{aligned} \text{AS:} \quad & \Pr(a_k \rightarrow j) = p_{ij}^{(k)} \\ \text{ACS:} \quad & \Pr(a_k \rightarrow j) = \begin{cases} 1 & \text{if } j = \arg \max_J p_{iJ}^{(k)} \\ 0 & \text{otherwise} \end{cases} \quad \text{if } r < q_0 \\ & \quad \quad \quad p_{ij}^{(k)} \quad \quad \quad \text{if } r \geq q_0 \end{aligned} \quad (10.11)$$

where  $r$  is a random number taken from a uniform distribution on  $[0, 1]$ , and  $q_0 \in [0, 1]$  is a tuning parameter. In standard AS, probabilities are derived using pheromone amounts, and ant  $k$  decides which city to go to based on those probabilities (see Figures 10.6 and 10.12). However, in ACS, there is a  $q_0$  probability that ant  $k$  goes to the city with the highest probability (that is, with the largest amount of pheromone leading from the current city to it, denoted by the  $\arg \max$  function in Equation (10.12)); and there is a  $(1 - q_0)$  probability that ant  $k$  uses the standard AS rule to decide which city to go to. This biases the ants to explore highly promising options in their solution construction. This is conceptually equivalent to increasing the probability of high-pheromone paths, which is equivalent to increasing  $\alpha$  in Figures 10.6 and 10.12.

The ACS probabilities of Equation (10.11) when  $r \geq q_0$  are only approximately accurate. For better accuracy, they should be normalized so that they sum to 1 (see Problem 10.7).

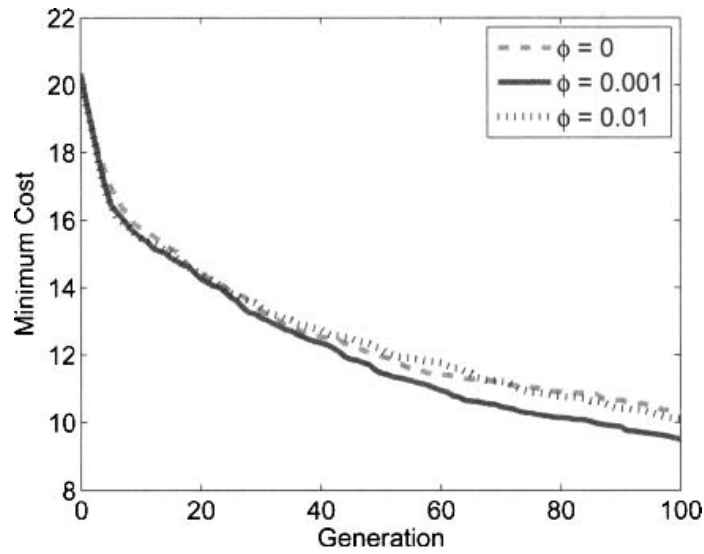
#### ■ EXAMPLE 10.4

In this example we investigate the use of the local pheromone decay constant  $\phi$  in ACS. As in earlier examples in this chapter, we minimize the Ackley function with  $n = 20$  dimensions. We use the following parameters:

- $N = 40$ ;
- $\alpha = 1$ ;
- $Q = 20$ ;
- $\rho = 0.9$ ;
- $\tau_0 = 10^{-6}$ ;
- Two elite solutions each generation;
- Mutation rate 1% per dimension per individual per generation;
- Number of intervals  $B_i = 20$  for  $i \in [1, n]$ ;
- $\tau_{\min} = 0$  and  $\tau_{\max} = \infty$ ;
- The best four ants deposit pheromone each generation;
- Exploration constant  $q_0 = 0$ .

Figure 10.15 shows the best solution at each generation, averaged over 20 Monte Carlo simulations, for  $\phi = 0$ , 0.001, and 0.01. We see that performance is noticeably better with a nonzero value for the local pheromone decay constant. A positive value of  $\phi$  encourages more exploration, which results in faster convergence. However, if  $\phi$  is too large, other ants are discouraged too strongly from exploring previously used paths, and performance becomes worse. To make more firm conclusions, we should perform statistical significance tests on the results of Figure 10.15 (see Appendices B.2.4 and B.2.5).

□

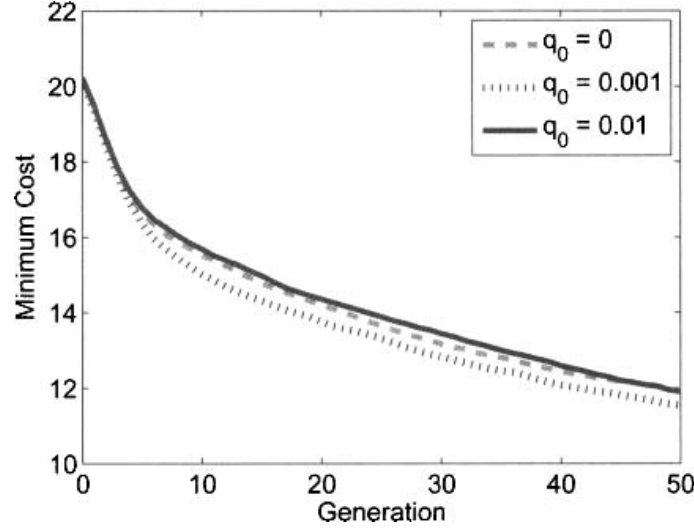


**Figure 10.15** Example 10.4: Performance of the ant colony system (ACS) on the 20-dimensional Ackley function. The traces show the best solution at each generation averaged over 20 Monte Carlo simulations for various values of the local pheromone decay constant. We get better performance with  $\phi > 0$ , but if  $\phi$  is too large then performance suffers.

### ■ EXAMPLE 10.5

In this example we investigate the use of the exploration constant  $q_0$  in ACS. As in earlier examples in this chapter, we minimize the Ackley function with  $n = 20$  dimensions. We use the same ACS parameters as in Example 10.4, except that we fix the local pheromone decay constant  $\phi = 0$  and test various values of  $q_0$ . Figure 10.16 shows the best solution at each generation, averaged over 100 Monte Carlo simulations, for  $q_0 = 0$ , 0.001, and 0.01. We see that performance is slightly better with a nonzero value for the exploration constant. A positive value of  $q_0$  provides a greater bias to the ants to use more favorable solution features. However, if  $q_0$  is too large, then the ACS does not have enough exploration and performance becomes worse. To make more firm conclusions, we should perform statistical significance tests on the results of Figure 10.16 (see Appendices B.2.4 and B.2.5). Also note that these

results are highly dependent on the particular problem that we solve, and on the other parameter settings listed in the previous example. Most ACS implementations use high values of  $q_0$ , such as  $q_0 = 0.9$  [Dorigo and Gambardella, 1997b].



**Figure 10.16** Example 10.5 performance of the ant colony system (ACS) on the 20-dimensional Ackley function. The traces show the best solution at each generation averaged over 100 Monte Carlo simulations for various values of the exploration constant. We get better performance with  $q_0 > 0$ , but if  $q_0$  is too large then performance suffers.

□

### 10.4.3 Even More Ant Systems

Space prevents us from going into detail about other ant systems, but there are a few notable variations that we mention briefly here. In the elitist ant system, the best solution deposits pheromone every time the other ants deposit pheromone [Dorigo and Stützle, 2004, Chapter 3]. The  $\Delta\tau$  calculation in Figure 10.12 is thus modified as follows:

$$\begin{aligned} \text{Standard AS: } \Delta\tau_{ij}^{(k)} &\leftarrow \delta_{ij}^{(k)} Q / L_k \\ \text{Elitist AS: } \Delta\tau_{ij}^{(k)} &\leftarrow \delta_{ij}^{(k)} Q / L_k + \delta_{ij}^{(\text{best})} Q / L_{\text{best}} \end{aligned} \quad (10.12)$$

where  $\delta_{ij}^{(k)} = 1$  if the  $i$ -th dimension of the  $k$ -th candidate solution lies in the  $j$ -th discretized interval, and  $\text{best}$  is the index of the best individual in the population. We see that in the elitist ant system, every time an ant deposits pheromone, the best ant also does so.

Ant-Q is a hybrid of AS and Q-learning [Gambardella and Dorigo, 1995]. In rank-based AS, the amount of pheromone deposited depends not only on an ant's solution quality, but also on its rank relative to the other ants [Dorigo and Stützle, 2004,

Chapter 3]. Approximated non-deterministic tree search (ANTS) specifies certain mechanisms to define how attractive a move is, and how to update pheromone [Maniezzo et al., 2004]. The best-worst AS deposits extra pheromone on the best solution, applies extra evaporation on the worst solution, and also uses mutation to encourage exploration [Cordon et al., 2000]. The hypercube ACO algorithm limits pheromone amounts to the interval  $[0, 1]$  to regularize the behavior of ACO on problems with different objectives, and to facilitate its theoretical investigation [Blum and Dorigo, 2004]. The population-based ACO maintains a population of pheromone histories rather than storing all information in a single pheromone map; it uses this population to modify the update algorithm [Guntch and Middendorf, 2002]. Beam ACO is a hybrid of ACO and beam search, which is a popular tree search algorithm [Blum, 2005b].

## 10.5 THEORETICAL RESULTS

Ever since experimental results first began to show that ACO works, researchers have been working on developing ACO theory to explain when, why, and how it works. The first convergence proofs of ACO were given in [Gutjahr, 2000]. Since then various convergence proofs for various types of ACO algorithms have been published [Dorigo and Stützle, 2004]. Most of these proofs claim something like, “Given enough time, ACO will eventually find the best solution to a combinatorial optimization problem.” Convergence results like this are mathematically interesting, but have limited practical interest. As long as the pheromone along each branch is maintained within lower and upper bounds as in the MMAS, there is always a nonzero probability of each ant exploring each possible branch of the solution space, so that given enough time, every branch will be explored. This means that eventually the optimal solution will be found. Of course, any stochastic search algorithm with a nonzero probability of searching each possible candidate solution will eventually converge. Even the simplest random search will eventually converge [Bäck, 1996].

More interesting theoretical results are along the lines of time to convergence [Gutjahr, 2008], [Neumann and Witt, 2009], probability of convergence within a given time, scalability with problem size, and descriptive mathematical models such as Markov models or dynamic system models (see Chapter 4 for mathematical models for GAs). Note that theoretical results for combinatorial problems are much different than theoretical results for continuous-domain problems. Also, if ACO could be shown to be equivalent to other optimization algorithms for which more interesting convergence proofs exist, then those convergence proofs might be able to be adapted to ACO to strengthen its theoretical foundations.

ACO has already been shown to be equivalent to the stochastic gradient ascent (SGA) and cross entropy (CE) optimization algorithms under certain conditions [Meuleau02 and Dorigo, 2002], [Zlochin et al., 2004], [Dorigo and Stützle, 2004]. SGA and CE are model-based optimization algorithms that construct solutions on the basis of a parameterized probability distribution over the search space. The evaluation of candidate solutions is used to modify the probability distribution so that it is biased toward better candidate solutions.

## 10.6 CONCLUSION

Some ACO researchers emphasize that ACO is not an algorithm but is instead a meta-heuristic because of its many variations. However, we could say that about any of the algorithms discussed in this book (GAs, EP, ES, GP, and so on); they all have many variations, and so they are all meta-heuristics. The difference between an algorithm and a meta-algorithm is one of degree, and so the difference is not black and white. Most ACO researchers emphasize that ACO is *not* an evolutionary algorithm because individuals do not exchange information with each other in the traditional sense of EAs. As we have seen in this chapter, although ACO solution construction parameters evolve over time, it is true that ACO individuals do not directly share information with each other.

Most of our discussion in this chapter has focused on trail pheromones. However, ants deposit other pheromones for purposes other than marking paths. The typical ant colony uses as many as 20 different pheromones [Hölldobler and Wilson, 1990, Chapter 7]. For example, ants deposit alarm pheromones when they are crushed. This can stimulate other ants to aggressively fight the predator that crushed their colleague [Šobotník et al., 2008]. These types of pheromones could be simulated in an ACO algorithm by having a poor solution broadcast information that discourages other individuals from repeating its poor strategy. This is similar to the negative reinforcement PSO discussed in Section 11.6.

Female ants deposit epideictic pheromones when they lay their eggs to signal other females of the same species to lay their eggs elsewhere [Gómez et al., 2005]. Animals deposit territorial pheromones to mark their territory [Horne and Jaeger, 1988]. Territorial pheromones are present in the urine of cats and dogs, which they deposit on the boundaries of their claimed territory. Animals release sex pheromones to communicate their availability for breeding [Wyatt, 2003]. Ants release recruitment pheromones to attract other ants to some place where work is required [Hölldobler and Wilson, 1990]. These types of pheromones could be simulated in ACO by having individuals broadcast information about previously explored territory in the search space to prevent other individuals from searching in regions that have already been explored, or to encourage other individuals to explore promising regions of the search domain. Ants can also release task-specific pheromones [Greene and Gordon, 2007]. This could be simulated in ACO for multi-objective optimization with different individuals pursuing the optimization of different sub-problems. We see that there are many opportunities for biologically-motivated extensions of ACO.

Additional reading about ACO can be found in books [Bonabeau et al., 1999], [Dorigo and Stützle, 2004], [Solnon, 2010]; book chapters [Maniezzo et al., 2004], [Dorigo and Stützle, 2010]; and tutorial papers [Blum, 2005a], [Blum, 2007]. Other directions for future research in ACO are similar to research priorities for other optimization algorithms [Dorigo et al., 2006]. How can ACO be applied to dynamic optimization problems for which the search space changes with time, and how can ACO be applied to stochastic optimization problems with noisy fitness function evaluations (see Chapter 21)? How can ACO be applied to multi-objective optimization problems (see Chapter 20)? How can ACO be hybridized with other evolutionary algorithms?

## PROBLEMS

Written Exercises

**10.1** Give an example in a real-world problem when the cost of traveling from node  $A$  to node  $B$  would be different than the cost of traveling from node  $B$  to node  $A$ .

**10.2** Let  $t$  be the total number of ants so that  $m_1 \approx p_1 t$  and  $m_2 \approx p_2 t$  in Equation (10.1).

- a) What are the equilibrium ratios of  $p_1/p_2$ ?
- b) Which of the equilibrium ratios are stable, and which are unstable?

**10.3** Suppose  $\beta = 1$  in the ant system of Figure 10.6. If two path segments have equal amounts of pheromone and segment 1 is half as long as segment 2, how much more likely is an ant to travel on segment 1 than segment 2? What if  $\beta = 2$ ? What if  $\beta = 3$ ?

**10.4** The ant system of Figure 10.6 sets the pheromone deposit of the  $k$ -th ant to  $\Delta\tau_{ij}^{(k)} = \delta_{ij}^{(k)} Q/L_k$ , where  $\delta_{ij}^{(k)} = 1$  if the  $k$ -th ant went from city  $i$  to city  $j$ , and  $\delta_{ij}^{(k)} = 0$  otherwise. Suppose we instead set it to  $\delta_{ij}^{(k)} \epsilon \tau_{ij}$ , where  $\epsilon$  is a tuning parameter.

- a) What range of  $\epsilon$  makes the pheromone update equation stable?
- b) What is the equilibrium value of  $\tau_{ij}$  in this case? Is this a desirable equilibrium value?

**10.5** In the standard continuous-domain AS of Figure 10.12, the  $m$ -th ant's pheromone deposit is  $\Delta\tau_{ij}^{(m)} = Q/L_m$ . Suppose we instead allow the  $m$ -th ant to deposit pheromone with probability  $p_m$ , where  $p_m$  decreases with increasing cost, as mentioned at the end of Section 10.4.1:

```

 $p_m \leftarrow \frac{1}{L_m} \sum_{r=1}^N L_r$ 
 $r \leftarrow U[0, 1]$  – that is,  $r$  is a random number uniformly distributed on  $[0, 1]$ 
If  $r < p_m$  then
     $\Delta\tau_{ij}^{(m)} \leftarrow Q_1/L_m$ 
else
     $\Delta\tau_{ij}^{(m)} \leftarrow 0$ 
End if

```

What value should we use for  $Q_1$  in the above algorithm so that the average pheromone amount deposited by the  $m$ -th ant is equal to that deposited in the standard AS of Figure 10.12?

**10.6** How does computational effort in the continuous-domain ant system of Figure 10.12 increase with the population size? How does it increase with the problem dimension? How does it increase with the number of discretized intervals in each dimension?

**10.7 Ant colony system probabilities:**

- a) Suppose we have an ACS with four cities and  $q_0 = 1/2$ . Suppose the  $k$ -th ant is in city 1, and that

$$\begin{aligned} p_{11}^{(k)} &= 0 \\ p_{12}^{(k)} &= 1/4 \\ p_{13}^{(k)} &= 1/4 \\ p_{14}^{(k)} &= 1/2 \end{aligned}$$

According to Equation (10.11), what is the probability that the  $k$ -th ant will proceed to each of the four cities? Do these probabilities sum to 1?

- b) Normalize the ACS probabilities  $\Pr(a_k \rightarrow j)$  of Equation (10.11) so that the sum from  $j = 1$  to  $n$  is 1.
- c) Use your answer to part (b) to calculate new probabilities for the scenario described in part (a). Do the new probabilities sum to 1?

**10.8** Propose a way to implement a rank-based AS such as the one mentioned in Section 10.4.3.

**10.9** Propose a way to implement a best-worst AS such as the one mentioned in Section 10.4.3.

## Computer Exercises

**10.10** This problem explores the effect of  $\beta$ , which is the heuristic sensitivity of an ant system, on AS performance. Simulate the ant system of Example 10.1 20 times, recording the best cost among all ants at each generation. Plot the average of the 20 Monte Carlo simulations as a function of generation number. Do this for  $\beta = 0.1, 1$ , and  $10$ . Discuss your results.

**10.11** Repeat Example 10.3 with  $M = 40$ . Run 20 Monte Carlo simulations for each of the following values of  $\tau_{\min}$ : 0, 0.001, 0.01, and 0.1. Plot the results. Comment on the effect of  $\tau_{\min}$  on AS performance.

**10.12** Repeat Example 10.3 with  $M = 40$ . Run 20 Monte Carlo simulations for each of the following values of  $\tau_{\max}$ : 1, 10, 100, and  $\infty$ . Plot the results. Comment on the effect of  $\tau_{\max}$  on AS performance.