Distributed House-Hunting in Ant Colonies*

Mohsen Ghaffari
MIT
Cambridge, MA, USA
ghaffari@csail.mit.edu

Cameron Musco
MIT
Cambridge, MA, USA
cnmusco@csail.mit.edu

Tsvetomira Radeva
MIT
Cambridge, MA, USA
radeva@csail.mit.edu

Nancy Lynch
MIT
Cambridge, MA, USA
lynch@csail.mit.edu

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 relocates to a new nest. The task of identifying and evaluating the quality of potential new nests is distributed among all ants. They must additionally reach consensus on a final nest choice and transport the full colony 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.

Categories and Subject Descriptors

F.2.m [**Analysis of Algorithms and Problem Complexity**]: Miscellaneous; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic Algorithms*

Keywords

distributed algorithms; randomized algorithms; social insect colonies; biology-inspired algorithms; house-hunting; *Temnothorax* ants

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, 19] 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 ants' 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. Researchers have studied ants' preferences and capabilities [16, 27], the general structure of the algorithms used by the ants [20, 23], variations due to colony size [13], and trade-offs between the speed and accuracy of the process [25]. 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 problem of consensus [12, 18]. 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

---

[*] This work is sponsored in part by AFOSR contract number FA9550-13-1-0042 and NSF award numbers 0939370-CCF and CCF-AF-0937274.

PODC'15, July 21–23, 2015, Donostia-San Sebastián, Spain.

ACM 978-1-4503-3617-8 /15/07 ...$15.00.
DOI: http://dx.doi.org/10.1145/2767386.2767426.

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 [16, 27] and, in general, it is assumed that nest assessments by an individual ant are not always precise or rational [26]. After some time spent assessing different nests, going back to the old nest and searching for new nests, an ant becomes sufficiently satisfied with a nest and proceeds 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 new 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* [24, 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 conclude the house-hunting process by bringing the full 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 [17].

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

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 nests, denoted n_i for $i \in \{1, ..., k\}$. Each nest n_i is assigned a quality $q(i) \in Q$, for some set Q. Throughout this paper we let $Q = \{0, 1\}$, where quality 0 indicates an unsuitable nest, and 1 a suitable one. 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 [4, 6].

The general behavior of the state machines is unrestricted but their interactions with the environment and other ants are limited to the 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 $\ell(a,r) \in \{0,1,...,k\}$; 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 in round r. Let $A(i,r) = \{a|\ell(a,r) = i\}$ denote the set of all ants located at nest n_i at the end of round r and let $c(i,r) = |A(i,r)|$.

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

search(): Returns a triple $< i, q(i), c(i,r) >$ where i is chosen uniformly at random from $\{1, \cdots, k\}$. 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 $i \in \{1, \cdots, k\}$ 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 $i \in \{1, \cdots, k\}$ such that there exists a round $r' < r$ in which $\ell(a,r') = i$. Returns a pair $< j, c(0,r) >$ where $j \in \{1, \cdots, k\}$. 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*$(1,\cdot)$.

```
Return value $j$ of recruit$(\cdot,\cdot)$ for all ants $a \in R$
$M$: a set of pairs $(a, a')$ of ants, initially $\emptyset$
for $i = 1$ to $|R|$ do
    if $a_{P(i)} \in S$ and $(\cdot, a_{P(i)}) \notin M$ then
        $a'$ := uniform random ant from $R$
        if $(a', \cdot) \notin M$ and $(\cdot, a') \notin M$ then
            $M := M \cup (a_{P(i)}, a')$
for $i = 1$ to $|R|$ do
    if $(a^*, a_{P(i)}) \in M$ then
        return $j$ to ant $a_{P(i)}$ where $j$ is input to
        recruit$(1, j)$ called by $a^*$
    else return $j$ to ant $a_{P(i)}$ where $j$ is input to
    recruit$(\cdot, j)$ called by $a_{P(i)}$
```

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. The algorithm 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 first element of a pair 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 [22], and since we focus on asymptotic behavior, we do not model this action separately.

Next, we prove a general statement about the probability that a recruiting ant is successful.

LEMMA 2.1. *Let a be any ant that executes recruit$(1,\cdot)$ in some round r, and assume $c(0,r) \geq 2$. Then, with probability at least $1/16$, $(a,\cdot) \in M$.*

PROOF. With probability 1/2 ant a is located in the first half of the random permutation. Conditioning on this, the probability that a is not recruited by some ant with higher precedence in round r is at least: $\left(1 - \frac{1}{c(0,r)}\right)^{c(0,r)/2} \geq 1/2$, where $c(0,r) \geq 2$ is the number of ants at the home nest. Further, with probability 1/2, ant a chooses to recruit an ant a' located in the second half of the permutation, and with probability $\geq \left(1 - \frac{1}{c(0,r)}\right)^{c(0,r)/2} \geq 1/2$, a' is not recruited by any ant with higher precedence than a. So, in total a recruits successfully with probability at least 1/16. ☐

Problem Statement: An algorithm $\mathcal{A}$ solves HOUSE-HUNTING with k nests in $T \in \mathbb{N}$ rounds with probability $1 - \delta$, for $0 \leq \delta \leq 1$, if with probability $1 - \delta$, taken over all executions of $\mathcal{A}$, there exists a nest $i \in \{1, \cdots, k\}$ with $q(i) = 1$ and $\ell(a,r) = i$ for all ants a and 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 is similar to lower bounds on spreading a rumor in a complete graph [17] 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 that 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 knows 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 is guaranteed to remain ignorant. In the second case, ant a calls **recruit**(). Let $X_r \leq c(0,r)$ 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 $\mathbf{recruit(1, w)}$. The probability that ant a is ignorant after round r is at least $\left(1 - \frac{1}{c(0,r)}\right)^{c(0,r)} \geq \frac{1}{4}$ when $c(0, r) \geq 2$. 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 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, at least one ant is ignorant.

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

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 $(1/4)^r \geq 12c/\sqrt{n}$. 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. These indicator random variables are not independent, so we cannot directly apply a Chernoff bound. We instead show that their sum is bounded by defining random variables that dominate the $\bar{X}_r^a$ variables.

Let 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 $\Pr[Y_r < 6c\sqrt{n}] \leq 1/n^c$.

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,

$$\Pr[\bar{X}_r^a = 1 | \text{all } \bar{X}_r^{a'} \text{ for } a' \text{ before } a \text{ in } P] \geq (1/4)^r,$$

where P is the random permutation used to model the recruitment process. So, by the definition of Y_r^a:

$$\Pr[\bar{X}_r^a = 1 | \bar{X}_r^{a'} \text{ for all } a' \text{ before } a \text{ in } P] \geq \frac{1}{4^r} = \Pr[Y_r^a = 1].$$

By Lemma 1.18 in [5][Chapter 1], $\Pr[\bar{X}_r < x] \leq \Pr[Y_r < x]$ for any $x \leq n$. In particular, for $r = (\log_4 n)/2 - \log_4(12c)$, it follows that $\Pr[\bar{X}_r < 6c\sqrt{n}] \leq \Pr[Y_r < 6c\sqrt{n}] \leq \frac{1}{n^c}$.

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 remains 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 in a 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 HouseHunting is solvable in optimal time in our model.

4.1 Algorithm Pseudocode and Description

The pseudocode of the algorithm is presented in Algorithm 1. 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.

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.

The *search* subroutine is executed only once, during the first round, and the $final$ subroutine is used by active ants 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 of these subroutines takes exactly four rounds, labeled as R1, R2, R3, or R4 in the pseudocode. The subroutines are carefully scheduled such 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. The *active* and *passive* subroutines are padded with $recruit(0, \cdot)$ and $go(nest)$ calls to achieve such interleaving (lines 12, 17–18, 32–33, 39 are such padding rounds).

Search (lines 6–11): In the first round, each ant searches for a nest. If the nest has quality 0, the ant moves to the *passive* state; otherwise, it moves to the *active* state.

Passive (lines 12–19): An ant is passive if it either found a bad nest or its nest has dropped-out. 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 passive ant can only be recruited by an ant in the $final$ state. Once recruited, the passive ant moves to the $final$ state.

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 $recruit(1, nest)$ coincides with the call to $recruit(0, nest)$ 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 to its nest by executing $recruit(1, nest)$. It then goes to the resulting nest to assess its population. Based on the nest ($nest_t$) and population ($count_t$), we consider three cases:

Algorithm 1: Optimal HouseHunting Algorithm

```
1  $state$ : $\{search, active, passive, final\}$, initially $search$
2  $nest$ : nest index $i \in \{0, \cdots, k\}$, initially 0
3  $count$ : an integer in $\{0, \cdots, n\}$, initially 0
4  $quality$ : a boolean in $\{0, 1\}$, initially 0
5  case $state = search$
6      $< nest, count, quality >:=$ search()
7      if $quality = 0$ then
8          $state := passive$
9      else
10         $state := active$
11 case $state = passive$
12     go(nest) (R1)
13     $< nest_t, \cdot >:=$ recruit(0, nest) (R2)
14     if $nest_t \neq nest$ then
15         $nest := nest_t$
16         $state := final$
17     go(nest) (R3)
18     go(nest) (R4)
19 case $state = final$
20     $< nest, \cdot >:=$ recruit(1, nest) (R1, R2, R3, R4)
21 case $state = active$
22     $< nest_t, \cdot >:=$ recruit(1, nest) (R1)
23     $count_t :=$ go($nest_t$) (R2)
24     if $(nest_t = nest)$ and $(count_t \geq count)$ then
25         $count := count_t$
26         go(nest) (R3)
27         $< \cdot, count_h >:=$ recruit(0, nest) (R4)
28         if $count_h = count$ then
29             $state := final$
30     else if $(nest_t = nest)$ and $(count_t < count)$ then
31         $state := passive$
32         recruit(0, nest) (R3)
33         go(nest) (R4)
34     else
35         $nest := nest_t$
36         $count_n :=$ go(nest) (R3)
37         if $count_n < count_t$ then
38             $state := passive$
39         go(nest) (R4)
```

Case 1 (lines 25–31): $nest_t = nest$ and the number of ants in *nest* has not decreased, so the nest remains competing. 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, if the number of ants at the home nest is equal to the number of ants at the competing nest, all ants have been recruited to a single winning nest and the ant switches to the *final* state.

Case 2 (lines 32–36): $nest_t = nest$ but the population has decreased so the nest drops out. The ant becomes passive and spends a round at the home nest, which coincides with the round active ants spend at competing nests in Case 1.

Case 3 (lines 37–42): $nest_t \neq nest$, so the ant got recruited to another nest. Although it already knows the population ($count_t$) of the new nest, the ant updates that value ($count_n$) to determine if 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 nest; if $count_t > count_n$, the nest is dropping out because the ants in Case 2 already determined a decrease in population and are spending this round at the home nest.

4.2 Correctness Proof and Time Bound

As written, Algorithm 1 never terminates; after all ants are in the *final* state, they continuously recruit each other. 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. For simplicity we just assume the algorithm terminates once all ants are in the *final* state.

Proof Overview: The proof of correctness and time bound analysis of Algorithm 1 are structured as follows. In Lemma 4.1, we show 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 1 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_1 be the set of round numbers r such that $r \in R_1$ iff $(r-1) \mod 4 = 1$; these rounds, labeled $R1$ in the pseudocode, are exactly the rounds in which only ants from active nests are located at the home nest and recruiting each other. Let $C(i, r)$ for each $r \in R_1$ and each nest n_i denote the set of active ants for which $nest = n_i$. This implies that $|\cup_{i \in [1,k]} C(i, r)| = c(0, r)$ for $r \in R_1$.

Let random variable X_r^a, for each ant a and round $r \in R_1$, take on values in $\{-1, 0, 1\}$. 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 $\mathbf{o}_r$ be a vector of length $c(0, r)$, where $\mathbf{o}_r(a) = X_r^a$. Let random variable Y_r^i denote the change in the number of ants at nest n_i after round r, where $r \in R_1$. $Y_r^i = \sum_{a \in C(i,r)} X_r^a$.

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

Recall that, as specified by the recruitment model, Y_r^i is the result of a random 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.

Lemma 4.1. *Let nest n_i be a competing nest in round $r \in R_1$. Then,* $\Pr[Y_r^i < 0] = \Pr[Y_r^i > 0]$.

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 -1s and 1s, it is possible to choose some (non-random) permutation P' that swaps the locations of the -1s and 1s 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 $\Pr[Y_r^i < 0] = \Pr[Y_r^i > 0]$. □

Lemma 4.2. *Let nest n_i be a competing nest in round $r \in R_1$. If $|C(i,r)| < c(0,r)$, then $\Pr[Y_r^i < 0] \geq 1/66$.*

Proof. Let P_1 be the uniform random permutation in in the recruitment process that determines Y_r^i. Let permutation P_2 swap the position of a fixed ant a^* such that $a^* \notin C(i,r)$ with the position of an ant a chosen uniformly from $C(i,r)$. The existence of such an ant is guaranteed by the assumption that $|C(i,r)| < c(0,r)$.

By Lemma 2.1, $\Pr[X_r^{a^*} = 1] \geq 1/16$. 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,

$$\Pr[Y_r^i = 0] \quad \leq 1 - \tfrac{1}{32} \Pr[Y_r^i = 0] \text{ so } \Pr[Y_r^i = 0] \leq \tfrac{32}{33}.$$

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

Theorem 4.3. *For any constant $c > 0$, with probability at least $1 - 1/n^c$, Algorithm 1 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 at each good nest is n/k. Assuming $k \leq n/(12(c+1)\log n) = \mathcal{O}(n/\log n)$, by a Chernoff bound, with probability at least $1 - 1/n^{c+1}$, at least $n/(2k) = \Omega(\log n)$ ants visits each good nest.

Let k_r be a random variable denoting the number of competing nests in round $r \in R_1$. Suppose $k_r > 1$, and so $|C(i,r)| < c(0,r)$. By Lemma 4.2, $\Pr[Y_r^i < 0] \geq 1/66$ 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 $r = \log_{66/65} k + (c+1)\log_{66/65} n = \mathcal{O}(\log n)$, it follows that $\mathbb{E}[k_r] = 1/n^{c+1}$. By a Markov bound, $\Pr[k_r \geq 1] \leq 1/n^{c+1}$, so with probability at least $1 - 1/n^{c+1}$, $k_r \leq 1$. It is never possible for all nests to decrease in population, so for each round r, $k_r \geq 1$. So, overall, after $\mathcal{O}(\log n)$ rounds, with probability $1 - 1/n^{c+1}$, there is exactly one competing nest.

Once there is only one competing nest in some round $r \in R_1$, all ants from that nest switch to the *final* state and start recruiting the passive ants every fourth round. The recruited ants transition to the final state, and in at most $\mathcal{O}(\log n)$ rounds all ants are recruited 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 located at a single good nest in $\mathcal{O}(\log n)$ rounds. ☐

5. SIMPLE ALGORITHM

In this section, we present a simple algorithm 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].

Algorithm 2: Simple House-Hunting

```
$state$ : $\{active, passive\}$, initially $active$
$< nest, count, quality >:=$ search()
if $quality = 0$ then
    $state := passive$
case $state = active$
    $b := 1$ with probability $count/n$, 0 otherwise
    $nest :=$ recruit(b, nest)
    $count :=$ go(nest)
case $state = passive$
    $nest_t :=$ recruit(0, nest)
    if $nest_t \neq nest$ then
        $state := active$
        $nest := nest_t$
        $count :=$ go(nest)
```

5.1 Algorithm Description and Pseudocode

In each round of Algorithm 2, each ant can be in one of two states: *active* or *passve*, 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**$(\mathbf{1}, \cdot)$ or passive **recruit**$(\mathbf{0}, \cdot)$), 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 becomes active again. When an active ant gets recruited to a different nest, it starts recruiting to that new nest.

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 proof of correctness and runtime analysis of the algorithm are structured as follows. In Lemmas 5.1, 5.2, and 5.3, we show some basic bounds on the expected number of ants recruited and the change of the value of $p(i,r)$ for a single nest in each round. 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 the widening population gaps lead to only a single nest remaining 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 $d \geq 64$. In the analysis, we 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 is not necessary.

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. So, 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 Nest in One Round*

We first study how the population of a single nest changes in a 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. So we expect a total change of $p(i, r)(p(i, r) - \Sigma^2(r))$. Qualitatively, we show this in Lemma 5.3, up to constant factors; the main difficulty is handling dependencies in the recruiting process. Before proving Lemma 5.3, we give 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.

The following lemma applies to an ant a recruiting with any probability p in a given round r. 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 for ants from different nests in 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 a random variable indicating whether some ant a chooses to recruit (executes $\mathbf{recruit(1, \cdot)}$) 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 $(P, \{A_r^{a'} | a' \neq a\}, \{1, \cdots, n\}^n)$ and it takes on values from the set $\mathcal{B}$ of all such triples. The expected value of X_r^a is:

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

Fix some value $B = b$. We consider several cases based on all the possible ways the value of A_r^a can affect X_r^a. We can immediately eliminate some of these cases. 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_x^b be the probability of case x occurring given $B = b$. The expected value of X_r^a is:

$$\begin{aligned}&\sum_{b \in \mathcal{B}} \Pr[B = b] \cdot \left(p\left(p_2^b + p_4^b - p_1^b\right) + (1 - p)\left(-p_1^b - p_2^b\right)\right) \\ &= \sum_{b \in \mathcal{B}} \Pr[B = b] \left(p\left(2p_2^b + p_4^b\right) - \left(p_1^b + p_2^b\right)\right) \\ &= p\mathbb{E}_B\left[2p_2^b + p_4^b\right] - \mathbb{E}_B\left[p_1^b + p_3^b\right].\end{aligned}$$

Now, we need to show that $\mathbb{E}_B\left[2p_2^b + p_4^b\right] \geq \xi$ for some fixed constant $\xi > 0$. By Lemma 2.1, each recruiting ant succeeds with constant probability. 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, $\mathbb{E}[X_r^a] = p\xi_1(i, r) - \xi_2(i, r)$, for constants $\xi_1(i, r) = \mathbb{E}_B[2p_2^b + p_4^b] \geq \xi > 0$ and $\xi_2(i, r) = \mathbb{E}_B[p_1^b + p_2^b]$. $\square$

Note that the values of $\xi_1(i, r)$ and $\xi_2(i, r)$ are slightly different for different nests. 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. In our full version [14], we prove a bound on the $\xi_1(i, r)$ and $\xi_2(i, r)$ values of two different nests that will be useful in later proofs.

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, $\xi_1(i, r) \cdot p(i, r) - \xi_2(i, r) \leq \xi_1(j, r) \cdot p(i, r) - \xi_2(j, r)$.*

It is simple to use Lemma 5.1 to calculate the expected change in $p(i, r)$. The calculation is given in [14].

Lemma 5.3. *For each nest n_i and each round $r \in R_1$, $\mathbb{E}[p(i, r + 2)] = p(i, r) \cdot [1 + \xi_1(i, r) \cdot p(i, r) - \xi_2(i, r)]$.*

Lemma 5.3 shows that the population change of a nest depends *quadratically* on $p(i, r)$ so larger nests will '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 measure of the population gap between two nests.

Definition 1. *For any 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. Define:*

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

$\epsilon(i,j,r)$ is the relative population gap between the larger and smaller of nests n_i and n_j in round r. In [14] we show:

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

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 bound this ratio by a linear function, and use Lemma 5.3 to compute its expected change. We first show how to use the Taylor series expansion in our setting. In [14] we compute:

LEMMA 5.5. *For 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:*

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

For simplicity of notation, write $p_H = p(n_H(i,j,r),r)$, $p_L = p(n_L(i,j,r),r)$. Also, $\Delta p_H = p(n_H(i,j,r),r+2) - p(n_H(i,j,r),r)$, $\Delta p_L = p(n_L(i,j,r),r+2) - p(n_L(i,j,r),r)$ 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 $\Delta p_L < p_L$ in 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:

$$\Delta\left(\frac{p_H}{p_L}\right) = \left(\frac{\Delta p_H}{p_L} - \frac{p_H \Delta p_L}{p_L^2}\right) \sum_{i=0}^{\infty} \left(-\frac{\Delta p_L}{p_L}\right)^i.$$

The value $\frac{\Delta p_L}{p_L}$ is strictly less than 1 by the assumption that $\Delta p_L < p_L$, so the infinite sum above converges to $\frac{1}{1+\Delta p_L/p_L} \geq 1/2$. $\square$

We now use $\Delta(p_H/p_L)$ to calculate the expected change in $\epsilon(i,j,r)$ for two nests after one recruitment round.

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, $\mathbb{E}\left[\epsilon(i,j,r+2)\right] \geq (1+\xi/(2dk))\,\mathbb{E}\left[\epsilon(i,j,r)\right]$ for fixed constant $\xi > 0$.*

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 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, $\epsilon(i,j,r+2) = \frac{p(n_H(i,j,r),r+2)}{p(n_L(i,j,r),r+2)} - 1$. Alternatively, 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. Fixing $p(i,r)$ and $p(j,r)$, we have:

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

So we have reduced our question of how we expect the *absolute difference* between the nests to change to the question of how we expect the population ratio to change. By Lemma 5.6, it follows that:

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

By Lemma 5.3 we have:

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

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:

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

5.2.3 Changes in Populations of Multiple 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. We first give a lemma, proven in [14], that once the population of a nest drops below a certain threshold, ants will recruit at a slow enough rate that the population will remain small for a number of rounds. We then show that over these rounds, the population will in fact drop to 0 with high probability.

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 $r' \in [r, r + 64(c+4)k\log n)]$.*

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:

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

Therefore, the expected value of X_r^a is:

$$\mathbb{E}[X_r^a] \le \frac{1}{dk} - \left(1 - \frac{1}{dk}\right)\mathbb{E}\left[\sum_{a' \notin A(j,r)} Y_r^{a'}\right]$$
$$= \frac{1}{dk} - \left(1 - \frac{1}{dk}\right)\left(\frac{\Sigma^2(r)}{16} - \frac{1}{dk}\right) \le -\frac{1}{64k},$$

where the bound follows from the assumption $d \ge 64$. So $\mathbb{E}[p(i, r+2)] = p(i,r)(1 + \mathbb{E}[X_r^a]) \le p(i,r)\left(1 - 1/64k\right)$.

By Lemma 5.8, $p(i, r') \le 1/(dk)$ 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 $\mathbb{E}[p(i,r')] \le 1/n^{c+3}$ 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 $1 - 1/n^{c+3}$, $c(i, r') = 0$. ☐

Next, for any pair of nests, both with population proportions $\ge 1/(dk)$, 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 $q(i) = q(j) = 1$ and $\xi > 0$ be a fixed constant. For $r' = (6d/\xi + 64(c+6))k\log n$, 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 $r \in [1, 6dk\log n/\xi)]$:

Case 1: In some round $r \in [1, 6dk\log n/\xi)]$ either n_i of 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 after $64(c+4)k\log n$ rounds.

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

$$\mathbb{E}[\epsilon(i,j,6dk\log n/\xi)] \ge [1 + \xi/(2dk)]^{6dk\log n/\xi}\,\mathbb{E}[\epsilon(i,j,1)]$$
$$\ge n^3 \cdot \mathbb{E}[\epsilon(i,j,1)].$$

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

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 $c(j, r') = 0$. ☐

Theorem 5.11. *With probability at least $1 - 1/n^c$, Algorithm 2 solves* HouseHunting *in $\mathcal{O}(k \log n)$ rounds.*

Proof. By our bound on k ($k \le \sqrt{n/(8d^2(c+6)\log n)} = \mathcal{O}(\sqrt{n/\log n})$), 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. So, with high probability, since only ants at nests with quality 1 recruit, there is always at least one ant recruiting 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.

Union bounding over the initial search phase and subsequent competition between nests, with probability at least $1 - 1/n^c$, HouseHunting is solved in $\mathcal{O}(k\log n)$ rounds. ☐

6. DISCUSSION AND FUTURE WORK

Extensions to the Model: For the analysis, we make many simplifying assumptions about the house-hunting process. We are confident that some of these assumptions can be weakened. Possible changes include assuming only approximate knowledge of n, allowing arbitrary values of k, and allowing non-binary nest qualities along with some measure of performance based on the chosen nest quality. Distinguishing between direct transport and tandem runs may also be interesting, paired with more fine-grained runtime analysis.

Additionally, real ants can only assess nest quality and population approximately. For example, they may estimate nest area by random walking inside the nest and counting how often they cross their own path [21]. They seem to estimate nest population by tracking encounter rates with other ants, with more encounters indicating higher population [15, 22]. Adding noisy measurements to our model and designing algorithms that handle this noise would be very interesting. It may even be possible to explicitly model low level behavior and implement modular algorithms for nest assessment, population measurement, recruitment, and search that give various runtime and error guarantees.

Extensions to the Algorithms: We believe that Algorithm 2 may be a good starting point for work on more realistic house-hunting models. Below we discuss some possible extensions to the algorithm. Some seem to simply require a more sophisticated analysis, while others 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 2 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, allowing 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 $\mathcal{O}(\text{polylog}\, n)$ rounds.

Non-binary nest qualities: Assuming a real-valued nest quality in the range $(0,1)$ affects Algorithm 2 because ants no longer have the notion of a good nest. However, it should be possible to incorporate nest quality 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: The analysis of Algorithm 2 does not require each ant to recruit with a specific probability, but just that the total number of ants recruiting to a nest is proportional to its population. So, it should be resilient to noisy quality and population measurements. If ants have unbiased estimators of these values, we believe the algorithm should remain correct, with some runtime dependence on estimator variance.

Fault tolerance: Similarly, Algorithm 2 should be fault tolerant. 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, Algorithm 2 currently works in synchronous rounds and relies on this 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 over time stays close to the distribution in the synchronous model, Algorithm 2 can be extended to work in a partially-synchronous model, potentially at some runtime cost.

7. 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*, 2014.

[8] Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 114–123, 2014.

[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 ACM Symposium on Principles of Distributed Computing*, 2012.

[12] Michael Fischer, Nancy Lynch, and Michael Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 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] Mohsen Ghaffari, Cameron Musco, Tsvetomira Radeva, and Nancy Lynch. Distributed house-hunting in ant colonies. *Preprint. arXiv:1505.03799*, 2015.

[15] Deborah M Gordon. *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.

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

[17] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[18] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 1998.

[19] 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 ACM Symposium on Principles of Distributed Computing*, 2014.

[20] E Mallon, S Pratt, and N Franks. Individual and collective decision-making during nest site selection by the ant Leptothorax albipennis. *Behavioral Ecology and Sociobiology*, 50(4):352–359, 2001.

[21] 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.

[22] 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.

[23] S.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.

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

[25] 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.

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

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