The Power of Pheromones in Ant Foraging

Christoph Lenzen, Tsvetomira Radeva
{clenzen,radeva}@csail.mit.edu

Abstract

Consider the following problem. An unknown number k of ants, initially located at the nest, searches the plane for a treasure. The identical, deterministic ants take steps asynchronously, until eventually an ant moves onto the location of the treasure. The only means to record the progress of the search is by marking each explored grid point with a pheromone. It is well-known that ants have this capability, as well as being able to determine the direction in which the nest lies.

Under these assumptions, we provide an extremely simple algorithm that is factor 5.5-optimal in terms of the worst-case number of asynchronous rounds required to find the treasure. This shows that the proposed mechanism of marking searched areas by pheromone is very powerful: Feinerman et al. proved that, without it, ants need to be given a constant-factor approximation of k to achieve asymptotic optimality, even when randomization is available.

1 Introduction

Their strength lying in numbers, ants need to be highly collaborative. Yet, despite the apparent simplicity of individual ant behavior and substantial research efforts, we lack understanding of how ants jointly solve tasks that exceed the capacity of individuals by far. Foraging, building, collecting, raising the offspring, and carrying large objects are examples of tasks that not only require a large number of ants, but must also be executed efficiently and reliably to ensure the survival of the colony.

In a recent paper, Feinerman et al. [3] approached one of these tasks, foraging for food, using distributed computing theory as a lens. The hope here is that casting the problem into a simple model and examining the performance algorithms can achieve in this abstract setting, one might identify algorithmic principles employed by the ants. More concretely, the authors argued that, according to current knowledge, little or no communication seems to take place between the foraging ants outside the nest, and therefore it should be asked how the ants coordinate to search efficiently.

Rephrasing this question, they examined how many bits of information the ants need to receive before the start of the search to find a food item hidden in the plane (*after* the auxiliary bits are determined) efficiently. If the food is at distance at most D from the nest and there are k ants, clearly $\Omega(D + D^2/k)$ rounds are required (in expectation) to find the food: It takes $\Omega(D)$ steps for an ant to move to the food, and $\Omega(D^2)$ grid points need to be searched. If k and D are known, it is straightforward to match this bound asymptotically by a randomized algorithm: each ant loops through (i) moving to a uniformly random grid point within distance at most D from the nest and (ii) doing a spiral search covering $\Theta(D^2/k^2)$ grid points. Feinerman et al. discuss how to generalize this approach by using a double loop (the outer loop increasing a guess of D by factor 2 in each step) to work for unknown D; the inner loop can be dropped at the expense of accepting to miss the treasure with arbitrarily small constant probability $\varepsilon > 0$. An advantage of such an algorithm is its robustness to ant faults; since ants are identical and do not rely on communication, even if some ants die, subsequent ants perform the search instead. However, the main result of [3] is that, for any constant $\varepsilon > 0$, any $\mathcal{O}(\log^{1-\varepsilon} k)$-competitive algorithm requires $\Omega(\log \log k)$ bits of auxiliary information. Since k

is the only relevant piece of information that is known a priori, this translates to a constant approximation of k—and the aforementioned algorithm works just as well with such an approximation.

Some aspects of the work by Feinerman et al. raise questions. First, while ants appear to be capable of estimating the size of their colonly, it is not clear whether they have a notion of the number of ants that are actually foraging. Is such a subroutine is really necessary for foraging? Second, the looping behavior of their algorithms, in particular double loops, seems somewhat unlikely to evolve without necessity, and is not backed up by observations. Considering the lower bound from [3], we believe it unlikely that such a fairly involved, yet in its early stages probably inefficient foraging strategy could have evolved if other options were available. And we know that this is indeed the case: ants are known to make use of pheromones, permitting to communicate without any direct ant-to-ant interaction.

In this paper, we provide an extremely simple foraging algorithm that works asynchronously with identical ants, based on marking visited grid points (by pheromone). Our algorithm is deterministic and 5.5-competitive, and simple modifications and use of randomization could potentially improve this ratio significantly. While the approach is lacking robustness to "faults" (e.g., ants being eaten during the search), we see it as proof-of-concept for the marking technique. It demonstrates that there is a *simple* mechanism that avoids the complexity issue arising in the model of Feinerman et al., and therefore is a more promising candidate for a naturally evolved algorithmic foraging mechanism.

Further Related Work. Other works solving ant/graph exploration fall into two main categories: theoretical work of graph exploration by single or multiple agents [1], [4], [5], and experimental results that show how different exploration strategies compare to each other in specific models [2], [6].

In the first line of work, most of the results deal with specific types of graphs: in [1] the authors consider solving the p-traveling salesman problem on a tree; in [4], the problem at hand is determining how many agents are necessary to explore a line graph; in [5], the authors provide an algorithm and lower bounds for the problem of collective exploration of a tree by multiple agents with different models of communication.

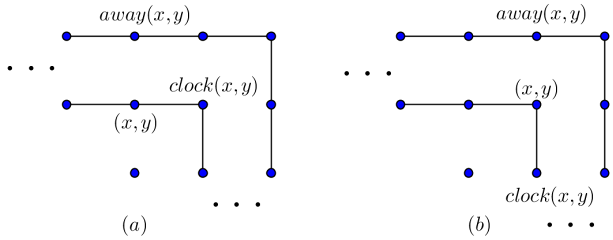
In the second line of work, the main results are established through simulation, so they do not provide precise complexity bounds. The authors of [6] evaluate the performance of four different heuristics for real-time search methods. Each of them is given by a limited local look-ahead rule which the agents use to determine their next step in the search. Other experimental results focus on other biological species, like honeybees [2], to determine the importance of communication during foraging in different situations.

2 Model

We consider an infinite two-dimensional square grid with coordinates chosen from $\mathbb{Z}^2$ with a predetermined unique vertex s (the nest). For notational convenience, we assume that $s = (0, 0)$, i.e., the origin. The grid will be explored by identical, deterministic agents (ants) by visiting the vertices of the grid. Initially, all $k \in \mathbb{N}$ ants are on vertex s. Ants can move to adjacent vertices, that is, the closest four grid points. When an ant visits a vertex on the grid, the vertex is marked as visited (the ant releases a pheromone at that spot). Each vertex of the grid may be occupied by multiple ants at the same time. We assume that the motion of the ants in the grid and the arrival of new ants proceeds asynchronously. We measure the time complexity of algorithms in terms of *asynchronous rounds*. Round $i \in \mathbb{N}$ is complete once all ants took at least r steps.

Our exploration algorithm is going to be deterministic and ants will decide on where to move next independently, based on their local views. When an ant is at a given vertex in the grid, it can sense whether the neighboring four vertices have been visited. Moreover, ants know the direction in which the nest s lies, and can distinguish "right from left", i.e., between turning clockwise and counter-clockwise. An algorithm is given by a decision rule stating the move of an ant that takes a step solely based on the above information (the order in which ants take steps is determined by an adversarial scheduler).

We hide a target (i.e., a food item) at some vertex $t = (t_1, t_2)$. We say that the food is at distance



(a) (b)

Figure 1: Vertices (x_1, y_1) and (x_1, y_2) connected by an edge are located at the same distance from the nest in the sense that $\max\{|x_1|, |y_1|\} = \max\{|x_2|, |y_2|\}$. In both figures, $clock(x, y)$ is located at the same distance as (x, y) and to the clockwise direction of it. In Figure (a), vertex (x, y) is not located in a corner, so $\text{away}(x, y)$ is located at distance one more than (x, y) from the nest. In Figure (b), vertex (x, y) is located at a corner, so there are two neighboring vertices at distance one more that the distance of (x, y) from the nest; therefore, $\text{away}(x, y)$ is defined as just one of these vertices (always chosen consistently).

$D := \max\{|t_1|, |t_2|\}$ from the nest. The goal of an exploration algorithm is to find the food as quickly as possible, i.e., move an ant onto vertex t in as few rounds as possible. Note that in order for a deterministic algorithm to guarantee the target is visited, it has to search all vertices in the grid up to distance D. Therefore, the same algorithm is also capable of locating multiple targets within distance D.

3 Algorithm

In this section, we present a very simple algorithm and we show that it performs optimally in terms of the number of rounds needed to reach vertex t. Informally, at each step, each ant moves clockwise, if the respective vertex has not been visited yet, and otherwise moves away from the nest. The pseudocode for the algorithm is presented in Figure 1. The definition of the functions $\text{clock}(x, y)$ and $\text{away}(x, y)$ is illustrated in Figure 1. We remark that the precise definition of $\text{away}(x, y)$ is irrelevant, as long as the distance from the nest is exactly by one larger than the one of (x, y).

Algorithm 1: Ant exploration algorithm for an ant located at vertex (x, y)

```
if $\text{clock}(x, y)$ is not visited then move to $\text{clock}(x, y)$ else move to $\text{away}(x, y)$;
```

Analysis. We denote vertex (x, y) as *spearhead*, iff (x, y) is visited but $\text{clock}(x, y)$ is not.

Lemma 1. *At each time, all spearheads are occupied by ants.*

Proof. Suppose (x, y) is a spearhead. Thus, (x, y) has been visited by some ant. If this ant has not taken another step since it moved to (x, y), the claim of the lemma holds for this spearhead. Hence, assume for contradiction that the ant took another step. It either moved to $\text{clock}(x, y)$, thereby visiting $\text{clock}(x, y)$, or $\text{clock}(x, y)$ was already visited when it took its step. In both cases, (x, y) is not a spearhead anymore. $\square$

Lemma 2. *If an ant is at distance D from the nest, at most $8D$ rounds later all vertices at distance at most D from the nest have been visited.*

Proof. If an ant is at distance D from the nest, at least one vertex of each layer $1 \leq d \leq D$ has been visited. Clearly, each such layer that has not been fully explored yet contains a spearhead. By Lemma 1, on each spearhead (x, y) we have an ant. After one asynchronous round, $\text{clock}(x, y)$ will be visited (either because the ant on the spearhead moves there or another one does first). Hence, in each layer $d \leq D$ that is not fully explored yet, in each round at least one new vertex will be explored. As layers at distance $d \leq D$ from the nest have at most $8D$ vertices, the claim of the lemma follows. ☐

Theorem 3. *The number of rounds needed for the algorithm to terminate is smaller than* $9D+4D(D+1)/k$. *This is optimal up to factor* 5.5.

Proof. Initially, the number of unvisited vertices at distance at most D from the nest is exactly $\sum_{d=1}^{D} 8d = 4D(D+1)$. In each asynchronous round, each ant increases its distance from the nest or explores a new vertex (and does not decrease its distance from the nest). Therefore, after $D+4D(D+1)/k$ rounds, there is an ant at distance at least D from the nest. By Lemma 2, at most $8D$ asynchronous rounds later, all vertices at distance D from the nest are explored. Hence the treasure must have been found.

To show the bound on the competitive ratio, observe that, in the worst case, it takes at least (i) $2D$ asynchronous rounds to reach, e.g., vertex (D, D) and (ii) $\lceil 4D(D+1)/k \rceil$ rounds to explore all vertices at distance at most D from the nest. If $2D \geq 4D(D+1)/k$, then $9D + 4D(D+1)/k \leq 11D = 11/2 \cdot 2D$. Otherwise, $9D + 4D(D+1)/k \leq 11/2 \cdot 4D(D+1)/k$. ☐

We remark that requiring ants to also explore the grid in counter-clockwise direction reduces the competitive ratio to 3.5, but requires more careful reasoning. Moreover, the arguments for the lower bound are simplistic. The type of algorithms we consider is restricted to memoryless agents with a local view, which entails that (except for the first few ants) randomization or some other means of symmetry breaking is necessary to send the ants into different directions from the nest. We conjecture that without such symmetry breaking information, it is not possible to improve the competitive ratio further.

4 Further Research Questions

So far we have only considered simple deterministic algorithms. If we use randomization together with the ants' capability to communicate through pheromones, it may be possible to achieve an even better competitive ratio. Our conjecture is that if ants determine the initial direction to leave the nest randomly and then move both clockwise and counter-clockwise, we can achieve an near-optimal competitive ratio.

In addition to the simple model described in this paper, there are many further issues to consider.

- Use of multiple pheromones in concurrent or repetitive searches (e.g., to increase robustness).
- Obstacles in the plane.
- Removal of agents during the algorithm.
- Ants may start the search at different times.
- Ants may refuse to move to occupied grid points.
- Multiple targets appear over time.

References

[1] Igor Averbakh and Oded Berman. $(p-1)(p+1)$-approximate Algorithms for p-traveling Salesmen Problems on a Tree with Minmax Objective. *Discrete Applied Mathematics*, 75(3):201–216, 1997.

[2] Peter Bailis, Radhika Nagpal, and Justin Werfel. Positional Communication and Private Information in Honeybee Foraging Models. In *Proc. 7th Conference on Swarm Intelligence (ANTS)*, pages 263–274, 2010.

[3] Ofer Feinerman, Amos Korman, Zvi Lotker, and Jean-Sebastien Sereni. Collaborative Search on the Plane without Communication. In *Proc. 31st Symposium on Principles of Distributed Computing (PODC)*, pages 77–86, 2012.

[4] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. How Many Oblivious Robots Can Explore a Line. *Information Processing Letters*, 111(20):1027–1031, 2011.

[5] Pierre Fraigniaud, Leszek Gasieniec, Dariusz R. Kowalski, and Andrzej Pelc. Collective tree exploration. In *Proc. 6th Latin American Theoretical Informatics Symposium (LATIN)*, pages 141–151, 2004.

[6] Sven Koenig and Yaxin Liu. Terrain Coverage with Ant Robots: a Simulation Study. In *Proc. 5th Conference on Autonomous Agents (AGENTS)*, pages 600–607, 2001.