

# Ant Foraging Revisited

Liviu A. Panait and Sean Luke

George Mason University, Fairfax, VA 22030  
lpanait@cs.gmu.edu, sean@cs.gmu.edu

## Abstract

Most previous artificial ant foraging models have to date relied to some degree on *a priori* knowledge of the environment, in the form of explicit gradients generated by the nest, by hard-coding the nest location in an easily-discoverable place, or by imbuing the artificial ants with the knowledge of the nest direction. In contrast, the work presented solves ant foraging problems using two pheromones, one applied when searching for food and the other when returning food items to the nest. This replaces the need to use complicated nest-discovery devices with simpler mechanisms based on pheromone information, which in turn reduces the ant system complexity. The resulting algorithm is orthogonal and simple, yet ants are able to establish increasingly efficient trails from the nest to the food in the presence of obstacles. The algorithm replaces the blind addition of new amounts of pheromones with an adjustment mechanism that resembles dynamic programming.

## Introduction

Swarm-behavior algorithms are increasingly popular approaches to clustering and foraging tasks multi-agent and robotics research (Beckers et al., 1994; Deneubourg et al., 1991). We are interested in global communication among swarm agents, and the focus of this paper is the application of pheromones to swarm foraging. Ants and other social insects use pheromones very successfully to mark trails connecting the nest to food sources (Holldobler and Wilson, 1990) and to recruit other ants to the foraging task. For example, leafcutter ants are able to organize trails connecting their nest with food sources located as far as hundreds of meters away (Holldobler and Wilson, 1990).

We are interested in the capabilities of pheromones as a guide for artificial agents and robots. In some sense pheromones may be viewed as a mechanism for inter-agent communication that can help reduce the complexity of individual agents. But pheromones cannot be viewed as essentially a blackboard architecture. While pheromones are *global* in that they may be deposited and read by any ant, and generally last long periods of time, they are *local* in that ants may only read or change the pheromone concentrations local to themselves in the environment.

However, nearly all existing pheromone models for artificial agents have eschewed a formalism such as this, concentrating instead on approaches inspired by biology if not biologically plausible. These models assume a single pheromone in order to mark locations to food, and use an ad-hoc mechanism for getting back home (for example, compass information). However, this problem is very easily represented in a symmetric fashion under the assumption of *two pheromones*, one for finding the food and one leading back to the nest. The use of multiple pheromones not only eliminates the ad-hoc nature of previous work, but also permits obstacles and other environment features.

## Previous Work

The problem discussed in this paper is known as “central place food foraging”, and consists of two phases: leaving the nest to search for food, and returning to the nest laden with food (Sudd and Franks, 1987). There are several ant foraging applications already suggested in the literature. One of the earliest ones (Deneubourg et al., 1990) concerns modeling the path selection decision making process of the Argentine ant *Linepithema humile*. Given two paths of identical lengths connecting the nest and the food source, Deneubourg *et al* show that the ants collectively choose one of the two paths by depositing pheromones over time. When paths of different lengths connect the two sites, the ant system gradually learns to forage along the shortest one. Bonabeau *et al* suggest that this happens because larger amounts of pheromones accumulate on the shorter paths more quickly (Bonabeau, 1996; Bonabeau and Cogne, 1996). Deneubourg *et al* also present a model of the foraging patterns of army ants (Deneubourg et al., 1989). Monte Carlo simulations of their model produces patterns strikingly similar to the ones observed in nature.

All models presented thus far have used a single type of pheromone. A foraging model that uses two kinds of pheromones is presented in (Resnick, 1994): ants deposit one pheromone to mark trails to the food sources, but a second type of pheromone is released by the nest itself. This second pheromone diffuses in the environment and creates a

gradient that the ants can follow to locate the nest. Resnick’s ants learn to forage from the closest food source. Moreover, when food sources deplete, the ants learn to search for other more distant food sources and establish paths connecting them to the nest, “forgetting” about the previously established trails to depleted sites. Further investigation of this model is presented in (Nakamura and Kurumatani, 1997).

The models discussed so far all make the assumption that the ants have an ad-hoc oracle — essentially a compass — which leads them back to the nest. This assumption is either stated explicitly, or it is adopted implicitly by using specific environment models and transition functions. The main justifications are not computational, but biological: ants are believed to employ sophisticated navigational techniques (including orientation based on landmark memorization and the position of the sun) to return to the nest (Holldobler and Wilson, 1990).

We know of two papers which do not rely on ad-hoc methods to return to the nest. The first such work (Wodrich and Bilchev, 1997) is similar to our own, using two pheromones to establish gradients to the nest and to a food source in the presence of an obstacle. The ants use simple pheromone addition to create trails and so the authors rely on high rates of pheromone evaporation to maintain the gradients. Another work (Vaughan et al., 2000) proposes agent behaviors using *directed pheromones* (pheromones that indicate a direction), and show successful foraging in simulation and on real robots. Aside from using directed pheromones, their work also assumes that the agents can deposit pheromones at any location in the environment, even if they are located far away from it (similar to a random-access memory).

## Method

The method described in this paper uses two different pheromones, one used to locate the food source, and the other to locate the nest. These pheromones are deposited by the ants and may evaporate and diffuse. The algorithm assumes that both types of pheromones can co-exist at the same location.

Upon leaving the nest, ants follow food-pheromone gradients to the food source, but also leave “home” (toward-nest) pheromones behind to indicate the path back home. When the ants are at the food source, they pick up a food item and start heading back to the nest. To do so, they now follow the trail of home pheromones back to the nest, while leaving a trail of food pheromones behind to mark the trail toward the food source.

Ants have one of eight possible orientations: N, NE, E, SE, S, SW, W, NW. When determining where to move, the ants first look at three nearby locations: directly in front, to the front-left, and to the front-right. We will refer to these three locations as the *forward locations*. For example, an ant with orientation N considers its NE, N and NW neighbors. Each location receives a weight based on the amounts

Parameter	Value
Duration of simulation	5000 time steps
Environment	100x100, Non-Toroidal
Nest location and size	(70,70), single location
Food source location and size	(20,20), single location
Maximum ants in simulation	1000
Maximum ants per location	10
Life time for each ant	500
Number of initial ants at nest	2
Ants borne per time step	2
Min amount of pheromone	0.0
Max amount of pheromone	1000.0
Evaporation ratio	0.1%
Diffusion ratio	0.1%
Obstacles	none
K (for Select-Neighbor)	0.001
N (for Select-Neighbor)	10.0

Table 1: Parameters for the experiments

of pheromones it contains. Ants searching for food are more likely to transition to locations with more food pheromones, while ants returning to the nest are attracted by more home pheromones. Transition decisions are stochastic to add exploration. An ant does not consider moving to a neighboring location that is too crowded (more than ten ants) or that is covered by an obstacle. If none of the three forward locations are valid, the ant then considers moving to one of the five remaining (non-forward) locations. If none of these locations are valid either, then the ant stays in its current location until the next time step.

At each location visited, ants “top off” the current pheromone level to a desired value. If there is already more pheromone than the desired value, the ant deposits nothing. The desired value is defined as the maximum amount of pheromones in the surrounding eight neighbors, minus a constant. Thus as the ant wanders away from the nest (or food), its desired level of deposited nest (or food) pheromone drops, establishing a gradient. When reaching a goal (nest or food) the desired concentration is set to a maximum value.

## The Algorithm

Foraging ants execute the Ant-Forage procedure at each time step. The pseudocode is shown next:

```

Ant-Forage
  If HasFoodItem
    Ant-Return-To-Nest
  Else
    Ant-Find-Food-Source

```

### Ant-Return-To-Nest

If ant located at food source  
 Orientation  $\leftarrow$  neighbor location with max home pheromones  
 X  $\leftarrow$  forward location with max home pheromones  
 If X = NULL  
 X  $\leftarrow$  neighbor location with max home pheromones  
 If X  $\neq$  NULL  
 Drop-Food-Pheromones  
 Orientation  $\leftarrow$  heading to X from current location  
 Move to X  
 If ant located at nest  
 Drop food item  
 HasFoodItem  $\leftarrow$  FALSE

### Ant-Find-Food-Source

If ant located at nest  
 Orientation  $\leftarrow$  neighbor location with max food pheromones  
 X  $\leftarrow$  Select-Location(forward locations)  
 If X = NULL  
 X  $\leftarrow$  Select-Location(neighbor locations)  
 If X  $\neq$  NULL  
 Drop-Home-Pheromones  
 Orientation  $\leftarrow$  heading to X from current location  
 Move to X  
 If ant located at food source  
 Pick up food item  
 HasFoodItem  $\leftarrow$  TRUE

### Select-Location(LocSet)

LocSet  $\leftarrow$  LocSet - Obstacles  
 LocSet  $\leftarrow$  LocSet - LocationsWithTooManyAnts  
 If LocSet = NULL  
 Return NULL  
 Else  
 Select a location from LocSet, where each location is chosen with probability  $(K + \text{FoodPheromonesAtLocation})^N$

### Drop-Home-Pheromones

If ant located at nest  
 Top off home pheromones to maximum level  
 Else  
 MAX  $\leftarrow$  max home pheromones of neighbor locations  
 DES  $\leftarrow$  MAX - 2  
 D  $\leftarrow$  DES - home pheromones at current location  
 If D > 0  
 Deposit D home pheromones at current location

### Drop-Food-Pheromones

Same as Drop-Home-Pheromones, but modify “home”  $\rightarrow$  “food” and “nest”  $\rightarrow$  “food source”.

	Ph. Adjust	Ph. Increment
Mean	<b>10450.70</b>	4396.04
Std. Dev.	1850.69	2835.90

Table 2: Mean and standard deviation of the food items foraged when adjusting or incrementing the amount of pheromone at the current location.

Evaporation Rate	Mean	Std. Dev.
0.0	<b>9783.38</b>	2477.07
0.001	<b>10450.70</b>	1850.69
0.01	1203.98	409.62

Table 3: Mean and standard deviation of the food items foraged when different evaporation rates are used

## Experiments

The experiments were performed using the MASON multi-agent system simulator (Luke et al., 2003). Unless stated otherwise, the experiments use the settings described in Table 1. Claims of better or worse performance are verified with a Welch’s two-sample statistical test at 95% confidence over samples of 50 runs each.

The first experiment compares the algorithm using the “top off” mechanism with one that simply deposits a fixed amount of pheromones, incrementing the amount existing at the current location. Table 2 shows the means and standard deviations of the amounts of food items collected with the two algorithms. The pheromone adjustment algorithm is significantly better.

Next, we analyzed the sensitivity of the “top off” algorithm to evaporation and diffusion rates (Tables 3 and 4). The results show that the amount of food foraged is relatively sensitive to both parameters. Lower evaporation rates do not significantly affect the results, while higher rates deplete information too rapidly for successful foraging. Diffusion is different: a zero diffusion rate significantly decreases performance. This suggests that without diffusion, pheromone information can spread to neighboring locations only through ant propagation. As expected, too much diffusion again depletes information.

We then investigated the effectiveness of the foraging behaviors at different exploration-exploitation settings. In the Selection-Location procedure,  $N$  is an exploitation factor, where larger values influence greedier decisions. The results shown in Table 5 compare the impact of the  $N$  parameter on the performance. We found that increasing the greediness led to a direct improvement in performance. We then replaced Select-Location with Greedy-Select-Location to examine extreme greediness ( $N \rightarrow \infty$ ). The revised algorithm is shown here:

Diffusion Rate	Mean	Std. Dev.
0.0	6133.2	1159.17
0.001	<b>10450.70</b>	1850.69
0.01	6162.52	1145.94

Table 4: Mean and standard deviation of the food items foraged when different diffusion rates are used

N	Mean	Std. Dev.
2	4111.3	648.02
5	8435.36	1514.27
10	<b>10450.70</b>	1850.69

Table 5: Mean and standard deviation of the food items foraged when different values for the  $N$  parameter are used in the `Select-Location` procedure. Results are significantly improving with increasing  $N$  from 2 to 5 and then 10.

```

Greedy-Select-Location(LocSet)
  LocSet ← LocSet - Obstacles
  LocSet ← LocSet - LocationsWithTooManyAnts
  If LocSet = NULL
    Return NULL
  Select from LocSet the location with max food
  pheromones

```

This increased the total number of food items returned to the nest by about three times (Table 6). However, if the algorithm uses the fixed-incrementing method, greedy selection method shows no statistically significant improvement (a mean of 4926.88, standard deviation of 2425.68 — compare to the fixed-incrementing results in Table 2). Not only does a “topping-off” method do better than more traditional approaches, but it improves with more greedy behavior.

We also observed that the foraging trails for the greedy foraging are usually optimally short and straight. Our final experiment marked a clearly suboptimal path before the simulation, and adjusted the neighborhood transition probabilities to reflect the fact that diagonal transitions are longer than horizontal and vertical ones (this adjustment was used only for this experiment). The progress of the foraging process is presented in Figure 1. The ants collectively change the path until it becomes optimal, which suggests that our foraging technique also exhibits emergent path optimization.

## Discussion

The adjustment of pheromones as expressed in this algorithm greatly resembles a dynamic programming technique, and in particular Dijkstra’s single source shortest path graph algorithm. The main difference is that the information is encoded in the environment by multiple autonomous agents via pheromones that evaporate and diffuse over time, provid-

	( $N = 10$ ) Select-Neighbor	Greedy- Select-Neighbor
Mean	10450.70	<b>28369.98</b>
Std. Dev.	1850.69	849.16

Table 6: Increasing the greediness of the algorithm. The `Greedy-Select-Neighbor` method shows an increased performance by 171%

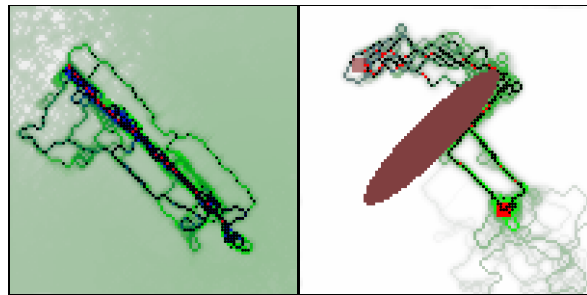


Figure 2: Ant foraging with no obstacles (left), and with an obstacle placed symmetrically on the shortest path (right)

ing a greater challenge due to increased unreliability in the information. The pheromone level adjustment algorithm is particularly useful in domains where the shortest path to the origin (nest location) is unknown. This is usually the case in environments not explored before and where strong existing pheromone information is not available.

We have also tested the behavior of the pheromone adjustment algorithm in more difficult domains in our simulator. Two examples are shown in Figure 2. Ants are represented by black dots. Shades of gray represent pheromones: the lighter the shade, the smaller the amount of pheromones.

The left snapshot in Figure 2 shows the foraging process after some 1250 time steps. The nest and food source locations are located close to the two ends of the dense center trail. There are many ants on the shortest path connecting the nest and the food source. Additionally, most space has a light gray color: ants have previously performed an extensive exploration for the food source.

The right snapshot in Figure 2 shows the foraging task in the presence of an obstacle centered on the path. The ants discover a path on one side of the obstacle and concentrate the foraging in that direction. Taken later in the run, the image shows most pheromones have already evaporated throughout the environment, suggesting that the ants are mainly exploiting the existing food source and they are not exploring for alternatives.

Figure 3 shows the same algorithm applied in an environment where the only obstacle is slightly modified such that one of its sides offers a shorter path than the other does. In an early (left) snapshot, the ants discover several paths on both sides of the obstacle. Later, the ants concentrate on the

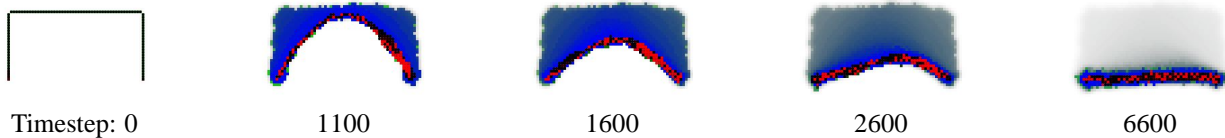


Figure 1: Foraging sequence with predefined suboptimal path, showing local path optimization.

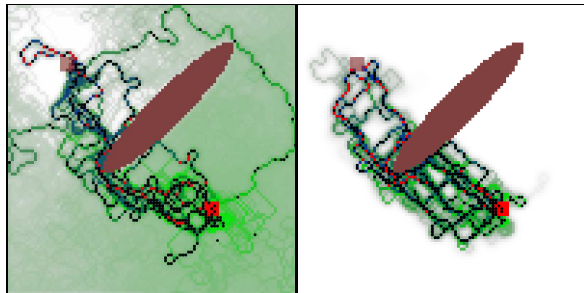


Figure 3: Ant foraging with an obstacle placed non-symmetrically: early (left) and late (right) snapshots of the simulation

shorter path and the pheromones disappear in most of the space (right snapshot). It is important to note that, because in this environment, diagonal transitions cost the same as others, most of the paths shown are in fact optimal.

Figure 4 shows the exploration with two obstacles placed such that the shortest path goes between them. Several paths are initially found, but their number decreases with time. If more time were allowed, the longer trail close to the top-right corner of the image would probably disappear.

## Conclusions and Future Work

We proposed a new algorithm for foraging inspired from ant colonies which uses multiple pheromones. This model does not rely on ad-hoc methods to return to the nest, and is essentially symmetric. We showed its efficacy at solving basic foraging tasks and its ability to perform path optimization. We then experimented with the effects of different parameter settings (degrees of evaporation and diffusion; greediness) on the rate of food transfer in the model.

For future work we plan to extend the method and apply it to other problem domains involving defense of foraging trails and nest, competition for resources with other ant colonies, and depleting and moving food sources. We are also interested in formal investigations of the properties of the model. In particular, we wish to cast the model in a form based on dynamic programming.

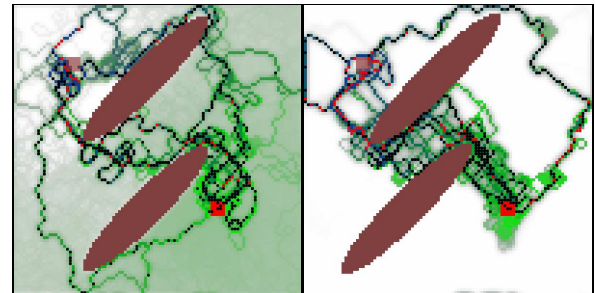


Figure 4: Ant foraging with two obstacles: early (left) and late (right) snapshots of the simulation

## Acknowledgements

We would like to thank Dr. Larry Rockwood, Elena Popovici, Gabriel Balan, Zbigniew Skolicki, Jeff Bassett, Paul Wiegand and Marcel Barbulescu for comments and suggestions.

## References

- Beckers, R., Holland, O. E., and Deneubourg, J.-L. (1994). From local actions to global tasks: Stigmergy and collective robotics. In *Artificial Life IV: Proceedings of the International Workshop on the Synthesis and Simulation of Living Systems*, third edition. MIT Press.
- Bonabeau, E. (1996). Marginally stable swarms are flexible and efficient. *Phys. I France*, pages 309–320.
- Bonabeau, E. and Cogne, F. (1996). Oscillation-enhanced adaptability in the vicinity of a bifurcation: The example of foraging in ants. In Maes, P., Mataric, M., Meyer, J., Pollack, J., and Wilson, S., editors, *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From Animals to Animats 4*, pages 537–544. MIT Press.
- Deneubourg, J. L., Aron, S., Goss, S., and Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Insect Behavior*, 3:159–168.
- Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting: Robot-like ants and ant-like robots. In *From Animals to Animats: Proceedings of the First Inter-*

- national Conference on Simulation of Adaptive Behavior*, pages 356–363. MIT Press.
- Deneubourg, J. L., Goss, S., Franks, N. R., and Pasteels, J. (1989). The blind leading the blind: Modeling chemically mediated army ant raid patterns. *Insect Behavior*, 2:719–725.
- Holldobler, B. and Wilson, E. O. (1990). *The Ants*. Harvard University Press.
- Luke, S., Balan, G. C., Panait, L. A., Cioffi-Revilla, C., and Paus, S. (2003). MASON: A Java multi-agent simulation library. In *Proceedings of Agent 2003 Conference on Challenges in Social Simulation*.
- Nakamura, M. and Kurumatani, K. (1997). Formation mechanism of pheromone pattern and control of foraging behavior in an ant colony model. In Langton, C. G. and Shimohara, K., editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 67–76. MIT Press.
- Resnick, M. (1994). *Turtles, Termites and Traffic Jams*. MIT Press.
- Sudd, J. H. and Franks, N. R. (1987). *The Behavioral Ecology of Ants*. Chapman & Hall, New York.
- Vaughan, R. T., Støy, K., Sukhatme, G. S., and Mataric, M. J. (2000). Whistling in the dark: Cooperative trail following in uncertain localization space. In Sierra, C., Gini, M., and Rosenschein, J. S., editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 187–194. ACM Press.
- Wodrich, M. and Bilchev, G. (1997). Cooperative distributed search: The ants' way. *Control and Cybernetics*, 26.