New Perspectives on Algorithmic Robustness Inspired by Ant Colony House-Hunting (Extended Abstract)[*]

Tsvetomira Radeva
MIT
radeva@csail.mit.edu

Cameron Musco
MIT
cnmusco@mit.edu

Nancy Lynch
MIT
lynch@csail.mit.edu

June 12, 2017

Abstract

In past work, we presented an abstract model for the house-hunting process, in which *Temnothorax* ant colonies search for, decide on, and move to a new nest site when their current nest becomes unsuitable [5]. We gave an optimal algorithm under our model in terms of runtime – a colony of n ants can chose between $k < n$ potential nest sites and relocate to the chosen site in $O(\log n)$ rounds of computation, matching an $\Omega(\log n)$ round lower bound.

However, this optimal algorithm is not very 'natural'. It requires ants to count the number of other ants in a given nest site exactly, and completely breaks down when population estimates are computed noisily. In light of this, we proposed a more natural algorithm which used the number of ants in a candidate site to obtain a probability of recruiting to this site. Intuitively, this algorithm seems much less sensitive to noisy population estimates.

In this work, we formalize this intuition, showing that our simple house-hunting process is robust to noise. Even when ants can only obtain noisy population measurements, our algorithm allows an n ant colony to decide between k candidate nests in $O(k^3 \log^{1.5} n)$ rounds, nearly matching our lower bound in the natural case that $k << n$.

Our model of noise is very general – our algorithm works given population estimates that are drawn from *any distribution* which is bounded in a reasonable range and correct in expectation (i.e., unbiased). We demonstrate, for example, that randomized algorithms for population estimation in ant colonies [10] produce estimates satisfying our requirements, and thus can be used as subroutines in our house-hunting algorithm. We hope that this approach to demonstrating algorithmic robustness using a 'probabilistic adversary' which can draw values from any of a broad class of distributions, is useful in studying distributed algorithms in engineered systems operating in noisy or fault prone environments.

[*] This extended abstract is an overview of work largely contained in Tsvetomira Radeva's Ph.D. thesis [14], which can be accessed at http://groups.csail.mit.edu/tds/papers/Radeva/Radeva-phdthesis.pdf.

1 Introduction

In this work we give an example of how studying models of biological computation and natural algorithms allows us to better understand distributed algorithm design. Through theoretical investigation of house-hunting in ant colonies, we develop new methods for proving the robustness of distributed algorithms which must handle noisy inputs, imprecise subroutines, or random failures.

1.1 House-Hunting in Ant Colonies

House-hunting is the process through which some species of ants (and bees) choose a new nest for the colony to move to when their current nest is destroyed or otherwise becomes unsuitable. We focus on *Temnothorax* ants, whose house-hunting process has been studied extensively [4, 11, 13].

In the first phase of house-hunting, some ants begin exploring their surroundings searching for possible new nests and assessing the quality of these nests [6, 15, 16]. After an ant becomes satisfied with some nest, it moves on to the recruitment phase, which consists of *tandem runs* – an ant leading another ant from the old to a new nest. The recruited ant learns the candidate nest location and can assess the nest itself and begin performing tandem runs if the nest is acceptable.

At this point many nest sites may have ants recruiting to them, so a decision has to be made in favor of one nest. From a distributed computing perspective, this decision-making phase of house-hunting is closely related to the fundamental problem of consensus [3, 8]. It is believed that *Temnothorax* ants solve consensus using a *quorum threshold* [11, 12] – once a large enough population of ants is detected at a candidate nest, the ants recruiting to that nest decide that it should be chosen as the new home. They then complete the house-hunting process, transporting the rest of the colony to the nest by physically picking up individual ants and carrying them there.

1.2 Prior Work on House-Hunting Algorithms

In earlier work [5], we presented a model for *Temnothorax* house-hunting, which we overview in Section 2. Briefly, we consider synchronous computation, where, in each round, an ant may perform one of a few basic tasks – search for a new nest and assess its quality, go to a previously visited nest, or attempt to recruit other ants to a nest via tandem running. For simplicity we assume that all nests have binary qualities – i.e., are either acceptable or unacceptable. This seems to be the hardest case – any variation in quality makes breaking symmetry between candidate sites easier.

We showed that no strategy can solve house-hunting (ensure that all n ants in the colony are located in a single acceptable nest) in $o(\log n)$ rounds in our model, and we give an optimal algorithm which uses $\mathcal{O}(\log n)$ rounds. Unfortunately, this algorithm relies heavily on the ability of the ants to detect very small population changes in nests between rounds, which is unrealistic. It is believed that nest population is estimated via the number of physical encounters ants have with each other over a certain time interval, an inherently noisy approximation [11].

To handle this issue, in [5] we presented an alternative algorithm that solves house-hunting in nearly optimal time, yet seems to be more noise resilient. In each round, an ant that has located a candidate nest recruits other ants to the nest with probability proportional to the nest's current population. Thus, nests with higher populations are recruited to with higher probability, and this positive feedback eventually leads to a single distinguished nest to which all ants are recruited. Since nest populations are used simply as recruiting probabilities, rather than as rigid decision mechanisms, intuitively, this algorithm seems more robust to noise in population estimation.

1.3 Our Results: Robust Algorithm for House-Hunting

In this work, we make this intuition rigorous. We first give a novel proof that our alternative algorithm solves house-hunting with k candidate nests in $\mathcal{O}(k^3 \log^{1.5} n)$ rounds. This proof uses tools developed in studying the population protocol model of primitive distributed computing [2].

We then demonstrate that this proof still holds when ants base their recruiting probabilities on noisy estimates of the candidate nest populations in each round. Specifically, our algorithm still succeeds when population estimates are drawn from *any distribution* satisfying a few simple assumptions. For a nest with population p, we require that for some $0 < \epsilon < 1$ and $c > 2$, the estimated population is in the range $[(1-\epsilon)p, (1+\epsilon)p]$ with probability at least $(1 - 1/n^c)$ and that it never exceeds the colony size n; moreover, we assume the population estimate is *correct in expectation* (i.e., unbiased). Under these assumptions, we show that the algorithm's running time increases by a factor of $\mathcal{O}(1/(1-\epsilon)^2)$ compared to the case of no uncertainty, and the probability of solving the problem within this time decreases by $1/n^{c-2}$. Thus, even if $\epsilon = 1/2$ or some other large constant, our algorithm still runs in $\mathcal{O}(k^3 \log^{1.5} n)$ rounds and succeeds with high probability.

The above result allows us to combine our house-hunting algorithm with theoretical work on encounter-rate-based population density estimation in ant colonies [10]. We show that an encounter-rate-based algorithm can be used as a black-box subroutine to estimate the number of ants at each candidate nest during the house-hunting process, giving correctness and efficiency guarantees nearly matching what is achievable when we assume that nest populations can be measured exactly.

1.4 Insights into Robustness of Distributed Algorithms

Algorithmic robustness is often studied using *adversarial models* [1, 7, 9]. While noise and fault occurrences may be probabilistic and governed by a distribution, it is often impossible to determine this distribution or predict its basic properties. Thus, one assumes that noise or faults are selected by an adversary, who chooses the outcomes that make computation most difficult. The adversary, e.g., may choose noisy population estimates from an arbitrary distribution. They may also use more general strategies, including those that do not involve an underlying distribution at all.

Surprisingly, our house-hunting algorithm is *not robust* to a very simple adversary – one that, for a nest with population p is guaranteed to return an estimate $\tilde{p} \in [(1-\epsilon)p, (1+\epsilon)p]$ for some small ϵ to each ant. When there are two candidate nests with nearly equal populations, this adversary can give all ants the same population estimate in each round. The ants in both nests thus recruit with identical probabilities, and with good probability, the nest populations remain nearly equal.

As the results discussed in Section 1.3 demonstrate, this counterexample breaks when we consider a slightly more constrained adversary. This adversary *must chose population estimates from some distribution* which not only approximates p with high probability, but is also correct in expectation. This expectation constraint is key – roughly, in the case of two nests, even small differences in the populations force the adversary to draw their population estimates from slightly different distributions, and eventually positive feedback kicks in and the nest populations diverge.

The use of such a 'probabilistic adversary' allows us to balance the goal of adversarial analysis of being flexible to many models of noise, with the recognition that some very mild distributional assumptions (such as correctness in expectation) may allow for significantly stronger bounds. We are hopeful that this approach will be useful for studying distributed algorithms in engineered systems, such as in wireless network and message passing settings.

In house-hunting, the adversary determines population estimates, which in turn determine recruitment probabilities. In general, we hope that the idea of characterizing the robustness of randomized algorithms by their ability to tolerate perturbations in probabilities used can be extended to other settings. This seems especially important in biological systems, where individuals are believed to have limited access to true randomness and inaccurate estimates of environmental parameters, both of which can imply large perturbations of the intended probabilities used.

2 Basic Model and Algorithm

We briefly overview the model of house-hunting defined in [5] and [14]. The environment consists of a home nest and k candidate nests $\{1, ..., k\}$. Each candidate i has quality $q(i) \in \{0, 1\}$; 0 indicates an unsuitable nest, and 1 indicates a suitable one. We assume that $q(i) = 1$ for at least one i.

The colony consists of n identical probabilistic finite state machines, representing the ants. We assume n is somewhat larger than k (i.e., $k = \mathcal{O}(n/\log n)$) and that the ants do not know the value of k but do know the value of n.[1] The general behavior of the state machines is unrestricted, but their interactions with the environment and with other ants are limited to a few high-level functions defined below. We assume a synchronous model of execution, starting with all ants located at the home nest. In each round r, each ant performs a call to exactly one of the following:

- **search**(): The ant searches for a nest. The function returns a nest index $i \in \{1, \cdots, k\}$ of a uniformly random candidate nest, the nest's quality $q(i)$, and the number of ants currently in nest i.
- **go**($\mathbf{i}$): The ant revisits a candidate nest i. The function returns the number of ants currently in i.
- **recruit**($\mathbf{b}, \mathbf{i}$): This function takes a boolean $b \in \{0, 1\}$ and a nest index $i \in \{1, \cdots, k\}$, and returns a pair (j, c_j), where j is a nest index and c_j is the number of ants currently in nest j. If $b = 0$, the ant is waiting to be recruited by others, and if $b = 1$, the ant is attempting to recruit to i. j indicates the nest that the ant ends up at either after successfully recruiting to nest i or being recruited to a new nest. The recruitment process is randomized. Each ant with $b = 1$ chooses a uniformly random ant also calling **recruit**$(\cdot, \cdot)$; after all ants make their choices, we randomly resolve any recruitment conflicts (e.g. two ants trying to recruit each other). For a detailed definition, see [5].

We say that an algorithm $\mathcal{A}$ *solves the* HouseHunting *problem* with k nests in T rounds with error probability δ, if with probability at least $1 - \delta$, taken over all executions of $\mathcal{A}$, there exists a nest $i \in \{1, \cdots, k\}$ such that $q(i) = 1$ and all ants are located at nest i and for all times $r \geq T$.

2.1 House-Hunting Algorithm

Algorithm 1 describes the house-hunting algorithm of [5] that we study. In the first round, all ants search for nests. The algorithm then proceeds in subsequent rounds of recruitment by all ants, with each ant either actively recruiting by calling **recruit**$(\mathbf{1}, \cdot)$, or trying to get recruited by calling **recruit**$(\mathbf{0}, \cdot)$. In each round, each ant chooses to actively recruit with probability $(count{\cdot}q(nest))/n$, where *count* and $q(nest)$ are the assessed population and quality of the nest, and n is the colony size. Recall that the quality is binary, so if a nest is not good, ants simply do not recruit to it.

3 House-Hunting Under Uncertainty

We conclude by discussing our model of uncertainty and the performance of Algorithm 1 under this model. For proofs and details see [14]. Formally, we consider an adversary that determines the

[1] We conjecture that just approximate knowledge of n is required, which is more biologically plausible.

Algorithm 1: Simple House-Hunting Algorithm

```
initialize: $nest := 0$, $count := 0$, $b := 0$
$(nest, count) :=$ search()
while true do
    $b := 1$ with probability $(q(nest) \cdot count)/n$, 0 otherwise
    $(nest, count) :=$ recruit(b, nest)
```

population estimates of each ant in each round of execution by choosing a family of distributions from which the population estimates are drawn. For any $\epsilon \in (0, 1)$, $c' > 2$, and $n \in \mathbb{N}$, we define an (ϵ, c', n)-family of distributions to be a set of distributions $\mathcal{F} = \{F_x\}_{x \in \{1,\ldots,n\}}$, where each F_x is an x-variate distribution. For each $a \in \{1, ..., x\}$, let F_x^a denote the marginal probability distribution of the a^{th} element of F_x. Then, for all $a \in \{1, ..., x\}$, the following are satisfied:

1) $\sum_{y \in [(1-\epsilon)x, (1+\epsilon)x]} F_x^a(y) \geq 1 - 1/n^{c'}$, 2) F_x^a has mean x, 3) the domain of F_x^a is $[0, n]$.

In other words, sampling from distribution F_x in the (ϵ, c', n)-family of distributions results in a vector of x values, each of which is: (1) in the range $[(1-\epsilon)x, (1+\epsilon)x]$ with probability at least $1 - 1/n^{c'}$, (2) equal to x in expectation, and (3) guaranteed to be in the range $[0, n]$.

In our noisy house-hunting model, an adversary choses any (ϵ, c', n)-family of distributions $\mathcal{F}$. In any round r, all ants located in some nest i which has true population $p(i, r)$ receive estimates of this population drawn from $F_{p(i,r)}$ and execute Algorithm 1 based on these estimates.

Note that we do not make independence assumptions, so it is possible that the adversary chooses population estimates that are correlated between different ants. We do assume independence of the population estimates of ants in different nests and across different rounds. The independence between estimates across rounds is important for our analysis, while the independence between estimates in different nests is just a simplification. The independence is inspired by thinking of the population estimates as being determined by a randomized distributed algorithm that runs independently at each nest in each round. In [14] we prove the following main result:

Theorem 3.1. *Fix constants c, c' with $2 < c' < c$, and $\epsilon \in (0, 1)$, and consider any (ϵ, c', n)-family $\mathcal{F}$ of distributions. With probability at least $1 - 1/n^c - 1/n^{c'-2}$, Algorithm 1 using $\mathcal{F}$ solves the* HouseHunting *problem in the noisy model in $\mathcal{O}(1/(1-\epsilon)^2 \cdot k^3 \log^{1.5} n)$ rounds.*

As an example of its usefulness, Theorem 3.1 lets us compose house-hunting with randomized population estimation routines. Consider p ants, positioned uniformly at random on a grid with area A. The density of ants is defined to be $d = p/A$, and can be used to estimate the population p if A is known. From a biological perspective, it is reasonable to assume that ants know the approximate area of a candidate nest because they use that information to assess nest quality.

Musco et al. [10] consider a density estimation algorithm which involves each ant counting how many times it collides with other ants while randomly walking in the grid. They show that an ant's collision rate provides an unbiased estimate of the true density, which additionally is close to d with high probability. While the density estimates of different ants given by encounter-rate-based estimation are not independent, and while the result of [10] does not fully characterize the complex distribution of density estimates, we still have enough information to apply Theorem 3.1. This gives a specific example of how population approximation can be combined with house-hunting using our robustness analysis.

References

[1] Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*. John Wiley & Sons, 2004.

[2] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. Plurality consensus in the gossip model. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 371–390, 2015.

[3] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[4] Nigel R Franks, Anna Dornhaus, Charlotte S Best, and Elizabeth L Jones. Decision making by small and large house-hunting ant colonies: one size fits all. *Animal Behaviour*, 2006.

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

[6] Christiane I.M. Healey and Stephen C. Pratt. The effect of prior experience on nest site evaluation by the ant Temnothorax curvispinosus. *Animal Behaviour*, 76(3):893–899, 2008.

[7] Fabian Kuhn, Nancy Lynch, and Rotem Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 513–522, 2010.

[8] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.

[9] Nancy A. Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

[10] Cameron Musco, Hsin-Hao Su, and Nancy Lynch. Ant-inspired density estimation via random walks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 469–478. ACM, 2016.

[11] Stephen C. Pratt. Quorum sensing by encounter rates in the ant Temnothorax albipennis. *Behavioral Ecology*, 16(2):488–496, 2005.

[12] 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):117–127, 2002.

[13] Stephen C Pratt, David JT 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):1023–1036, 2005.

[14] Tsvetomira Radeva. *A Symbiotic Perspective on Distributed Algorithms and Social Insects*. PhD thesis, Massachusetts Institute of Technology, 2017.

[15] Takao Sasaki and Stephen C. Pratt. Emergence of group rationality from irrational individuals. *Behavioral Ecology*, 22(2):276–281, 2011.

[16] Takao Sasaki and Stephen C. Pratt. Ants learn to rely on more informative attributes during decision-making. *Biology Letters*, 9(6):20130667, 2013.