

Distributed House-Hunting in Ant Colonies

Mohsen Ghaffari Cameron Musco Tsvetomira Radeva Nancy Lynch

{ghaffari, cnmusco, radeva, lynch}@csail.mit.edu, MIT

Abstract

We introduce the study of the *ant colony house-hunting* problem from a distributed computing perspective. When an ant colony's nest becomes unsuitable due to size constraints or damage, the colony must relocate to a new nest. The task of identifying and evaluating the quality of potential new nests is distributed among all ants. The ants must additionally reach consensus on a final nest choice and the full colony must be transported to this single new nest. Our goal is to use tools and techniques from distributed computing theory in order to gain insight into the house-hunting process.

We develop a formal model for the house-hunting problem inspired by the behavior of the *Temnothorax* genus of ants. We then show a $\Omega(\log n)$ lower bound on the time for all n ants to agree on one of k candidate nests. We also present two algorithms that solve the house-hunting problem in our model. The first algorithm solves the problem in optimal $\mathcal{O}(\log n)$ time but exhibits some features not characteristic of natural ant behavior. The second algorithm runs in $\mathcal{O}(k \log n)$ time and uses an extremely simple and natural rule for each ant to decide on the new nest.

1 Introduction

Some recent work in distributed computing theory has focused on biological problems inspired by algorithmic tasks carried out by ant colonies; for example, [7, 9, 11, 18] study collaborative food foraging and [3] models and proposes algorithms for task allocation within a colony. Often computer scientists study biologically-inspired algorithms with the aim of engineering better computing systems. An alternative line of work uses tools developed to analyze and design distributed computer systems to better understand the behavior of biological systems [10]. In this paper, we follow the second approach in an attempt to better understand the house-hunting behavior of the *Temnothorax* ant. The essence of this process is a collaborative search, consensus decision, and relocation of the entire ant colony to a new home, with an emphasis on the decision making component.

We first model the ant colony and its computational constraints, and formally define an algorithmic problem that it solves. The challenge is to develop a model that is simultaneously tractable to theoretical analysis and close enough to reality to give meaningful insights into ant behavior. For example: Is ant behavior in some sense optimal given their biological constraints? One hypothesis may be that behavior has been optimized through evolution. Lower bounds matching the performance of algorithms seen in nature can provide mathematical support for this hypothesis. Alternatively, results may show that behavior is far from optimal, suggesting the existence of hidden constraints or goals yet to be fully understood. Can we identify why certain behavioral patterns have developed and what environmental and biological constraints they are adaptations to?

One motivation for studying the house-hunting process of *Temnothorax* ants is that it has received significant attention from biologists. Various aspects of the process have been studied,

including ants’ preferences and capabilities [15, 26], the general structure of the algorithms used by the ants [19, 23], variations due to colony size [13], and trade-offs between the speed and accuracy of the process [24]. This work forms a wide basis of scientifically tested hypotheses that we can use for the foundation of our model and inspiration for our algorithms.

From a distributed computing perspective, house-hunting is closely related to the fundamental and well-studied problem of consensus [12, 17]. This makes the problem conceptually different from other ant colony inspired problems studied by computer scientists. Task allocation and foraging are both intrinsically related to parallel optimization. The main goal is to divide work optimally amongst a large number of ants in a process similar to load balancing. This is commonly achieved using random allocation or *negative feedback* [1] against work that has already been completed. In contrast, the house-hunting problem is a decision problem in which all ants must converge to the same choice. Both in nature and in our proposed algorithms, this is achieved through *positive feedback* [1], by reinforcing viable nest candidates until a single choice remains. At a high level, our work is related to previous work on rumor spreading in biological populations [8].

1.1 The House-Hunting Process

Temnothorax ants live in fragile rock crevices that are frequently destroyed. It is crucial for colony survival to quickly find and move to a new nest after their home is compromised. This process is highly distributed and involves several stages of searching for nests, assessing nest candidates, recruiting other ants to do the same, and finally, transporting the full colony to the new home.

In the search phase, some ants begin searching their surroundings for possible new nests. Experimentally, this phase has not been studied much; it has been assumed that ants encounter candidate nests fairly quickly through random walking. In the assessment phase, each ant that arrives at a new nest evaluates it based on various criteria, e.g., whether the nest interior is dark and therefore likely free of holes, and whether the entrance to the nest is small enough to be easily defended. These criteria may have different priorities [15, 26] and, in general, it is assumed that nest assessments by an individual ant are not always precise or rational [25]. After some time spent assessing different nests, going back to the old nest and searching for new nests, an ant becomes sufficiently satisfied with some nest and moves on to the recruitment phase, which consists of *tandem runs* – one 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. The ants must solve the classic distributed computing problem of consensus. One strategy that ants are believed to use is a *quorum threshold* [22, 23] – a threshold of the number of ants in a candidate nest, that, when exceeded, indicates that the nest should be chosen as the new home. Each time an ant returns to the new nest, it evaluates (not necessarily accurately) whether a quorum has been reached. If so, it begins the transport phase – picking up and carrying other ants from the old to the new nest. These transports are generally faster than tandem runs and they conclude the house-hunting process by bringing the rest of the colony to the new nest.

1.2 Main Results and Organization

Our main results are a mathematical model of the house-hunting process, a lower bound on the number of rounds required by any algorithm solving the house-hunting problem in the given model, and two house-hunting algorithms.

Our model (Section 2) is based on a synchronous model of execution with n probabilistic ants and communication limited to one ant leading another ant (tandem run or transport), chosen

randomly from the ants at the home nest, to a candidate nest. Ants can also search for new nests by choosing randomly among all k candidate nests. We do not model the time for an ant to find a nest or to lead a tandem run; each of these actions are assumed to take one round.

Our lower bound (Section 3) states that, under this model, no algorithm can solve the house-hunting problem in time sub-logarithmic in the number of ants. The main proof idea is that, in any step of an algorithm’s execution, with constant probability, an ant that does not know of the location of the eventually-chosen nest remains uninformed. Therefore, with high probability, $\Omega(\log n)$ rounds are required to inform all n ants. This technique closely resembles lower bounds for rumor spreading in a complete graph, where the rumor is the location of the chosen nest [16].

Our first algorithm (Section 4) solves the house-hunting problem in asymptotically optimal time. The main idea is a typical example of positive feedback: each ant leads tandem runs to some suitable nest as long as the population of ants at that nest keeps increasing; once the ants at a candidate nest notice a decrease in the population, they give up and wait to be recruited to another nest. With high probability, within $\mathcal{O}(\log n)$ rounds, this process converges to all n ants committing to a single winning nest. Unfortunately, this algorithm relies heavily on a synchronous execution and on the ability to precisely count nest populations, suggesting that the algorithm is susceptible to perturbations of our model and most likely does not match real ant behavior.

The goal of our second algorithm (Section 5) is to be more natural and resilient to perturbations of the environmental parameters and ant capabilities. The algorithm uses a simple positive-feedback mechanism: in each round, an ant that has located a candidate nest recruits other ants to the nest with probability proportional to its current population. We show that, with high probability, this process converges to all n ants being committed to one of the k candidate nests within $\mathcal{O}(k \log n)$ rounds. While this algorithm is not optimal, except when k is assumed to be constant, it exhibits a much more natural process of converging to a single nest. In Section 6, we discuss in more detail possible modifications to the algorithm and various perturbations and faults to which it is resilient. Such robustness criteria are necessary in nature and generally desirable for distributed algorithms.

2 Model

Here we present a simple model of *Temnothorax* ants behavior that is tractable to rigorous analysis, yet rich enough to provide a starting point for understanding real ant behavior.

The environment consists of a home nest, denoted n_0 , along with k candidate new nests, identified as n_i for $i \in \{1, \dots, k\}$. Each nest n_i is assigned a quality $q(i) \in Q$, for some set Q . Throughout this paper we let $Q = \{0, 1\}$, with quality 0 indicating an unsuitable nest, and 1 a suitable one. Additionally, we assume that there is always at least one nest with $q(i) = 1$.

The colony consists of n identical probabilistic finite state machines, representing the ants. We assume n is significantly larger than k , with $k = O(n/\log n)$. Additionally, ants know the value of n but not k , so the state machines may be parameterized by n but must be uniform for all k . This assumption is based on evidence that real *Temnothorax* ants and other species are able to estimate colony size and change their behavior in response [6, 4].

The general behavior of the state machines is unrestricted but their interactions with the environment and with other ants are limited to the high-level functions **search()**, **go()**, and **recruit()**, defined below. Ants execute synchronously in numbered rounds starting with round 1. In each round, each ant may perform unlimited local computation (transition through an arbitrary sequence of states), along with exactly one call to one of the functions: **search()**, **go()**, or **recruit()**.

At the end of each round r , each ant a is located at a nest, denoted by $\ell(a, r) \in \{0, 1, \dots, k\}$; initially, before round 1, all ants are located at the home nest. The value of $\ell(a, r)$ is set by the

calls to **search()**, **go()**, or **recruit()** made by the ant in round r . Let $c(i, r) = |\{a | \ell(a, r) = i\}|$ denote the number of ants located in nest n_i at the end of round r .

In each round, each ant a performs exactly one call to the following functions:

- **search()**: Returns a triple $\langle i, q(i), c(i, r) \rangle$ where i is chosen uniformly at random from $\{1, \dots, k\}$. Sets $\ell(a, r) := i$. This function represents ant a searching randomly for a nest; the return value of the function includes the nest index, the nest's quality and the number of ants at the nest.
- **go(i)**: Takes input $i \in \{1, \dots, k\}$ such that there exists a round $r' < r$ in which $\ell(a, r') = i$. Returns $c(i, r)$. Sets $\ell(a, r) := i$. The function represents ant a revisiting a candidate nest n_i ; the function returns the number of ants at nest n_i at the end of round r .
- **recruit(b, i)**: Takes input $b \in \{0, 1\}$ and $i \in \{1, \dots, k\}$ such that there exists a round $r' < r$ in which $\ell(a, r') = i$. Returns a pair $\langle j, c(0, r) \rangle$ where $j \in \{1, \dots, k\}$. Sets $\ell(a, r) := 0$. The return value j is determined as follows. Let R be the set of all ants that call $recruit(\cdot, \cdot)$, and let P be a uniform random permutation of all ants in R . Let $S \subseteq R$ be the set of ants that call $recruit(1, \cdot)$.

Algorithm 1: Return value j of **recruit**(\cdot, \cdot) for all ants $a \in R$

<pre> 1 M: a set of pairs (a, a') of ants, initially \emptyset 2 for $i = 1$ to P do 3 if $a_{P(i)} \in S$ and $(\cdot, a_{P(i)}) \notin M$ then 4 $a' :=$ uniform random ant from R 5 if $(a', \cdot) \notin M$ and $(\cdot, a') \notin M$ then 6 $M := M \cup (a_{P(i)}, a')$ </pre>	<pre> 7 for $i = 1$ to P do 8 if $(a^*, a_{P(i)}) \in M$ then 9 return j to ant $a_{P(i)}$ where j is 10 input to $recruit(1, j)$ called by a^* 11 else return j to ant $a_{P(i)}$ where j is 12 input to $recruit(\cdot, j)$ called by $a_{P(i)}$ </pre>
---	--

In short, all actively recruiting ants in S randomly choose an ant to recruit. P simply serves as tie breaker to avoid conflicts between recruitments. It is important to note that this process is not a distributed algorithm executed by the ants, but just a modeling tool to formalize the idea of ants recruiting other ants randomly without introducing dependency chains between the ordered pairs of recruiting and recruited ants. Algorithm 1 can be thought of as a centralized process run by the environment of the home nest in order to pair ants appropriately. We believe our results also hold under other natural models for randomly pairing ants.

An ant *recruits successfully* if it is the recruiting ant (first element) in one of the pairs in M . If an ant recruits successfully or is not recruited, **recruit()** just returns the input nest id i . Otherwise, it returns the id of the nest that the ant is recruited to.

We intentionally define **recruit()** to locate ants back to the home nest, where recruitment happens. Additionally, **go(i)** is only applicable to a candidate nest n_i for $i \neq 0$, so calling **recruit()** is the only way to return to the home nest. Since ants are required to call one of the three functions in each round, if an ant does not search (call **search()**) or go to a nest (call **go()**), it is required to stay at the home nest. In this way, all ants located in the home nest participate in recruitment in each round, either actively recruiting ($b = 1$) or waiting to be recruited ($b = 0$).

Recruitment encompasses both the tandem runs and direct transport behavior observed in *Temnothorax* ants. Since direct transport is only about three times faster than tandem walking [21], and since we focus on asymptotic behavior, we do not model this action separately.

Next, we prove a general statement about the recruitment process that will be used in the proofs of our algorithms.

Lemma 2.1. *Let a be an arbitrary ant that executes $\text{recruit}(1, \cdot)$ in some round r , and assume $c(0, r) \geq 2$. Then, with probability at least $1/16$, $(a, \cdot) \in M$; that is, ant a succeeds in recruiting another ant.*

Proof. One requirement for ant a to succeed recruiting is that it does not get recruited by another ant positioned earlier in the random permutation P than a . With probability $1/2$ ant a is located in the first half of the random permutation, so conditioning on that, the probability that a is not recruited by some ant with higher precedence in some round r is at least:

$$\left(1 - \frac{1}{c(0, r)}\right)^{c(0, r)/2} \geq \frac{1}{4^{(1/2)}} \geq \frac{1}{2},$$

where $c(0, r) \geq 2$ is the number of ants at the home nest in round r .

Another requirement for ant a to succeed recruiting is to choose to recruit an ant that has not itself successfully recruited and has not been recruited by the preceding ants. Recall that we conditioned on ant a being located in the first half of the permutation P . With probability $1/2$, ant a chooses to recruit an ant a' located in the second half of the permutation; therefore, a' appears after a in the permutation and so a' has not recruited yet when a recruits. With probability at least $\left(1 - \frac{1}{c(0, r)}\right)^{c(0, r)/2} \geq 1/2$, ant a' is not recruited by any ant other than a in round r .

Therefore, in total, the probability that ant a recruits successfully is at least: $1/2$ to be located in the first half of the permutation, $1/2$ to not be recruited by another ant, $1/2$ to recruit an ant in the second half of the permutation, and $1/2$ for this chosen ant to not be recruited by any other ant. This gives a total probability of at least $1/16$. \square

Problem Statement: An algorithm \mathcal{A} solves the HOUSEHUNTING problem with k nests in $T \in \mathbb{N}$ rounds with probability $1 - \delta$, for $0 < \delta \leq 1$, if with probability $1 - \delta$, taken over all executions of \mathcal{A} , there exists a nest $i \in \{1, \dots, k\}$ such that $q(i) = 1$ and $\ell(a, r) = i$ for all ants a and for all rounds $r \geq T$.

3 Lower Bound

In this section, we present a lower bound on the number of rounds required for an algorithm to solve the house-hunting problem. The key idea of the proof is similar to the lower bounds on spreading a rumor in a complete graph [16] where neighbors contact each other randomly. Assuming a house-hunting process with a single good nest, its location represents the rumor to be spread among all ants and communication between random neighbors is analogous to the recruiting process.

Assume only a single nest, n_w , has quality 1, and so it is the only option for the ants to relocate to. Additionally, assume that each ant is able to recognize nest n_w once it knows its id. These assumptions only restrict the environment and add to the ants' power, so a lower bound under these assumptions is sufficient. Note that each ant has only two ways of reaching a nest: (1) by searching, or (2) by getting recruited to it. Throughout this section, an ant is considered to be *informed* if it know the id of the winning nest n_w ; otherwise, an ant is considered to be *ignorant*.

Lemma 3.1. *With probability at least $1/4$ an ant that is ignorant at the beginning of round r remains ignorant at the end of round r .*

Proof. In round r , ant a may either (1) not participate in recruitment or search (2) participate in recruitment at the home nest or (3) search. In the first case, ant a does not call either **recruit**() or **search**() and is guaranteed to remain ignorant.

In the second case, ant a calls **recruit**($\mathbf{0}, \cdot$). Let X_r be the number of informed ants at the beginning of round r that are recruiting at the home nest to the winning nest; these ants are calling **recruit**($\mathbf{1}, \mathbf{w}$). Then, the probability that ant a remains ignorant after round r is at least:

$$\left(1 - \frac{1}{c(0, r)}\right)^{X_r} \geq \left(1 - \frac{1}{c(0, r)}\right)^{c(0, r)} \geq \frac{1}{4}$$

where $c(0, r) \geq 2$ is the number of ants at the home nest at the beginning of round r . Note that if $c(0, r) < 2$, ant a is forced to recruit itself, so it remains ignorant.

In the third case, for $k \geq 2$, the probability that ant a remains ignorant after one round of searching is $(k-1)/k \geq 1/2$.

Therefore, the overall probability that an ignorant ant remains ignorant in round r , after either searching or participating in recruitment, is at least $1/4$ (since $1/2 > 1/4$). \square

Let random variable \bar{X}_r denote the number of ignorant ants at the beginning of round r . In order for the algorithm to solve the house hunting problem, it is necessary that $\bar{X}_r = 0$ for some round r . Otherwise, there is at least one ant that is not committed to the winning nest n_w .

Theorem 3.2. *For any constant $c > 0$, let \mathcal{A} be an algorithm that solves the HOUSEHUNTING problem with $k \geq 2$ nests in T rounds with probability at least $1/n^c$. Then, $T = \Omega(\log n)$.*

Proof. By Lemma 3.1, with probability at least $1/4$ an ant that is ignorant at the beginning of round r remains ignorant at the end of round r . The probability that an ignorant ant remains ignorant for $r = (\log_4 n)/2 - \log_4(12c)$ consecutive rounds is at least $(1/4)^r \geq 12c/\sqrt{n}$, where $c > 1$ is an arbitrary constant¹. So the expected number of ignorant ants at the beginning of round $r = (\log_4 n)/2 - \log_4(12c)$ is $\mathbb{E}[\bar{X}_r] \geq (12c/\sqrt{n})(n) = 12c\sqrt{n}$.

Note that the random variable \bar{X}_r can be expressed as a sum of indicator random variables \bar{X}_r^a , where for each ant a , \bar{X}_r^a is 1 if ant a is ignorant, and 0 if it is informed, at the beginning of round r . Unfortunately, these indicator random variables are not independent, so we cannot directly apply a Chernoff bound. However, we can still show that their sum is bounded by defining random variables that dominate the \bar{X}_r^a variables.

Let random variable Y_r^a be a random variable such that $P[Y_r^a = 1] = (1/4)^r$ for each a and each r . Let Y_r be the sum of these independent random variables for all ants a . Note that $\mathbb{E}[Y_r] = n(1/4)^r$, so for $r = (\log_4 n)/2 - \log_4(12c)$, $\mathbb{E}[Y_r] = 12c\sqrt{n}$. Therefore, by a Chernoff bound:

$$P[Y_r < 6c\sqrt{n}] < e^{-(12c\sqrt{n})/12} \leq e^{-c \ln n} \leq \frac{1}{n^c}.$$

By Lemma 3.1, the probability that an uninformed ant remains uninformed is at least $1/4$, regardless of the other ants' actions in round r . Therefore, $P[\bar{X}_r^a = 1 | \text{all } \bar{X}_r^{a'} \text{ for } a' \text{ before } a \text{ in } P] \geq (1/4)^r$, where P is the random permutation used to model the recruitment process. So, by the definition of Y_r^a :

$$P[\bar{X}_r^a = 1 | \text{all } \bar{X}_r^{a'} \text{ for } a' \text{ before } a \text{ in } P] > \left(\frac{1}{4}\right)^r = P[Y_r^a = 1]$$

By Lemma 1.18 in [5][Chapter 1] (also stated as Lemma A.1 in the Appendix), $P[\bar{X}_r < x] \leq P[Y_r < x]$ for any $x \leq n$. In particular, for $r = (\log_4 n)/2 - \log_4(12c)$, it follows that :

$$P[\bar{X}_r < 6c\sqrt{n}] \leq P[Y_r < 6c\sqrt{n}] < \frac{1}{n^c}$$

¹We assume a constant is asymptotically smaller than \sqrt{n} .

Therefore, with probability at least $1 - 1/n^c$, at least $6c\sqrt{n}$ ants are ignorant at the end of round $r = (\log_4 n)/2 - \log_4(12c)$. These ants are not informed of the id of the winning nest, and so cannot be located at this nest.

Since algorithm \mathcal{A} solves the HOUSEHUNTING problem with probability at least $1/n^c$ in T rounds, then, with probability at least $1/n^c$ after T rounds, all ants are informed of the winning nest. We showed that with probability at least $1 - 1/n^c$, at least $6c\sqrt{n}$ ants are ignorant at the end of round $r = (\log_4 n)/2 - \log_4(12c)$. Therefore, $T \geq r = \Omega(\log n)$. \square

4 Optimal Algorithm

We present an algorithm that solves the HOUSEHUNTING problem and is asymptotically optimal. In the key step of the algorithm, each ant tries to recruit other ants to the nest it found after searching; after each round of recruiting, each ant checks if the number of ants at its nest has increased or decreased. Nests with a non-decreasing population continue competing while nests with a decreasing population drop out. In each round, the population of at least one nest is non-decreasing, so at least one nest will remain in the competition. Additionally, other nests drop out at a constant rate, meaning a single winning nest will be identified quickly.

This algorithm relies heavily on the synchrony in the execution and the precise counting of the number of ants at a given nest, which makes it sensitive to perturbations of these values, and therefore, not a natural algorithm that resembles ant behavior. However, the algorithm demonstrates that the HOUSEHUNTING problem is solvable in optimal time in the model of Section 2.

4.1 Algorithm Pseudocode and Description

The pseudocode of the algorithm is presented in Algorithm 2. Each call to the functions from Section 2 (in bold) takes exactly one round. The remaining lines of the algorithm are considered to be local computation and are executed in the same round as the preceding function call.

Throughout the algorithm's execution, each ant is in one of four states: *search*, *active*, *passive*, or *final*, initially *search*. Based on the state of the ant, it executes the corresponding case block from the pseudocode. An ant is said to be *committed* to a nest n_i if $nest = i$.

Note that the *search* subroutine is executed only once, during the first round, and the *final* subroutine is what active ants do at the end of the execution to recruit all ants to the winning nest. The other two subroutines, *active* and *passive*, represent the actions of active (recruiting) ants, and ants from bad/dropped-out nests, respectively. Each one of these subroutines takes exactly four rounds; each round is labeled as either R1, R2, R3, or R4 in the pseudocode. The subroutines are carefully scheduled in such a way that these two types of ants do not meet until the end of the competition process when a single winning nest remains; otherwise, the competition between ants from competing nests would be slowed down by ants from dropped-out nests. Therefore, the *active* and *passive* subroutines are padded with *recruit(0, ·)* and *go(nest)* calls to achieve such interleaving (lines 13, 18–19, 35–36, 42 are such padding rounds).

- **Search (lines 6–11):** In the first round, each ant searches for a nest and commits to it. If the nest has quality of 0, the ant moves to the *passive* state; otherwise, it moves to the *active* state.
- **Passive (lines 12–19):** A passive ant is an ant committed to either a bad or a dropped-out nest. The ant spends a round at its (non-competing) nest, then it goes home to be recruited. This call to *recruit(0, nest)* never coincides with a *recruit(1, nest)* of an active ant, so a

Algorithm 2: Optimal HOUSEHUNTING Algorithm

```

1 state : {search, active, passive, final},
2   initially search
3 nest : nest index  $i \in \{0, \dots, k\}$ , initially 0
4 count : an integer in  $\{0, \dots, n\}$ , initially 0
5 quality : a boolean in  $\{0, 1\}$ , initially 0
6 case state = search
7   |  $\langle \textit{nest}, \textit{count}, \textit{quality} \rangle := \textit{search}()$  (R1)
8   | if quality = 0 then
9     | state := passive
10  | else
11  |   | state := active
12 case state = passive
13  | go(nest) (R1)
14  |  $\langle \textit{nest}_t, \cdot \rangle := \textit{recruit}(0, \textit{nest})$  (R2)
15  | if  $\textit{nest}_t \neq \textit{nest}$  then
16  |   | nest :=  $\textit{nest}_t$ 
17  |   | state := final
18  | go(nest) (R3)
19  | go(nest) (R4)
20 case state = final
21  |  $\langle \textit{nest}, \cdot \rangle := \textit{recruit}(1, \textit{nest})$  (R1)
22 case state = active
23  |  $\langle \textit{nest}_t, \cdot \rangle := \textit{recruit}(1, \textit{nest})$  (R1)
24  | countt := go(nestt) (R2)
25  | if ( $\textit{nest}_t = \textit{nest}$ ) and
26  |   (countt ≥ count) then
27  |   | count := countt
28  |   | go(nest) (R3)
29  |   |  $\langle \cdot, \textit{count}_h \rangle := \textit{recruit}(0, \textit{nest})$  (R4)
30  |   | if  $\textit{count}_h = \textit{count}$  then
31  |   |   | state := final
32  |   | else if ( $\textit{nest}_t = \textit{nest}$ ) and
33  |   |   (countt < count) then
34  |   |   | state := passive
35  |   |   | recruit(0, nest) (R3)
36  |   |   | go(nest) (R4)
37  |   | else
38  |   |   | nest :=  $\textit{nest}_t$ 
39  |   |   | countn := go(nest) (R3)
40  |   |   | if  $\textit{count}_n < \textit{count}_t$  then
41  |   |   |   | state := passive
42  |   |   | go(nest) (R4)

```

passive ant can only get recruited by an ant in the *final* state calling $\textit{recruit}(1, \textit{nest})$. Once successfully recruited, the passive ant moves to the *final* state and commits to the new nest.

- **Final (lines 20–21):** An ant in the final state is aware that a single winning nest remains, so it recruits to it in each round. This call to $\textit{recruit}(1, \textit{nest})$ coincides with the call to $\textit{recruit}(0, \textit{nest})$ of passive ants, so once a single nest remains, passive ants are recruited to it in every fourth round.
- **Active (lines 22–42):** An active ant tries to recruit other ants to its competing nest by executing $\textit{recruit}(1, \textit{nest})$. The ant then goes to the resulting nest to count the number of ants there. Based on the values of the resulting nest (\textit{nest}_t) and count (\textit{count}_t), we consider three cases:

- *Case 1 (lines 25–31):* \textit{nest}_t is the same as the nest that the ant is committed to and the number of ants in that nest has not decreased; therefore, the nest remains competing. As a result, the ant updates the new count and spends an extra round at the nest that has a special purpose with respect to Cases 2 and 3 below. Finally, the ant checks if the number of ants at the home nest is the same as the number of ants at the committed nest; if this is the case, it means that all ants have been recruited to a single winning nest and the ant switches to the *final* state.
- *Case 2 (lines 32–36):* \textit{nest}_t is the same as the nest that the ant is committed to but the number of ants has decreased; therefore the nest is about to drop out. As a result, the

ant sets its state to *passive* and spends a round at the home nest, which coincides with the round an active ant spends at the committed nest in Case 1.

- *Case 3 (lines 37–42):* $nest_t$ is different from the nest that the ant is committed, which indicates the ant got recruited to another nest. Although it already knows the number of ants ($count_t$) at the new nest, the ant updates that count ($count_n$). The reason for this is to determine whether this new nest is about to compete or drop out. If $count_t = count_n$, the nest is competing because the active ants in Case 1 are spending the same round at the committed nest; if $count_t > count_n$, the nest is dropping out because the ants in Case 2 already determined a decrease in the number of ants and are spending this round at the home nest.

4.2 Correctness Proof and Time Bound

As written, Algorithm 2 never terminates; after all ants are committed to the same nest and in the *final* state, they continue to recruit (each other) in every round. This issue can easily be handled if ants check whether the number of ants at the home nest is the same as the number of ants at the candidate nest. However, for simplicity, we choose not to complicate the pseudocode and consider the algorithm to terminate once all ants have reached the *final* state and, thus, committed to the same unique nest.

Proof Overview: The correctness proof and time bound of Algorithm 2 are structured as follows. By defining a slightly different but equivalent recruitment process, we show, in Lemma 4.1, that a competing nest is equally likely to continue competing and to drop out. Consequently, as we show in Lemma 4.2, each competing nest has a constant probability of dropping out. We put these lemmas together in Theorem 4.3 to show that, with high probability, Algorithm 2 solves the HOUSEHUNTING problem in $\mathcal{O}(\log n)$ rounds: $\mathcal{O}(\log k)$ rounds to converge to a single nest and $\mathcal{O}(\log n)$ rounds until all passive ants are recruited to it.

Let R_3 be the set of round numbers r such that $r \in R_3$ iff $r \bmod 4 = 3$; these rounds are exactly the rounds in which only ants committed to active nests are located at the home nest and try to recruit each other. Also, let $C(i, r)$ for each $r \in R_3$ and each nest n_i denote the set of ants committed to nest n_i and competing at the home nest. This implies that $|\cup_{i \in [1, k]} C(i, r)| = c(0, r)$ for $r \in R_3$.

Let random variable X_r^a , for each ant a and each round $r \in R_3$, take on values $-1, 0$ or 1 as follows. If ant a gets recruited by another ant in round r , then $X_r^a = -1$; if ant a successfully recruits another ant, then $X_r^a = 1$; otherwise, $X_r^a = 0$.

Let random variable Y_r^i denote the change in the number of ants at nest n_i after one round of recruiting, where $r \in R_3$:

$$Y_r^i = \sum_{a \in C(i, r)} X_r^a.$$

Informally speaking, the change in population of nest n_i is simply the sum of identically distributed $\{-1, 0, 1\}$ random variables with mean 0 that take on non-zero values with constant probability. Therefore, the sum of these variables is negative with constant probability. However, proving this fact requires a more rigorous argument because the X_r^a variables are not independent.

As specified by the recruitment model, Y_r^i is the result of a randomized recruitment process in which all ants are ordered by a random permutation. Then, in order of the permutation, ants choose uniformly at random other ants at the home nest to recruit. The random variables involved in this process are the random permutation P as well as the set of random choices of ants. Consider

the vector \mathbf{o}_r , of length $c(0, r)$, where $\mathbf{o}_r(a) = -1$ if ant a is chosen by another ant, $\mathbf{o}_r(a) = 1$ if ant a chooses an ant a' such that a' had not already chosen an ant and a' was not already chosen by another ant, and $\mathbf{o}_r(a) = 0$ otherwise.

Lemma 4.1. *Let nest n_i be a competing nest in round $r \in R_3$. Then, $P[Y_r^i < 0] = P[Y_r^i > 0]$.*

Proof. Random variable Y_r^i is simply the sum of values of $\mathbf{o}_r(a)$ corresponding to ants $a \in C(i, r)$. Since \mathbf{o}_r has exactly the same number of -1 s and 1 s, it is possible to choose some (non-random) permutation P' that swaps the locations of the -1 s and 1 s in \mathbf{o}_r . Choosing a random permutation P in the recruitment process and then applying the swapping permutation P' negates the value of Y_r^i . Moreover, choosing a uniform random permutation P and then applying this swapping permutation P' still yields a uniform random permutation. So, $-Y_r^i$ is distributed according to the exact same distribution as Y_r^i . Therefore, Y_r^i is symmetric around 0 and $P[Y_r^i < 0] = P[Y_r^i > 0]$. \square

Lemma 4.2. *Let nest n_i be a competing nest in round $r \in R_3$. If $|C(i, r)| < c(0, r)$, then $P[Y_r^i < 0] \geq 1/66$.*

Proof. Let P_1 and P_2 be uniform random permutations. Consider using P_1 as the uniform permutation in the recruitment process that determines Y_r^i . Permutation P_2 swaps the position of a fixed ant a^* not committed to nest n_i with the position of an ant a chosen uniformly at random among the ants $C(i, r)$ (the ants committed to n_i). The existence of such an ant is guaranteed by the assumption that $|C(i, r)| < c(0, r)$.

By Lemma 2.1, $P[X_r^{a^*} = 1] \geq 1/16$. Conditioning on $Y_r^i = 0$, at most $1/2$ the ants $a \in C(i, r)$ have $X_r^a = 1$. Therefore, with probability at least $1/2$, the ant a chosen uniformly at random by permutation P_2 has $X_r^a < 1$. Conditioning on $X_r^{a^*} = 1$ can only increase this probability. So with probability at least $(1/2)(1/16)$, applying the composition of P_1 and P_2 to compute Y_r^i increases its value by at least 1. Since the composition of P_1 and P_2 is also a uniform random permutation, the distribution of Y_r^i remains exactly the same as the case when only P_1 is applied. Therefore,

$$\begin{aligned} P[Y_r^i = 0] &\leq 1 - \frac{1}{32}P[Y_r^i = 0] \\ P[Y_r^i = 0] &\leq \frac{32}{33}. \end{aligned}$$

By Lemma 4.1, $P[Y_r^i < 0] = P[Y_r^i > 0]$. Therefore, $P[Y_r^i < 0] \geq 1/66$. \square

Theorem 4.3. *For any constant $c > 0$, with probability at least $1 - 1/n^c$, Algorithm 2 solves the HOUSEHUNTING problem in $\mathcal{O}(\log n)$ rounds.*

Proof. In the first round, all ants search for nests, so the expected number of ants located each good nest is n/k . Assuming $k \leq n/(12(c+1)\log n) = \mathcal{O}(n/\log n)$, by a Chernoff bound, it follows that, with probability at least $1 - 1/n^{c+1}$, at least $n/(2k)$ ants visits each good nest.

Let k_r be a random variable denoting the number of competing nests in round $r \in R_3$. Suppose $k_r > 1$, and so $|C(i, r)| < c(0, r)$. By Lemma 4.2, $P[Y_r^i < 0] \geq 1/66$ for each nest n_i among the k_r competing nests. Therefore, $\mathbb{E}[k_{r+4}] \leq (65/66)k_r$. Also, note that for any round r , $k_r \leq k$. For $r = \log_{66/65} k + (c+1)\log_{66/65} n = \mathcal{O}(\log n)$, where $c > 0$ is an arbitrary constant, it follows that $\mathbb{E}[k_r] = 1/n^{c+1}$. By a Markov bound, $P[k_r \geq 1] \leq 1/n^{c+1}$, so with probability at least $1 - 1/n^{c+1}$, $k_r \leq 1$. Note that in any given round, it is not possible for all nests to experience decrease in population, so for each round r , we have $k_r \geq 1$. So, overall, after $\mathcal{O}(\log n)$ rounds, with probability $1 - 1/n^{c+1}$, there is exactly one competing nest.

After there is exactly one competing nest in some round $r \in R_3$, all ants committed to it switch to the *final* state and start recruiting the passive ants during every fourth round. Each recruited ant also transitions to the final state, resulting in at most $\mathcal{O}(\log n)$ rounds until all ants are committed to the winning nest.

Therefore, in total, with probability at least $1 - 1/n^c$ each nest is discovered by at least one ant and all ants are committed to a single good nest in $\mathcal{O}(\log n)$ rounds. \square

5 Simple Algorithm

We now give a very simple algorithm (Algorithm 3) that solves the HOUSEHUNTING problem in $\mathcal{O}(k \log n)$ rounds with high probability. The main idea of the algorithm is that all ants initially search for nests and those that find good nests simply continuously recruit to their nests with probability proportional to nest population in each round. Ants in larger nests recruit at higher rates, and eventually their populations swamp the populations of smaller nests. This process is similar to the well-known *Polya's urn* model [2].

5.1 Algorithm Description and Pseudocode

In each round of Algorithm 3, each ant can be in one of two states: *active* or *passive*, initially starting in the *active* state. In the first round of the algorithm, all ants search for nests; the ants that find good nests remain in the *active* state, and the ants that find bad nests switch to the *passive* state. Then, the algorithm proceeds in alternating rounds of recruitment by all ants at the home nest (either active **recruit**(**1**, \cdot) or passive **recruit**(**0**, \cdot)), and population assessment at candidate nests. In each round of population assessment, each ant chooses to recruit actively in the next round with probability $count/n$, where $count$ is the assessed population at the candidate nest, and n is the total number of ants. When a passive ant gets recruited to a nest, it commits to the new nest and becomes active again. When an active ant gets recruited to a different nest, it commits to the new nest and starts recruiting for that new nest.

Algorithm 3: Simple House-Hunting

<pre> 1 <i>state</i> : {<i>active</i>, <i>passive</i>}, initially <i>active</i> 2 $\langle nest, count, quality \rangle := \mathbf{search}()$ 3 if <i>quality</i> = 0 then 4 <i>state</i> := <i>passive</i> 5 case <i>state</i> = <i>active</i> 6 <i>b</i> := 1 with probability $\frac{count}{n}$, 0 otherwise 7 <i>nest</i> := recruit(b, <i>nest</i>) 8 <i>count</i> := go(<i>nest</i>) </pre>	<pre> 9 case <i>state</i> = <i>passive</i> 10 <i>nest</i>_{<i>t</i>} := recruit(0, <i>nest</i>) 11 if <i>nest</i>_{<i>t</i>} \neq <i>nest</i> then 12 <i>state</i> := <i>active</i> 13 <i>nest</i> := <i>nest</i>_{<i>t</i>} 14 <i>count</i> := go(<i>nest</i>) </pre>
---	--

5.2 Correctness Proof and Time Bound

For each nest n_i and each round r let random variable $p(i, r) = c(i, r)/n$ denote the proportion of ants at nest n_i in round r . By the pseudocode, ants located at nest n_i in round r will recruit with probability $p(i, r)$ in round $r + 1$. Define $\Sigma^2(r) = \sum_{i=1}^k p(i, r)^2$, the expected proportion of ants that will recruit in total, in round r . Since $\sum_{i=1}^k p(i, r) = 1$, by the ℓ_1 versus ℓ_2 norm bound $\Sigma^2(r) \geq 1/k$.

Proof Overview: The correctness proof and time bound of the algorithm are structured as follows. First, in Lemmas 5.1, 5.2, and 5.3, we show some basic bounds on the expected number of ants recruited in each round and the change of the value of $p(i, r)$ for a single nest in each round of recruiting. Then, in Lemmas 5.5, 5.6, and 5.7, we use a Taylor series expansion of the ratio between the populations of two nests to show that in expectation this ratio increases multiplicatively by $(1 + \Omega(1/k))$ in each round of recruiting, provided that both nests have a $\Omega(1/k)$ fraction of the total population. On the other hand, if a nest has less than a $\Omega(1/k)$ fraction of the total population, in Lemmas 5.8 and 5.9, we show that the ants in such a nest recruit so slowly that the nest completely drops out within $\mathcal{O}(k \log n)$ rounds with high probability. Finally, in Theorem 5.11, we consider all $\binom{k}{2}$ pairs of nests to show that only a single nest will remain in each pair within $\mathcal{O}(k \log n)$ rounds, ensuring, by a union bound, that a single nest remains overall.

Throughout this section, let $c > 0$ be an arbitrary constant and let d be an arbitrary constant such that $d \geq 64$. For the sake of analysis, assume that $k \leq \sqrt{n/(8d^2(c+6) \log n)} = \mathcal{O}(\sqrt{n/\log n})$. While we feel that this assumption is reasonable, we are also hopeful that it could be removed. Let R_1 be the set of all odd rounds, excluding round 1; by the pseudocode, in these rounds, ants are located at the candidate nests. Therefore, for each $r \in R_1$ and each ant a , $\ell(a, r) = i \neq 0$; for each round $r' \notin R_1$ and for each ant a , $\ell(a, r') = 0$.

5.2.1 Change in Population of a Single Nest in One Round

We first study how the population of a single nest changes in a single round. Intuitively, we expect a $p(i, r)$ proportion of the ants in n_i to recruit, and a $\Sigma^2(r)$ proportion to be recruited. Therefore, we expect the population to change by $p(i, r) [p(i, r) - \Sigma^2(r)]$. Qualitatively, we show this in Lemma 5.3, modulo constant factors; the main technical difficulty is handling dependencies in the random recruiting process. Before we are ready to prove Lemma 5.3, we show two lemmas that state the expected outcome of a single ant recruiting.

As before, define random variable X_r^a to take on value -1 if ant a is recruited away from its current nest in round r , 1 if it successfully recruits another ant, and 0 otherwise.

Note that the following lemma is stated in a slightly generalized way: it applies to an ant a recruiting with a fixed probability p in a given round r . By the pseudocode of the algorithm it is clear that whenever an ant recruits, it is always with probability $p(i, r)$, so the majority of the time we will apply the lemma for $p = p(i, r)$; however, the general statement of the lemma helps reason about the expected value of X_r^a with respect to two ants from two different nests in the proof of Lemma 5.2.

Lemma 5.1. *Let n_i be any nest and let ant a be located in nest n_i in some round $r \in R_1$. Also, suppose ant a recruits with probability p in round r . Then, there exist functions ξ_1 and ξ_2 such that $\mathbb{E}[X_r^a] = p\xi_1(i, r) - \xi_2(i, r)$ and $\xi_1(i, r) \geq \xi$ for a fixed constant $\xi > 0$.*

Proof. Let A_r^a be an indicator variable indicating whether some ant a chooses to recruit (executes **recruit**($\mathbf{1}, \cdot$)) in round $r + 1$. By assumption, $\Pr[A_r^a = 1] = p$. Whether ant a is actually successful in recruiting another ant depends on: (1) the order of the random recruiting permutation P , (2) the choices of other ants to recruit or not, and (3) the choices by recruiting ants of whom to recruit. Let B be a random variable encompassing all these random variables that affect X_r^a , excluding A_r^a . Therefore, B is a triple $(P, \{A_r^{a'} | a' \neq a\}, \{1, \dots, n\}^n)$ and it takes on values from the set \mathcal{B} of all such triples. The expected value of X_r^a is:

$$\mathbb{E}[X_r^a] = \sum_{b \in \mathcal{B}} \Pr[B = b] \cdot (p\mathbb{E}[X_r^a | B = b, A_r^a = 1] + (1 - p)\mathbb{E}[X_r^a | B = b, A_r^a = 0]).$$

Fix some value $B = b$. We consider several cases based on the nine possible pairs of values of A_r^a and X_r^a (A_r^a has two possible values and X_r^a has three possible values). However, note that some permutations of these values are not allowed; that is, if $A_r^a = 0$ (the ant chooses not to recruit), it is not possible that $X_r^a = 1$ (the ant succeeds in recruiting). Also, since $B = b$ is already fixed, ant a is either chosen by another ant or not, regardless of the value of A_r^a . This fact rules out two more cases: (1) the case where $X_r^a = 0$ if $A_r^a = 0$, and $X_r^a = -1$ if $A_r^a = 1$, and (2) the case where $X_r^a = -1$ if $A_r^a = 0$, and $X_r^a = 0$ if $A_r^a = 1$. In (1), since the choices of the other ants are already fixed and included in $B = b$, it is not possible that ant a gets recruited if it chooses to recruit but it does not get recruited otherwise. Similarly, in (2), it is not possible that ant a gets recruited when it chooses not to recruit but does not get recruited when it fails to recruit.

All remaining cases are listed below:

Case 1: $X_r^a = -1$ for both $A_r^a = 0$ and $A_r^a = 1$. That is, the ant is recruited by another ant no matter its decision.

Case 2: $X_r^a = -1$ if $A_r^a = 0$, and $X_r^a = 1$ if $A_r^a = 1$. That is, if the ant chooses to recruit it succeeds and if not, it is recruited by another ant.

Case 3: $X_r^a = 0$ for both $A_r^a = 0$ and $A_r^a = 1$. That is, whether the ant chooses to recruit or not, it will not be part of a successful recruitment.

Case 4: $X_r^a = 0$ if $A_r^a = 0$, and $X_r^a = 1$ if $A_r^a = 1$. That is, if the ant chooses to recruit it succeeds, but if not, it is not recruited by another ant.

Let p_c^b be the probability of case c occurring given $B = b$. The expected value of X_r^a is:

$$\begin{aligned}\mathbb{E}[X_r^a] &= \sum_{b \in \mathcal{B}} \Pr[B = b] \cdot \left(p \left(p_2^b + p_4^b - p_1^b \right) + (1 - p) \left(-p_1^b - p_2^b \right) \right) \\ &= \sum_{b \in \mathcal{B}} \Pr[B = b] \left(p \left(2p_2^b + p_4^b \right) - \left(p_1^b + p_2^b \right) \right) \\ &= p \mathbb{E}_B \left[2p_2^b + p_4^b \right] - \mathbb{E}_B \left[p_1^b + p_2^b \right].\end{aligned}$$

Now, we need to show that $\mathbb{E}_B [2p_2^b + p_4^b] \geq \xi$ for some fixed constant $\xi > 0$. By Lemma 2.1, each recruiting ant succeeds with some probability lower bounded by a constant. A recruiting ant only succeeds in cases 2 and 4 so we have $\mathbb{E}_B [p_2^b + p_4^b] \geq \xi$ for some constant $\xi > 0$. Therefore, $\mathbb{E}_B [2p_2^b + p_4^b] \geq \mathbb{E}_B [p_2^b + p_4^b] \geq \xi$.

Overall, $\mathbb{E}[X_r^a] = p\xi_1(i, r) - \xi_2(i, r)$, where $\xi_1(i, r) = \mathbb{E}_B [2p_2^b + p_4^b]$ is any constant greater than $\xi > 0$ and $\xi_2(i, r) = \mathbb{E}_B [p_1^b + p_2^b]$ is some unspecified value. \square

Note that the values of $\xi_1(i, r)$ and $\xi_2(i, r)$ are slightly different for different nests because the behavior of an ant depends on the nest it is committed to. For example, the different probabilities corresponding to the cases in Lemma 5.1 vary based on the population of a given nest in a given round and the populations of the remaining nests in that round. Next, we prove a bound on the $\xi_1(i, r)$ and $\xi_2(i, r)$ values of two different nests.

Fix a constant ξ and functions ξ_1 and ξ_2 such that $\xi_1(i, r) \geq \xi$ for all rounds r and all nests n_i .

Lemma 5.2. *Let n_i and n_j be two nests with $p(i, r) \leq p(j, r)$ in some round $r \in R_1$. Then, $\xi_1(i, r) \cdot p(i, r) - \xi_2(i, r) \leq \xi_1(j, r) \cdot p(i, r) - \xi_2(j, r)$.*

Proof. Let a_i and a_j be two ants located in nests n_i and n_j , respectively, in round $r \in R_1$. Ant a_i recruits with probability $p(i, r)$ and ant a_j recruits with probability $p(j, r)$ where $p(i, r) \leq p(j, r)$. By Lemma 5.1, applied to ant a_i , recruiting with probability $p(i, r)$, and nest n_i , it follows that $\mathbb{E}[X_r^{a_i}] = p(i, r)\xi_1(i, r) - \xi_2(i, r)$.

Suppose we swap the locations of ants a_i and a_j and let both ants recruit with probability $p(i, r)$; since the ants just switch locations, the populations of the two nests remain the same, so the rest of the ants still recruit with probability $p(i, r)$ in nest n_i and $p(j, r)$ in nest n_j ². By Lemma 5.1, applied to ant a_i , recruiting with probability $p(i, r)$, and nest n_j , it follows that the expected value of $X_r^{a_i}$ in this hypothetical situation is $\mathbb{E}'[X_r^{a_i}] = p(i, r)\xi_1(j, r) - \xi_2(j, r)$.

As a result of switching the positions of ants a_i and a_j , we are essentially lowering a_j 's probability to recruit. Therefore, the probability that ant a_i succeeds recruiting, if it chooses to recruit, increases, and the probability that ant a_i is recruited, if it chooses not to recruit, decreases.

Therefore, $\mathbb{E}[X_r^{a_i}] \leq \mathbb{E}'[X_r^{a_i}]$, and so $p(i, r)\xi_1(i, r) - \xi_2(i, r) \leq p(i, r)\xi_1(j, r) - \xi_2(j, r)$. \square

Next, we use Lemma 5.1 to calculate the expected change in the value of $p(i, r)$ after one round of recruiting.

Lemma 5.3. *For each nest n_i and each round $r \in R_1$:*

$$\mathbb{E}[p(i, r + 2)] = p(i, r) \cdot [1 + \xi_1(i, r) \cdot p(i, r) - \xi_2(i, r)].$$

Proof. By linearity of expectation,

$$\mathbb{E}[p(i, r + 2)] = p(i, r) + \frac{1}{n} \sum_{\{a | \ell(a, r) = i\}} \mathbb{E}[X_r^a] = p(i, r) \cdot (1 + \mathbb{E}[X_r^a]),$$

where in the last equation a is a fixed ant in nest n_i and the equality follows from the fact that X_r^a is identically distributed for each ant in the same nest.

By Lemma 5.1, $\mathbb{E}[X_r^a] = p(i, r)\xi_1(i, r) - \xi_2(i, r)$, so:

$$\mathbb{E}[p(i, r + 2)] = p(i, r) \cdot (1 + p(i, r)\xi_1(i, r) - \xi_2(i, r)). \quad \square$$

Lemma 5.3 shows that the population change in a single nest depends *quadratically* on $p(i, r)$. With this fact, we can show that larger nests will tend to ‘swamp’ smaller nests, causing their populations to drop to 0.

5.2.2 Relative Changes in the Populations of Two Nests in One Round

We first define a measurement of the population gap between two nests.

Definition 1. *For any two nests n_i and n_j , let $n_H(i, j, r) \in \{i, j\}$ be the id of the nest with the higher population in round r . Let $n_L(i, j, r)$ be the nest with the lower population. For simplicity of notation let $p_H(i, j, r) = p(n_H(i, j, r), r)$ and $p_L(i, j, r) = p(n_L(i, j, r), r)$. Define:*

$$\epsilon(i, j, r) = \frac{p_H(i, j, r)}{p_L(i, j, r)} - 1.$$

That is, $\epsilon(i, j, r)$ is the relative population gap between the larger and smaller of nests n_i and n_j in round r .

²Here it is not important to which nests ants are recruiting and bringing other ants; we just want to reason about the value of $X_r^{a_i}$. Therefore, it is not alarming that ant a_i is located at nest n_j and recruiting with probability $p(i, r)$.

Lemma 5.4. For any two nests n_i and n_j , $\mathbb{E}[\epsilon(i, j, 1)] \geq 1/(3(n-1))$.

Proof. We need to bound the expected difference between two nests after the first round of searching. If nests n_i and n_j receive the same number of ants after the initial round of searching, then, by definition, $\epsilon(i, j, 1) = 0$. Let $2 \leq x \leq n$ be the total number of ants that find either nest n_i or nest n_j in the first round of searching. Note that if $x = 1$, it is not possible for both nests to receive the same number of ants, and if $x = 0$, then the nests are already empty. The probability that both nests receive exactly $x/2$ ants each is:

$$\Pr[\epsilon(i, j, 1) = 0] = \binom{x}{x/2} \left(\frac{1}{2}\right)^{x/2} \left(\frac{1}{2}\right)^{x/2} \leq \left(\frac{e2^x}{\pi\sqrt{x}}\right) \frac{1}{2^x} \leq \frac{e}{\pi\sqrt{x}} \leq \frac{e}{\pi\sqrt{2}} < \frac{2}{3},$$

where in the second step we use the Stirling approximation $\sqrt{2\pi}x^{x+1/2}e^{-x} \leq x! \leq ex^{x+1/2}e^{-x}$. Conditioning on $\epsilon(i, j, 1) \neq 0$, the smallest possible value of $\epsilon(i, j, 1)$ is $1/(n-1)$, and so $\mathbb{E}[\epsilon(i, j, 1)|\epsilon(i, j, 1) \neq 0] \geq 1/(n-1)$. Therefore,

$$\mathbb{E}[\epsilon(i, j, 1)] \geq \mathbb{E}[\epsilon(i, j, 1)|\epsilon(i, j, 1) \neq 0] \cdot \Pr[\epsilon(i, j, 1) \neq 0] \geq \frac{1}{3(n-1)}. \quad \square$$

Since $\epsilon(i, j, r)$ is a ratio of two populations, it is not immediately clear how to compute its expected change between rounds. However, using a Taylor series expansion we are able to *linearize* this ratio, and then use Lemma 5.3 to compute its expected change. We first show how to use the Taylor series expansion in our setting.

Lemma 5.5. Consider positive numbers x_0, x_1, y_0 , and y_1 . Let $\Delta x = x_1 - x_0$, $\Delta y = y_1 - y_0$, and $\Delta\left(\frac{x}{y}\right) = \frac{x_1}{y_1} - \frac{x_0}{y_0}$. Then:

$$\Delta\left(\frac{x}{y}\right) = \left[\frac{\Delta x}{y_0} - \frac{x_0\Delta y}{y_0^2}\right] \cdot \sum_{i=0}^{\infty} \left(\frac{-\Delta y}{y_0}\right)^i \quad (1)$$

Proof. We can expand $\Delta\left(\frac{x}{y}\right)$ using a Taylor series for $f(x, y) = \frac{x}{y}$. To save notation, we write $\frac{\partial^k f}{\partial x^i \partial y^{k-i}}$ to denote $\frac{\partial^k f}{\partial x^i \partial y^{k-i}}(x_0, y_0)$. We have:

$$\Delta\left(\frac{x}{y}\right) = \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{1}{2!} \left[\frac{\partial^2 f}{\partial x^2}\Delta x^2 + 2\frac{\partial^2 f}{\partial x\partial y}\Delta x\Delta y + \frac{\partial^2 f}{\partial y^2}\Delta y^2\right] + \frac{1}{3!} \left[\frac{\partial^3 f}{\partial x^3}\Delta x^3 + \dots\right] + \dots$$

We will show by induction that, for all k :

$$\frac{1}{k!} \left[\frac{\partial^k f}{\partial x^k}\Delta x^k + k\frac{\partial^k f}{\partial x^{k-1}\partial y}\Delta x^{k-1}\Delta y + \dots + \frac{\partial^k f}{\partial y^k}\Delta y^k\right] = \left[\frac{\Delta x}{y_0} - \frac{x_0\Delta y}{y_0^2}\right] \left[\frac{-\Delta y}{y_0}\right]^{k-1}. \quad (2)$$

Plugging into the Taylor series expansion, this immediately gives (1). For $k = 1$, we differentiate $f(x, y) = \frac{x}{y}$ to get our base case:

$$\frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y = \frac{\Delta x}{y_0} - \frac{x_0\Delta y}{y_0^2}.$$

Now for $k > 1$, by the inductive assumption we have:

$$\frac{1}{(k-1)!} \left[\frac{\partial^{k-1} f}{\partial x^{k-1}}\Delta x^{k-1} + \dots + \frac{\partial^{k-1} f}{\partial y^{k-1}}\Delta y^{k-1}\right] = \left[\frac{\Delta x}{y_0} - \frac{x_0\Delta y}{y_0^2}\right] \left[\frac{-\Delta y}{y_0}\right]^{k-2} = \frac{\Delta x(-\Delta y)^{k-2}}{y_0^{k-1}} + \frac{x_0(-\Delta y)^{k-1}}{y_0^k}.$$

Differentiating gives:

$$\begin{aligned} \frac{1}{k!} \left[\frac{\partial^k f}{\partial x^k} \Delta x^k + \dots + \frac{\partial^k f}{\partial y^k} \Delta y^k \right] &= \frac{1}{k} \left[\frac{(-\Delta y)^{k-1}}{y_0^k} \Delta x - \frac{(k-1)\Delta x (-\Delta y)^{k-2}}{y_0^k} \Delta y - \frac{kx_0 (-\Delta y)^{k-1}}{y_0^{k+1}} \Delta y \right] \\ &= \left[\frac{\Delta x \cdot (-\Delta y)^{k-1}}{y_0^k} + \frac{x_0 \Delta x (-\Delta y)^k}{y_0^{k+1}} \right] = \left[\frac{\Delta x}{y_0} - \frac{x_0 \Delta y}{y_0^2} \right] \left[\frac{-\Delta y}{y_0} \right]^{k-1}. \end{aligned}$$

□

For simplicity of notation, write $p_H = p(n_H(i, j, r), r)$, $p_L = p(n_L(i, j, r), r)$. Additionally, write $\Delta p_H = p(n_H(i, j, r), r+2) - p(n_H(i, j, r), r)$, $\Delta p_L = p(n_L(i, j, r), r+2) - p(n_L(i, j, r), r)$ and $\Delta \left(\frac{p_H}{p_L} \right) = \frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - \frac{p(n_H(i, j, r), r)}{p(n_L(i, j, r), r)}$. We can now apply Lemma 5.5 to show:

Lemma 5.6. *Let n_i and n_j be two nests with $\Delta p_L < p_L$ for some round $r \in R_1$. Then, $\Delta \left(\frac{p_H}{p_L} \right) \geq \frac{1}{2} \left(\frac{\Delta p_H}{p_L} - \frac{p_H \Delta p_L}{p_L^2} \right)$.*

Proof. Note that we are not considering expectations for now, just the actual values. Applying Lemma 5.5, we can write $\Delta \left(\frac{p_H}{p_L} \right)$ in terms of Δp_H and Δp_L . We have:

$$\Delta \left(\frac{p_H}{p_L} \right) = \left(\frac{\Delta p_H}{p_L} - \frac{p_H \Delta p_L}{p_L^2} \right) \left(1 - \left(\frac{\Delta p_L}{p_L} \right) + \left(\frac{\Delta p_L}{p_L} \right)^2 - \left(\frac{\Delta p_L}{p_L} \right)^3 + \dots \right).$$

The value $\frac{\Delta p_L}{p_L}$ is strictly less than 1 by the assumption that $\Delta p_L < p_L$. So:

$$\sum_{i=0}^{\infty} \left(-\frac{\Delta p_L}{p_L} \right)^i = \frac{1}{1 + \Delta p_L/p_L} \geq \frac{1}{1+1} = 1/2. \quad \square$$

Next, we use the value of $\Delta(p_H/p_L)$ in order to calculate the expected change in $\epsilon(i, j, r)$ for two nests n_i and n_j after one round of recruiting.

Lemma 5.7. *Let n_i and n_j be two nests with $\Delta p_L < p_L$, $p(i, r) \geq 1/(dk)$ and $p(j, r) \geq 1/(dk)$ for some round $r \in R_1$. Then, $\mathbb{E}[\epsilon(i, j, r+2)] \geq (1 + 1/(2dk)) \mathbb{E}[\epsilon(i, j, r)]$.*

Proof. First we show that:

$$\epsilon(i, j, r+2) \geq \frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - 1.$$

This is true because, if the larger of the two nests in round r remains larger in round $r+2$, then $n_H(i, j, r) = n_H(i, j, r+2)$ and $n_L(i, j, r) = n_L(i, j, r+2)$, so, by the definition of ϵ , $\epsilon(i, j, r+2) = \frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - 1$. On the other hand, if the two nests flip positions, then $\frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - 1$ must be smaller than $\epsilon(i, j, r+2)$ since it contains the population of the larger nest in the denominator of the fraction. Fixing $p(i, r)$ and $p(j, r)$, we have:

$$\begin{aligned} \mathbb{E}[\epsilon(i, j, r+2)] &\geq \mathbb{E} \left[\frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - 1 \right] = \epsilon(i, j, r) + \mathbb{E} \left[\frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - 1 - \epsilon(i, j, r) \right] \\ &= \epsilon(i, j, r) + \mathbb{E} \left[\frac{p(n_H(i, j, r), r+2)}{p(n_L(i, j, r), r+2)} - \frac{p(n_H(i, j, r), r)}{p(n_L(i, j, r), r)} \right] \end{aligned}$$

So we have reduced our question of how we expect the *absolute difference* between the nests to change in a single round to the question of how we expect the ratio of population to change.

By Lemma 5.6, it follows:

$$\mathbb{E}[\epsilon(i, j, r + 2)] \geq \epsilon(i, j, r) + \frac{1}{2} \left(\mathbb{E} \left[\frac{\Delta p_H}{p_L} - \frac{p_H \Delta p_L}{p_L^2} \right] \right) \geq \epsilon(i, j, r) + \frac{1}{2} \left(\frac{p_L \mathbb{E}[\Delta p_H] - p_H \mathbb{E}[\Delta p_L]}{p_L^2} \right).$$

By Lemma 5.3 we have:

$$\begin{aligned} p_L \mathbb{E}[\Delta p_H] - p_H \mathbb{E}[\Delta p_L] &= p_L p_H^2 \xi_1(n_H, r) - p_L p_H \xi_2(n_H, r) - p_H p_L^2 \xi_1(n_L, r) + p_L p_H \xi_2(n_L, r) \\ &= p_L p_H [p_H \xi_1(n_H, r) - \xi_2(n_H, r) - p_L \xi_1(n_L, r) + \xi_2(n_L, r)] \\ &\geq p_L p_H \xi_1(n_L, r) \cdot (p_H - p_L) \geq \xi_{p_L p_H} (p_H - p_L), \end{aligned}$$

where the inequality follows from the guarantee of Lemma 5.2 that $p_L \xi_1(n_L, r) - \xi_2(n_L, r) \leq p_L \xi_1(n_H, r) - \xi_2(n_H, r)$. Noting that $\epsilon(i, j, r) = \frac{p_H - p_L}{p_L}$ we have:

$$\mathbb{E}[\epsilon(i, j, r + 2)] \geq \epsilon(i, j, r) + \frac{1}{2} \left(\frac{p_L p_H (p_H - p_L)}{p_L^2} \right) \geq \epsilon(i, j, r) \left(1 + \frac{p_H}{2} \right) \geq \epsilon(i, j, r) \left(1 + \frac{1}{2dk} \right). \quad \square$$

5.2.3 Changes in Populations of Nests over $O(k \log n)$ Rounds

We now show that once the population of a nest is very small, it will quickly drop to zero.

Lemma 5.8. *Let n_i be any nest with $p(i, r) \leq 1/(dk)$ in some round $r \in R_1$. Then, with probability at least $1 - 1/n^{c+4}$, $p(i, r') \leq 1/(dk)$ for all rounds $r' \in [r, r + 64(c + 4)k \log n]$.*

Proof. It suffices to show that if $p(i, r) \leq 1/(dk)$ then $p(i, r+2) \leq 1/(dk)$ with probability $1 - 1/n^{c+5}$. We can then union bound over all $O(k \log n)$ rounds to get that, with probability $1 - 1/n^{c+4}$, $p(i, r') \leq 1/(dk)$ for all rounds $r' \in [r, r + 64(c + 4)k \log n]$.

Consider two possible cases:

Case 1: $p(i, r) < 1/(2dk)$. Then, even if all ants successfully recruit, the nest cannot more than double in size, so $p(i, r + 2) < 1/(dk)$.

Case 2: $p(i, r) \geq 1/(2dk)$. The expected number of ants to recruit to nest n_i is $n \cdot p(i, r)^2 \geq n/(4d^2 k^2)$. By the assumption that $k \leq \sqrt{n/(8d^2(c+6) \log n)} = \mathcal{O}(\sqrt{n/\log n})$, this value is in $\Omega(\log n)$. By a Chernoff bound, with probability at least $1 - 1/n^{c+6}$, at most $2n \cdot p(i, r)^2 = n/(2d^2 k^2)$ ants choose to recruit.

Similarly, the expected the number of ants to be recruited from nest n_i is:

$$np(i, r) (1 - p(i, r)) \Sigma^2(r) \geq n \frac{1}{dk} \left(1 - \frac{1}{2dk} \right) \frac{1}{k} \geq \frac{n}{dk^2} \left(1 - \frac{1}{2dk} \right),$$

where in the second step we use the fact that $1/(dk) > p(i, r) \geq 1/(2dk)$ and $\Sigma^2(r) > 1/k$. This value is also in $\Omega(\log n)$.

Next, we would like to apply a Chernoff bound to the number of recruited ants from nest n_i ; however, recruitments are not independent, so we need to do a simple domination trick, similarly to the proof of the lower bound.

Let random variable Y_r^a have value 1 with probability $1/k(1 - 1/(2dk))$, and let Y_r be the sum of $n/(dk)$ such independent random variables. So, $\mathbb{E}[Y_r] = (n/(dk^2))(1 - 1/(2dk))$, and by a Chernoff bound, $P[Y_r < (3/2)(n/(dk^2))(1 - 1/(2dk))] < n^{-(c+6)}$. We can see that:

$$P[a \text{ is recruited} \mid \text{the choices of all other ants}] \geq (1-p(i, r))\Sigma^2(r) \geq \left(1 - \frac{1}{2dk}\right) \frac{1}{k} = P[Y_r^a = 1]$$

Therefore, by Lemma A.1, $P[\# \text{ of ants recruited from nest } n_i < x] < P[Y_r < x]$ for some $x \leq n/(dk)$. Plugging in $x = (3/2)(n/(dk^2))(1 - 1/(2dk))$, it follows that:

$$P[\# \text{ of ants recruited from nest } n_i < (3/2)(n/(dk^2))(1 - 1/(2dk))] < n^{-(c+6)}.$$

Therefore, with probability at least $1 - 1/n^{c+6}$, at least $(3/2)(n/(dk^2))(1 - 1/(2dk))$ ants will be recruited from nest n_i . Recall that, with probability at least $1 - 1/n^{c+6}$, at most $n/(2d^2k^2)$ ants from nest n_i choose to recruit. Since $d \geq 64$, it is easy to see that the number of ants recruiting is smaller than the number of ants recruited away from nest n_i , so with probability at least $1 - 1/n^{c+5}$, the total population of n_i decreases, indicating $p(i, r+2) < 1/(dk)$.

So overall, if $p(i, r) < 1/(dk)$, then, with probability at least $1 - 1/n^{c+4}$, $p(i, r') < 1/(dk)$ for all $r' \in [r, r + \mathcal{O}(k \log n)]$. \square

Lemma 5.9. *Let n_i be a nest with $p(i, r) \leq 1/(dk)$ in some round $r \in R_1$. Then, with probability at least $1 - 1/n^{c+3}$, $c(i, r') = 0$ for $r' = 64(c+4)k \log n$.*

Proof. We calculate the expected change in the number of ants in nest n_i by first calculating $\mathbb{E}[X_r^a]$ for some ant a in nest n_i and some round $r \in R_1$ such that $p(i, r) \leq 1/(dk)$.

Suppose ant a does not recruit. Let random variable $Y_r^{a'}$ have value 1 if ant a' successfully recruits ant a , and value 0 otherwise. By Lemma 2.1, it follows that $\mathbb{E}[Y_r^{a'}] \geq p(i', r)/(16n)$ where $i' = \ell(a', r)$. Therefore:

$$\begin{aligned} \mathbb{E}\left[\sum_{a' \neq a} Y_r^{a'}\right] &= \sum_{a' \neq a} \mathbb{E}[Y_r^{a'}] \geq \frac{1}{16n} \sum_{j=1}^k \sum_{\{a' \neq a \mid \ell(a', r) = j\}} p(j, r) \\ &\geq \frac{1}{16n} \left(\sum_{j=1}^k \sum_{\{a' \mid \ell(a', r) = j\}} p(j, r) \right) - \frac{1}{dk} \\ &\geq \frac{1}{16n} \left(\sum_{j=1}^k np(j, r)^2 \right) - \frac{1}{dk} \geq \frac{\Sigma^2(r)}{16} - \frac{1}{dk} \end{aligned}$$

Therefore, the expected value of X_r^a is:

$$\mathbb{E}[X_r^a] \leq \frac{1}{dk} - \left(1 - \frac{1}{dk}\right) \mathbb{E}\left[\sum_{\{a' \mid \ell(a', r) \neq i\}} Y_r^{a'}\right] = \frac{1}{dk} - \left(1 - \frac{1}{dk}\right) \left(\frac{\Sigma^2(r)}{16} - \frac{1}{dk}\right) \leq -\frac{1}{64k},$$

where the last step follows from the assumption that $d \geq 64$. Therefore,

$$\mathbb{E}[p(i, r+2)] = p(i, r) + \frac{1}{n} \sum_{\{a \mid \ell(a, r) = i\}} \mathbb{E}[X_r^a] \leq p(i, r)(1 + \mathbb{E}[X_r^a]) \leq p(i, r) \left(1 - \frac{1}{64k}\right)$$

By Lemma 5.8, $p(i, r') \leq 1/(dk)$ for all $r' \in [r, r + 64(c + 4)k \log n]$. Therefore, for $r' = 64(c + 4)k \log n$, it is true that $\mathbb{E}[p(i, r')] < 1/n^{c+3}$ and, by a Markov bound, nest n_i has at least one ant with probability at most $1/n^{c+3}$. Therefore, with probability at least $1 - 1/n^{c+3}$, $c(i, r') = 0$. \square

Next, we show that, for any pair of nests both with population proportions bounded below by $1/(dk)$, we can use Lemma 5.7 to argue that the populations of these nests diverge quickly. As soon as a nest drops below the $1/(dk)$ threshold we can use Lemma 5.9 to show that it will not be the winning nest.

Lemma 5.10. *Let n_i and n_j be two nests with $q(i) = q(j) = 1$. Then, for $r' = (6d+64(c+6))k \log n$, with probability at least $1 - 1/n^{c+2}$, at least one of the following is true: $c(i, r') = 0$ or $c(j, r') = 0$.*

Proof. Note that if at any point a nest has no ants in it, it remains having no ants thereafter. We consider two possible cases based on how many ants are in each nest in each round $r \in [1, 6dk \log n]$:

Case 1: In some round $r \in [1, 6dk \log n]$ either n_i or n_j has fewer than $n/(dk)$ ants. Then, by Lemma 5.9, with probability at least $1 - 1/n^{c+3}$, this nest has no ants committed to it after $64(c + 4)k \log n$ rounds.

Case 2: In every round $r \in [1, 6dk \log n]$ both n_i and n_j have at least $n/(dk)$ ants. We are going to show that, with probability at least $1 - 1/n^{c+3}$ this case does not happen. Suppose, in contradiction, that in every round $r \in [1, 6dk \log n]$ both n_i and n_j have at least $n/(dk)$ ants. First, we want to show that $\Delta p_L < p_L$, so that we can apply Lemma 5.7. Since both nests have at least $n/(dk)$ ants, it follows that $p_L \geq 1/(dk)$, so the number of ants recruiting for that nest is at least $n/(d^2k^2) = \Omega(\log n)$, by our assumption that $k \leq \sqrt{n/(8d^2(c+6)\log n)} = \mathcal{O}(\sqrt{n/\log n})$. By a Chernoff bound, with probability at least $1 - 1/n^{c+3}$, the number of recruiting ants is at most $2n/(d^2k^2)$. This indicates that, with probability at least $1 - 1/n^{c+3}$, not all ants from the nest with lower population recruit, and so $\Delta p_L < p_L$.

This means that by Lemma 5.7,

$$\mathbb{E}[\epsilon(i, j, 6dk \log n)] \geq \left(1 + \frac{1}{2dk}\right)^{6dk \log n} \mathbb{E}[\epsilon(i, j, 1)] \geq n^3 \cdot \mathbb{E}[\epsilon(i, j, 1)].$$

By Lemma 5.4, $\mathbb{E}[\epsilon(i, j, 1)] \geq 1/(3(n - 1))$. Therefore, after $6dk \log n$ rounds we have $\mathbb{E}[\epsilon(i, j, 6dk \log n)] \geq n$. However, this is a contradiction to the fact that $\epsilon(i, j, 6dk \log n) \leq n - 1$, by definition, and consequently $\mathbb{E}[\epsilon(i, j, 6dk \log n)] \leq n - 1$.

Each case holds with probability at least $1 - 1/n^{c+3}$, so union bounding them, we get that, with probability at least $1 - 1/n^{c+2}$, either $c(i, r') = 0$ or $c(j, r') = 0$. \square

Theorem 5.11. *With probability at least $1 - 1/n^c$, Algorithm 3 solves the HOUSEHUNTING problem in $\mathcal{O}(k \log n)$ rounds.*

Proof. By our bound on k ($k \leq \sqrt{n/(8d^2(c+6)\log n)} = \mathcal{O}(\sqrt{n/\log n})$), with probability at least $1 - 1/n^{c+1}$, in the first round of the algorithm (a round of searching), at least some ant will arrive at a nest with quality 1. Further, since only ants at nests with quality 1 choose to recruit, at any point, at least one ant is committed to a good nest.

By Lemma 5.10, with probability at least $1 - 1/n^{c+2}$, for each pair of nests n_i and n_j , at least one nest contains no ants by the end of $\mathcal{O}(k \log n)$ rounds. Union bounding over all, at most

$\binom{k}{2} < k^2 < n$ (by the bound on k), pairs of good nests, we conclude that, with probability at least $1 - 1/n^{c+1}$, after $\mathcal{O}(k \log n)$ rounds, each pair contains at least one nest with no ants. This can only be true if all nests have no ants (not possible) or if all ants are located at one good nest.

Finally, union bounding the initial search phase and the subsequent competition between nests, we get that with probability at least $1 - 1/n^c$, the house hunting problem is solved in $\mathcal{O}(k \log n)$ rounds. \square

6 Discussion and Future Work

Extensions to the Model For the sake of analysis, we have made many simplifying assumptions about the house-hunting process. We are confident that many of these assumptions can be weakened to make the model more realistic and natural. Some obvious modifications include assuming ants know only an approximation of n , allowing values of k larger than $\mathcal{O}(n/\log n)$, and allowing non-binary nest qualities, variability in the ants' quality sensing, along with some measure of algorithmic performance based on the quality of the chosen nest. Distinguishing between direct transport and tandem runs may also be interesting, paired with a more fine-grained runtime analysis.

Additionally, real ants can only assess nest quality and population approximately. For example, they may estimate nest size (one measure of quality) by randomly walking within the nest and counting how many times they cross their previous path [20]. They seem to estimate nest population by measuring encounter rates with other ants, with a higher encounter rate indicated a higher population at the nest [14, 21]. Adding noisy measurements to our model and designing algorithms that handle this noise would be a very interesting future direction. It may even be possible to explicitly model lower level behavior and implement subroutines for nest assessment, population measurement, recruitment, and search which give various runtime and error guarantees.

Extensions to the Algorithms We believe that Algorithm 3 may be a good starting point for work on more realistic house-hunting models. Below we discuss some interesting possible extensions to the algorithm. Some seem to simply require a more involved analysis, while others seem to require trade-offs in the algorithm's running time and its level of simplicity.

Improved running time: The $\mathcal{O}(k \log n)$ runtime of Algorithm 3 is required because, on average each nest initially contains n/k ants, so ants only recruit with probability $1/k$. $\mathcal{O}(k)$ time is required to amplify population gaps by a constant factor. Ideally, ants would all recruit with a probability lower bounded by a constant, but still linearly dependent on the nest populations. This would allow convergence in $\mathcal{O}(\log n)$ rounds. If ants keep track of the round number, they can map this to an estimate $\tilde{k}(r)$ of how many competing nests remain, allowing them to recruit at rate $\mathcal{O}(c(i, r)/n \cdot \tilde{k}(r))$. We believe that such a strategy should yield a relatively natural algorithm converging in $\mathcal{O}(\log^c n)$ rounds.

Non-binary nest qualities: Assuming a real-valued nest quality in the range $(0, 1)$ affects the correctness of Algorithm 3 because ants no longer have the notion of a good nest. However, it should be possible to incorporate the quality of the nest into the recruitment probability in order to make the algorithm converge to a high-quality nest, without significantly affecting runtime.

Approximate counting, nest assessment, and knowledge of n : Since the analysis of Algorithm 3 does not require each ant to recruit with a specific probability, but rather that the total number of ants choosing to recruit from a nest is proportional to its population, it should be resilient to noisy quality and population measurements. As long as ants have unbiased estimators of these values, we believe that Algorithm 3 should remain correct, perhaps with some runtime cost dependent on estimator variance.

Fault tolerance: Similarly, Algorithm 3 should support some degree of fault tolerance. A small number of ants suffering from crash-faults or even malicious faults, should not affect the overall populations of recruiting ants and the algorithm’s performance.

Asynchrony: Finally, note that Algorithm 3 currently works in synchronous rounds and relies on that assumption to get the correct number of ants at a given nest. However, we believe that, as long as the distribution of ants in candidate nests throughout time stays close to the distribution in the synchronous model, Algorithm 3 can be extended to work in a partially-synchronous model, potentially at the cost of some extra running time.

References

- [1] Karl Johan Aström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- [2] Fan Chung, Shirin Handjani, and Doug Jungreis. Generalizations of Polya’s urn problem. *Annals of combinatorics*, 7(2):141–153, 2003.
- [3] Alejandro Cornejo, Anna Dornhaus, Nancy Lynch, and Radhika Nagpal. Task allocation in ant colonies. In *Distributed Computing*, pages 46–60. Springer, 2014.
- [4] Adam L Cronin. Synergy between pheromone trails and quorum thresholds underlies consensus decisions in the ant *myrmecina nipponica*. *Behavioral Ecology and Sociobiology*, 67(10):1643–1651, 2013.
- [5] Benjamin Doerr and Anne Auger. *Theory of Randomized Search Heuristics*. World Scientific, 2011.
- [6] A Dornhaus and NR Franks. Colony size affects collective decision-making in the ant *temnothorax albipennis*. *Insectes sociaux*, 53(4):420–427, 2006.
- [7] Yuval Emek, Tobias Langner, Jara Uitto, and Roger Wattenhofer. Solving the ANTS problem with asynchronous finite state machines. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, 2014.
- [8] Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Efficient and noise-resilient rumor-spreading in large, anonymous populations. *arXiv preprint arXiv:1311.3425*, 2013.
- [9] Ofer Feinerman and Amos Korman. Memory lower bounds for randomized collaborative search and implications for biology. In *Distributed Computing*, pages 61–75. Springer, 2012.
- [10] Ofer Feinerman and Amos Korman. Theoretical distributed computing meets biology: A review. In *Distributed Computing and Internet Technology*, pages 1–18. Springer, 2013.
- [11] Ofer Feinerman, Amos Korman, Zvi Lotker, and Jean-Sébastien Sereni. Collaborative search on the plane without communication. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, pages 77–86. ACM, 2012.
- [12] 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.

- [13] 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*, 72(3):611–616, 2006.
- [14] Deborah M Gordon. *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.
- [15] Christiane IM 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.
- [16] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 565–574. IEEE, 2000.
- [17] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.
- [18] Christoph Lenzen, Nancy Lynch, Calvin Newport, and Tsvetomira Radeva. Trade-offs between selection complexity and performance when searching the plane without communication. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, 2014.
- [19] E Mallon, S Pratt, and N Franks. Individual and collective decision-making during nest site selection by the ant *Leptothorax albigipennis*. *Behavioral Ecology and Sociobiology*, 50(4):352–359, 2001.
- [20] Eamonn B Mallon and Nigel R Franks. Ants estimate area using Buffon’s needle. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1445):765–770, 2000.
- [21] S.C. Pratt. Nest site choice in social insects. In Michael D. Breed and Janice Moore, editors, *Encyclopedia of Animal Behavior*, pages 534 – 540. Academic Press, Oxford, 2010.
- [22] Stephen C Pratt. Quorum sensing by encounter rates in the ant *Temnothorax albigipennis*. *Behavioral Ecology*, 16(2):488–496, 2005.
- [23] 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 albigipennis*. *Behavioral Ecology and Sociobiology*, 52(2):117–127, 2002.
- [24] Stephen C Pratt and David JT Sumpter. A tunable algorithm for collective decision-making. *Proceedings of the National Academy of Sciences*, 103(43):15906–15910, 2006.
- [25] Takao Sasaki and Stephen C Pratt. Emergence of group rationality from irrational individuals. *Behavioral Ecology*, 22(2):276–281, 2011.
- [26] Takao Sasaki and Stephen C Pratt. Ants learn to rely on more informative attributes during decision-making. *Biology letters*, 9(6):20130667, 2013.

A Math Preliminaries

Lemma A.1. *Let X_1, \dots, X_n be arbitrary binary random variables. Let X_1^*, \dots, X_n^* be random variables that are mutually independent and such that for all i , X_i^* is independent of X_1, \dots, X_{i-1} . Assume that for all i and all $x_1, \dots, x_{i-1} \in \{0, 1\}$,*

$$P[X_i = 1 | X_1 = x_1, \dots, X_{i-1} = x_{i-1}] \geq P[X_i^* = 1]$$

Then, for all $k \geq 0$, we have,

$$P \left[\sum_{i=1}^n X_i < k \right] \leq P \left[\sum_{i=1}^n X_i^* < k \right],$$

and the latter term can be bounded by Chernoff bounds for independent random variables.