A MODEL AND ANALYSIS OF HOUSE-HUNTING IN ANT COLONIES

EMILY Y. ZHANG*, JIAJIA ZHAO, AND NANCY LYNCH

Massachusetts Institute of Technology
*eyzhang@mit.edu

Abstract. We study the problem of house-hunting in ant colonies, where ants reach consensus on a new nest and relocate their colony to that nest, from a distributed computing perspective. We propose a house-hunting algorithm that is biologically inspired by *Temnothorax* ants. Each ant is modelled as a probabilistic agent with limited power, and there is no central control governing the ants. We show a $\Omega(\log n)$ lower bound on the running time of our proposed house-hunting algorithm, where n is the number of ants. Further, we show a matching upper bound of expected $O(\log n)$ rounds for environments with only one candidate nest for the ants to move to. Our work provides insights into the house-hunting process, giving a perspective on how environmental factors such as nest qualities or a quorum rule can affect the emigration process. In particular, we find that a quorum threshold that is high enough causes transports to the inferior nest to cease to happen after $O(\log n)$ rounds when there are two nests in the environment.

1. Introduction

Recently, there has been an interest in the distributed computing community on studying biologically-inspired algorithms [1]. Tissues found within the human body and insect colonies of ants and bees are good examples of naturally occurring systems where there are many agents with limited power, a global goal, and no central control. Interestingly, an ant colony as a whole exhibits a high level of collective intelligence and is able to achieve global goals, such as relocating to new nests. It is puzzling how the distributed system is able to quickly reach consensus through local communications, especially given the high noise levels observed in nature.

The house-hunting process in ant colonies is a naturally occurring algorithmic task that is closely related to consensus, a fundamental problem in distributed computing theory. The goal of the ants is to relocate the colony of ants to a new nest with superior quality. During the house-hunting process, a colony is able to reach consensus on a new nest and execute the move of the entire colony, even though each individual actively-scouting ant has information about only a small subset of the new candidate nests.

In 2015, Ghaffari et al. [3] modeled the ant colony house-hunting process as a distributed algorithm on independent random agents. They also showed theoretical guarantees on the number of rounds required for various house-hunting algorithms under their model to converge. Recently, Zhao et al. [13] developed a simulator that closely mimics the ants' behaviors on both individual and colony levels. The simulator is based on the agent-based model of ants' house-hunting process that Pratt et al. [10] created by studying videotaped behavior of ants. Zhao et al. showed that their simulator is biologically plausible in that it accurately reflects many behaviors observed in real ant colonies, and their simulator is also useful for predicting some of the behaviors of ants that are harder for biologists to directly study in experiments. The algorithm presented by Zhao et al. is more biologically plausible than those studied by Ghaffari et al., but one remaining challenge is that there are no theoretical bounds on the convergence speed of the algorithm by Zhao et al.

We present an algorithm for house-hunting that is structured like the model considered by Ghaffari et al. [3] and takes into account the different behavior of ants in different phases observed by biologists Pratt et al. [10] in ant labs and modelled by Zhao et al. [13] in computer simulations. Thus, our algorithm is biologically plausible while still tractable to rigorous analysis. Our model is a mathematical agent-based model, with transition probabilities derived from the data that Pratt et al. collected in their observations of ant colonies and from computer simulations run by Zhao et al. We show that the theoretical guarantees on the running time of our algorithm are similar to those of the algorithms considered by Ghaffari et al.

Our work has many implications for both the biology community and the computer science community. Natural algorithms have evolved over time to have many advantageous properties. For example, algorithmic tasks carried out by collections of living beings are usually highly adaptive to different types of environments, robust to noise, and also optimized in terms of their speed and accuracy. Thus, insights from these biological algorithms can inspire more robust, efficient algorithms for distributed computer systems, such as robot

swarms [5]. Additionally, using mathematical tools to analyze the house-hunting algorithm can allow for a better understanding of the qualities of ant colonies that are harder for biologists to directly observe, such as the dependence on various environmental parameters.

1.1. **The House-Hunting Process.** *Temnothorax* ants often search for and move to new nests, as living in favorable nests is important to the survival of their colony. Their moving process is highly distributed, as each individual ant has limited power, and there is no central control governing the emigration process.

Ant colonies are typically composed of roughly equal populations of active and passive ants. Active ants execute the emigration while passive ants, such as brood items or inactive adult ants, are transported to new nests by active ants.

Biologists have observed that the house-hunting process involves several stages. Active ants search for nests, assess nests, recruit other ants, and also transport other ants. Once an active ant has found a new nest of satisfactory quality, it moves on to the recruitment phase, where it recruits other active ants to the new nest via *tandem runs* [6, 11]. Should the population of active ants in a new nest surpass a *quorum threshold*, then active ants will commit to the new nest and begin transporting (i.e. picking up and carrying) other ants from the old nest to the new nest [9]. These transports speed up emigration to the new nest.

2. Model

We present a model of *Temnothorax* ants' house-hunting process that is both tractable to analysis and biologically plausible. This algorithm is primarily inspired by the agent-based model for house-hunting in ant colonies introduced in [13]. Like the model in [13], our algorithm is parameterized by many constants, which can be tuned to reflect the changing environmental conditions and varied behaviors of ants observed in nature. Our model differs in that we reduce the number of internal variables stored for each ant and the number of possible states that the ants can be in, thus simplifying the rules for how ants change locations. These simplifications make the model tractable for proofs of theoretical guarantees.

In our algorithm, active and passive ants in the colony play different roles in the emigration process. Active ants transition through many states, including searching for a new nest, evaluating a nest, and recruiting other ants to the nest by lead forward or transport runs. Passive ants, on the other hand, change location only when transported by an active ant. The biological insights for these design decisions come from [10]. Each ant changes both state and location at most once on each discrete time step, or round.

2.1. **Framework.** The environment consists of at least two nests, one of which is the original home of the colony. Each nest has an associated *quality*, a nonnegative real number. The ants are modeled as identical finite state machines that execute computations synchronously in rounds. In each round, an active ant performs at most one call to each of the functions **select_action**(), **select_ant**(), and **transition**(), which are defined in Section 2.2.

We let n denote the total number of ants, and we let n_a and n_p denote the number of active ants and passive ants, respectively, with $n_a = \theta(n)$, $n_p = \theta(n)$, and $n_a + n_p = n$. The location of an ant a is denoted $a.location$, which is one of the nests in the environment. Every active ant a has an associated state, denoted $a.state$, which is one of 9 possible states: At Nest_i, Search_i, Quorum Sensing, Lead Forward, and Transport, for $i \in \{E, C, T\}$. The subscripts E, C, and T stand for Exploration, Canvassing, and Transport, three different phases of active ants described in [10]. For every ant a, the value of $a.state$ begins as At Nest_E and the value of $a.location$ begins as the original home nest of the colony. These values are updated by calls to the helper function **transition**().

Like the model from [13], our model is further parameterized by adjustable constants. The parameters μ_q and μ_p are the *quality coefficient* and *population coefficient*, respectively. They represent the relative weight that ants give to the quality and population of a nest when evaluating that nest. We denote by θ the *quorum threshold*, or the proportion of active ants that must be in a nest before an active ant can commit to that nest and begin transporting ants to that nest. The constants c_s, c_f, c_ℓ, and c_t denote the *search constant*, the *follow constant*, the *lead forward constant*, and the *transport constant*. These constants parameterize the probability that the corresponding type of action succeeds. Finally, λ controls for how noisy individual ants' decision-making is, with higher λ values corresponding to lower individual noise level. All constants other than λ range between 0 and 1; λ ranges from 1 to 16 [13].

2.2. **Helper Functions.** This subsection defines the helper functions that are called in the house-hunting Algorithm, which is given in Section 2.3.

select_action(**a**): The input is an ant a. Let n' be a nest chosen uniformly at random from all of the nests other than $a.location$. Let q be the quality of $a.location$, and let p and p_a be the number of ants and active ants in that nest, respectively. Finally, let q' and p' denote the quality and population of nest n'. The ant probabilistically chooses an action to take: *advance* or *hold*. The probability distribution over these actions depends on the state transition probabilities given in (1):

$$
\begin{aligned}
&\Pr[\text{At Nest}_i \to \text{Search}_i] = 1 - \left(1 + e^{-\lambda\left(\mu_q \cdot q + \mu_p \cdot \frac{p}{n}\right)}\right)^{-1} \quad \text{for } i \in \{\text{E, C, T}\} \\
&\Pr[\text{Search}_i \to \text{At Nest}_i] = c_s \cdot \left(1 + e^{-\lambda\left(\mu_q \cdot (q'-q) + \mu_p \cdot \frac{p'-p}{n}\right)}\right)^{-1} \quad \text{for } i \in \{\text{E, C, T}\} \\
&\Pr[\text{Quorum Sensing} \to \text{Transport}] = \begin{cases} 1 & \text{if quorum has been met, that is } p_a > \theta \cdot n_a \\ 0 & \text{otherwise} \end{cases} \\
&\Pr[\text{Transport} \to \text{At Nest}_\text{T}] = c_t \\
&\Pr[\text{Lead Forward} \to \text{At Nest}_\text{C}] = \begin{cases} c_\ell & \text{if } q' > q \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

The function **select_action**(**a**) returns (advance, n') if the ant a probabilistically chooses to transition along a solid arrow in the state transition diagram (Figure 1); otherwise, this function returns (hold, n').

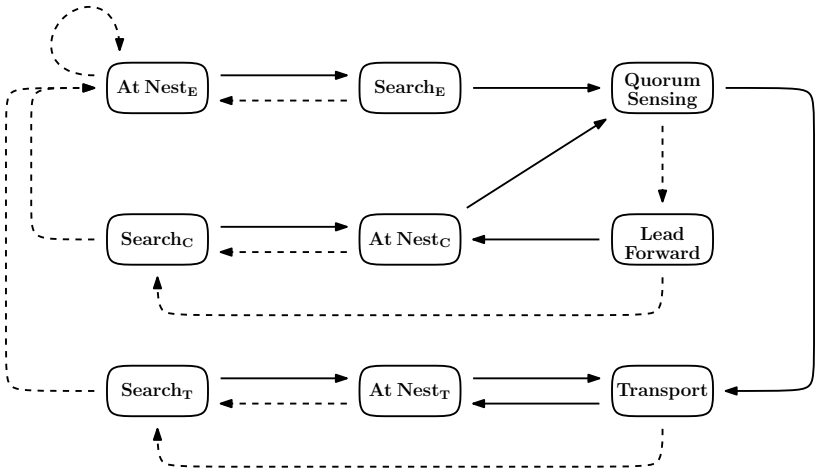

FIGURE 1. The State Transition Diagram. The solid arrows denote the ant choosing to *advance* — that is, make progress towards emigration. In contrast, the dashed arrows denote the ant choosing to *hold*. The transition probabilities are given in (1).

select_ant(**a**, **n′**, **action**): This function takes as input an ant a, a nest n', and an action *action*. If *action* is hold, then this function immediately returns *null*. First, we set a' to null. If $a.state$ is Lead Forward and there is at least one active ant in nest n', then let a' be an active ant chosen uniformly at random from n'. If $a.state$ is Transport and there is at least one passive ant in nest n', then let a' be a passive ant chosen uniformly at random from n'; if there are only active ants in n', let a' be an active ant chosen uniformly at random from n'. We return a' with probability c_f and *null* otherwise. Note that the returned ant cannot be a because a is not in nest n'.

transition(**a**, **a′**, **n′**, **action**): This function takes as input ants a and a', a nest n', and an action *action*. If $a.state$ is Search$_i$ for any $i \in \{E, C, T\}$ and *action* is advance, then we set $a.location$ to n'. If $a.state$ is Lead Forward or Transport and *action* is advance, then we set $a'.location$ to $a.location$. We set $a.state$ to the state obtained by starting from $a.state$ and following the arrow corresponding to *action* in the state transition diagram (Figure 1). If a' is not null, we set the state of ant a' to At Nest$_E$.

2.3. **Algorithm.** With the helper functions defined, we are ready to present the house-hunting algorithm. Let P be a permutation of all of the active ants chosen uniformly at random. Algorithm 1 shows one round of the house-hunting algorithm. We repeat the procedure given in Algorithm 1 until the house-hunting algorithm *converges*. The house-hunting algorithm converges when all of the passive ants have moved to a different nest with superior quality, called the *winning nest*.

In Algorithm 1, the set M serves to make sure that each ant transitions at most once: either actively by initiating an action or passively by getting recruited. If there are conflicts with the recruitment actions of advancing from lead forward or transport, then P serves as a tie breaker between the conflicting actions.

Algorithm 1: One Round of the HouseHunting Algorithm

```
M: a set of ants, initially ∅
for i = 1 to |P| do
    if a_{P(i)} ∉ M then
        action, n' := select_action(a_{P(i)})
        a' := select_ant(a_{P(i)}, n', action)
        if a' ∈ M then
            a' ← null
        transition(a_{P(i)}, a', n', action)
        M := M ∪ {a_{P(i)}} ∪ {a'}
```

3. Our Results

In [3], Ghaffari et al. proved a lower bound of $\Omega(\log n)$ on the running time of their house-hunting algorithm. Their proof depended on a lemma that stated that all ants in an inferior nest do not find the winning nest with at least constant probability during each round; we showed that this result is true for $\frac{n_p}{2}$ of the passive ants. The discrepancy by a constant factor does not affect the asymptotic behavior of the algorithm. Thus, the reasoning from the proof of Theorem 3.2 from [3] implies the following theorem:

Theorem 3.1. *If $n_p \geq 4$, then for any constant $c > 0$, our proposed house-hunting algorithm requires $\Omega(\log n)$ rounds for all of the passive ants to move from the original home nest to the winning nest with probability at least $\frac{1}{n^c}$.*

For the rest of this section, we consider an environment with only two nests: Nest 0 (with quality q_0), and Nest 1 (with quality $q_1 > q_0$). All of the ants start in Nest 0 in the At_Nest$_E$ state. Since an active ant will change locations if it advances from a Search state, the frequency at which ants visit Search states is of interest to us.

Definition 3.2. For every integer $k > 0$, random variable $R_0^{(k)}$ is the fraction of active ants in Nest 0 that are in a Search state at the beginning of round k if Nest 0 has active ants at the beginning of round k; and 0 otherwise. $R_1^{(k)}$ is similarly defined for Nest 1. Let $f_0 := \min_k \mathbb{E}\left[R_0^{(k)}\right]$ and $f_1 := \max_k \mathbb{E}\left[R_1^{(k)}\right]$

We do not explicitly compute f_0 and f_1 since this would require intensive computations. Instead, we make use of the following useful property:

Lemma 3.3. *We have $f_0, f_1 > \epsilon$ for some constant $\epsilon > 0$ that is independent of n.*

We use the values f_0, f_1 to state our second result, an upper bound on the expected number of rounds that the house-hunting algorithm takes to converge when the environment consists of two nests. Recall from Section 2 that we say that the algorithm converges when all of the passive ants are in the winning nest.

Theorem 3.4. *Let*

$$a(\epsilon) := n_a \left(\frac{\frac{f_0}{f_1} - \epsilon}{\frac{f_0}{f_1} + e^{\alpha}} \right),$$

where $\alpha = -\lambda(\mu_q(q_1 - q_0) - \mu_p)$ and $\epsilon > 0$ is any small constant less than $\frac{f_0}{f_1}$.

Let random variable R_ϵ denote the number of rounds required for at least $a(\epsilon)$ of the active ants and all of the passive ants to move from Nest 0 to Nest 1.

If the quorum threshold satisfies $1 - \frac{a(\epsilon)}{n_a} < \theta < \frac{a(\epsilon)}{n_a}$, then $\mathbb{E}[R_\epsilon] = O(\log n)$.

The lower bound condition on the quorum threshold in Theorem 3.4 allows us to prove that there will be no more backward transports (transports from the winning nest to the inferior nest) after expected $O(\log n)$ rounds. There is a lot of work in the biology community studying the role of the quorum threshold in the house-hunting process. As we see with our result, mathematical analyses such as ours can provide possible explanations for why ant colonies have evolved to use a quorum threshold.

References

[1] *Biological Distributed Algorithms (BDA) 2019*, snl.salk.edu/∼navlakha/BDA2019/.

[2] Nigel R. Franks, Jonathan P. Stuttard, Carolina Doran, Julian C. Esposito, Maximillian C. Master, Ana B. Sendova-Franks, Naoki Masuda, and Nicholas F. Britton. How ants use quorum sensing to estimate the average quality of a fluctuating resource. *Scientific Reports*, **5(1)** (2015), 1–12.

[3] Mohsen Ghaffari, Cameron Musco, Tsvetomira Radeva, and Nancy Lynch. Distributed house-hunting in ant colonies. *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, (2015), 57–66.

[4] Simone M. Glaser and Christoph Grüter. Ants (temnothorax nylanderi) adjust tandem running when food source distance exposes them to greater risks. *Behavioral Ecology and Sociobiology*, **72(3)** (2018), 1–8.

[5] Michael J. B. Krieger, Jean-Bernard Billeter, and Laurent Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, **406(6799)** (2000), 992–995.

[6] Michael Möglich. Social organization of nest emigration in Leptothorax (Hym., Form.). *Insectes Sociaux*, **25(3)** (1978), 205–225

[7] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. *In Proceedings 41st Annual Symposium on Foundations of Computer Science*, IEEE, (2000), 565–574.

[8] Stephen C. Pratt. Efficiency and regulation of recruitment during colony emigration by the ant temnothorax curvispinosus. *Behavioral Ecology and Sociobiology*, **62(8)** (2008), 1369–1376.

[9] Stephen C. Pratt, Eamonn B. Mallon, David J. Sumpter, and Nigel R. Franks. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis. *Behavioral Ecology and Sociobiology*, **52(2)** (2002), 117–127.

[10] Stephen C. Pratt, David J. Sumpter, Eamonn B. Mallon, and Nigel R. Franks. An agent-based model of collective nest choice by the ant Temnothorax albipennis. *Animal Behaviour*, **70(5)** (2005), 1023–1036.

[11] Thomas O. Richardson, Philippa A. Sleeman, John M. McNamara, Alasdair I. Houston, and Nigel R. Franks. Teaching with evaluation in ants. *Current biology*, **17(17)** (2007), 1520–1526.

[12] Lili Su, Martin Zubeldia, and Nancy Lynch. Collaboratively learning the best option on graphs, using bounded local memory. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **3(1)** (2019), 1–32.

[13] Jiajia Zhao, Nancy Lynch, and Steven Pratt. The Power of Social Information in Ant-Colony House-Hunting: A Computational Modeling Approach. *bioRxiv*, (2021), 2020–10.