ADAPTIVE FORAGING IN ROBOTIC SWARMS

by

Adam Lein
B.Sc., Simon Fraser University, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2010

APPROVAL

Name: Adam Lein

Degree: Master of Science

Title of thesis: Adaptive foraging in robotic swarms

Examining Committee: Dr. Andrei Bulatov, Associate Professor
School of Computing Science
Chair

Dr. Richard Vaughan, Associate Professor
School of Computing Science
Senior Supervisor

Dr. Brian Funt, Professor
School of Computing Science
Supervisor

Dr. Sasha Fedorova, Assistant Professor
School of Computing Science
Examiner

Date Approved:

Abstract

Foraging is a canonical problem in robotics as it can be used to represent many other tasks. Bucket-brigading is a known control strategy for reducing interference in large swarms of robots.

In this thesis, we examine the foraging problem and bucket-brigading solution from the perspective of the spatial distributions of robots and pucks in the foraging task. We allow these distributions to inform the robots' foraging behavior. We give a variation on bucket-brigading—adaptive ranging—in which the controller adapts to varying and nonuniform robot distributions. We provide a second variation—relocation—in which the swarm adapts to varying and nonuniform puck distributions. We describe a custom swarm foraging simulation tool, with which we show that these enhancements outperform the conventional bucket-brigading approach.

We discuss robotic foraging in the context of the ideal free distribution from behavioral ecology, and propose a statistical test for measuring how well the distributions are matched.

To my mother and father

Acknowledgements

I would like to acknowledge the patience, sound advice, and experience of my supervisor, Dr. Richard Vaughan, as well as the encouragement and support I received from my colleages in the Autonomy Lab.

I also thank Dr. Alexandra Fedorova and Dr. Brian Funt for their professional criticism and helpful suggestions in the final stages of this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The *foraging problem* or *foraging task*, as the term is used in this work, is a task in which one or more agents must find and collect spatially distributed target objects. The term "central place foraging" introduces an additional requirement that foragers deliver collected objects to a "home area". Specific instantiations of this problem may place other constraints, such as unknown starting locations of the workers or their work sites, unknown distributions of target objects, restrictions on the agents' ability to sense their environment, limited time or energy, *etc.* The foraging task is considered canonical in cooperative multi-robot research[5].

Research in foraging can address any of several aspects of the problem: general strategies for where and how to search on a large scale (for example, a broad sweep of the search area *vs.* sequential thorough searches of small regions), local strategies of how the agent should move in order to conduct its search (moving in a spiral pattern *vs.* an alternating back-and-forth scan *vs.* a random walk), problems that arise as obstacles or noise are introduced, and others. In the case when the agents are physically instantiated as robots, the engineering requirements of the robots (energy, localization, sensing and manipulation of target objects) are also studied. All of these topics are common to single- and multi-agent foraging. However, having multiple agents foraging in the same space introduces a new dimension to the problem: the interaction among agents.

Interaction among agents has both positive and negative aspects. Intuitively, bringing in more workers to work on the same job will increase productively, but only up to a point. In addition to spending time and energy on the task at hand (searching, carrying target objects around, etc.), agents in multiple-agent tasks suffer from interference in that they compete for resources (generally, space and objects to collect).

1.1 Goal and motivation

In this thesis, we will examine and extend a well-known solution to the foraging problem that is specifically designed to mitigate interference: bucket-brigading. Following Shell[27], we will define bucket-brigading as a foraging strategy in which pucks are transfered from their original position to the home zone in multiple stages, in each of which a robot carries the puck from where it is found a predetermined distance towards the home zone. We will then describe two enhancements (in Chapters 4 and 5) to the bucket-brigading algorithm given by [27].

In this work we will frequently discuss two distributions that essentially define the state of the foraging problem at any given time. First, there is the distribution of robots, which determines the amount of spatial interference among robots in that regions with greater densities of robots will experience more interference. Second there is the distribution of pucks, which determines the suitability of the environment for foraging in that greater densities of pucks will provide a greater reward to foragers.

The unifying idea in this work is the recognition, first that the actions of the foragers affects these distributions, and second that the nature of the distributions informs the robots' future foraging behavior. Chapter 4 follows the work of Shell and deals solely with the issue of the distribution of robots and effects of interference. Chapter 5 extends this by also considering the distribution of pucks and finding more rewarding work sites for robots.

1.2 Contributions

This thesis makes novel contributions to the field of multi-robot research in general and foraging robots in particular:

- A bucket-brigading algorithm that adapts to different degrees of interference between robots.
- A bucket-brigading algorithm that adapts to nonuniform distributions of target objects.
- Identification of a formal, statistical test for determining the degree of similarity between the distributions of robots and pucks.
- A simulator, specifically designed to test foraging strategies in large swarms of robots.

1.3 Thesis outline

The purpose of this work is to examine the foraging problem, with specific focus on the phenomena unique to multi-robot systems, especially interference and optimal work allocation. We discuss previous solutions and propose novel algorithms of our own to improve on existing methods. Finally, we present an analysis of these problems and make recommendations for future progress in the subfield of foraging.

- Chapter 2, **Foraging background**: an overview of the state of research in foraging from two perspectives: natural systems (ecology and ethology) and artificial systems (robotics), and an outline of how this work fits into the present body of knowledge.

- Chapter 3, **Simulator**: a detailed description of the software used to produce the experimental results cited in this work, including motivation and technical details. Along with making the source code of this package available in the public domain, this section enables other researchers to truly reproduce these results.

- Chapter 4, **Adaptive ranging**: a summary of the first adaptation, that of the bucket-brigading algorithm (which has an adjustable parameter - the zone size or work area radius) to unknown distributions of robots. The work in this chapter was originally published as [17].

- Chapter 5, **Adaptive relocation**: a summary of the second adaptation, that of the bucket-brigading algorithm to nonuniform distributions of pucks. The work in this chapter was originally published as [18].

- Chapter 6, **Mathematical model**: a discussion of the theoretical (biological) model of foraging motivating this research, and the question of how well the algorithms from the previous sections produce the theoretical results.

- Chapter 7, **Conclusion**: drawing together the different adaptations with an overall assessment of the bucket-brigading approach.

Chapter 2

The study of foraging

2.1 Robotics

Foraging is a common analogy for a wide variety of robot tasks, including exploration and mapping, search, and object retrieval.

Beckers *et al.* noted the use of *stigmergy* in insect swarms[2]. Stigmergy is a process by which insects (in general, agents) communicate implicitly and indirectly by modifying the environment—their coworkers' future behavior is affected by these changes. The authors built robots which utilized stigmergy to collect objects and gather them into a pile—a version of the foraging task discussed in this work. They found that increasing the number of robots decreases the mean time required to complete the task, *but only up to a certain point* (three robots), after which the mean time increased, due to interference between robots. This thesis attempts to directly deal with that problem.

Holland and Melhuish examined stigmergy and self-organization in physical robots[14]: using very simple behavioral rules, the robots were able to cluster and sort Frisbees, despite possessing no memory or capacity for spatial orientation. The authors argued that their results hinged on the robots' behavior taking advantage of real-world physics. The robots in this thesis have similar sensorimotor capabilities, except that they are able to orient themselves in space with respect to an external reference frame.

Wawerla and Vaughan applied the rate-maximizing foraging model to a single robot performing the task of foraging over a long period of time[31]. The robot had a finite energy supply, and was required to travel to a charging station to recharge its batteries. While recharging, and while traveling between the work site and the charging site, the robot is not doing work. The authors presented a scalable, online, heuristic algorithm for the robot to recharge efficiently, maximizing the proportion of its time it spends working. While this thesis does not consider the energy needs of the foraging robots, it is an important avenue to explore in future work of foraging swarms, and the work of Wawerla and Vaughan will shed light on the subject.

Rybski *et al.* performed experiments in which real robots perform a foraging task using a variety of simple communication methods[25]. Robots communicated by flashing a light bulb under various circumstances. The authors showed that communication can reduce the variance in the robots' performance. In contrast, this research does not use explicit communication between robots; communication is implicit, however, in that robots must alter their behavior in the short term in response to the presence of other robots (collision avoidance), and in the long term by adapting a parameter of their behavior (discussed later). On the other hand, the results suggest that communication may be necessary to allow the robots to fully adapt to the pucks' distribution, as discussed in Chapter 6.

Other authors are closing the natural/artificial gap between ants and robots. Kube and Bonabeau modeled a group of ants cooperatively carrying a single food item with real robots in a cooperative box-pushing task[16]. They produced a coordinated movement effect without direct communication between robots.

Trail-following behavior in robotics research is well-studied. Vaughan *et al.* in [30] implemented trail-following in robots in a transportation task (similar to foraging) using low-bandwidth wireless communication. Real-world implementation of ant-like trail following shows that mutual spatial interference is a severe limitation. Later work by the same authors addressed this problem directly by quickly resolving conflicts over navigation space[33].

2.1.1 Interference in foraging

In [34], building on [4], Zuluaga and Vaughan addressed the problem of spatial interference in multirobot systems through the use of aggressive display behaviors. Several robots were required to perform a transportation task (akin to our foraging task) in shared space. Robots selected an "aggression level" based on the amount of work they had invested up to that point. The discrepancy in aggression levels between interfering robots was used to break the symmetry that would otherwise have lead to deadlock. The authors showed their approach to be effective, both in simulation and in a real-world implementation. Superficially the problem seems dissimilar to the one studied in this thesis: in our environment, there are no obstacles, but narrow passages created by obstacles are the situations that give rise to interference and deadlock in the environment studied in [34] and [4]. However, this thesis is investigating techniques for avoiding interference caused by competition for space, and during experiments it was observed that, sometimes, deadlock occurred when too many robots tried to get to the same space (the home zone) at the same time. No robot was able to get close enough to drop off its puck, and instead of yielding to other robots in the same situation, continually attempted to get closer. Bucket-brigading attempts to prevent this situation from ever occurring, but offers no advice on what to do when it does occur. Thus deadlock *recovery* is still an essential feature of any multirobot foraging scheme.

Lerman and Galstyan formally modeled the effect of interference on the performance of a swarm of foraging robots[19]. Their model was formulated as a system of coupled first-order nonlinear differential equations. They found that group performance grows sublinearly with group size, so that individual performance actually decreases with increasing group size. Simulations using the Player/Stage system[12] verified the predictions of their model. The variables in their model were the number of robots in each state (searching, homing, *etc.*; see Chapter 4 for details on what these states mean). Instead of modeling *states* of foragers with differential equations, the modeling done in this thesis treats the foragers' and pucks' *locations* as random variables and considers their distribution.

Then, in [11], the same authors present an adaptive algorithm for robots foraging for two distinct puck types, red pucks and green pucks. The robots' behavior changes dynamically – without communication or global information – so that the number of robots searching for each type of puck reflects the prevalence of that type of puck in the environment. A detailed mathematical model for the probability of finding robots in either the red state or the green state is rigorously derived, and this model is verified against experiments in simulation. Note that [11] compares the distribution of robots and pucks in the framework of two discreet classes, red and green, whereas this thesis is interested more in the spatial properties of the pucks and workers themselves and compares the spatial distributions of the objects. Also, in this thesis, there is only one type of puck, but they may be distributed nonuniformly.

Foraging strategies that seek to reduce spatial interference have been studied extensively. Liu *et al.* produced an adaptive controller to dynamically decide between "resting" and "foraging" operations, in order to maximize energy income[20]. Here interference was controlled by adapting the size of the foraging population. In our work, we adapt other aspects of the population, but the inspiration for an adaptive controller is the same.

Fontán and Matarić [9] investigated foraging with small teams (2, 3, and 4 robots) of real robots which forage in separate territories to reduce spatial interference. Territories were assigned *a priori* and covered all available space without overlap. In contrast, in our work, work areas (the analogue of territories) are assigned dynamically and adaptively. There are similarities between the motivation for Fontán and Matarić's work and the following work, and [9] should be considered an example of bucket-brigading in small teams.

Goldberg and Matarić, as well as Østergaard *et al.*, describe *bucket brigading*[13, 22], in which each robot is restricted to a finite search area, and instead rely on their coworkers to both deliver pucks into their search area, and remove pucks out of it. By restricting each robot to a finite area whose size is determined a priori, interference is ameliorated. In [22], the expected performance of multi-robot, space-constrained systems is described as a curve to which the $R = 40$m curve in Figure 4.2 roughly corresponds; the curve has a local maximum, after which the marginal performance is negative.

Notably, Schmickl and Crailsheim simulated robots acquiring food from a localized source *and* investigated the possibility of *positive* effects of spatial interference[26],. Their robots communicated using trophallaxis (transferring food from one robot to the next directly) and avoided non-trophallactic collisions with one another. The use of trophallactic contact induced a gradient of food concentrations, and robots could measure the local gradient and use the information to navigate uphill. However, their research did not explicitly look at the effect that using a localized source had on the performance of their algorithm. They did not address alternate distributions of the "dirt", such as the uniform distribution. In our work, interference is also used, in a sense, as a vector of communication: experiencing interference causes robots to change their behavior to lessen the likelihood of further interference.

Shell and Matarić [27] pioneered the bucket-brigade approach to foraging in large-scale multi-robot systems. They discussed interference in multi-robot systems. Instead of allowing all foragers to explore the entire environment, they restricted each forager to straying no further from its starting location (work-site) than a universal, preset value: the search radius. That study found a relationship between the number of workers and the performance of the team as a function of search radius. In this thesis, we reproduce and expand upon this algorithm to allow robots to adapt both their work-area radius and their work sites. The work of Shell and Matarić is the foundation on which this thesis is based.

2.2 Behavioral ecology

The literature on foraging in animals is too broad and deep to be completely surveyed in this section. Instead, this review will focus on work concerning the specific subtopics of interference, bucket-brigading, clustering, and work-site selection in animals.

Bucket-brigade foraging has been studied in insect societies as a form of task partitioning. Anderson *et al.*, in [1], described a form of bucket-brigading seen in several species of ant and termite in which insects pass resources *directly* from one to the next (as opposed to the *indirect* transfer in this thesis, in which pucks are dropped off by one robot and then, at a possibly distant future time, picked up by another robot) until it reaches the nest. They define a bucket-brigade as a "multistage partitioned transport scheme that uses only direct transfer between individual workers and without any predetermined transfer locations, other than the first or last stages." This definition is similar to the one used in this thesis, except in that transfer is indirect and there is no predetermined start location. They gave situations in which bucket-brigading would be especially effective, such as situations in which material must be passed along a narrow passageway, or when many insects foraging around the same source create a bottleneck. This is reminiscent of the studies of interference and deadlock in robots[4, 34].

The distinction between direct and indirect transfer is an important one as it highlights some important differences between natural and artificial agents, and raises the question of why bucket-brigading is beneficial to foraging. The act of passing a morsel of food from the mandibles of one ant to another seems so natural, and yet requires a degree of articulation not commonly found in the grippers of simple robots used in foraging experiments (simulated or otherwise). Indeed, when one pictures a team of humans putting out a fire as a bucket-brigade, it is common that the humans pass buckets of water down the line from hand to hand, rather than by walking with them for some distance, putting them down, and walking back to find more buckets. One could speculate that this is done to minimize the chance of spilled buckets and to conserve the firefighter's energy, due to features of buckets (filled with liquid and open on top) and of firefighters (avoiding using their legs to move around) that pucks and robots do not share. As such, [1] is a particularly interesting resource, since it shows that foraging insects use bucket-brigading for the same reason foraging robots do: avoiding interference.

Other forms of bucket-brigade foraging in ants have also been studied. Hubbell *et al.* observed leaf-cutter ants using a two-stage foraging process[15]. Normally, ants marched along a regularly used trail to find food. Occasionally, an ant would break off from the "main trail" and, upon discovering a new patch of food, would bring that food back only as far as the main trail, where the load would be transferred to an ant following the main trail. Since the ants use pheromones to lay trails, more ants would be led to this intrepid ant's newfound patch until gradually the two trails are "consolidated" and the new trail is part of the main trail. In this case, the frequency of bucket-brigading (transferring food to ants on the main trail) decreased as the trails consolidated, and so it is thought that bucket-brigading was only used to speed up the consolidation process by keeping some ants working on the new trail.

The same work also mentions another occurrence of two-stage bucket-brigading: in a certain species of leaf-cutter, workers are specialized into heavy, strong "carrier" ants and smaller "harvester" ants who climb to the tops of trees, clip leaves, and allow them to fall to the ground where they are collected by carriers who bring the leaves to the nest. Bucket-brigading is not used here for interference avoidance but as a side-effect of a division of labor.

Fretwell and Lucas introduced the notion of the ideal free distribution in the context of foraging in birds[10]. This work considered the allocation of foragers to any one of several patches (clusters), defined the notions of "ideal" and "free" foragers, and gave an explanation of the theory from the perspective of evolutionary adaptation and reproductive success.

Chapter 3

Foraging in simulation

3.1 Motivation

Our goal was to study the dynamics of teams of robots in the foraging task, especially large numbers of robots. We were interested in the average behavior of such "swarms", independent of the robots' particular starting conditions or the precise location of the pucks to be foraged. This required a large number of experiments, and we wanted the ability to quickly modify the parameters of those experiments, including number and distribution of robots, number and distribution of pucks, location of the home area, and sensory capabilities of the robots. For these reasons, and fundamentally due to the large number of robots (however simple) involved, we deemed experimentation with actual robots to be impractical and opted for simulation.

At the time of the research behind the work on adaptive ranging, there was a well-known and robust robot simulator in Player/Stage. Player abstracts disparate (real) hardware implementations into functional categories such as "position" and "rangefinder". Robotics researchers may implement their controllers in a device-independent fashion using Player. Stage is a general-purpose simulator providing experimenters with a virtual environment, and exposing a Player interface to virtual actuators and sensors of a variety of types.

However, at the time our research was underway, this simulator package performed slowly for large numbers of robots. Also, while the broad array of features provided by Player/Stage was not necessary for our research, one specific feature—an actuator for gripping and carrying objects found on the floor—was not available and would have to be separately implemented. Since then, Stage has been shown to scale approximately linearly up to 100,000 robots and operate 1,000 faster than real time[29]. As a result, and due to the ubiquity of Stage, that simulator would therefore be a preferable medium in which to continue this research.

For this research, a new simulator was implemented from scratch. This simulator provided a simple virtual environment for robots and pucks, localization with an odometry error model, sensing of obstacles and pucks, physical constraints of non-overlapping of robots, and locomotion. As this researched focused on the foraging task, we had the luxury of simplicity: the simulator did not need to be modular or easily adapted to provide support for new hardware or virtual environments significantly different from the original intent, both important requirements for an all-purpose package like Player/Stage. Perhaps most crucially, the low overhead of our simulator allowed experiments to run at a time ratio of over 300 simulated seconds per CPU-time second, even with large swarms (> 500) of robots.

3.2 Features of the foraging simulator

While the simulator evolved as we pursued new areas of research, several features were common to all experiments:

- A "spatial hash table" data structure to store robots and pucks and streamline sensation and collisions, which are local interactions. Up to a point, increasing the granularity of this table reduced the size of the neighborhood of any cell C and thereby reduced the number of comparisons needed to, for instance, completely specify the current sensory data of a robot in C. (By "granularity" we mean the number of cells into which an environment of fixed size is subdivided.)

  The limit to this speedup is twofold. Firstly there is overhead involved in maintaining the table: as a robot or puck moves from one cell to a neighboring cell, it is necessary to update the table. Though this is a constant-time operation, it is an operation that must be performed more and more often as the table's granularity increases.

  Second is the problem of dealing with a table so fine-grained that the boundary of the eight-cell neighborhood of a robot's current cell intersects, or is strictly inside, the robot's sensory radius. In this case, cells beyond the eight-cell neighborhood must be explored for possible interactions. Our implementation, while simple, leaves room for performance improvement: the simulator explores the smallest set of cells *containing the robot's sensory field*.

  When robots and pucks are uniformly distributed, this fixed-size, one-level table performs adequately. In cases where pucks or robots are clustered tightly in a small area, however, the benefits of the subdivision are lost and all that remains is the overhead of table maintenance.

- Locomotion. Robots can drive forward or back up to a maximum speed of 0.1m/s; they can turn either direction at a maximum rate of 0.3πm/s. Within this interval, robots can drive and turn at any rate (up to floating-point precision), though in practice there is a discrete set of drive and turn speeds used. Robots slow down by a fixed amount when avoiding obstacles or turning to stay within a work area.

- Sensing. Robots can sense the presence of a puck if that puck's center is inside the area covered by the robot's grippers. Robots can determine the range to obstacles (walls or other robots) using twelve sensors located at the robot's center and oriented in twelve evenly-spaced directions. When performing rangefinding, robots can tell the difference between a wall and another robot; this is necessary when a robot wants to determine whether spatial interference is taking place. Finally, robots can determine whether or not they are in a special circular region in the environment called the home area.

  Interference among the sensors of different robots, such as occurs with ultrasonic rangefinding, is not modeled. Also, sensors are not noisy.

- Localization. A basic requirement for most of the foraging controllers in our research is knowledge about "where to go to start searching". Initially, this location is the location in which the robot found itself when the simulation started. Knowledge about this location is kept as a displacement in Cartesian coordinates from the robot's current location. The robot updates its knowledge about this location using odometry: at every time step, the robot computes the vector of its physical displacement in that time step using its current speed and heading, and subtracts it from the displacement to the work-site. To simulate imperfect odometry, the simulator adds white noise to this displacement.

  Later, we considered that a Gaussian model for noisy odometry would be more realistic. By that time, however, many experiments had been carried out using the white noise model and it was decided to continue using that model rather than varying the controller and ending up with results that would be difficult to compare with earlier data. In the end, the dependence of our algorithms on noise should not depend on the type of noise used: adding noise to the estimate of the robot's position ensures every point in the environment is almost certain to eventually fall within a robot's search area (see Chapter 4 for details).

  Each robot is equipped with a compass for determining heading, though compass deviation and other errors are not modeled.

  A more realistic simulation of imperfect odometry would amplify the added noise proportionally with the distance to the work-site.

- Gripping. Robots can detect, retrieve, and carry up to one puck at a time. It takes four seconds for a pickup to take place, and no time for a puck to be dropped off. With a small probability, robots may be mistaken about whether they have successfully picked up a puck.

- Pucks. Pucks can be "created" and "destroyed". A puck is destroyed when a robot drops it off in the home zone; it is created when it is placed into the environment. Every time a puck is a destroyed, there is a configurable probability with which a new puck will be created (in all experiments, this probability was 1.)

Pucks can be distributed uniformly at random through the environment, or they can be localized in "clusters". There can be any number of such clusters, and they can be located in fixed places or randomly. A cluster is nothing more than a non-uniform way of randomly locating a new puck in the environment. Three different non-uniform distributions are implemented:

– The "hat" distribution (bounded): pucks are located uniformly at random within a fixed distance from the center of the cluster.

– The "tent" distribution (bounded): pucks are linearly more likely to be located near the center of the cluster than near the boundary of the cluster.

– The gaussian distribution (unbounded): the density of clusters follows a gaussian distribution.

Each distribution is parametrized by a "clustering" parameter. The exact interpretation of the parameter is different for each distribution shape, but a higher value always indicates a tighter or denser cluster, and the value is considered relative to the world size (*i.e.*, a given clustering value produces a larger, less dense cluster in a larger world). In the "hat" and "tent" cases, clustering is the ratio of the width of the world to the diameter of the cluster. In the "gaussian" case, the clustering parameter is the reciprocal of the standard deviation of the normal distribution used.

- Reporting. The simulator keeps track of how many pucks have been delivered, and reports this value, along with parameters specifying the experiment, at regular intervals or, optionally, in a condensed format at the end of the run. When running multiple experiments, these data will then be combined into mean and variance measurements.

3.3 Challenges

The first problem encountered in modeling a large number of robots and pucks – and the first motivation for developing a simulator from scratch – was dealing with the quadratic growth inherent in modeling the interactions between such a large number of objects. At each timestep, it must be ensured that

- no two robots occupy the same space;
- robots' range finders produce valid and complete data that could be generated by potentially any wall or any other robot;
- robots' puck sensors report whether any one of many hundreds of pucks is available for retrieval.

Some assumptions were made to simplify the task.

- The original goal of the research was to reproduce and expand upon the results of Shell and Matarić[27], which was done in a square arena with no obstacles and a home zone located in a corner of the world. We could therefore assume that there was a constant number of walls to be checked for sensor detection, and that in any direction, there must be exactly one wall (though possibly out of the range of the sensors).

- the morphological details of the robots were not relevant to the problem. Therefore for the purpose of collision detection, the robots could be treated as discs of uniform size.

The solution was to divide the world into a grid and and restrict searches to grid cells that could possibly contain relevant targets. For example, when filling sensor data, only cells in which at least one point was within the robot's sensor range were checked for sensible objects (walls or other robots). This was effective as the ratio of simulation time to real time was upwards of 8:1 for a large simulation (500 robots, 64×64 meter world, four pucks per square meter, clustering of 15) to 20:1 for a moderate-sized simulation, to 130:1 for a small simulation (36 robots, 25×25 meter world, no clustering). Since each robot had many sensors, the number of robots was the largest factor in determining running speed; number of pucks had little effect.

Chapter 4

Adaptive ranging

4.1 Introduction

In Chapter 1 we introduced the idea that the distribution of robots affects interference and therefore should inform the robots' foraging behavior. In this chapter, we explain what that means and show how to use that idea to improve foraging performance.

Our starting point in this chapter is the bucket-brigade foraging algorithm developed by Shell and Matarić in [27]: restricting robots to forage within a fixed distance from a certain position (which we call the robot's *work site* helps to lessen interference when the work sites are uniformly distributed. In experiments, the authors were able to find a dependency of the foragers' performance on this fixed distance, known as the *range* ("search radius" in [27]).

We reproduced the findings of Shell and Matarić and then developed an adaptive algorithm, described in this chapter, that allowed each robot to have its own range which it could increase or decrease in response to interference with other robots. This is what we mean when we say that the distribution of robots "informs the robots' foraging behavior": that distribution affects interference, which the robots measure (indirectly) to determine their range.

4.1.1 Basic bucket-brigading algorithm

Robots can be in one of three states: searching, homing, or returning. The initial state is searching.

- A robot in the searching state searches its "work area", the circle defined by the robot's initial location and a fixed work-radius. The search method is implementation-specific; the details of how a robot goes about locating a puck are not relevant to the application of the bucket-brigade strategy. In our implementation, robots followed a random walk except when avoiding obstacles and steering to stay within the work area.

On detecting a puck, the robot picks it up and transitions from the searching state to the homing state.

- A robot in the homing state drives in the direction of the global home location (supplied to the robot) while avoiding collisions with other robots.

  On reaching the home zone, or on leaving its work area, the robot drops its puck and transitions to the returning state.

- A robot in the returning state, drives towards the center of its work zone. On arriving within *half* their search radius from this location, the robot returns to the searching state.

In any state S, if a robot must avoid an obstacle it will transition to an avoiding state until obstacle avoidance is no longer necessary, at which point it will return to state S, as per a subsumption architecture [3].

The result of this strategy is that very few robots actually deliver pucks to the home zone. Assuming robots are uniformly distributed in space, and assuming a sufficiently large population, one or more robots will have a work area that overlaps the home zone. These robots will transport pucks just outside the home zone into the home zone. Other robots will have work areas that overlap these robots' work areas, and this pattern of overlap will continue so that most of the environment is covered by a work area that is "connected" to the home zone, and pucks will travel toward the home zone in a manner suggested by the algorithm's name—a bucket-brigade.

See Figure 4.1 for an illustration of the bucket-brigading algorithm on a simple example. Four robots were placed the diagonal of the arena, approximately evenly spaced. A single puck was placed in the work area of the robot most distant from the home zone.

In experiments, we investigated an approach to bucket brigading that does away with *a priori* ranges by allowing robots to adapt their ranges in response to interference with other robots, improving overall performance.

4.2 Simulation

In order to reproduce the results of [27], and compare them to the performance of the adaptive bucket-brigadiers, we used the simulator described in Chapter 3 to model a scenario similar to that in [27]. A description follows:

Robots are located in an arena, a 64 meter-square plane scattered with pucks, in the northeast corner of which is a 3 m quarter circle "home area". Robots are equipped with grippers which they can use to retrieve these pucks, and when a puck is dropped off in the home area, it is said to have been foraged.

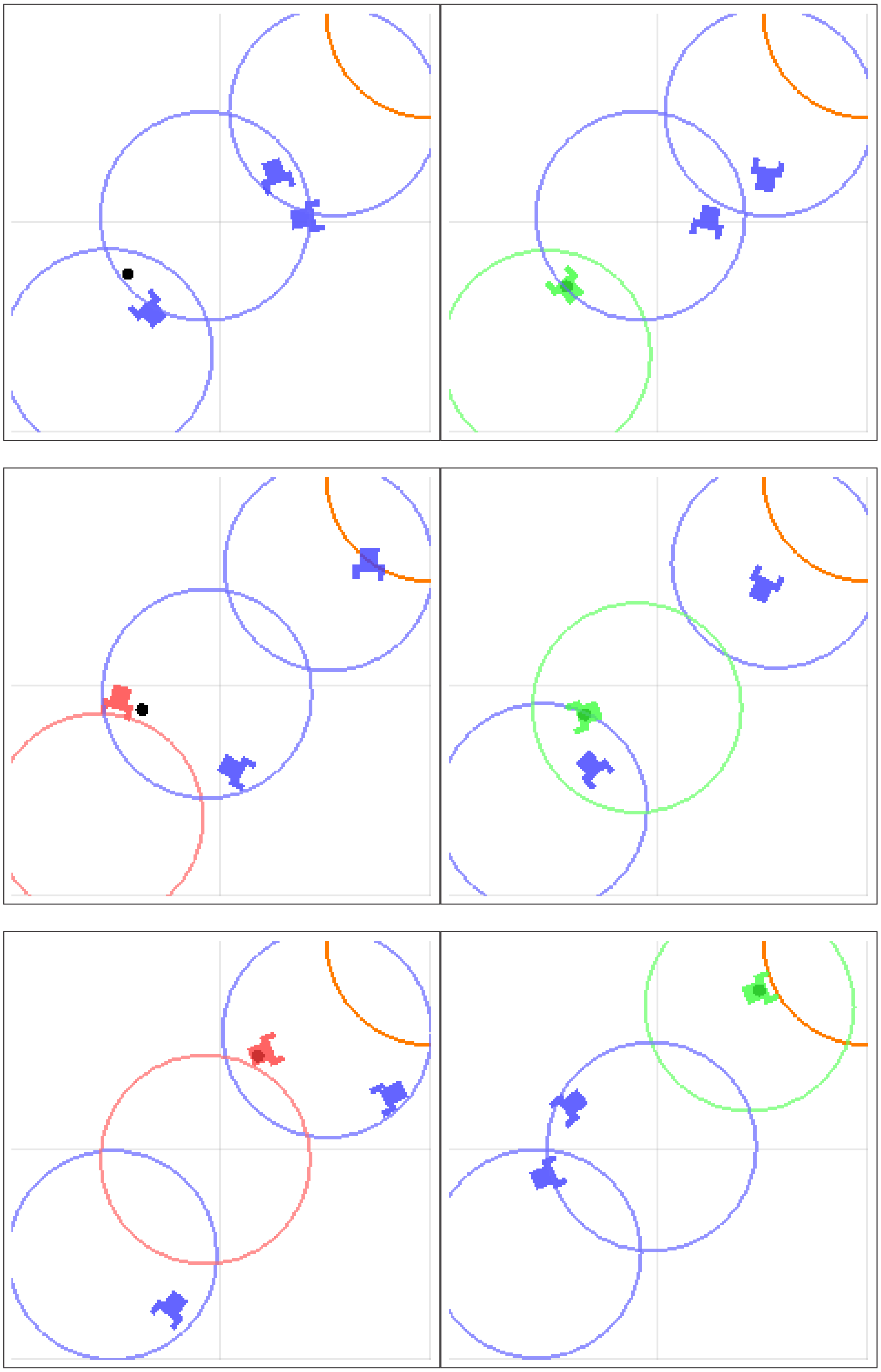


Figure 4.1: A single puck being delivered to the home zone by three robots using bucket brigading. Each robot is displayed within its zone. Blue robots/zones indicate the robot is in the searching; green indicates homing; and red indicates homing. The order is left to right, then down.

Robots can move forward at a rate of 0.1 m/sec, and can turn to either side at a rate up to once every five seconds. It takes four seconds to retrieve a puck, but a carried puck can be dropped instantaneously.

Robots can sense walls and other robots within 0.6 m of their centers without error (compared with 0.5 m in [27]), via the use of 12 radially oriented sensors. These sensors report the range in meters to the nearest obstacle (wall or robot). Specifically, each robot is idealized as a line segment, and if the center or either endpoint of that line segment is within sensor range, the sensor most closely oriented towards that point reports the distance to that point.

Robots cannot sense pucks unless the puck in question is located directly within the grip of their grippers, and this sensing is binary and also without error. Robots can determine the direction towards the home zone at any time; Shell and Matarić, in [27] explain this as the use of a “four-bit compass”. Robots use the data from their proximity sensors to avoid walls and other robots.

Shell and Matarić add a variety of noise to each parameter in their simulation. For simplicity, we have dispensed with this noise, except in the case of odometry noise (described in greater detail below). While this does detract from the physical realism of the experiments, the conclusions in this work are drawn by comparing two controllers in identical worlds; we do not attempt to compare our new controller with one tested under different conditions.

4.2.1 Parametrized bucket brigading

The purpose of the overall robot system is to retrieve pucks and deliver them into the home area – to forage for the pucks. In the bucket brigade approach to this problem, individual robots do not attempt to carry a puck all the way to the home zone themselves, but rather merely to shift the distribution of pucks towards the home area.

Each robot will attempt to stay within a fixed distance from its initial location. This zone is known as the robot’s “work area”. Through odometry, the robot can determine how far it is from the center of this zone, and can tell the direction towards the center of the zone. However, this odometry is noisy; as a result, the center of each robot’s work area drifts on a random walk at a rate of 0.01 m/sec.

The robot searches via a naïve algorithm within its work area. If it ever leaves the work area, it will drop off any puck it may be carrying, return towards its work area, and continue searching. If it ever discovers a puck, it will retrieve it and head towards the home area. The effect is that a “brigade” of similarly-behaving robots will incrementally transport pucks from one robot (work area) to the next, bringing the puck closer to the home area. Of course, ultimate delivery of the puck requires a connected sequence of overlapping work areas ending in the home area. This may be achieved over time, even if never simultaneously: due to the random-walk work areas, the probability that a puck is *never* inside at least one robot’s work area is zero.

Shell and Matarić's robots all share the same work area radius, or "range". In the following sections, we will explore other approaches to assigning these ranges to robots. In any given experiment, every robot uses the same approach to range selection.

4.2.2 Hypothesis

In all experiments, pucks are initially distributed at random. However, it can clearly be seen that as soon as the robots interact with the pucks, the distribution becomes less random, biased toward the home area — the system has a form of entropy that decreases as a result of the work of the robots. Since the optimal range for a robot depends on the density of pucks (demonstrated in [27]), once the density of pucks changes, the robot's original choice for range may no longer be optimal.

Ideally, robots would know and select the optimal choice for range on an ongoing basis, but in these experiments, robots are not given enough information to achieve this ideal.

In the next section, we propose a method for approximating this ideal. We propose that robots using an adaptive method (informed by interference and therefore by the distribution of robots) to individually adjust the range will perform better than a population of robots using a fixed, universal range.

4.2.3 Adaptive range selection

The aforementioned ideal puts an extra burden on the robot — it must be constantly aware of the distance between the center of its zone and the center of the home area. Alternatively, it must be able to measure the local puck density. In place of these assumptions we may also allow robots to adaptively select their range parameter using purely local information. In *adaptive range selection*, a robot will continuously increase its range parameter at rate dR^+, except while it is avoiding a collision with another robot, to which situation the robot will react by shrinking its zone at rate dR^-, thus making it less likely to interfere with other robots in the future.

Consider the extreme example of a robot alone in the 64×64 meter-square arena. The robot's work area has some initial radius - say, 10m. The robot will remove pucks from his work area and carry them outside in the direction of the home area. In parametrized bucket-brigade foraging, once the robot has removed all pucks from his work area, there will be no work for it to do, but it will not know this, since it cannot sense pucks, or their absence, from afar – so it will continue searching. Its restricted range does not improve efficiency since there are no robots to interfere with its navigation. Adding adaptive ranges into the picture, the robot's range grows at a rate of dR^+ and never shrinks (in practice, we limit the growth so that the radius of any search area is at most the diagonal of the arena).

Now consider the addition of a second robot into our example. As long as the two robots stay outside of each others' sensor ranges, their search areas will continue to grow as before; this scenario will be the same as the above-described scenario. However, each robot is sensitive to other robots that come within a certain range, less than their sensor range. If such an encounter occurs, each robot will take action to avoid colliding with the other. At the same time (*i.e.*, as long as the collision-avoidance behavior persists), each robot's search area will decrease at the rate of dR^-. Eventually, it may happen that one robot's search area shrinks so that the robot is no longer inside it; at this point, instead of continuing to search for pucks, that robot will try to return to his search area, thus lessening the chance that he will encounter and interfere with the other.

As mentioned above, no robot's search area will grow without bound – there is a maximum useful radius (the diagonal of the arena). In addition, no robot's search area is allowed to decrease below the space needed for the robot to drive in a full circle (at fixed forward/turn speeds).

Results for adaptive range selection are included in Figures 4.2 and 4.3.

4.2.4 Experimental design

Initially, we followed [27] in experimental design. The following parameters were varied: puck density ($0.781/m^2$ and $3.125/m^2$), search area radius (5, 10, 20, 30, 40, or 50 m), and number of robots (20, 40, 60, 80, ..., 500). The task was simulated for each combination of parameters for 2000 simulated seconds, and the number of pucks foraged after that time was recorded. Twenty such trials were run, each with a different initial distribution of pucks; to control for robot position, robots were initially placed on a square lattice. The reported results, in Figures 4.2 and 4.3, are the averages of those twenty trials. Error bars indicate the standard deviations of the twenty-trial experiments.

Next, we tested our adaptive range selection controller using the same experimental setup. For these experiments, ranges were allowed to increase by $dR^+ = 0.1$ m/s and decrease by $dR^- = 0.05$ m/s, biasing the robots towards the limit of homogeneous foraging in the absence of significant interference. Each robot began with a small range of $R = 5m$. For each parameter set, twenty trials were run, and mean performances were plotted with standard deviations shown.

While it is possible that the choice of the initial value R affects the performance of the adaptive-ranging robots, this was not explored in our experiments.

4.3 Results

Our first step was to reproduce the results in [27]. Inspection of the data in Figures 4.2 and 4.3 indicates that this was accomplished, in that increasing the radius of robots' search areas in the fixed-range regime led to an increase in the marginal benefit of adding robots (*i.e.*, to the slope of the curves in those graphs), but only up to a point: eventually, adding more robots decreases the performance of the system as overcoming interference begins to dominate the robots' behavior. These critical points are clearly visible as the significant local maxima in the $R = 30$ m, $R = 40$ m, and $R = 50$ m curves.

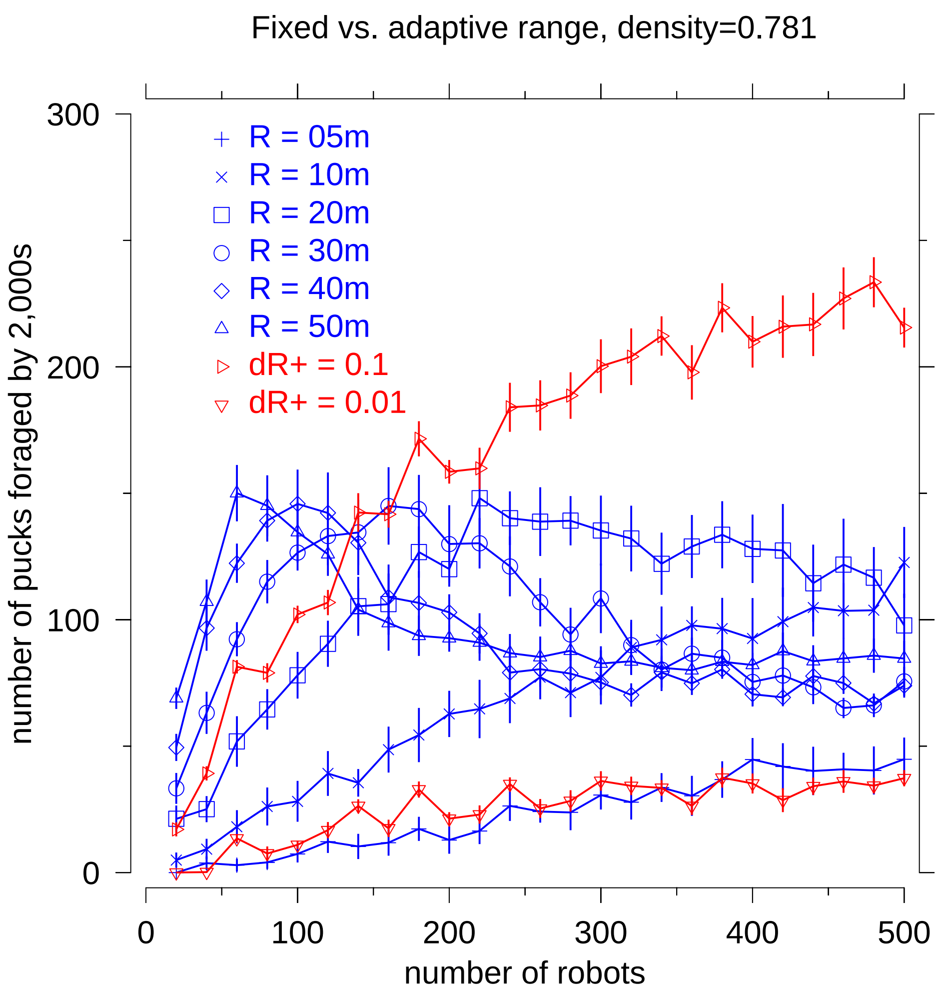


Figure 4.2: Performance with fixed and adaptive ranges at 0.781 pucks/m^2. R is the range of each robot in the trial. The curves labeled dR^+ show the results for adaptive range selection.

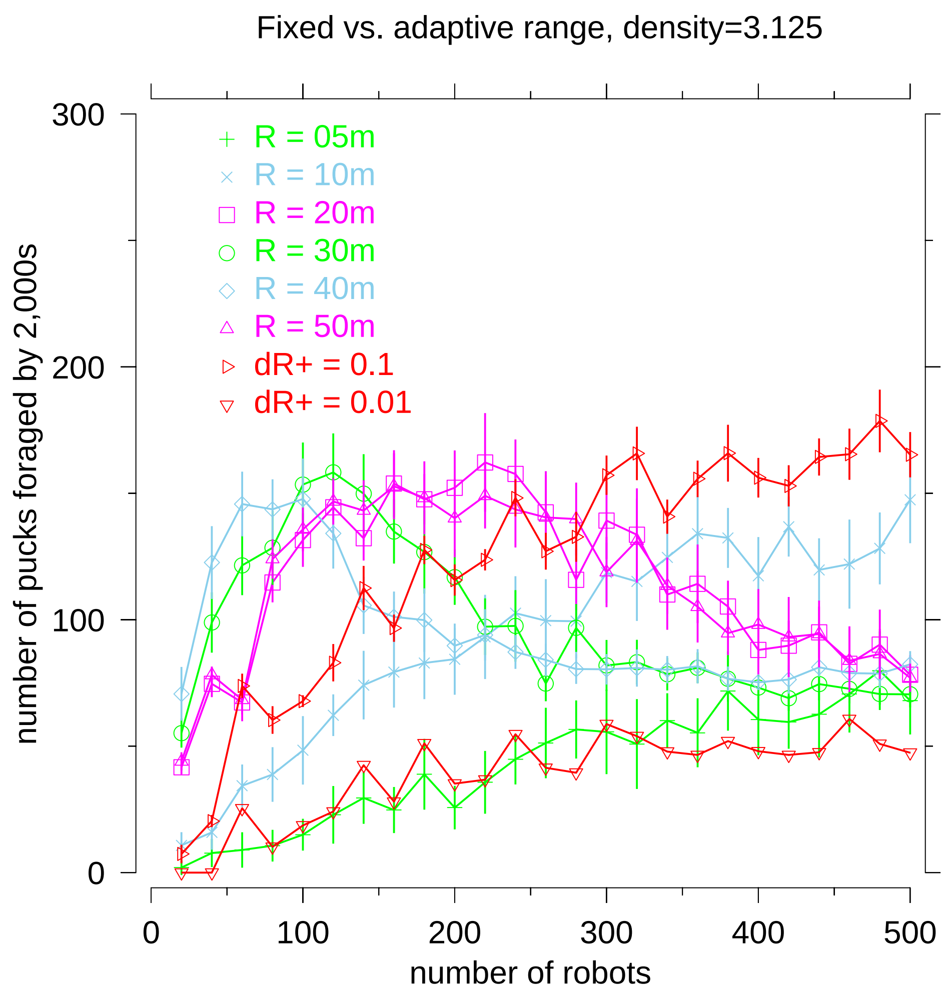


Figure 4.3: Performance with fixed and adaptive ranges at 3.125 pucks/m^2. R is the range of each robot in the trial. The curves labeled dR^+ show the results for adaptive range selection.

There is a noteworthy distinction in that robots in our experiments only foraged approximately half the pucks that those of [27] did. This indicates more a quantitative difference in the efficacy of the robots' controller programs than a qualitative failure of our simulations to produce the same behavior. The two sets of results agree at the heart of the matter: that interference affects a growing population later when the individuals' foraging spaces are larger, which is indicated by the relative shapes of the curves.

For appropriate parameter settings, our adaptive range selection algorithm performed at least as well as the fixed ranged controllers in simulation, and scaled better with respect to the fixed range solution. Fixed range algorithms suffered from one of two problems: robots with small search areas did not gather many pucks, and robots with large search areas interfered too much and the critical point at which the marginal benefit of increasing the number of robots was reached when the number of robots was still small. While the robots using adaptive range selection did not gather as many pucks when the number of robots was small as did robots with large, fixed ranges, increasing the number of robots always increased the performance of the group.

Also noteworthy is that the adaptive controllers performed more consistently, as indicated by tighter error bars on those curves than on the fixed-range performance curves. Since interference informs the robot that another robot is nearby, a form of implicit communication, this is in agreement with the findings of [25].

Adaptive range selection was sensitive to variations in dR^+ and dR^-. If dR^+ was too small, adaptive selection underperformed the fixed range foragers. Figure 4.2 shows results when $dR^+ = 0.01$, a fifth of dR^-. In that case, the adaptive controller performs no better than the worst-performing fixed-range controller we tested. This is not altogether surprising, since the range in the worst fixed-range controller and the initial range in the $dR^+ = 0.01$ m adaptive controller were both 5 m.

Figure 4.3 shows qualitatively similar improvements; however, in this scenario, where puck density is 3.125 pucks per square meter, the improvement is only slight, even for large group sizes. It might be the case that, had we increased group size even further, we would have continued to see increasing performance. More work is needed to determine why the effect of adaptive ranging was not as large in this case.

The two distributions

In the next chapter, we will move on from the robots' distribution and discuss how the distribution of pucks is affected by and in turn informs the robots' behavior. So far we have assumed that the distribution of pucks is uniform and have not dealt with violations of that assumption (even though we recognize that the assumption won't hold). Our next chapter will explain why nonuniform distributions of pucks are important to study, and will present a new adaptive extension to bucket-brigading that will deal with such distributions.

4.4 Conclusions

To summarize the contributions of this chapter: we replicated the results of [27], confirming their findings with an independent implementation. Further, we propose a simple modification of their foraging scheme in which each robot's foraging area is adapted in response to interference. When the new method's parameters are tuned, the method was shown to improve performance, particularly in large population sizes.

As a result, the robots' behavior is informed, indirectly, by the distribution of the robots. In regions where robots are more densely clustered, robots will tend to stay closer to their work-sites with the goal of avoiding interference.

In the next chapter, we will continue with the theme of behavior informed by the two relevant distributions, looking now at the distribution of pucks.

Chapter 5

Adaptive work-site relocation

5.1 Introduction

The theme of the thesis is adaptive foraging informed by indirect measurements of distributions relevant to the foraging task. In the last chapter, we recognized that the local distribution of robots affects the degree of interference among the robots, and therefore had the robots modify their behavior in response to interference (high interference being a proxy for a dense distribution of robots), thus "closing the loop" (Figure 5.1) and improving performance.

Continuing with that theme we now look to the distribution of pucks and seek a way for robots with limited sensory and communication abilities to adapt their behavior by measuring that distribution, possibly indirectly. We will take a cue from the animal foraging literature.

In animal foraging literature, resources are frequently described as occurring in "patches" or "food-source locations", or otherwise referred to as clusters implicitly in discussions of isolated food sources distant from a nest site [21, 15, 6]. Similar language and environmental setups are used in ant-inspired artificial agent research, often in the context of pheromone trail-following [23, 8].

Here we allow robots to relocate their work zones to more productive areas. This allows the distribution of robots to reflect the current distribution of pucks, while still separating them in space to maintain resistance to interference.

This chapter narrows the gap between previous work in bucket brigading, characterized by its assumption of high risk of spatial interference, and uniformly distributed resources, and ant-like foraging, with its assumptions of relatively very low spatial interference and tightly clustered resources. We discuss some ways that clusters may form spontaneously in a foraging system, further motivating cluster-friendly methods.

We describe and examine our novel adaptive forager in a series of simulation experiments.

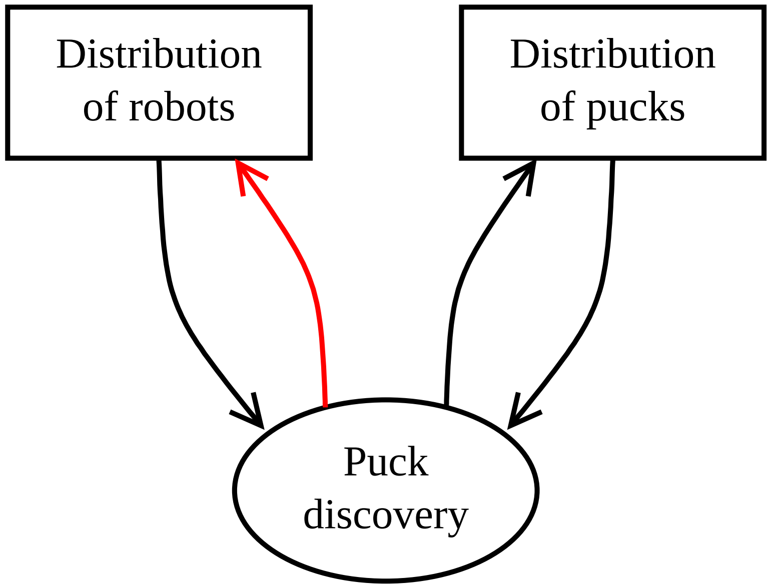


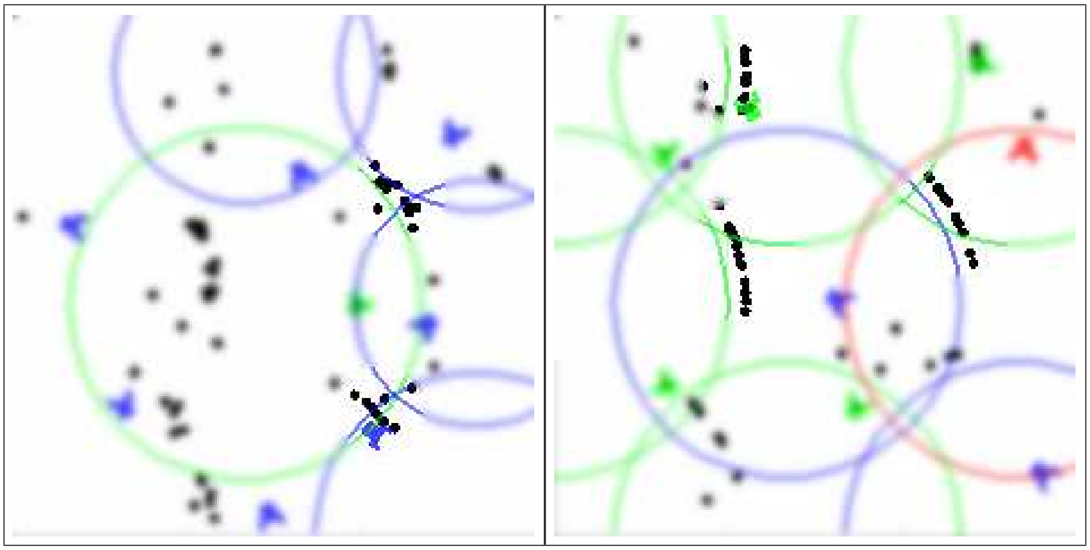
Figure 5.1: "Closing the loop" by enabling the discovery of pucks (an indirect measurement of the puck distribution) to inform the distribution of robots through relocation. The dashed arrow represents the interaction introduced in this chapter.

5.2 Simulated robotic foraging

Our experiments were performed in the dedicated multi-robot foraging simulator as described in Chapter 3. The environment is a square region with a home location in the northeast corner. Space is approximately continuous, and there are no obstacles in the environment apart from the robots themselves; robots avoid collisions with the environment's boundaries and with one another. Pucks are distributed throughout the environment, either uniformly or in a cluster depending on the experiment; pucks are modeled as points and do not "interfere" with one another. Robots are initially placed at random intervals throughout the environment.

Robots are equipped with 12 short-range proximity sensors capable of detecting walls and other robots up to 1m away; the sensors can differentiate between walls and robots. Each robot has a gripper which can lift, carry and drop a single puck. Another sensor detects when a puck is under the gripper and therefore grippable. This is the only way the robots can sense pucks. The robots measure 15cm $\times$ 15cm, approximately 1/28,000 the area of the world.

Except for the ability to place clusters of pucks (see below), this is the same experimental setup as used in [17].



(a) (b)

Figure 5.2: Clusters forming spontaneously (a) in neglected regions and (b) near the boundaries of foragers' work areas. The large circles indicate the boundaries of robots' work areas. The areas outside the clusters are blurred to highlight the clusters.

5.3 Clustering

Much of the previous work with foraging in robotic swarms assumes that the distribution of pucks is initially uniform. However, in real-world foraging scenarios, it is possible that the resources to be collected are distributed non-uniformly, especially in patches or clusters. In this work, we examine scenarios where there is initially a single cluster of pucks. Clusters are of interest for two reasons, first because some resources naturally or inherently are produced or deposited in clusters (apples on a tree; a bucket of tennis balls; a pile of rocks), and secondly because clusters may emerge as a side-effect of the actions of the robots and the peculiarities of their controllers.

The controller developed in [27] allowed robots to keep track of the distance and bearing to their work-site, through odometry. The inevitable error and unbounded drift in odometry caused the robots' work areas to drift over time. The authors noted that this had the added effect of ensuring coverage of otherwise neglected areas of the environment.

In our original experiments with global, non-drifting localization, we noticed that some areas of the environment were indeed neglected (see Figure 5.2a), in that no robot's work area covered them. Clusters of pucks spontaneously formed in these neglected areas. Even when total coverage is achieved, if some areas are more frequently visited for drop-off than pick-up then a clustering effect will exist.

As an example of this effect, we observed that clusters of pucks tended to spontaneously form near the edges of robots' work zones, visible in the simulator screenshot depicted in Figure 5.2(b).

The existence of non-uniform resource distributions, typically clusters caused either by the asymmetry inherent in the initial supply of resources, or by asymmetrical pick-up and drop-off frequency, motivated us to study adaptive approaches to foraging that work well under these conditions.

In our experiments, we looked at two distributions of pucks:

- the "hat" distribution, in which pucks are uniformly distributed, but only appear within a fixed circular area (the "cluster") of radius $\frac{1}{c} \times 25$m, and
- the uniform distribution (effectively $c = 0$).

Here c is the "clustering parameter", which indicates how tightly clustered the pucks are. A larger parameter indicates a tighter cluster. This parameter is dimensionless: it gives the size of cluster only when multiplied by the size of the world (25 meters in this case).

Ideally we would have explored a much wider variety of distribution shapes, such as multimodal distributions. However, the parameter space for such an experiment would have been much larger, and the results much harder to analyze. Focusing on single-cluster distributions which vary only in their size and density made for simplier analysis, and is also representative of many foraging tasks in both robotics and ecology.

5.4 Foraging in a bucket-brigade

In the bucket brigade foraging controllers of Shell and Matarić[27] and Chapter 4, each robot stays within the same fixed radius of its work-space location (subject only to drift). This restriction means that robots approximately maintain their initial uniform distribution, keeping the robots spread out and limiting interference. In this chapter, an adaptive method allows the robot's distribution to change over time in response to the environment. We will still need to use adaptive ranging as described in Chapter 4.

5.4.1 Prerequisite: adaptive ranging

If we suppose that the spatial distribution of the work-sites of robots using the adaptive controller described in this chapter will become exactly that of the pucks, and if we further assume that the pucks are clustered, possibly tightly, we create a new problem for ourselves: our many robots, all trying to work in the same space, will suffer a large amount of spatial interference. Therefore we will require the adaptation of the previous chapter to keep the robots apart.

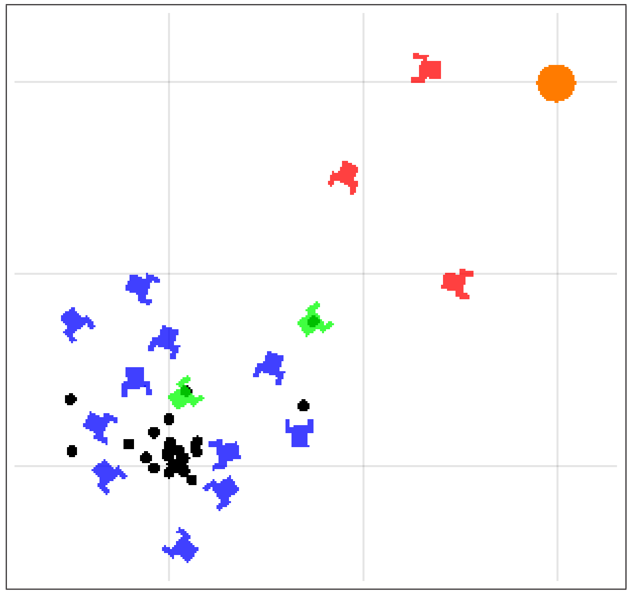


Figure 5.3: Robots in simulation foraging around a cluster of pucks. Robots are indicated by boxes with “arms”, and pucks by solid black circles. Blue/dark robots are searching, green/light robots are homing, and red/medium robots are returning to their work areas. The larger circle in the northeast corner is the home zone.

5.4.2 Effects of clustering on the bucket-brigade controller

Given the adaptive-ranging bucket-brigade controller described in Chapter 4, we found that if the initial placement of pucks is clustered instead of uniformly distributed, then performance decreases. Using the simulation environment shown in Figure 5.3 we ran 30 trials for a range of cluster parameters c, and for two robot populations. Figure 5.4 shows the results. The more tightly clustered the pucks (larger c), the fewer pucks are delivered over the length of the experiment. The success of the non-adaptive bucket brigade method depends on exploiting the uniform distribution of pucks.

5.4.3 Where to start searching

Recall that the bucket-brigade foraging algorithms require each robot to maintain an approximation of its start location. They begin their puck search near this point each time. When puck density is uniform, one location is as good as another for puck searching, and the start location serves only to spread robots out. In non-uniform puck distributions, however, placing the center of the work zone in a puck-rich neighborhood is likely to reduce search time and improve performance. Our second adaptive modification attempts to improve the position of the work zone center over time.

As in Shell and Matarić’s original method [27], our robots’ estimates of their work areas drift on a slow random walk. We improve on this by adding non-random, deliberate work-zone *relocation* as

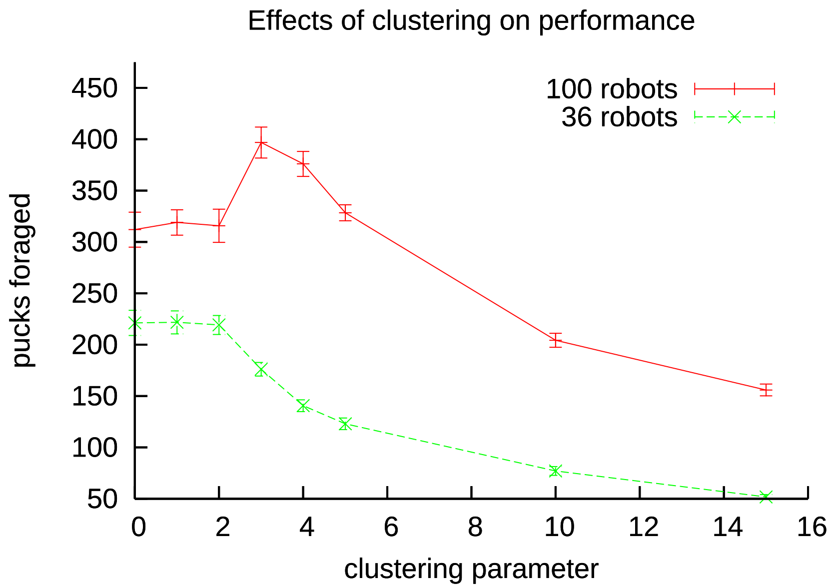


Figure 5.4: Performance of the standard, non-adaptive bucket-brigade algorithm degrades significantly as the degree of clustering increases. As the "clustering parameter" increases, the cluster gets smaller but has the same number of pucks in it. Mean performance over thirty trials are plotted with 95% confidence interval.

follows.

Upon finding a puck, a robot will relocate its estimate of its work-site a certain portion of the way along the line from its current position to the location of the puck. This proportion is an adjustable *relocation parameter* ρ; a value of $\rho = 0$ (no relocation) corresponds to non-relocating bucket-brigade foraging, in which the robot's estimate of its work-site remains fixed (upon finding pucks; it is still subject to drift). A value of $\rho = 1$ (complete relocation) indicates that robots will return to the last place they found a puck before they begin searching again. All robots use the same value of ρ (though there is no a priori reason requiring this). This process is illustrated in Figure 5.5 and described algorithmically in Algorithm 1. A proof of this algorithm is given in Appendix A.

Algorithm 1 Procedure for robots in the searching state.

Require: $0 \leq \rho \leq 1$ is the *relocation parameter*
Require: $r_Z \geq 0$ is the *range to zone center* from odometry
  if I found a puck **then**
    Pick up the puck.
    $r_Z \leftarrow (1 - \rho) r_Z$
    Switch to the homing state.
  end if

This tends to cause robot work zones to move towards areas of high puck density, reducing search

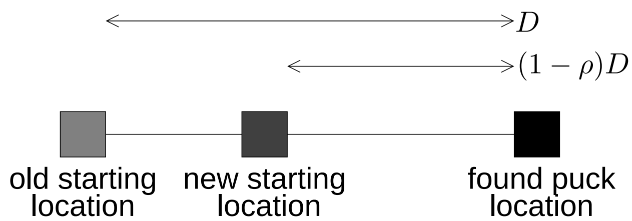


Figure 5.5: An illustration of work-site relocation. If D is the distance between the old work-site and the newly found puck, then ρD is the new distance; ρ (for $0 \leq \rho \leq 1$) is the relocation parameter.

time and potentially improving performance.

5.5 Experiments

5.5.1 Simulated environment

We performed a series of simulation experiments using the simulator described in Chapter 3 to examine the new method. Our hypothesis is that relocation can improve performance by adapting to the current puck distribution.

The foraging environment was a 25m $\times$ 25m enclosure with a quarter-circular home area, of radius 1m, in the northeast corner. Robots were initially positioned at random locations (different in each trial).

Pucks are initially placed at random locations throughout the environment (different in each trial), either in a uniform distribution, or according to the clustering model described in Section 5.3. In either case, on average there were two pucks per square meter (for a total of 1,250 pucks available to be foraged). Upon delivery to the home zone, a puck is replaced at random according to the same initial distribution.

5.5.2 Parameter space

Experiments were run with all possible combinations of the following parameter values:

- Clustering parameter: we tested several small values ($c = 1, \ldots, 5$), and a few larger values ($c = 10$, $c = 15$).
- Robot population: could be either a "small swarm" of 36 (0.25 robots per square meter) or a "large swarm" of 100 (0.69 robots per square meter).
- Relocation parameter: could be 0.00, 0.25, 0.50, 0.75, or 1.00.

5.5.3 Performance metric

We make two claims; first, that clustering of pucks hurts the performance of existing algorithms, such as bucket-brigading, which ignore the distribution of the pucks. The data discussed in Section 5.4.2 and displayed in Figure 5.4 support this claim. The second claim is that we can adapt the foraging algorithm to work well in the presence of non-uniformity by using the relocation system described in Section 5.4.3 and illustrated in Figure 5.5. To support the second claim, we compare the performance of foragers in clustered-resource situations with and without relocation.

In each experiment, robots foraged for one hour of simulated time. At the end of the hour, the total number of pucks delivered to the home zone was counted.

Performance is reported as the average number of pucks collected over thirty trials for each possible choice of parameters as given in Section 5.5.2. A 95% confidence interval was determined, assuming that real performance is normally distributed.

5.6 Conclusion

We found that relocation can have either a beneficial or an adverse effect, depending on the setting. More research is needed to find a range of measureable environmental factors for which relocation improves foraging. Ideally ρ would be determined online by the robots themselves.

The results support our hypothesis that relocation would allow robots to adapt to a tightly clustered distribution of pucks: swarms employing complete relocation in patchy environments outperformed those that did not relocate their work sites, nearly doubling performance in the small-swarm (36 robots) scenario. These results are statistically significant; see Table 5.1.

The results are illustrated graphically in Figure 5.6.

We found that the performance did increase when relocation was used, and we suggest that the observed improvements were due to the robots' search areas adapting to the puck distribution. Figure 5.7 illustrates the progress of this adaptation.

The two distributions

This chapter concludes the discussion of our extensions of the bucket-brigading algorithm to allow the robots' behavior to be informed (indirectly) by the distributions of pucks and robots. In our final chapter, we will focus on relocation and provide a statistical test for how well the two distributions are "matched".

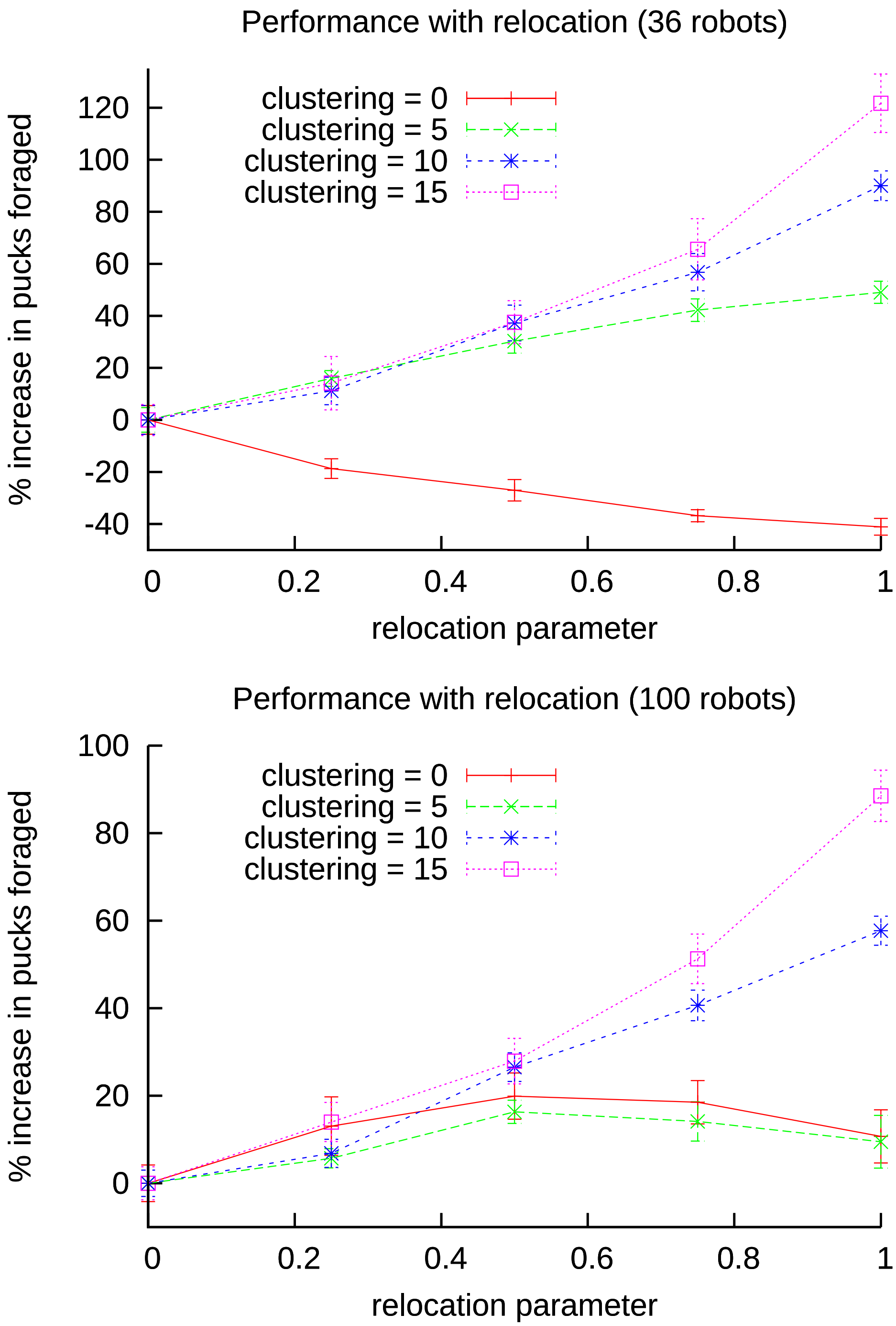
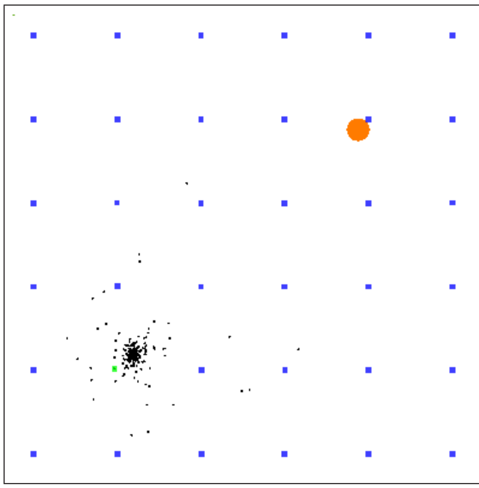
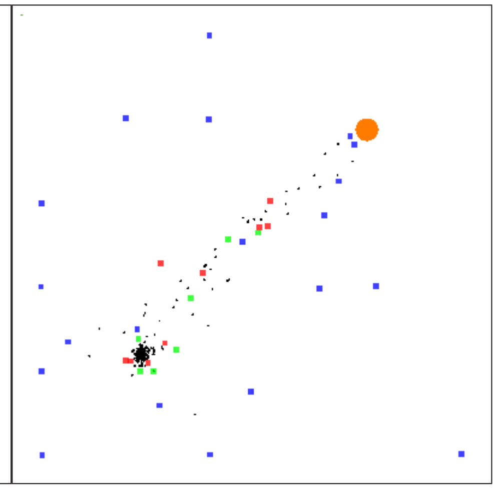


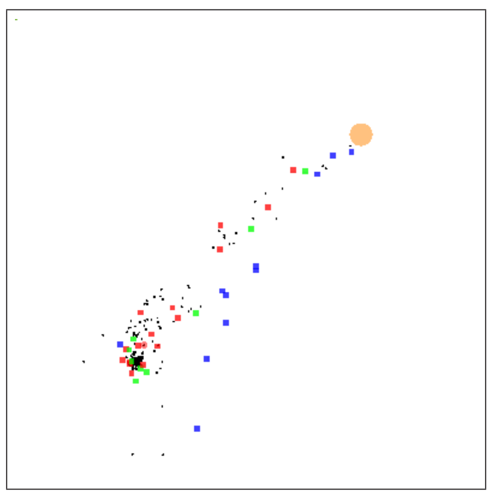
Figure 5.6: Performance of relocating compared to non-relocating foragers, averaged over thirty trials, with 95% confidence interval, over a range of relocation parameters. In tightly clustered puck distributions, the graph shows a upward trend in performance as the relocation parameter is increased.



(a) Initial



(b) 20 minutes



(c) 40 minutes

Figure 5.7: Distribution of pucks (dark circles) and robots' work sites (colored squares) initially and after 20 and 40 minutes. As pucks are spread out via bucket-brigading, the foragers automatically relocate to the "trail" of pucks from the cluster to the home zone. For the clarity of this illustration, robots were initially placed on a grid. Robots are not shown in this figure.

#robots	Clustering	ρ	Performance μ	Performance σ	p
36	0	0	221.000	33.837	< 0.0001
36	0	1	130.300	19.701	
36	5	0	120.633	15.940	< 0.0001
36	5	1	179.800	14.372	
36	10	0	76.900	11.973	< 0.0001
36	10	1	146.133	12.260	
36	15	0	56.100	9.386	< 0.0001
36	15	1	124.367	17.604	
100	0	0	313.833	36.545	0.0022
100	0	1	347.433	53.187	
100	5	0	335.667	36.456	0.0002
100	5	1	367.500	33.470	
100	10	0	208.033	17.629	< 0.0001
100	10	1	328.067	19.311	
100	15	0	150.033	15.751	< 0.0001
100	15	1	282.833	24.552	

Table 5.1: Analysis of performance for extreme values of the relocation parameter ρ. p-values for the significance of the difference between the $\rho = 0$ and $\rho = 1$ cases' performances are shown (mean over thirty trials). A more complete table (for intermediate values of ρ) is given in Appendix B.

Chapter 6

Foraging model

6.1 Ideal free distribution

This work has approached the problem of distributing workers "properly" in the foraging task, on the one hand by preventing them from interfering with one another, and on the other hand, by sending them to regions where there is more work to be done. In both cases, we have shown that the enhancements made to the basic foraging algorithm are associated with statistically significant improvements in foraging performance. And yet, it is not clear whether these algorithms allow robots to distribute themselves in the "best" possible way; indeed, that concept is not even formally defined in this work.

In animal foraging, the concept of the *ideal free distribution* is used to describe a certain distribution of workers in a "patchy" foraging task.

Definition 1. A *patch* is a homogeneous resource-containing area separated from others by areas containing little or no resources. An environment or habitat characterized by resources primarily located in patches is called *patchy.*[7]

The clusters used in the experiments in Chatper 5 qualify as patches by the above definition: clusters and patches both refer to certain nonuniformities in the distribution of target objects in a foraging environment, regions in which there are substantially more target objects to be foraged than in the surrounding environment.

Each patch can be assigned a *quality.* For example, this might refer to the inherent quality of goods to be acquired from the patch. Or, patch quality can refer to the density of the patch (if patches are of uniform size) or size of the patch (if they are of uniform density). In any case, any understanding of the quality of a patch should reflect the benefit that a forager will gain from working at that patch, and therefore the desirability of working there. Naturally, the quality of any given patch may change over time as the resources in the patch are exploited. Let us quantify the quality of the ith patch at time t as $Q_i(t)$.

During the process of foraging, at any given time each patch will have a certain number of workers foraging at it. In the context of this work, "foraging" corresponds to the process of searching for pucks, retrieving them to the home zone, and returning to search again. However, it will be difficult to say whether a given robot is foraging in a given patch; this will be discussed later. For now, let us say that the number of foragers at the ith patch at time t is well-defined, and let us call that quantity $N_i(t)$.

It has been shown[7, 10] that, under certain assumptions, members of a foraging population will arrange themselves in such a way that the number of foragers at each patch is proportional to the quality of that patch; that is, that

$$\frac{N_i}{N_j} = \frac{Q_i}{Q_j}$$

for all i, j. In other words, $N_i/Q_i = k$ for some constant k that is independent of which cluster is being considered. This distribution is called the ideal free distribution. The assumptions for this theory to hold are that

- the foragers are *ideal*, in that they always make the best decision about which patch to choose and have perfect access to information about their environment including quality of patches; and
- the foragers are equally *free* to select, travel to, and forage at any given patch *without cost*.

Another assumption is that the population density is high enough that Allee's principle—that the reproductive and survival suitability of a habitat increases with increasing population density—does not hold.[10, 7]

Note that in reality it is impossible for either of these assumptions to hold perfectly; the model should be considered only a formal description of what some idealized foragers would do.

The question of whether any particular example of a group of foragers is ideal free is a common one in behavioral ecology. For the rest of this chapter, we will discuss this question as it pertains to the specific example of the robots in the foraging task described in this work, concluding in the end that the robots are not ideal free, and that their distribution at best approximates an ideal free distribution.

6.2 Comparison of experimental data with the ideal free distribution: statistical testing

Recall the discussions of Section 5.3 in which we argued that the actions of the robots using the bucket-brigade strategy affects the distribution of the pucks. In simulation, it can be observed that, regardless of the initial arrangement of pucks, as pucks are carried towards the home zone and dropped off, the density of pucks near the home zone increases. This is especially evident in Figure 5.7 as a "trail" of pucks leading from a dense cluster to the home zone is formed. This effect makes it difficult to address the question of the ideal free distribution, since the "boundaries" of clusters are blurred. However, as was noted above, we are not relying on any hard-and-fast definition of "patch" or "cluster" anyway. As such, it makes intuitive sense to discuss not the distribution not of clusters but of pucks.

Furthermore, note that a given "cycle" in the robot's foraging activity involves the robot travel to many remote places; the home zone itself, for example. Thus it is hard to state whether a robot is "in" any given patch or cluster. Instead, since we require the robot to return to his work-site (the center of his work area), and since it is upon these work sites that Algorithm 1 operates, we are really more interested in the distribution of these work sites than of the robots themselves.

For these reasons, we cannot directly calculate the number of robots foraging in each cluster and compare this to what would be predicted by the ideal free distribution, as is common in the behavioral ecology literature. Instead, the question of whether the robots are in an ideal free distribution is really that of whether the distribution *of their work-sites* "is proportional to" the distribution of pucks, in a sense that we will now more formally define.

Let us suppose that the location of any given puck is a random variable P drawn from a region E, which is the area of the foraging environment, according to probability density function $f_P(x, y)$. Let us suppose further than the location of any given robot's work-site is a random variable R also drawn from this region E, according to probability density function $f_R(x, y)$. We would therefore like to determine whether $f_P(x, y) \propto f_R(x, y)$, or in other words, how similar the two distributions are. In the following, we will examine the statistical test used to determine whether we should reject the following null hypothesis:

> H_0 = the distribution of robots' work sites, created by Algorithm 1, is proportional to the distribution of pucks.

Note that we claim that the distributions should be *proportional*, not *equal*. In other words there is some constant k so that

$$f_R(x, y) = k f_P(x, y),$$

for all x and y. What should the value of this constant of proportionality be? If the above equation holds for all x and y then it must also hold for the integrals of these pdfs, so that

$$\iint f_R(x, y)\, dx\, dy = k \iint f_P(x, y)\, dx\, dy.$$

But since f_R and f_P are the *densities* of robots' work-sites and pucks, the integrals above are the total *numbers* of robots' work-sites and pucks. Therefore

$$k = \frac{\text{number of robots}}{\text{number of pucks}} = \frac{\text{density of robots}}{\text{density of pucks}}.$$

One approach to comparing the distributions is to use the *chi-square test*[32], a goodness-of-fit that involves dividing up the environment into "bins", and counting the number of pucks and work sites that fall into each bin. We would then simply compare the number R_i of work sites observed in bin i to the number we would expect if f_R and f_P are proportional, that is, $E_i = kP_i$ where P_i is the number of pucks in bin i. The random variable

$$\chi^2 = \sum_i \frac{(R_i - E_i)^2}{E_i}$$

has a well-known distribution, the *chi-square distribution*, and this could be used to perform a p-test on the null hypothesis.

Unfortunately, this test cannot be used in this case. Recall that our simulations use "hat"-shaped clusters, that is, distributions of pucks in which all pucks are uniformly distributed on a disc, and there are no pucks outside the disc. Thus the division by E_i will often be a division by zero, and so χ^2 will not be defined. In fact, the chi-square test is often not appropriate in situations where there will be few or zero data points in any bin.

Instead, we use the Kolmogorov-Smirnov test [28]. Like the chi-square test, this is a goodness-of-fit test that can be used to determine whether two distributions are the same. Due to the two-dimensional nature of the data, a variation on the conventional Kolmogorov-Smirnov test is required [24]. Instead of comparing the numbers of pucks and work sites in each bin, we compute the *cumulative distribution function* (cdf) over the entire environment, for each of the work-site and puck distributions, F_R and F_P. In fact, there are four ways of computing these cdfs:

$$\begin{aligned}
F_R^{(1)}(x,y) &= \Pr(X > x, Y > y), \\
F_R^{(2)}(x,y) &= \Pr(X > x, Y < y), \\
F_R^{(3)}(x,y) &= \Pr(X < x, Y > y), \\
F_R^{(4)}(x,y) &= \Pr(X < x, Y < y),
\end{aligned}$$

where X and Y are the x- and y-coordinates of the work sites, respectively. F_P has four similar definitions. The resulting statistic will be the least upper bound of the differences between measured work-site density F_R and expected work-site density kF_P:

$$D_n = \sup_{x,y,a} \left| F_R^{(a)}(x,y) - kF_P^{(a)}(x,y) \right|,$$

where a indexes over the four possible ways in which the the cdf can be computed. In the discrete case (using the same binning scheme as with the chi-square test), we have

$$D_n = \sup_{i,a} \left| R_i^{(a)} - kP_i^{(a)} \right|.$$

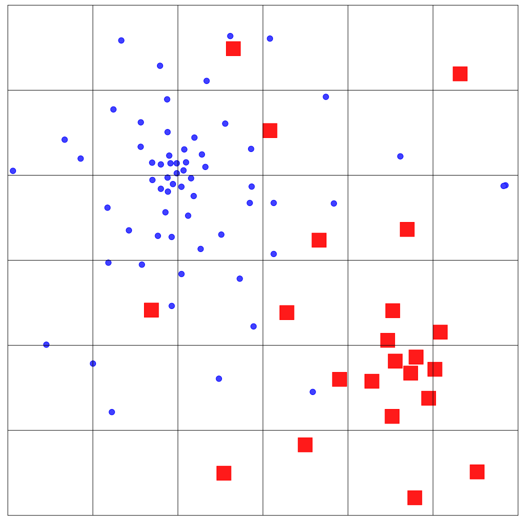


Figure 6.1: Two poorly matched distributions (22 red squares and 63 blue circles). Here, $D_n = 0.58$ and $p = 0.0012$, so we should reject the claim that the distributions are proportional.

An example

Consider the examples of the distributions of red squares and blue circles in Figures 6.1 and 6.2. Superimposed on these distributions is the grid of "bins". Each bin contains some number of each type of object. The objects in each bin are counted to get the frequency in that bin, and the resulting frequency is divided by the total number of objects to get the probability density in that bin. For example, the density of blue circles in the third bin from the left in the second row from the top (in either figure) is

$$\frac{8 \text{ blue circles in the bin}}{63 \text{ blue circles in total}} \approx 12.7\%.$$

In Figure 6.1 there are no red squares in that bin (0%), whereas in Figure 6.2 there are three red squares in the bin, so the density is

$$\frac{3 \text{ red squares in the bin}}{22 \text{ red squares in total}} \approx 13.6\%.$$

Analysis of experimental data from this research

Values of the Kolmogorov-Smirnov statistic D_n for the different scenarios tested in Chapter 5 are tabulated in Table B.5.

Ideally, if the algorithm always worked perfectly, then H_0 would always eventually be true, and therefore D_n would be zero (eventually). Do the observed values of D_n disprove H_0 and therefore the algorithm, or can we "explain away" those differences as due to chance? We will use a p-test to determine this. Namely, what is the probability p that such an extreme value of D_n will be observed,

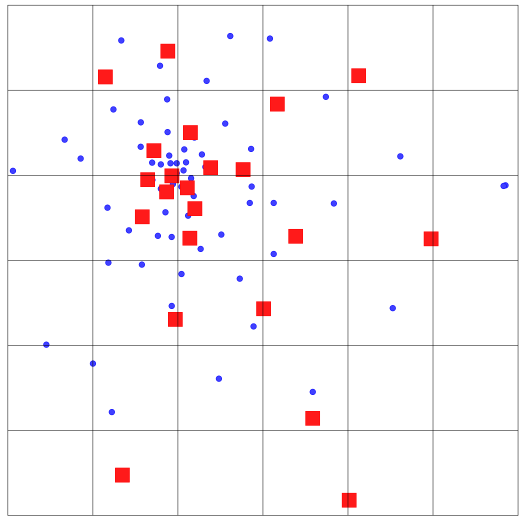


Figure 6.2: Two well-matched distributions (22 red squares and 63 blue circles). Here, $Dn = 0.12$ and $p = 1.9973$, so we cannot reject the hypothesis that the distributions are proportional.

given H_0? If p is very small – that is, if such a value of D_n is very unlikely – then we should reject H_0. If $p \geq 1/2$, then the difference can be assumed to be due to chance.

Peacock[24] gives a procedure for the use of the Kolmogorov-Smirnov test for comparing the distributions of two finite samples in two dimensions by finding a p-value. In the symbols we have used thus far, the procedure is:

1. Find D_n, the maximum absolute difference in the two cdfs, evaluating all four possible ranking combinations.

2. Set $Z_n = \sqrt{n}D_n$, where $n = n_R n_P/(n_R + n_P)$. Here, n_R and n_P are the numbers of work-sites and pucks, respectively.

3. Convert Z_n to Z_∞ by $1 - Z_n/Z_\infty = 0.53n^{-0.9}$. This effectively normalizes the variable by removing the "error" caused by the sample size (including numbers of pucks and of work-sites). For example, in very small samples the expected difference is likely to be larger.

4. Calculate the significance from $Pr(> Z_\infty) = 2\exp[-2(Z_\infty - 0.5)^2)]$. This is the probability of observing such an extreme value of D_n given the null hypothesis H_0.

   Alternatively, given a significance level p and a number n of points, we can determine the critical value of D_n beyond which H_0 should be rejected. For $p = 0.05$, we should reject H_0 when

$$D_n > D_n^* = Z_\infty \frac{1 - 0.52n^{-0.9}}{\sqrt{n}},$$

where $Z_\infty = 0.5 + \sqrt{\frac{1}{2} \ln 40} \approx 1.8581$. Values for D_n^* relevant for experiments in this work are given below:

- $n_p = 1249$ pucks, $n_r = 36$ robots. $n = 44964/1285$. $D_n^* = 0.3073$.
- $n_p = 1249$ pucks, $n_r = 100$ robots. $n = 124900/1349$. $D_n^* = 0.1914$.
- $n_p = 1249$ pucks, $n_r = 200$ robots. $n = 249800/1449$. $D_n^* = 0.1408$.
- $n_p = 1249$ pucks, $n_r = 300$ robots. $n = 374700/1549$. $D_n^* = 0.1190$.

By picking out the largest individual difference between cumulative distribution functions, the Kolmogorov-Smirnov test may judge the samples too harshly; other tests may be more generous to my algorithms. Nevertheless, we adhere to the use of this test as it is already an established test for comparing distributions, and because it can be used even though there are many parts of the world with low or zero local puck density (*i.e.*, many "bins" with no pucks).

The results of this test for the same experimental parameters as used in the relocation experiments Chapter 5 (in terms of number of pucks, number of robots, clustering, and relocation) are given in the appendix in Table B.5 and plotted in Figures 6.3 through 6.23.

By way of explanation, one line of Table B.5 is reproduced in Table 6.1.

Parameter			Time						
robots	clustering	relocation	0:00	0:10	0:20	0:30	0:40	0:50	1:00
200	10	0.5	0.65	0.61	0.57	0.53	0.50	0.48	0.46

Table 6.1: Sample Kolmogorov-Smirnov test for 200 robots, cluster of radius $25 \div 10 = 2.5$ meters, relocation parameter 0.5. The first line shows values of D_n at the given time. At each point in time, $D_n > D_n^*$, so we reject H_0 at each point in time.

In the figures, values of D_n are shown for three values of the relocation parameter ρ. Also shown is the critical value D_n^*; recall that if $D_n > D_n^*$, we should reject H_0 with significance level 95%.

Note that in many cases, the values of D_n decrease over time; this shows that in those cases, the maximum absolute difference between the observed distribution of work sites and the distribution one would expect if H_0 is true—in a sense, the "error" in the distribution of work sites—decreases as the experiment progresses and as the robots apply the relocation algorithm. This, along with data for other values of the relocation parameter, indicates that the relocation algorithm does indeed cause the distributions of pucks and work sites to become more similar as time goes on, even if we usually have significant (95%) evidence to conclude that H_0 is not true in these experiments.

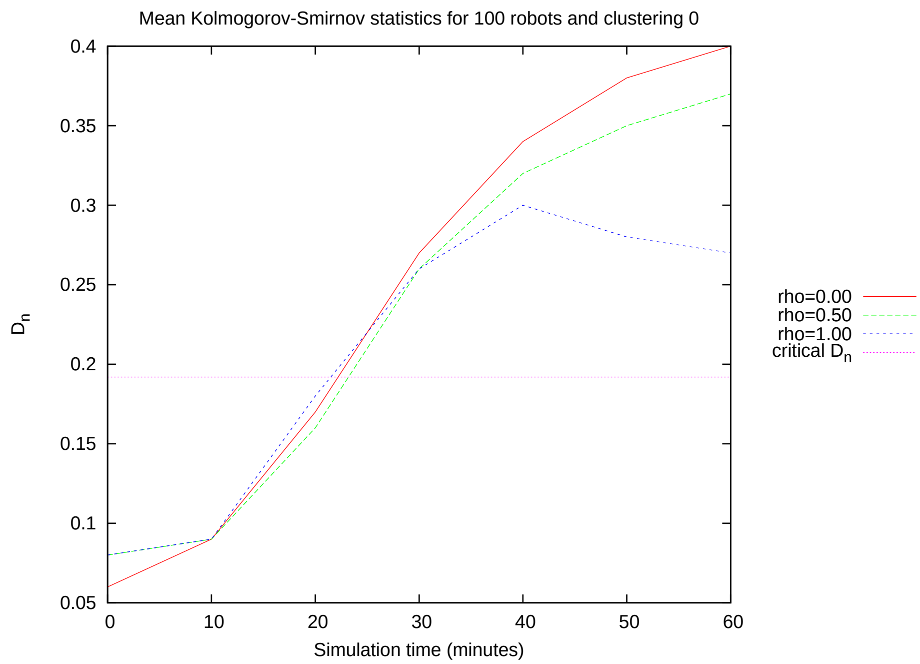


Figure 6.3: Kolmogorov-Smirnov statistics for 100 robots and clustering 0.

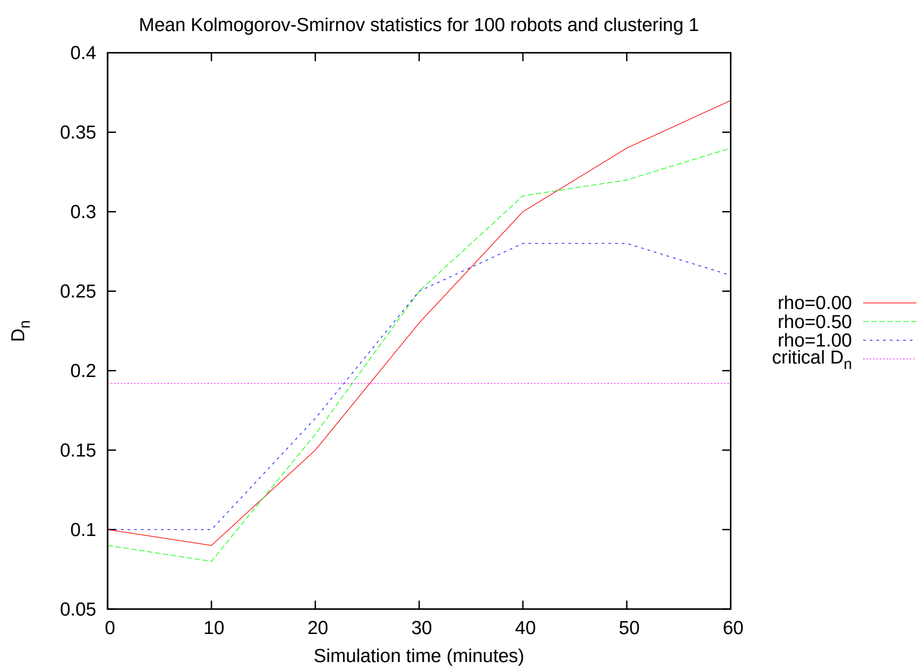


Figure 6.4: Kolmogorov-Smirnov statistics for 100 robots and clustering 1.

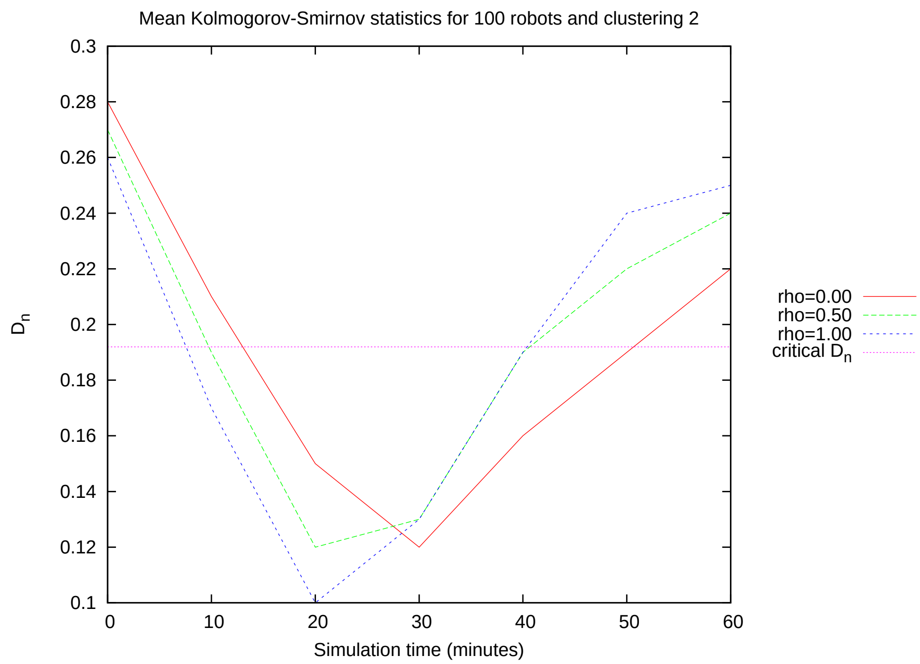


Figure 6.5: Kolmogorov-Smirnov statistics for 100 robots and clustering 2.

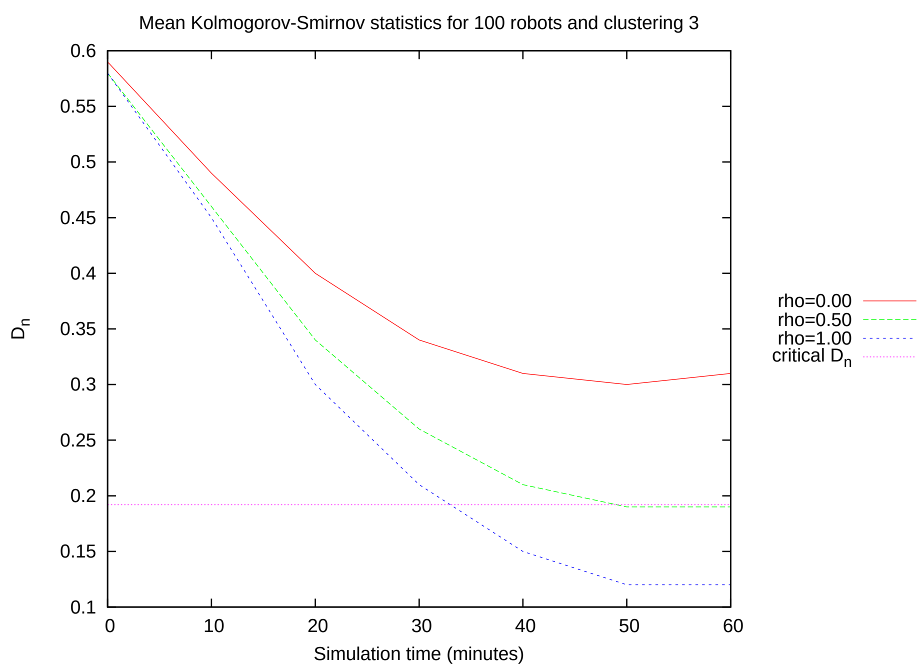


Figure 6.6: Kolmogorov-Smirnov statistics for 100 robots and clustering 3.

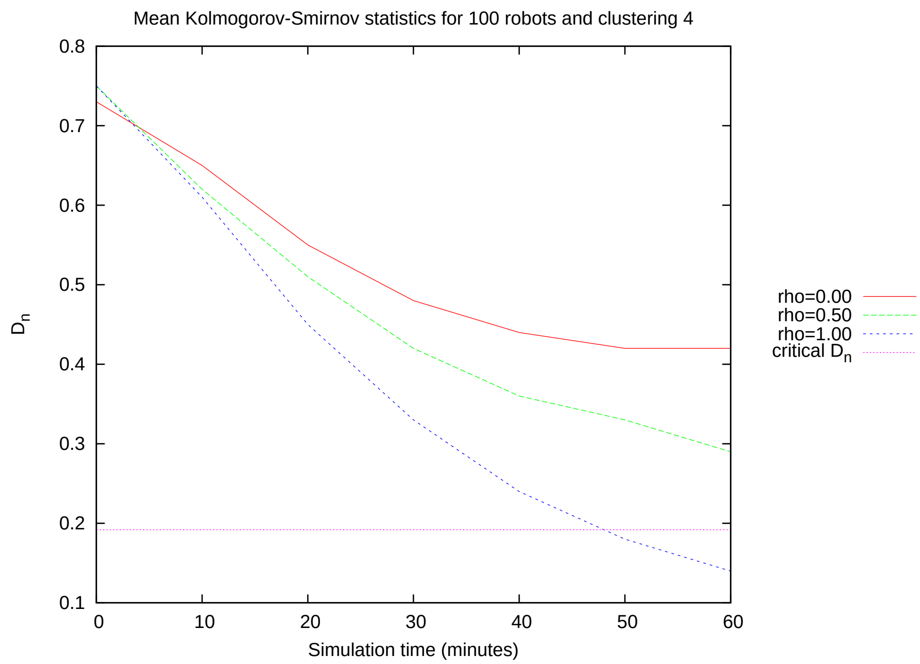


Figure 6.7: Kolmogorov-Smirnov statistics for 100 robots and clustering 4.

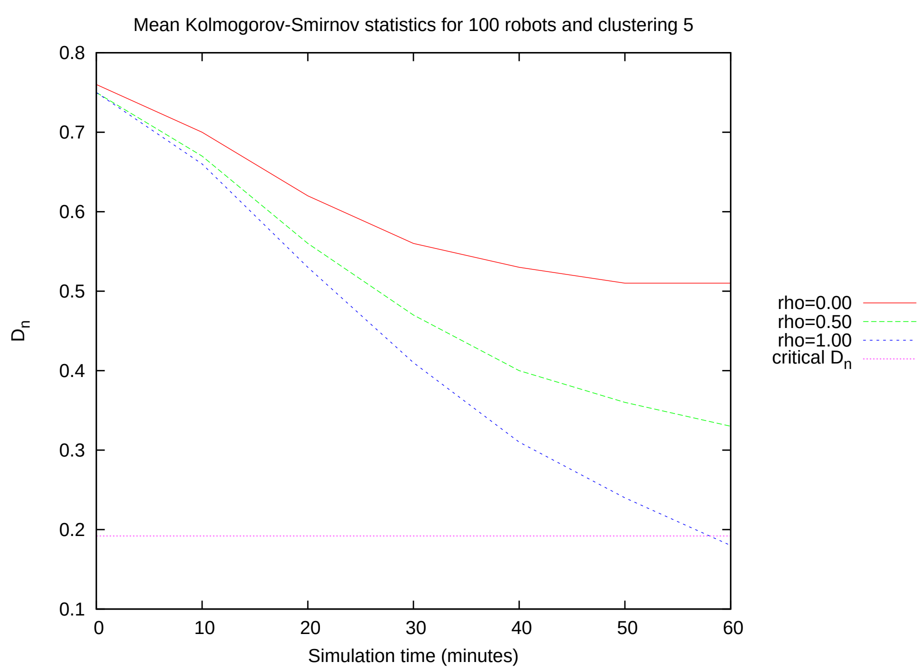


Figure 6.8: Kolmogorov-Smirnov statistics for 100 robots and clustering 5.

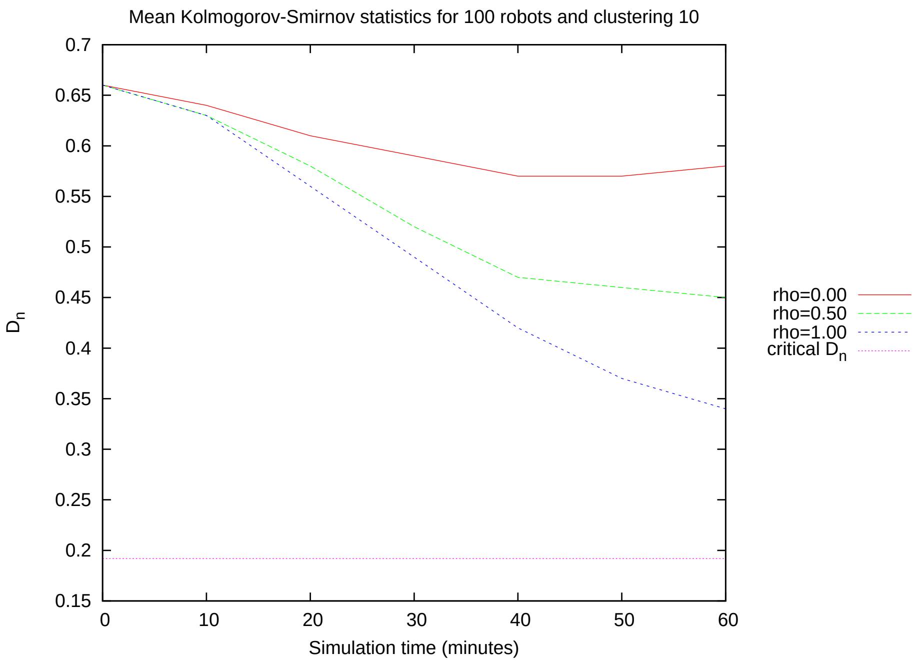


Figure 6.9: Kolmogorov-Smirnov statistics for 100 robots and clustering 10.

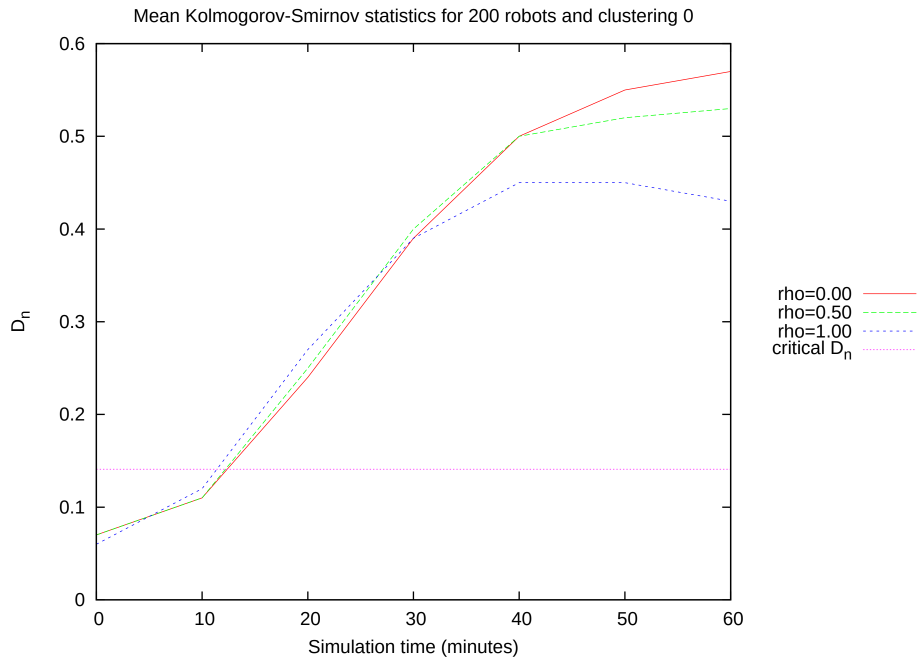


Figure 6.10: Kolmogorov-Smirnov statistics for 200 robots and clustering 0.

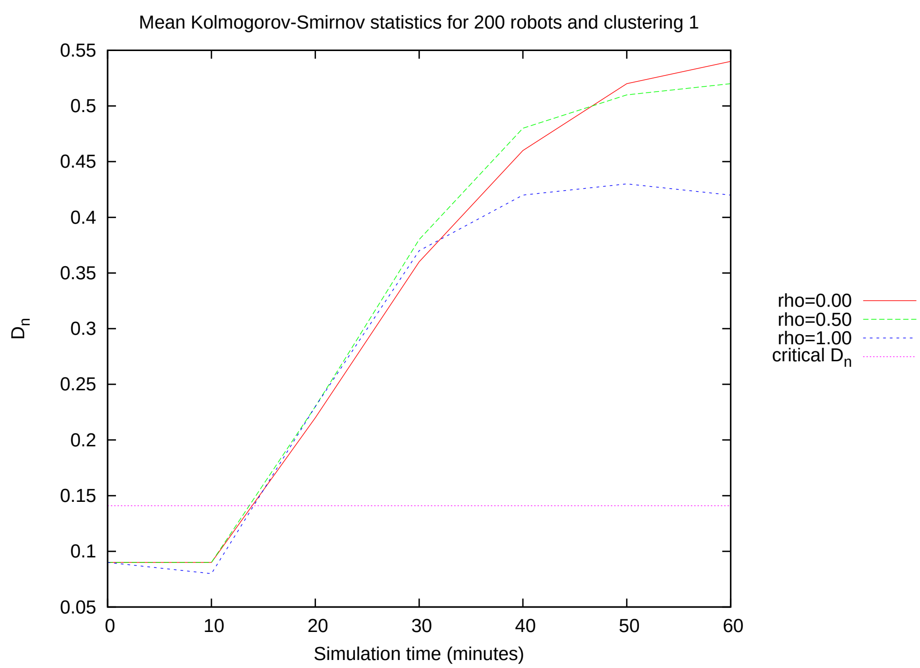


Figure 6.11: Kolmogorov-Smirnov statistics for 200 robots and clustering 1.

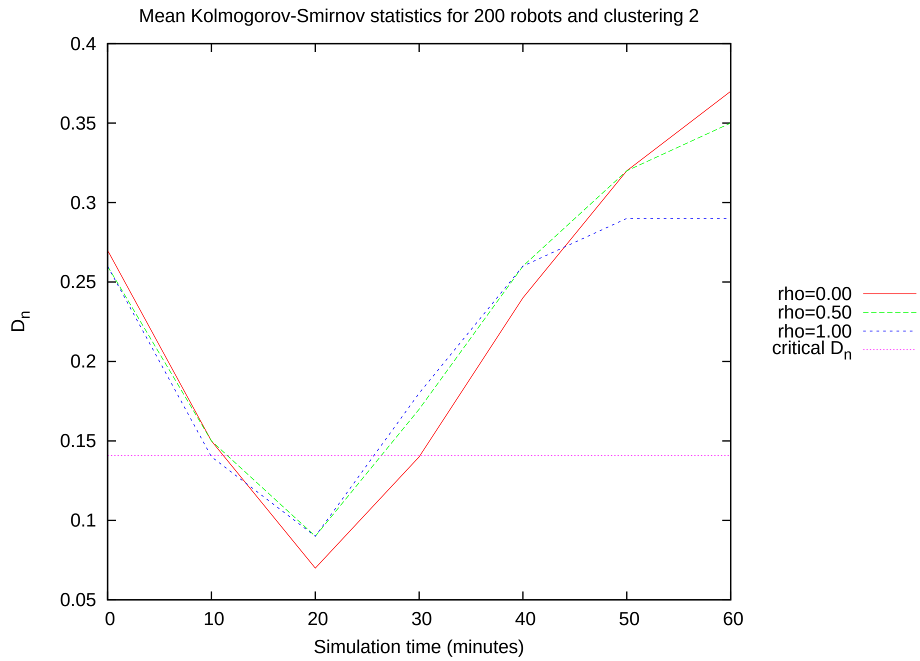


Figure 6.12: Kolmogorov-Smirnov statistics for 200 robots and clustering 2.

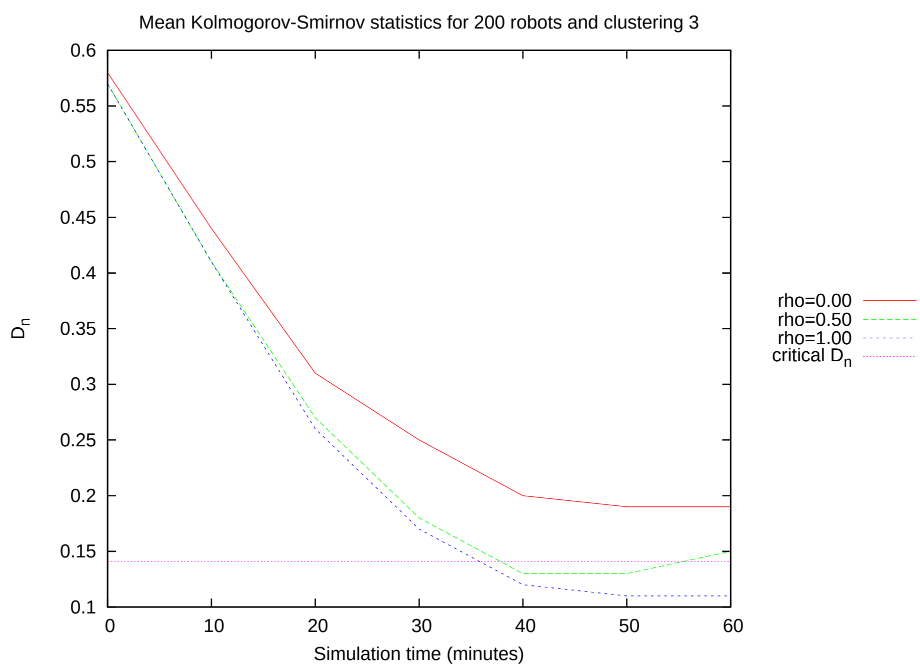


Figure 6.13: Kolmogorov-Smirnov statistics for 200 robots and clustering 3.

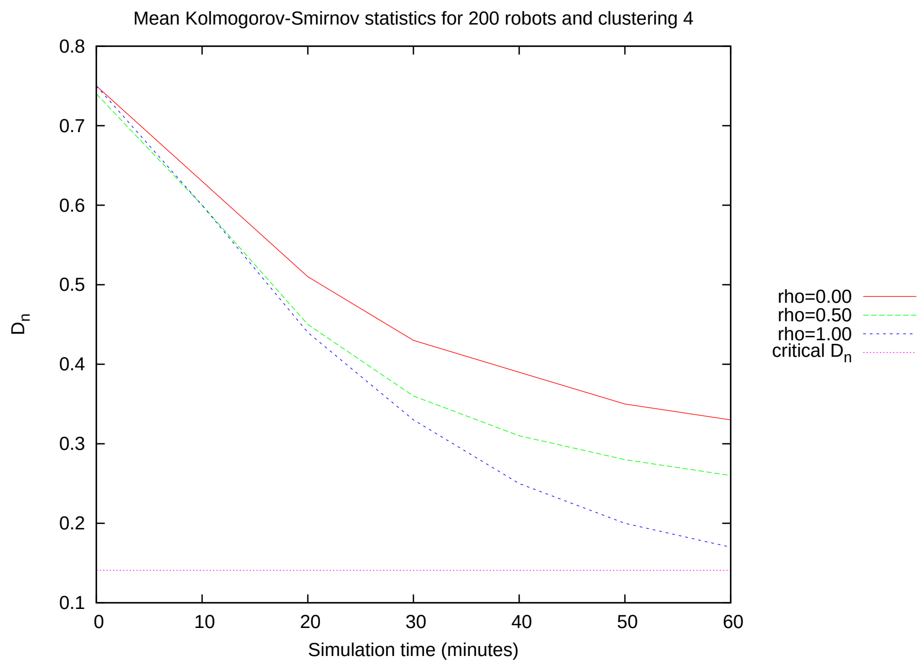


Figure 6.14: Kolmogorov-Smirnov statistics for 200 robots and clustering 4.

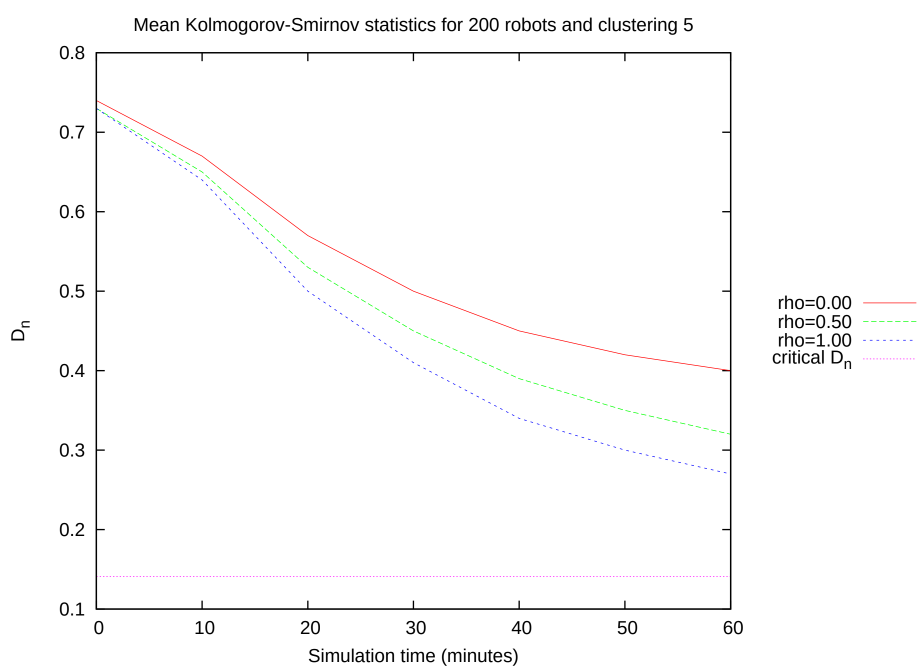


Figure 6.15: Kolmogorov-Smirnov statistics for 200 robots and clustering 5.

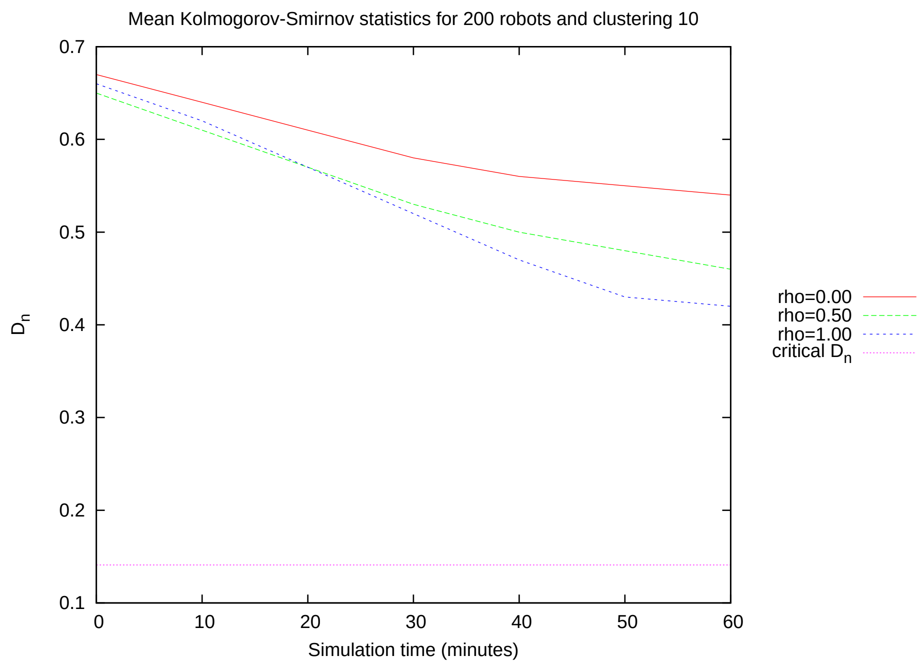


Figure 6.16: Kolmogorov-Smirnov statistics for 200 robots and clustering 10.

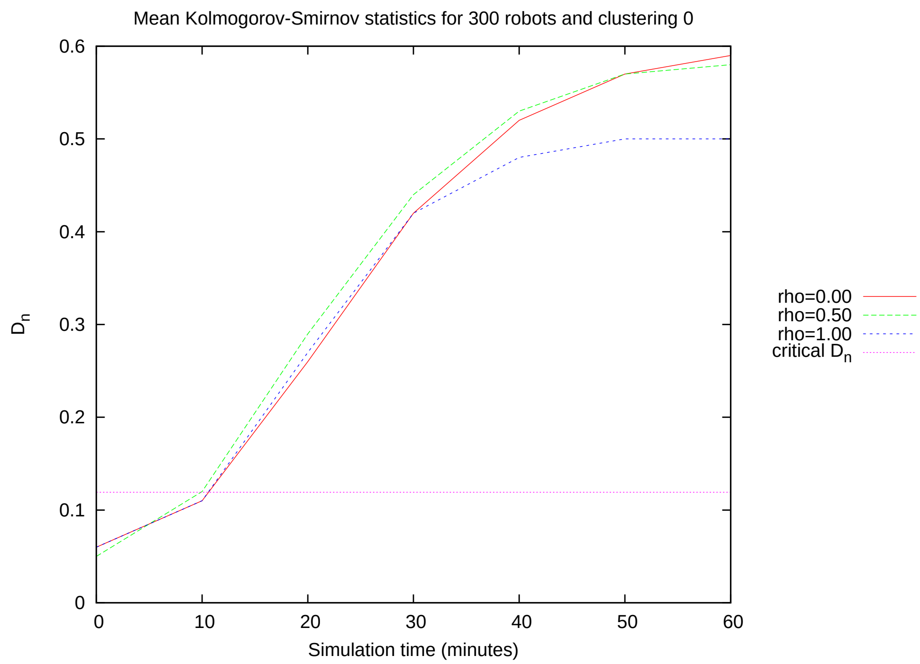


Figure 6.17: Kolmogorov-Smirnov statistics for 300 robots and clustering 0.

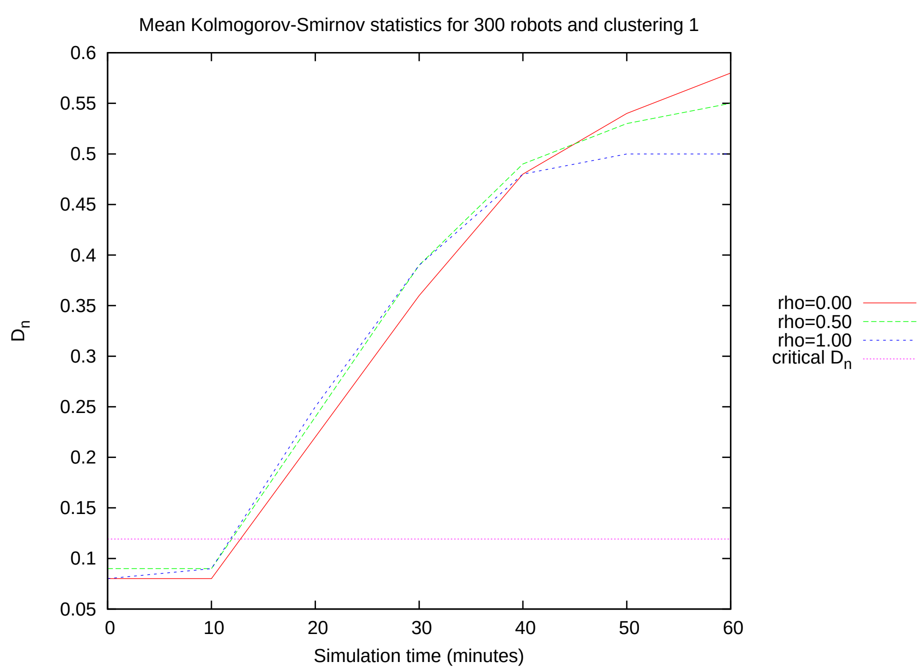


Figure 6.18: Kolmogorov-Smirnov statistics for 300 robots and clustering 1.

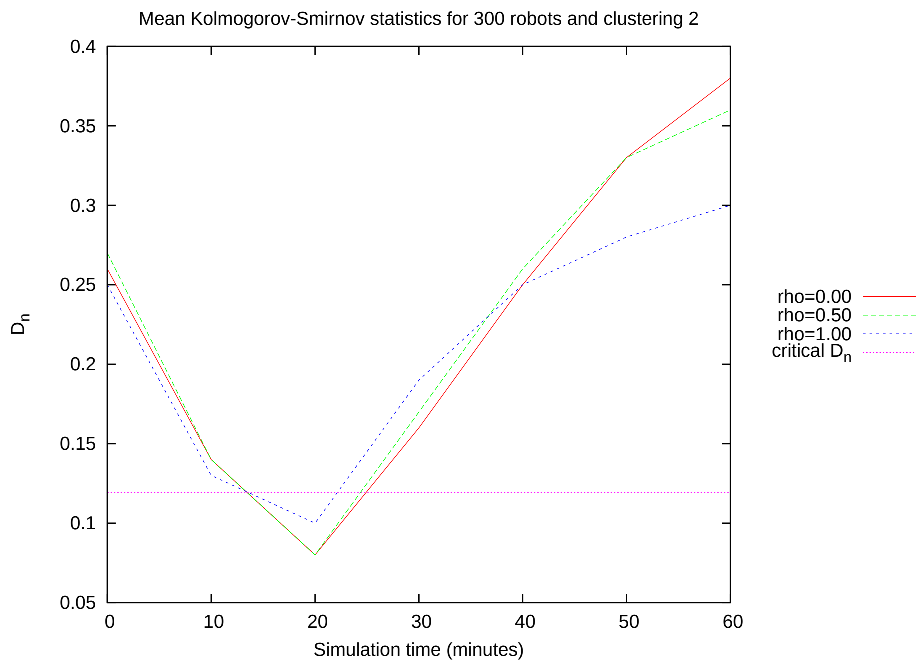


Figure 6.19: Kolmogorov-Smirnov statistics for 300 robots and clustering 2.

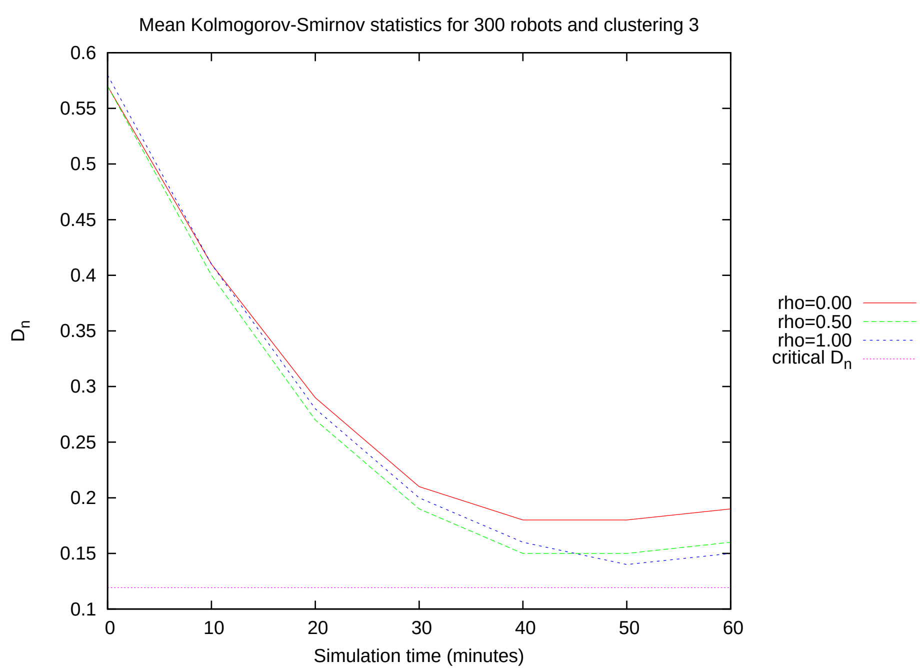


Figure 6.20: Kolmogorov-Smirnov statistics for 300 robots and clustering 3.

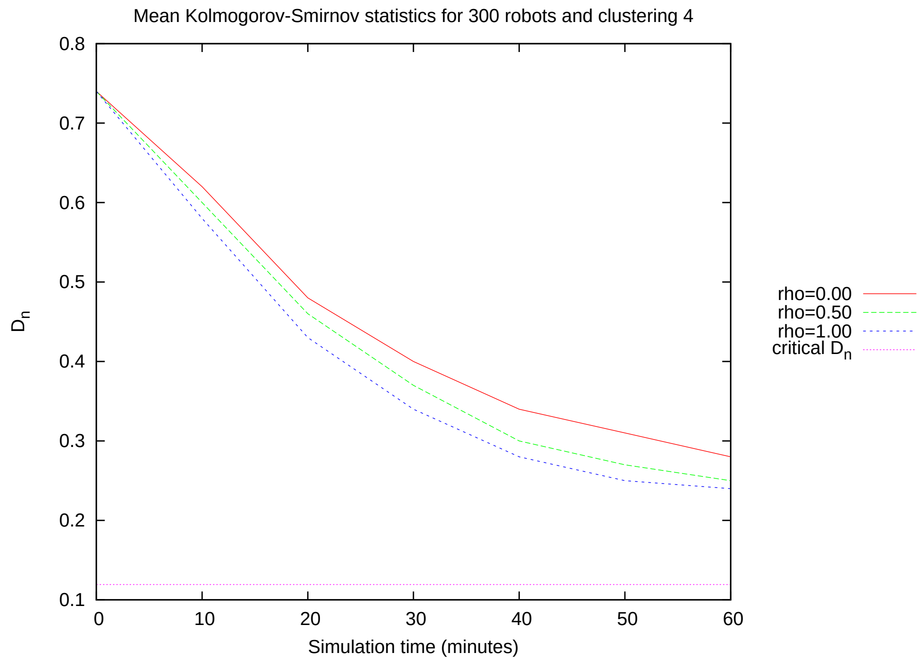


Figure 6.21: Kolmogorov-Smirnov statistics for 300 robots and clustering 4.

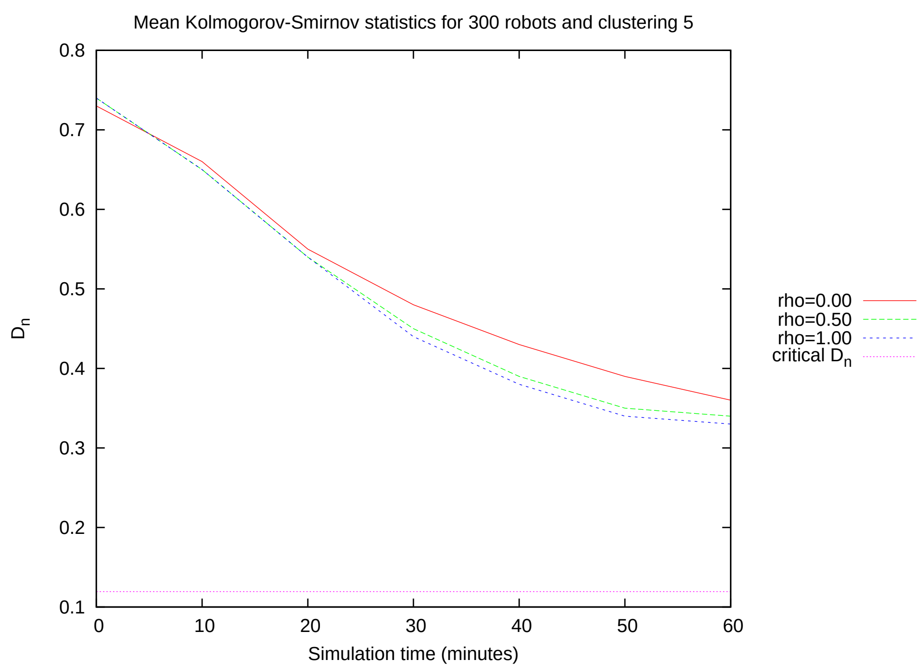


Figure 6.22: Kolmogorov-Smirnov statistics for 300 robots and clustering 5.

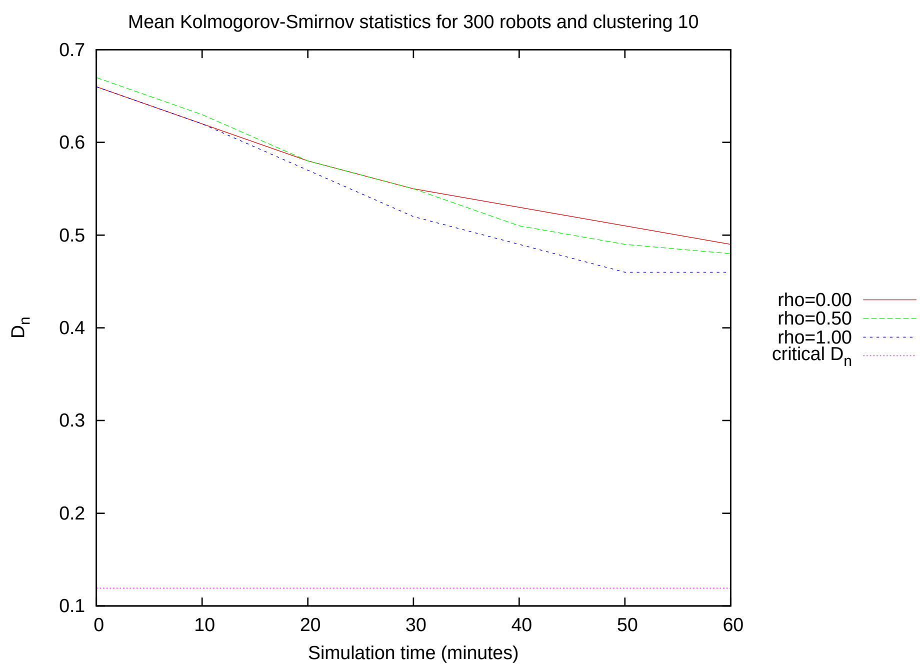


Figure 6.23: Kolmogorov-Smirnov statistics for 300 robots and clustering 10.

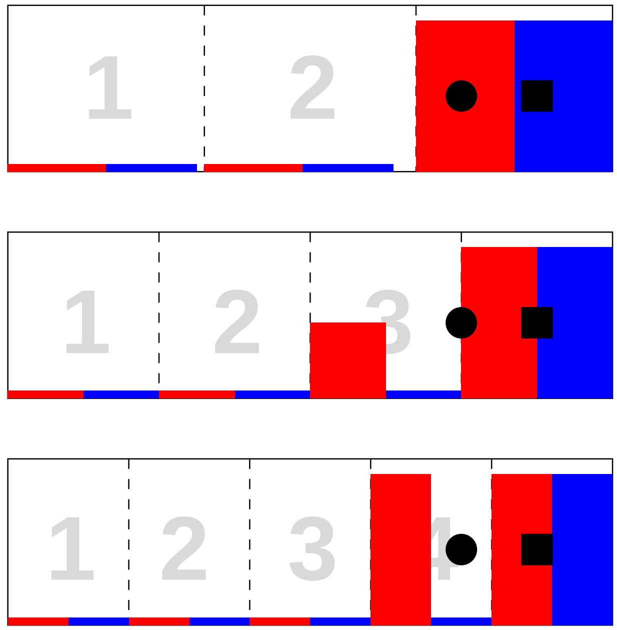


Figure 6.24: Cumulative distribution functions illustrating sensitivity of the Kolmogorov-Smirnov statistic to discretization. The circle is a patch of pucks and the square is a cluster of work sites. Dashed lines indicate boundaries between bins. The red histogram is the cdf for pucks, and the blue histogram is the cdf for work sites.

6.2.1 Limitations of the K-S test

One problem with this approach is its sensitivity to the discretization of the environment. Firstly, the value of D_n may depend on the number of bins used. By way of illustration, consider the following example: consider a one-dimensional environment with a very dense patch of pucks located at position 1 relative to the center of the environment and a similarly dense cluster of robots at position 1.5.

If the experimenter uses 4 bins then the patch will fall exactly on the intersection of bins 3 and 4 (counting from the left), and the cluster of robots will be entirely in bin 4, but the match will not be very good since half of the pucks will contribute to bin 3 and half to bin 4 ($D_n = 0.5$). If 5 bins are used, the patch will fall within bin 4, and the robots in bin 5, so there will be a very poor match ($D_n = 1$). And if 3 bins are used, the patch of pucks and cluster of robots will both fall within bin 3—a perfect match ($D_n = 0$). The cumulative distribution functions for this example are illustrated in Figure 6.24

Additionally, if the bins are translated or rotated within the environment, the measure may produce a different result. These latter problem may be partially alleviated by ensuring that each experiment is repeated several times, and the average result taken (as was done in the experiments in Chapter 5, though not in the example illustrated in Figure 5.7). If the statistic is averaged over many trials, we can expect that fluctuations due to slight changes in the position of bin boundaries will not contribute greatly to the average error.

Alternatively, we might measure D_n using a variety of bin sizes and aggregate the many results, for instance via averaging or simply looking at the worst-case scenario. However, we expect that this will not be necessary if a sufficiently large number of trials is performed. In any case, the issue is common among sampling problems.

6.3 Violation of the ideal free assumptions

Are adaptive bucket-brigading robot foragers ideal free? We argue that they are not, and that this explains the deviation of the observed distribution from the ideal free distribution.

First, the robots are not ideal. They cannot detect pucks, let alone clusters of pucks, at a distance. They have no memory aside from the location of their work-site, so upon discovering pucks or clusters of pucks they cannot store this information except by modifying the work-site (this is the inspiration for the relocation algorithm). Thus, far from having perfect information about the best location for their work sites, they have (almost) no information whatever.

Second, the robots are not free. The extensive study of interference in foraging shows that robots physically prevent one another from getting from place to place, including to and from clusters. In addition, it takes time to travel to distant clusters, and there is therefore an opportunity cost in terms of time not spent foraging. Having arrived at a cluster, interference now hinders the robot's work at foraging.

These problems are closely related to the assumptions made in the proof in Section A.1.1. In particular, the way in which the robots are not free – the fact that they must travel to the next location at which they will forage (search for pucks) – is a violation of the assumption that their search is random. The robot looks for pucks all along its route, which means the sequence of points checked for pucks is not (uniformly) random, but rather, the locus of a search is statistically dependent on the locus of prior searches.

Thus, the simplicity of the robots prevents them from being ideal, though additional sensing and computing resources would make them "more ideal". Interference prevents the robots from being free, though research in robot foraging (including this work) help to ameliorate interference and thus make robots "more free".

Chapter 7

Conclusions and future work

7.1 On bucket-brigade foraging

In Chapters 4 and 5 we found ways to improve the bucket-brigading approach to the foraging problem by following the biologically-inspired intuition that the distribution of robots should be similar to the distribution of pucks. Relocation should allocate more (or less) robots where more (or less) are needed, and adaptive ranging adjusts the bucket-brigading parameter to compensate for increased (or decreased) crowding.

We showed that robots using these enhancements outperformed conventional, nonadaptive bucket-brigadiers. In Chapter 6, however, we also found that the adaptations we developed were insufficient to arrange the robots in what our intuition told us was the best distribution – the ideal free distribution. This leads us to suspect that there is yet more room for improvement in the bucket-brigading model. We provided some suggestions for why our algorithm fails in this regard, which hopefully provides insight into what may be necessary to achieve the ideal free distribution.

7.2 Contributions

Bucket-brigading is a well-established, but rigid, algorithm. The two extensions provided in this thesis—adaptive ranging and relocation—are easy to implement and are shown to improve the performance of bucket-brigading. We would hope that researchers in the field of robotic foraging consider these enhancements whenever they consider bucket-brigading in their own work. Thus our algorithm is a contribution to the body of knowledge in the field of robotic foraging.

To my knowledge there has been no formal analysis of the question of whether the distribution of robots in a foraging scenario matches the distribution of pucks, or indeed a statistical analysis of the distribution of robots at all. A previous mathematical model of task allocation in foraging has been developed that compares distributions of robots and pucks[11], but in that work, the *nominal* characteristic of “puck type” is the property whose distribution is examined, whereas in this work, the distribution is of *position*. In this work we use the Kolmogorov-Smirnov test for samples in two dimensions to answer this question for my own robots, and we would hope that other researchers consider this test whenever they ask the question, “Are my robots arranged in such-and-such a distribution?” Thus my application of the K-S test is a practical contribution to the *methodology* of research in robotic foraging.

7.3 Future work

In analyzing the results of the experiments using relocation, we noticed two trends:

- When increasing the value of the relocation parameter, the performance of the foragers was observed to decrease when the clustering parameter was less than 5, but increase when the clustering parameter was greater than or equal to 5, in a monotonic fashion (as the clustering parameter increased from 0, the performance penalty decreased). Details are shown in Table B.4 and plotted in Figure 5.6.
- When increasing the value of the relocation parameter, the distribution of robots' work sites was observed to diverge from the puck distribution (measured using the Kolmogorov-Smirnov test) for clustering values up to 3, and to converge to the puck distribution for clustering values 3 and above. Details are in Table B.5 and plotted beginning in Figure 6.3.

The similarity between these trends hints at an avenue of further analysis: what exactly is the relationship between the performance of the foragers and the degree of their adaptation to the puck distribution? These results, coupled with the reasoning behind the ideal free distribution (Section 6.1) suggest that at the very least, the two trends are correlated, and that a causal relationship may be found.

Another avenue of future study would be to address the fact that currently, the two enhancements to bucket-brigading described in this thesis (adaptive ranging and adaptive work-site relocation) are some what ad-hoc and disconnected, in that the effects of each are studied without concern for any dependency on the effects of the other. For example, gathering more robots around a cluster of densely-distributed pucks would call for decreased work-area radii to be produced by the adaptive ranging algorithm, in order to thwart any increased interference caused by all the new traffic, and also for increased work-area radii in the case of robots who have not relocated near the cluster). While the robots in the experiments on adaptive relocation did use adaptive ranging as well, the converse was not the case. The interaction between the two algorithms merits further study, and a unified model of how the new parameters (rate of change of work site radius, and rate of relocation) simultaneously affect performance.

7.3.1 Verification of reasoning

In Chapters 4 and 5 we suggested that, since our adaptive foraging controllers improved performance over fixed-range and fixed-work-site controllers, our reasoning that adapting to the distributions of robots and pucks is "the right way to go" and that that was indeed what the robots were doing. In Chapter 6 we developed a statistical test to argue that the robots were indeed adapting their spatial distribution to the spatial distribution of pucks, albeit imperfectly for reasons given.

We have not done any such analysis for the results on adaptive ranging in Chapter 4. Therefore we suggest the following experiments to support the claims of that chapters:

1. Measure local interference (per-robot proportion of time spent avoiding collisions) and compare with local robot density. Hypothesis: strong correlation in robots without adaptive ranging, weak correlation in robots with adaptive ranging.

2. Measure global interference (proportion of total time spent avoiding collisions over all robots) and compare with performance. Hypothesis: strong negative correlation or inverse proportion.

3. Measure both local and global interference in the adaptive and non-adaptive cases and compare. Hypothesis: should see significantly less of both types in the adaptive case.

7.3.2 Balancing relocation and low interference

In Section 5.4.1 we noted that adaptive ranging is a prerequisite for relocation since the latter algorithm is likely to imply that many robots will be working close together, increasing interference. No experiments have been done to determine whether adaptive-range bucket-brigading is sufficient to avoid spatial interference in this scenario, so we propose that measurements of interference as suggested in the previous section be undertaken to determine whether this is the case.

We propose a modification of the relocation algorithm to help deal with this interference problem. Given that a variety of different values of the relocation parameter ρ (see Figure 5.5) enabled relocation, we propose that this parameter take a large value of little interference was experienced since the robot entered the `searching` state, and a small value if the the robot has experienced a large amount of interference since entering the `searching` state.

In essence, the above algorithm would (we conjecture) prevent robots from relocation too close to clusters that already have many robots near them.

7.4 Summary

This work was motivated by Dylan Shell and Maja Matarić's work in bucket-brigading [27]. Those authors developed a robot controller that suffered from less interference in very dense swarms than so-called homogeneous systems in which all robots explored the entire space and delivered found pucks all the way to the home zone.

This thesis begins with that bucket-brigading model and takes the following steps forward:

- added adaptive ranging, a novel algorithm, to bucket-brigading, allowing individual robots to search for the work-area size most appropriate for their local level of interference, which often improved performance and scaling;
- added relocation, a novel algorithm, to bucket-brigading, allowing the distribution of robots' work-sites to become similar to the distribution of pucks, which often improved performance; and
- analyzed the degree of adaptation due to relocation by applying a well-known statistical method.

Appendix A

Proof of algorithms

A.1 Proof of correctness of relocation algorithm

Algorithm 1 describes the process by which a robot using the *relocation* process updates its estimate of its work-site – *i.e.*, the place it will return to after it drops off a puck; also, the center of its work area. The algorithm's goal is to allow robots' work sites to dynamically adapt to the distribution of pucks, so that more resources (robots) are allocated to regions (clusters) with more work (pucks). By showing that the algorithm causes the work sites to travel uphill (*i.e.*, via gradient ascent), the following theorems and proofs demonstrate correctness of this algorithm, given some stated assumptions.

A.1.1 Normally-distributed pucks, random search

We will first assume that pucks are normally distributed and that the robot's search is ideally random. Here we ignore the fact that since a robot wishes to search a region for pucks, it must travel to that region, and there is a chance that a puck will be discovered along the way. As a result, the probability of discovering a puck in any given region is *dependent* on the current location of the robot; this dependency is ignored in this section.

Definition 2. A *random sequence* is a pairwise-independent sequence of points in $\mathbb{R}^2$.

Definition 3. A *random search* is a random sequence of locations inspected, each in constant time, for pucks.

Next we will formally define some terms used throughout this thesis:

Definition 4. A robot's *work-site* is a real-valued point in the foraging environment. The robot will return to this point after dropping off a puck and before beginning to search for new pucks. The initial location of a robot's work-site is the location of the robot when the experiment begins.

Definition 5. A robot's *work-area* is a disc centered on the robot's work-site. A bucket-brigading robot will drop off any puck it is carrying when it leaves its work-area. Unless otherwise specified, a bucket-brigading robot will stay within its work-area while searching for pucks.

The above definitions allow us to describe formally what we mean by an "ideally random" search.

Theorem 1. *If pucks are normally distributed, the random sequence* $(\mathbf{x}_i)$ *of work-sites of a robot using a random search and relocation with parameter* ρ *will perform gradient ascent on the puck distribution in the expected case.*

Theorem 1 says that, if we imagine each robot's search to be truly random, then we can expect its work-site to travel in the direction of the *gradient* of the distribution of pucks.

Proof. Let $p : S \to \mathbb{R}$ be the probability density function for the distribution of pucks in S, a normal distribution with mean $\mathbf{0}$ and standard deviation σ_p. A robot searching from work-site $\mathbf{x}_i$ searches using a random search until it encounters a puck at location $\mathbf{x}_e$.

Using relocation, the robot's new work-site is given by

$$\mathbf{x}_{i+1} = \rho\mathbf{x}_e + (1-\rho)\mathbf{x}_i.$$

The displacement of the robot's work-site at each step is given by

$$\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i = \rho(\mathbf{x}_e - \mathbf{x}_i).$$

Because of our assumption of random search, $\mathbf{x}_e$ is independent of $\mathbf{x}_i$, and this displacement $\Delta\mathbf{x}_i$ is a normally distributed random variable with mean

$$E(\Delta\mathbf{x}_i) = \rho(E(\mathbf{x}_e) - \mathbf{x}_i) = -\rho\mathbf{x}_i$$

and standard deviation

$$\sigma(\Delta\mathbf{x}_i) = \rho\sigma(\Delta\mathbf{x}_e) = \rho\sigma_p.$$

The vector $E(\Delta\mathbf{x}_i) = -\rho\mathbf{x}_i$ always points from the current work-site toward the mean of the puck distribution at the origin, and is therefore parallel to the gradient ∇p of the puck distribution. In other words, the vector $\Delta\mathbf{x}_i$ is parallel to the gradient ∇p in the expected case.

□

A.1.2 Arbitrarily distributed pucks

Instead of considering the hat distribution used in the experiments in Chapter 5 (since relocation events can only occur on the line segment connecting a robot's initial work-site $\mathbf{x}_i$ and a point inside the "hat", the displacement $\Delta\mathbf{x}_i$ will always point towards the cluster), we will now prove the algorithm for arbitrarily distributed pucks.

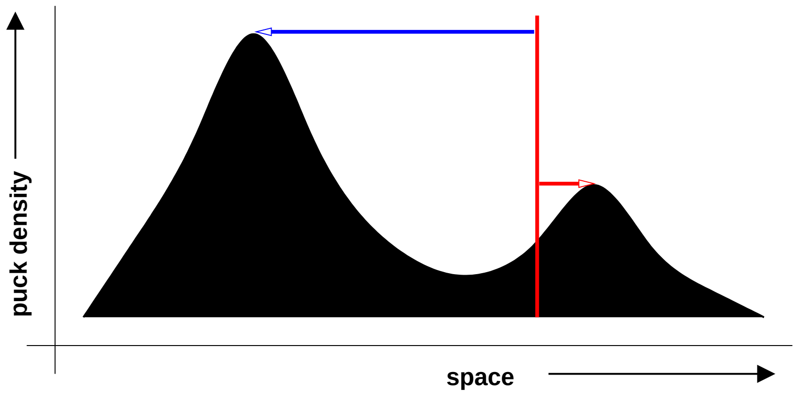


Figure A.1: Robot at the red/light line using a random search is more likely to relocate its work site along the blue/dark arrow than the red arrow.

Here we must abandon the simplifying assumption of random search as defined above. Since the distribution of pucks may be multimodal, it is possible that the most likely point for a robot's work-site after relocation is not in the direction of the local gradient. In Figure A.1, a robot (at the red/light line) following a random search is more likely to relocate its work-site along the blue/dark arrow than along the red arrow.

Instead we introduce a more realistic notion of search: the *locally random search*.

Definition 6. A *locally random search* is a random search confined to a circular neighborhood of radius ϵ, the work-area radius, centered on point $\mathbf{x}_i$, the work-site.

This notion of search is more realistic than that of a random search, since it recognizes that a robot may not detect (and therefore relocate its work-site towards) an arbitrarily distant puck in finite time.

Theorem 2. *Suppose a robot not located at a local extremum of the puck density function $f_p(\mathbf{x})$ is using a locally random search. Then, the robot will, in the expected case, relocate its work-site in the direction of ∇f_p.*

Proof. Choose a value of ϵ sufficiently small that:

- for $r = 1, \ldots,$ number of robots, robot r, searching at velocity v, for duration of at least ϵ/v, is using a locally random search of radius ϵ around a work-site $\mathbf{x}_i^{(r)}$; and
- the point in the ϵ-neighborhood around work-site $\mathbf{x}_i^{(r)}$ that maximizes the puck density distribution function lies in the direction ∇f_p. This will be possible as long as the puck distribution is sufficiently smooth.

At any given moment during a foraging task (real or simulated), the local (within ϵ of the work-site) distribution of pucks will be far from smooth since pucks are discrete objects. However, by the central limit theorem, the average of local puck distributions over a sufficiently large number of trials will be approximately normal and therefore smooth. In other words, over many trials in precisely the same local configuration of pucks with precisely the same work-site and work-area radius, we can say that the distribution of pucks is "normal on average".

Therefore, the conditions for Theorem 1 hold within ϵ of the work-site in the expected case, so we can conclude that the expected displacement of the work-site will be in the direction ∇f_p and therefore the sequence of work-sites will perform gradient ascent.

□

Appendix B

Experimental data

B.1 Adaptive ranging

puck density	robots	mean pucks foraged	std. dev.
0.781	20	17.100	5.549
0.781	40	39.250	5.635
0.781	60	81.400	5.287
0.781	80	79.000	7.621
0.781	100	102.100	6.893
0.781	120	106.800	10.002
0.781	140	142.400	15.280
0.781	160	141.700	10.602
0.781	180	171.550	13.910
0.781	200	158.500	9.340
0.781	220	159.850	16.325
0.781	240	184.000	19.385
0.781	260	184.750	19.853
0.781	280	188.650	18.388
0.781	300	200.250	21.296
0.781	320	204.000	22.395
0.781	340	212.200	15.553
0.781	360	197.800	21.536
0.781	380	223.350	19.428
0.781	400	209.900	20.384
0.781	420	215.900	24.675

puck density	robots	mean pucks foraged	std. dev.
0.781	440	216.750	24.997
0.781	460	227.050	24.527
0.781	480	233.450	19.802
0.781	500	215.500	15.754
3.125	20	7.350	7.328
3.125	40	20.400	4.787
3.125	60	73.850	9.659
3.125	80	60.350	10.919
3.125	100	67.800	4.661
3.125	120	83.000	14.773
3.125	140	112.550	17.466
3.125	160	96.650	10.779
3.125	180	127.600	11.234
3.125	200	115.700	12.336
3.125	220	123.700	8.438
3.125	240	148.200	14.658
3.125	260	127.100	14.571
3.125	280	132.800	19.774
3.125	300	157.150	15.547
3.125	320	165.750	21.145
3.125	340	140.750	13.529
3.125	360	155.700	14.456
3.125	380	165.850	22.486
3.125	400	156.150	15.667
3.125	420	152.950	16.291
3.125	440	164.350	14.695
3.125	460	165.450	20.245
3.125	480	178.600	24.772
3.125	500	165.250	17.915

Table B.1: Performance for robots in the adaptive ranging experiments in Chapter 4, averaged over thirty trials. Plotted in Figures 4.2 and 4.3 (red curves).

B.2 Effects of clustering

robots	clustering	pucks foraged (30 trials)	mean	std. dev.
36	0.00	190 193 228 309 278 210 188 200 230 238 207 195 198 311 222 198 226 183 259 171 237 224 217 231 187 194 241 246 242 184	221.233	34.328
36	1.00	261 260 232 194 208 197 204 193 272 234 267 207 230 224 204 216 188 310 207 227 208 277 179 218 213 187 233 232 183 188	221.767	31.409
36	2.00	218 265 191 203 240 185 232 210 172 204 232 201 172 244 241 245 237 235 230 249 237 197 224 243 258 195 239 182 191 205	219.233	25.677
36	3.00	170 176 178 161 179 191 148 183 164 162 138 169 204 198 171 169 185 212 179 159 198 215 177 178 168 153 174 203 175 147	176.133	18.583
36	4.00	136 167 131 129 113 126 151 159 146 141 139 132 162 109 162 133 136 130 139 141 129 144 163 113 153 135 124 156 171 149	130.633	16.003
36	5.00	136 130 124 111 101 98 112 147 121 154 105 127 108 104 141 145 125 110 120 137 124 106 110 145 126 129 114 124 153 103	123.000	15.933
36	10.00	79 55 112 74 96 73 73 57 79 76 89 80 91 84 66 91 67 58 68 66 80 74 74 76 73 75 81 91 69 84	77.033	11.932
36	15.00	50 43 64 48 62 55 55 51 48 43 58 49 48 47 59 59 39 58 62 60 50 46 41 55 40 47 55 58 52 49	51.700	6.842
100	0.00	266 309 372 238 398 321 353 276 269 275 319 276 326 225 248 256 336 312 300 275 256 308 395 359 349 299 378 359 346 362	312.033	47.368

robots	clustering	pucks foraged (30 trials)	mean	std. dev.
100	1.00	334 313 267 324 354 321 342 298 326 289 308 307 305 340 331 273 386 241 307 300 349 352 368 285 307 343 391 338 260 314	319.100	34.494
100	2.00	359 356 240 353 448 266 319 342 361 339 245 284 276 328 333 296 248 353 328 284 291 246 313 356 365 284 310 298 337 316	315.800	45.009
100	3.00	447 324 377 365 437 419 382 438 391 343 436 436 456 404 384 416 428 366 281 417 361 390 404 416 315 367 423 409 453 422	396.900	42.045
100	4.00	417 341 321 409 358 410 425 407 361 382 422 414 344 355 365 322 270 381 385 353 382 401 380 404 375 385 374 401 360 378	386.067	33.906
100	5.00	358 301 342 350 300 345 361 301 340 343 340 342 340 319 343 321 351 314 311 330 340 286 306 292 347 331 355 345 313 287	328.467	21.910
100	10.00	177 211 191 236 197 212 176 192 219 203 200 208 198 220 162 189 232 232 194 181 197 245 218 200 225 198 186 217 217 196	204.300	19.053
100	15.00	170 160 155 134 160 134 197 128 161 173 148 168 167 150 171 168 171 150 142 135 168 140 163 144 177 158 148 127 163 148	155.933	15.983

Table B.2: Data from experiments showing the effects of clustering on performance of the adaptive bucket-brigade algorithm.

B.3 Results of relocation experiments

robots	clustering	relocation	mean pucks	std. dev.	95% CI
36	0	0.00	221.000	33.837	12.108
36	0	0.25	179.667	23.223	8.310
36	0	0.50	161.367	25.380	9.082
36	0	0.75	139.700	14.372	5.143
36	0	1.00	130.300	19.701	7.050
36	1	0.00	227.867	30.167	10.795
36	1	0.25	181.200	21.430	7.668
36	1	0.50	159.800	21.931	7.848
36	1	0.75	139.000	14.613	5.229
36	1	1.00	124.267	15.590	5.579
36	2	0.00	209.267	24.610	8.807
36	2	0.25	178.133	18.242	6.528
36	2	0.50	168.767	16.206	5.799
36	2	0.75	147.467	16.498	5.904
36	2	1.00	142.667	13.516	4.837
36	3	0.00	175.433	17.853	6.388
36	3	0.25	177.333	15.460	5.532
36	3	0.50	182.067	14.313	5.122
36	3	0.75	178.100	14.981	5.361
36	3	0.00	175.400	12.891	4.613
36	4	0.00	141.500	14.525	5.198
36	4	0.25	156.767	18.843	6.743
36	4	0.50	169.933	12.819	6.487
36	4	0.75	176.533	16.041	5.740
36	4	1.00	178.300	12.163	4.353
36	5	0.00	120.633	15.940	5.704
36	5	0.25	139.867	10.151	3.633
36	5	0.50	157.233	15.760	5.640
36	5	0.75	171.567	14.562	5.211
36	5	1.00	179.800	14.372	5.143
36	10	0.00	76.900	11.973	4.285
36	10	0.25	85.467	11.313	4.048
36	10	0.50	105.567	14.739	5.274
36	10	0.75	120.567	15.422	5.519

36	10	1.00	146.133	12.260	4.387
36	15	0.00	56.100	9.386	3.359
36	15	0.25	64.033	16.053	5.744
36	15	0.50	77.167	12.975	4.643
36	15	0.75	92.900	18.414	6.590
36	15	1.00	124.367	17.604	6.299
100	0	0.00	313.833	36.545	13.077
100	0	0.25	354.833	58.541	20.949
100	0	0.50	376.300	46.280	16.561
100	0	0.75	371.967	43.498	15.566
100	0	1.00	347.433	53.187	19.033
100	1	0.00	309.200	36.456	13.046
100	1	0.25	382.333	53.440	19.123
100	1	0.50	383.000	48.908	17.502
100	1	0.75	369.833	32.132	11.498
100	1	1.00	356.933	33.470	11.977
100	2	0.00	323.600	39.514	14.140
100	2	0.25	337.833	56.128	20.085
100	2	0.50	367.167	49.042	17.549
100	2	0.75	356.767	49.153	17.589
100	2	1.00	350.100	47.938	17.154
100	3	0.00	388.033	48.904	17.500
100	3	0.25	364.867	55.950	20.021
100	3	0.50	347.000	70.085	25.080
100	3	0.75	337.567	61.040	21.843
100	3	1.00	312.267	81.942	29.323
100	4	0.00	368.600	24.683	8.833
100	4	0.25	387.033	22.852	8.178
100	4	0.50	383.033	45.910	16.429
100	4	0.75	357.467	66.353	23.744
100	4	1.00	334.133	69.102	24.728
100	5	0.00	335.667	14.614	5.229
100	5	0.25	354.833	20.682	7.401
100	5	0.50	390.467	24.936	8.923
100	5	0.75	383.167	42.119	15.072
100	5	1.00	367.500	56.409	20.186
100	10	0.00	208.033	17.629	6.308

100	10	0.25	222.300	18.666	6.679
100	10	0.50	263.233	18.956	6.783
100	10	0.75	292.600	20.287	7.260
100	10	1.00	328.067	19.311	6.910
100	15	0.00	150.033	15.751	5.636
100	15	0.25	171.033	18.754	6.711
100	15	0.50	191.933	21.828	7.811
100	15	0.75	226.967	23.714	8.486
100	15	1.00	282.833	24.552	8.786

Table B.3: Mean performance (in terms of number of pucks delivered over thirty trials), standard deviation, and width of the 95% confidence interval (assuming a normal distribution).

B.4 Relocation results (percent increase)

robots	clustering	relocation	mean pucks (%)	std. dev. (%)	95% CI (%)
36	0	0.00	0.000	15.311	5.479
36	0	0.25	-18.703	10.508	3.760
36	0	0.50	-26.983	11.484	4.110
36	0	0.75	-36.787	6.503	2.327
36	0	1.00	-41.041	8.915	3.190
36	1	0.00	0.000	13.239	4.737
36	1	0.25	-20.480	9.404	3.365
36	1	0.50	-29.871	9.624	3.444
36	1	0.75	-38.999	6.413	2.295
36	1	1.00	-45.465	6.842	2.448
36	2	0.00	0.000	11.760	4.208
36	2	0.25	-14.877	8.717	3.119
36	2	0.50	-19.353	7.744	2.771
36	2	0.75	-29.532	7.884	2.821
36	2	1.00	-31.825	6.459	2.311
36	3	0.00	0.000	10.176	3.642
36	3	0.25	1.083	8.813	3.154
36	3	0.50	3.781	8.159	2.920
36	3	0.75	1.520	8.539	3.056
36	3	1.00	-0.019	7.348	2.629
36	4	0.50	0.000	9.873	4.996

36	4	0.75	-1.305	8.968	3.209
36	4	1.00	-0.317	6.800	2.433
36	4	0.00	-20.891	8.121	2.906
36	4	0.25	-12.356	10.534	3.770
36	4	0.50	-4.994	7.167	3.627
36	5	0.00	0.000	13.214	4.729
36	5	0.25	15.944	8.415	3.011
36	5	0.50	30.340	13.064	4.675
36	5	0.75	42.222	12.071	4.320
36	5	1.00	49.047	11.914	4.263
36	10	0.00	0.000	15.570	5.572
36	10	0.25	11.140	14.711	5.264
36	10	0.50	37.278	19.167	6.859
36	10	0.75	56.784	20.055	7.177
36	10	1.00	90.030	15.943	5.705
36	15	0.00	0.000	16.730	5.987
36	15	0.25	14.141	28.615	10.240
36	15	0.50	37.552	23.128	8.276
36	15	0.75	65.597	32.824	11.746
36	15	1.00	121.687	31.380	11.229
100	0	0.00	0.000	11.645	4.167
100	0	0.25	13.064	18.654	6.675
100	0	0.50	19.904	14.747	5.277
100	0	0.75	18.524	13.860	4.960
100	0	1.00	10.706	16.947	6.065
100	1	0.00	0.000	11.790	4.219
100	1	0.25	23.652	17.283	6.185
100	1	0.50	23.868	15.818	5.660
100	1	0.75	19.610	10.392	3.719
100	1	1.00	15.438	10.825	3.874
100	2	0.00	0.000	12.211	4.370
100	2	0.25	4.398	17.345	6.207
100	2	0.50	13.463	15.155	5.423
100	2	0.75	10.249	15.190	5.435
100	2	1.00	8.189	14.814	5.301
100	3	0.00	0.000	12.603	4.510
100	3	0.25	-5.970	14.419	5.160

100	3	0.50	-10.575	18.062	6.463
100	3	0.75	-13.006	15.731	5.629
100	3	1.00	-19.526	21.117	7.557
100	4	0.00	0.000	6.696	2.396
100	4	0.25	5.001	6.200	2.219
100	4	0.50	3.916	12.455	4.457
100	4	0.75	-3.020	18.001	6.442
100	4	1.00	-9.351	18.747	6.709
100	5	0.00	0.000	4.354	1.558
100	5	0.25	5.710	6.161	2.205
100	5	0.50	16.326	7.429	2.658
100	5	0.75	14.151	12.548	4.490
100	5	1.00	9.484	16.805	6.014
100	10	0.00	0.000	8.474	3.032
100	10	0.25	6.858	8.972	3.211
100	10	0.50	26.534	9.112	3.261
100	10	0.75	40.651	9.752	3.490
100	10	1.00	57.699	9.283	3.322
100	15	0.00	0.000	10.498	3.757
100	15	0.25	13.997	12.500	4.473
100	15	0.50	27.927	14.549	5.206
100	15	0.75	51.277	15.806	5.656
100	15	1.00	88.514	16.364	5.856

Table B.4: Mean performance (in terms of number of pucks delivered over thirty trials), standard deviation, and width of the 95% confidence interval (assuming a normal distribution). In this table, values are reported in terms of the *percent increase* over the no-relocation case.

B.5 Kolmogorov-Smirnov statistics

Parameter			Time						
robots	clustering	relocation	0:00	0:10	0:20	0:30	0:40	0:50	1:00
100	0	0.00	0.06	0.09	0.17	0.27	0.34	0.38	0.40
100	0	0.50	0.08	0.09	0.16	0.26	0.32	0.35	0.37
100	0	1.00	0.08	0.09	0.18	0.26	0.30	0.28	0.27
100	1	0.00	0.10	0.09	0.15	0.23	0.30	0.34	0.37
100	1	0.50	0.09	0.08	0.16	0.25	0.31	0.32	0.34

Parameter			Time						
robots	clustering	relocation	0:00	0:10	0:20	0:30	0:40	0:50	1:00
100	1	1.00	0.10	0.10	0.17	0.25	0.28	0.28	0.26
100	2	0.00	0.28	0.21	0.15	0.12	0.16	0.19	0.22
100	2	0.50	0.27	0.19	0.12	0.13	0.19	0.22	0.24
100	2	1.00	0.26	0.17	0.10	0.13	0.19	0.24	0.25
100	3	0.00	0.59	0.49	0.40	0.34	0.31	0.30	0.31
100	3	0.50	0.58	0.46	0.34	0.26	0.21	0.19	0.19
100	3	1.00	0.58	0.45	0.30	0.21	0.15	0.12	0.12
100	4	0.00	0.73	0.65	0.55	0.48	0.44	0.42	0.42
100	4	0.50	0.75	0.62	0.51	0.42	0.36	0.33	0.29
100	4	1.00	0.75	0.61	0.45	0.33	0.24	0.18	0.14
100	5	0.00	0.76	0.70	0.62	0.56	0.53	0.51	0.51
100	5	0.50	0.75	0.67	0.56	0.47	0.40	0.36	0.33
100	5	1.00	0.75	0.66	0.53	0.41	0.31	0.24	0.18
100	10	0.00	0.66	0.64	0.61	0.59	0.57	0.57	0.58
100	10	0.50	0.66	0.63	0.58	0.52	0.47	0.46	0.45
100	10	1.00	0.66	0.63	0.56	0.49	0.42	0.37	0.34
200	0	0.00	0.07	0.11	0.24	0.39	0.50	0.55	0.57
200	0	0.50	0.07	0.11	0.25	0.40	0.50	0.52	0.53
200	0	1.00	0.06	0.12	0.27	0.39	0.45	0.45	0.43
200	1	0.00	0.09	0.09	0.22	0.36	0.46	0.52	0.54
200	1	0.50	0.09	0.09	0.23	0.38	0.48	0.51	0.52
200	1	1.00	0.09	0.08	0.23	0.37	0.42	0.43	0.42
200	2	0.00	0.27	0.15	0.07	0.14	0.24	0.32	0.37
200	2	0.50	0.26	0.15	0.09	0.17	0.26	0.32	0.35
200	2	1.00	0.26	0.14	0.09	0.18	0.26	0.29	0.29
200	3	0.00	0.58	0.44	0.31	0.25	0.20	0.19	0.19
200	3	0.50	0.57	0.41	0.27	0.18	0.13	0.13	0.15
200	3	1.00	0.57	0.41	0.26	0.17	0.12	0.11	0.11
200	4	0.00	0.75	0.63	0.51	0.43	0.39	0.35	0.33
200	4	0.50	0.74	0.60	0.45	0.36	0.31	0.28	0.26
200	4	1.00	0.75	0.60	0.44	0.33	0.25	0.20	0.17
200	5	0.00	0.74	0.67	0.57	0.50	0.45	0.42	0.40
200	5	0.50	0.73	0.65	0.53	0.45	0.39	0.35	0.32
200	5	1.00	0.73	0.64	0.50	0.41	0.34	0.30	0.27
200	10	0.00	0.67	0.64	0.61	0.58	0.56	0.55	0.54

Parameter			Time						
robots	clustering	relocation	0:00	0:10	0:20	0:30	0:40	0:50	1:00
200	10	0.50	0.65	0.61	0.57	0.53	0.50	0.48	0.46
200	10	1.00	0.66	0.62	0.57	0.52	0.47	0.43	0.42
300	0	0.00	0.06	0.11	0.26	0.42	0.52	0.57	0.59
300	0	0.50	0.05	0.12	0.29	0.44	0.53	0.57	0.58
300	0	1.00	0.06	0.11	0.27	0.42	0.48	0.50	0.50
300	1	0.00	0.08	0.08	0.22	0.36	0.48	0.54	0.58
300	1	0.50	0.09	0.09	0.24	0.39	0.49	0.53	0.55
300	1	1.00	0.08	0.09	0.25	0.39	0.48	0.50	0.50
300	2	0.00	0.26	0.14	0.08	0.16	0.25	0.33	0.38
300	2	0.50	0.27	0.14	0.08	0.17	0.26	0.33	0.36
300	2	1.00	0.25	0.13	0.10	0.19	0.25	0.28	0.30
300	3	0.00	0.57	0.41	0.29	0.21	0.18	0.18	0.19
300	3	0.50	0.57	0.40	0.27	0.19	0.15	0.15	0.16
300	3	1.00	0.58	0.41	0.28	0.20	0.16	0.14	0.15
300	4	0.00	0.74	0.62	0.48	0.40	0.34	0.31	0.28
300	4	0.50	0.74	0.60	0.46	0.37	0.30	0.27	0.25
300	4	1.00	0.74	0.58	0.43	0.34	0.28	0.25	0.24
300	5	0.00	0.73	0.66	0.55	0.48	0.43	0.39	0.36
300	5	0.50	0.74	0.65	0.54	0.45	0.39	0.35	0.34
300	5	1.00	0.74	0.65	0.54	0.44	0.38	0.34	0.33
300	10	0.00	0.66	0.62	0.58	0.55	0.53	0.51	0.49
300	10	0.50	0.67	0.63	0.58	0.55	0.51	0.49	0.48
300	10	1.00	0.66	0.62	0.57	0.52	0.49	0.46	0.46

Table B.5: Two-dimensional Kolmogorov-Smirnov statistics for distributions of pucks and work-area centers at ten-minute intervals during the simulation. K-S statistics are averaged over thirty trials. A larger value indicates greater difference between the two distributions. Data are plotted in Figures 6.3 through 6.23.

Appendix C

Source code

To aid in reproduction of the experimental results documented herein, the reader may find source code for the simulators used in the above research. The same simulator was used throughout; however, changes were made with each subsequent experiment. Versions of the code are provided that correspond to the programs used to generate the experimental data quoted or summarized in this work.

Chapter 4 is based on experiments performed with version 1.0 of the simulator.

Chapter 5 is based on experiments performed with version 2.0 of the simulator.

Chapter 6 reproduced the experimental data used in Chapter 5 but with the added feature of recording the positions of work-sites and pucks so that their distributions could be compared. Since the simulator is deterministic (influenced by a pseudorandom number generator with predetermined seed), the experiments themselves, in terms of locations of objects and behavior of robots, were identical. This is version 2.5 of the simulator.

Included with the source code is instructions for building, configuring, and running the code, along with a list of required software. All code was built and tested on 32- and 64-bit Linux computers.

The source code is available at the given URL or via the Autonomy Lab's website:

`http://autonomy.cs.sfu.ca`

Date	**Version**	**Description**
March 7, 2008	1.0	Adaptive bucket-brigade foraging simulation `http://www.cs.sfu.ca/~alein/personal/apbbf-1.0.tar.gz`
March 1, 2009	2.0	Foraging with clustering and relocation simulation `http://www.cs.sfu.ca/~alein/personal/apbbf-2.0.tar.gz`
September 8, 2009	2.5	Latest version as of publication `http://www.cs.sfu.ca/~alein/personal/apbbf-2.5.tar.gz`

Table C.1: URLs to various versions of simulator source code.

Bibliography

[1] C. Anderson, J. J. Broomsma, and III J. J. Bartholdi. Task partitioning in insect societies: bucket brigades. *Insected Sociaux*, 49:171–180, 2002.

[2] R. Beckers, O. Holland, and J.-L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 181–189. Cambridge, MA: MIT Press, 1994.

[3] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159, 1991.

[4] S. Brown, M. Zuluaga, Y. Zhang, and R. T. Vaughan. Rational aggressive behaviour reduces interference in a mobile robot team. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, pages 741–748, Seattle, Washington, July 2005.

[5] Y. Uny Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:226–234, 1997.

[6] S. Holmes D. M. Gordon, R. J. Schafer. Forager activation and food availability in harvester ants. *Animal Behaviour*, 71(4):815–822, April 2006.

[7] É. Danchin, L.-A. Giraldeau, and F. Cézilly, editors. *Behavioral Ecology.* Oxford University Press, New York, 2008.

[8] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior: From animals to animats*, pages 356–363, Cambridge, MA, USA, 1990. MIT Press.

[9] M. S. Fontán and M. J. Matarić. A study of territoriality: The role of critical mass in adaptive task division. In *In From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, pages 553–561, Cape Cod, MA, USA, 1996. MIT Press.

[10] S. D. Fretwell and H. L. Lucas. On territorial behavior and other factors influencing habitat distribution in birds. *Acta Biotheoretica*, 19(1):16–36, 1969.

[11] A. Galstyan and K. Lerman. Analysis of a stochastic model of adaptive task allocation in, robots. In *Engineering Self-Organizing Systems*, Lecture Notes in Computing Science, pages 167–179. Springer, Berlin, 2005.

[12] B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pages 317–323, Coimbra, Portugal, June 2003.

[13] D. Goldberg and M. J. Matarić. Maximizing Reward in a Non-Stationary Mobile Robot Environment. *Autonomous Agents and Multi-Agent Systems*, 6(3):287–316, 2003.

[14] O. Holland and C. Melhuish. Stigmergy, Self-Organization, and Sorting in Collective Robotics. *Artificial Life*, 5(2):173–202, 1999.

[15] S. P. Hubbell, L. K. Johnson, E. Stanislav, B. Wilson, and H. Fowler. Foraging by bucket-brigade in leaf-cutter ants. *Biotropica*, 12(3):210–213, 1980.

[16] C. R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30:85–101, 2000.

[17] A. Lein and R. T. Vaughan. Adaptive multi-robot bucket-brigade foraging. In *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, Artificial Life*, pages 337–342, Winchester, Southampton, UK, August 2008. MIT Press.

[18] A. Lein and R. T. Vaughan. Adapting to non-uniform puck distributions in robotic swarm foraging through work-site relocation. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems*, pages 601–606, October 2009.

[19] K. Lerman and A. Galstyan. Mathematical Model of Foraging in a Group of Robots: Effect of Interference. *Autonomous Robots*, 13(2):127–141, 2002.

[20] W. Liu, A. F. T. Winfield, J. Sa, J. Chen, and L. Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 15(3):289–305, 2007.

[21] P. Nonacs and J. L. Soriano. Patch sampling behaviour and future foraging expectations in argentine ants, *linepithema humile. Animal Behavior*, 55(3):519–527, March 1998.

[22] E. H. Østergaard, G. S. Sukhatme, and M. J. Matarić. Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In

Proceedings of the fifth international conference on Autonomous agents, pages 29–30. ACM Press New York, NY, USA, 2001.

[23] L. Panait and S. Luke. A pheromone-based utility model for collaborative foraging. In *Proceedings of the 2004 Conference on Autonomous Agents and Multiagent Systems*, pages 36–43, New York, NY, USA, 2004.

[24] J. A. Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notes of the Royal Astronomical Society.*, 202:615–627, 1983.

[25] P. E. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, and Maria Gini. Communication strategies in Multi-Robot Search and Retrieval: Experiences with MinDART. In *Proceedings of the Seventh International Symposium on Distributed Autonomous Robotic Systems (DARS-04)*, pages 301–310, Toulouse, France, 2004.

[26] T. Schmickl and K. Crailsheim. Trophallaxis among swarm-robots: A biologically inspired strategy for swarm robotics. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 377–382, 2006.

[27] D. A. Shell and M. J. Matarić. On foraging strategies for large-scale multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2717–2723, Beijing, China., Oct 2006.

[28] D. Sheskin. The Kolmogorov-Smirnov Test for Two Independent Samples. In *Handbook of parametric and nonparametric statistical procedures*, chapter 13, pages 453–463. Chapman & Hall, Boca Raton, FL, 3rd edition, 2004.

[29] R. T. Vaughan. Massively multi-robot simulations in stage. *Swarm Intelligence*, 2(2-4):189–208, 2008.

[30] R. T. Vaughan, K. Stóy, G. S. Sukhtame, and M. J. Matarić. Whistling in the dark: Cooperative trail following in uncertain localization space. In *Agents*, pages 187–194, Barcelona, Spain, 2000.

[31] J. Wawerla and R. T. Vaughan. Near-optimal mobile robot recharging with the rate-maximizing forager. In *Proceedings of the European Conference on Artificial Life*, pages 776–785, September 2007.

[32] T. H. Wonnacott and R. J. Wonnacott. *Introductory statistics*. John Wiley & Sons, Inc., Canada, 2nd edition, 1972.

[33] M. Zuluaga and R. T. Vaughan. Reducing spatial interference in robot teams by local-investment aggression. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2798–2805, Edmonton, AB, Canada, August 2005.

[34] M. Zuluaga and R. T. Vaughan. Reducing spatial interference in robot teams by local-investment aggression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2798–2805, Edmonton, AB, Canada, August 2005.