Central Place Foraging: Separating Search and Collection

A Thesis Presented to

The Faculty of the Computer Science Department

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

Nicholas Dolan-Stern

Advisor: Jason Isaacs

July 2018

© 2018
Nicholas Dolan-Stern
ALL RIGHTS RESERVED

APPROVED FOR MS IN COMPUTER SCIENCE

7/24/2018

Advisor: Dr. Jason Isaacs | Date

7/24/18

Dr. Brian Thoms | Date

7/24/2018

Dr. Houman Dallali | Date

APPROVED FOR THE UNIVERSITY

7/24/2018

Dr. Osman Özturgut | Date

Non-Exclusive Distribution License

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Central Place Foraging: Separating Search and Collection

Title of Item

Swarm Robotics, Search, Collection, Foraging

3 to 5 keywords or phrases to describe the item

Nicholas Dolan- Stern

Author(s) Name (Print)

Nicholas Dolan-Stern 7-25-2018

Author(s) Signature Date

This is a permitted, modified version of the Non-exclusive Distribution License from MIT Libraries and the University of Kansas.

Central Place Foraging: Separating Search and Collection

Nick Stern

July 25, 2018

Abstract

Central place foraging is a problem domain which consists of finding and delivering resources situated throughout an unknown environment to a singular collection depot. Foraging behaviors are the primary benchmark application of swarm robotics, which is the study of the complex group behavior that emerges from the local interactions of many simple individuals. A common issue within central place foraging approaches is inter-robot interference, a significant detractor from scalable group performance. To address this problem we propose a novel technique for central place foraging, the Multimodal approach. This technique separates a preliminary search phase from collection behavior, locating all of the resources within the environment before any are picked up, storing and sharing these locations amongst all of the agents. This information is then used by the collecting agents in order to select resources in areas in which there are no other agents, mitigating the effect of interference. The application of this approach to various simulated problem formulations resulted in a significant performance increase as compared to a baseline approach. This lends to our conclusion that a separation of search and collection can lead to the incorporation of more advanced routing techniques that further improve the performance of the foraging task.

Contents

List of Figures

1 Introduction

Swarm robotics is the study of the behavior that emerges from the interactions occurring between large numbers of simple embodied agents and their environment [2]. However, technological limitations have made such systems infeasible until only late in the last century. Swarm robotics is a relatively new field, emerging in the late 1980s as "the application of swarm intelligence to the study of multi-agent systems" [2]. The field is influenced heavily by biological systems of cells, insects, and animals that exhibit similar qualities to that which are trying to be replicated in these artificial systems [3]. The primary motivation of swarm robotics is the potential for swarms to solve large scale tasks that are difficult or impossible to be accomplished by single or small groups of agents. Additionally, swarm approaches offer scalability, robustness, and flexibility; meaning that they should be able to tolerate varying environments, group sizes, and individual failure rates without compromising the performance of the swarm [2]. Swarm robotics has a variety of studied applications, including patterned/formation movement, search, and foraging [4].

Foraging is used as the primary benchmark for evaluating swarm behavior due to the combination of tasks that it requires [5]. In order to successfully forage, agents must search and bring resources back to a collection depot while avoiding one another. This thesis is concerned primarily with central place foraging, when there exists only one collection depot. The overall task of central place foraging consists of finding desired resources within an unexplored environment and returning them to a collection depot location. Such depots are common features in organizational structures to reduce the complexity of sharing resources and communicating between the multiple agents acting within a logistical system. These depots, which are centers of activity, allow a fixed place for societal and commercial transactions as well as the opportunity for decisions to be distributed across chains of command. Common examples include ports, marketplaces, administrative capitals, as well as command posts. Such depots are abundant within nature, such as bee hives, bird nests, and anthills. The collection and delivery of resources to these depots therefore represents an important and useful logistical challenge for roboticists. Applications of central place foraging are found in exploration, land mine retrieval, as well as many other resource transportation problems [6–8]. While humans have been practicing foraging behavior for millenniums, encoding these practices into autonomous robotics systems creates many challenges.

Included amongst these challenges is the propensity of agents to get in each other's way, referred to throughout this thesis as the problem of interference. This has been cited by many researching within this domain as the primary obstacle of approach scalability [9–11]. One of the main goals of swarm robotics is to create systems capable of supporting large numbers of agents, numbering in the hundreds or even thousands. Typically, approaches thus far have found a critical number of agents, an amount after which performance continually decreases, in the single digits or dozens [6, 9, 12]. The reason for this has been consistently cited as being interference. Multiple agents operating within a shared space leads to interference [13], and central place foraging in particular concentrates agents around the collection depot area as well as resource clusters. Despite this, there exist no established and efficient techniques for interference mitigation.

Central place foraging concentrates agents around the collection depot as well as the pathways between the depot and resources, and each additional agent operating within these shared spaces increases the likelihood of agents obstructing each other [13]. The benefits of adding a second agent to help the first is readily apparent, however after many such additions there is a consistent detrimental effect on performance. Research in this area thus far has shown foraging efficiency quickly decreasing in proportion to the number of agents trying to forage beyond small numbers of agents. [6, 9, 12]. Despite foraging seeming to be a cooperative task, work in this area so far shows that agents seem to function best when kept far apart. Typically foraging behavior occurs in the following steps: search until a resource is found, collect it, bring it to the depot, followed by returning to search. This means that searching for and finding resources is integrated with collection behavior into a singular process. The spontaneous discovery of resources followed by the immediate need to bring them back to the activity center makes it extremely difficult for agents to route themselves in ways that both avoid others and as well as prevent the wasting of time that can occur from waiting for a clear path. In order to effectively address the problem of congestion, we propose that search and collection should be separated into two distinct modes of operation rather than switching constantly between the two behaviors. This allows agents to first discover information about the environment they are working in, as well as the locations of the resources, and then use this information to more effectively route and gather.

The present work thereby provides a novel approach to central place foraging; which separates search and collection into two modes in order to try and better manage interference. Prior to agents beginning collection, a preliminary search phase is used to determine the locations of the resources within the environment. Upon search completion, agents utilize these resource positions in conjunction with a sector locking congestion mitigation technique to efficiently collect the resources. Splitting search from collection allows agents to search uninterrupted in a deterministic pattern designed to quickly search an area of interest and use the locations gathered in the search phase to try and make better informed decisions about the order in which to collect the resources. This preliminary survey of the environment incurs an initial performance cost because no resources are collected until the survey is complete. We investigate the initial cost of such a search phase as well as its long term performance impact on various resource distributions. We demonstrate that this cost becomes insignificant due to the time saved by routing agents directly to and from resource locations during collection, as well as avoiding areas where other agents are collecting. The multimodal approach represents a proof of concept that the separation of search and collection can lead to the incorporation of more advanced mitigation techniques.

The remainder of this thesis is separated into the following chapters: Background, Problem and Multimodal Approach, Simulation Experiments, and Conclusions. The Background chapter provides the context in which our work is situated, both in field and problem domain. It begins with a historical overview of swarm robotics; much of which is shared by foraging applications as they represent the field's primary application. The foraging problem domain is the subject of the next few sections; these discuss the variations in problem formulations, the primary behaviors needing to be accomplished by foraging agents, as well as a discussion of the approaches most related to our work. This provides the reader with the context necessary to understand the problem formulation, and the approach described in the chapter of Problem and Approach. In Simulation Experiments, we describe the experiments designed to evaluate our approach as well as the data gathered from them. Results ends with a section analyzing this data and discussing their implications. The thesis is concluded with a summary of our findings, relating them to the aims of this thesis, described in the previous paragraph, as well as relating an agenda for future extension of this work.

2 Background

2.1 Historical Overview of Swarm Robotics

The conceptualization of swarm robotics, visualized in Figure 1, began in the late 80s when emerging technologies, control mechanisms, and decreasing hardware costs made the control of large numbers of robots seem to be reachable within a few years. The concept of behavior based design, one of the primary methodologies used in controlling swarm systems, was introduced by Brooks in 1986 [14]. This new method of robot controller design was not only robust, but also hardware resource efficient and scalable, which was extremely applicable to the limited resources of early mobile robots. Shortly after this time the uses of large scale, decentralized, simple robotics applications were proposed in two separate papers [15] and [16]. In 1987, [15] described how "gnat robots," could be controlled using controllers similar to that proposed by Brooks [14]. A year later, [16] introduced the term of "cellular robotics", describing a similar decentralized and self organizing system of robots using only limited communication, which attempted to address "the absence of a theoretical framework for dealing with distributed robotic systems" [16]. While the authors of [16] primarily describe the cellular robots in terms of mathematical modeling and properties, [15] addresses more of the hardware requirements and challenges of such systems. Together these represent the theoretical and practical basis for swarm robotics in its earliest forms. These swarms differentiated themselves from the more general field of multi-agent systems due to their decentralized control, simple components, and larger number of agents involved [2].

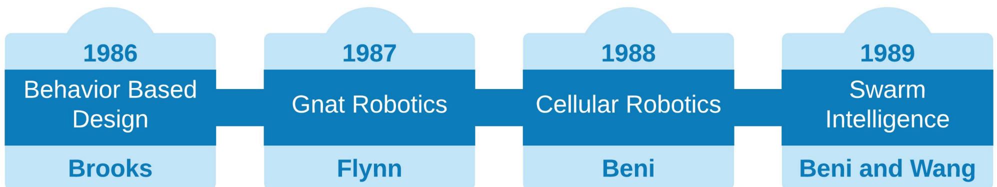


Figure 1: *The Conceptualization of Swarm Robotics*

Throughout the next several years, the study of early systems of cellular robotics and a recognition of their relatedness to biological systems helped bring about the transformation of swarm robotics from the academic "buzz words" of cellular and gnat robotics to a substantial component of multi-agent systems research, warranting its own study [2]. The separation of swarm robotics from the more general multi-agent systems seems to have occurred in between the early 1990s and 2000s. Throughout this time there existed a lack of agreement at this time as to what this area of research should be called, with researchers using different names such as "gnat robotics", "cellular robotics", "collective robotics", and "swarm robotics" [17]. In 1989, [18] coined the term "swarm intelligence" in order to describe the complex group behavior that emerges from simple individual cellular robots. While technological limitations made robotic applications in this field difficult to pursue at this time outside of small numbers of agents and simulation, biologists were quick to note the similarities between the study of social insect behavior and the goals of this emerging field [2]. Brook's controller design [14] provided a straightforward way for biologists to encode and organize behaviors they observed in the lab into robotic systems. In [2] it is described that when the term "swarm intelligence" was introduced, it was so heavily used by biologists that the term ended up "losing much of its original robotics context". This necessitated the invention of the term "swarm robotics" to differentiate between the natural and artificial research dealing with swarms.

Over the course of the next decade the primary area of application of research in swarm intelligence was centered around being able to model biological systems and determining how their interactions led to complex insect swarm behavior. This further tied the field to that of biological systems. In [3] it is described how the simple interactions of individual organisms results in societies that are robust, scalable, and flexible. Sahin uses these terms to describe the primary motivations of swarm robotics [2]. Any group of organisms that must interact together to achieve a common goal displays many of the characteristics swarm robotics seeks to achieve artificially. Throughout the early 1990s the field of swarm robotics systems exploded with approaches and ideas. In their review of cooperative robotics, Cao notes that "over the past 8 years (1987-1995) alone, well over 200 papers have been published in this field" [19]. However, Cao also notes that the field lacked a formal specification or corresponding means of discussion and that this was one of the areas that could be most improved. This caused Cao to note that the field remained very conceptual, with typical research involving new behaviors for the agents to perform. This provided an opportunity for biologists to have their behavioral observations on organisms in their research translated into artificial equivalents. This conceptual nature of swarm robotics research has produced an emphasis on producing desired individual behaviors rather than emergent group behavior that efficiently accomplishes the goal application.

This problem of a lack of top level design methodologies and general field formalization began to be addressed in the 1990s and early 2000s, visualized in Figure 2. In the early 90s, both [20] and [10] laid the groundwork for understanding how to design swarm behaviors, using Brook's method [14] to engineer specific interactions to occur among the individuals and create desired group behavior. This represented one of the first attempts to encode a general process for creating emergent behaviors. This was elaborated on by [21], recognizing the need for swarm behavior to be consistent and its performance to be provable from its design. In 2004, Dorigo and Sahin established a set of criteria to be "used as yardsticks for measuring the degree to which the term 'swarm robotics' applies" and intended to define the boundary between the more general multi-agent research, and that of swarm robotics [17]. In summary, these criteria indicate that swarm robotics research should involve the coordination of large groups (or small groups with purported scalability) of relatively homogeneous agents with limited sensory and communication capabilities [17]. While these established a means of identifying what constituted as swarm robotics research, refined methodologies for accomplishing these research programs remained undefined. In

2005, [22] formally defined the concept of "swarm engineering," previously introduced by [21], and identified the key processes needed to engineer dependable swarm behavior. Namely, they called for the study of mathematical modeling of swarms, design methods that produce reliable emergent behavior, as well as methodologies for testing and validation. A recent survey of swarm robotics [23] indicates that the field has continued to develop its design, analysis, and modeling techniques whereas "requirements analysis, maintenance, and performance measurement have received almost no attention". A substantial cause of these components being ignored is the lack of ongoing real-world swarm robotics projects. This delays the need for a system of requirements analysis to be developed, because swarms so far have almost always been deployed in simulation, or heavily controlled environments, allowing researchers to define project requirements to the needs of their agents, rather than the other way around.

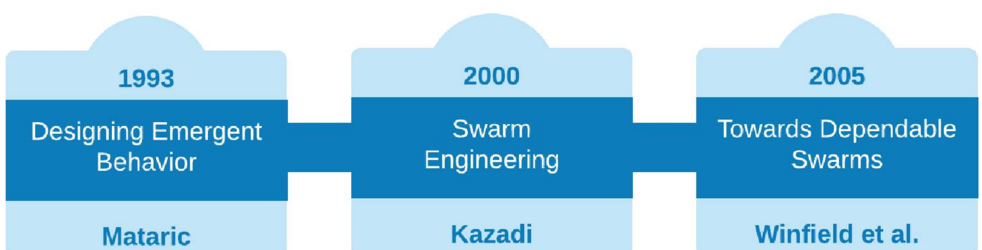


Figure 2: *The Formalization of Swarm Robotics*

Swarm robotics remains a relatively new field with many avenues of research needing to be explored before it may see widespread application.

In [23] it is plainly stated that “swarm robotics systems have never been used to tackle a real-world application and are still confined to the world of academic research”. The primary reason for this is that the body of research has not yet solidified, there are no consistent standards for the design of a swarm, nor are there clear “best” algorithms for any of its domains of applicability. One barrier to research in this field is that swarm robotics continues to have a very high cost of entry for new researchers. While the cost of electronic components have decreased since the fields inception, the financial and time cost, as well as expertise required, of acquiring, building, coding, and maintaining a swarm of physical robots is still large. This and the error prone nature of today’s hardware has led to swarm robotics research to primarily be conducted in either simulation, or on smaller physical swarms in controlled environments [23].

2.2 Central Place Foraging

The field of swarm robotics and its benchmark problem domain share many motivations as well as problems with lack of formalization. Swarm robotics is still in the process of formalization, and is heavily inspired by biological systems. The design of swarm behavior often relies on high level group behavior emerging from local individual interactions guided by simple controllers. These factors help to explain the prominence of the study of foraging within this field. The emphasis of biological inspiration in swarm robotics resulted in foraging being the testbed of swarm robotics applications. Foraging behavior represents a clear combination of many swarm behaviors which make it ideal as a goal for swarm systems.

The goal of central place foraging agents is to find and collect desired resources within an unexplored environment and return them to a singular collection location. Brambilla et al. and Winfield both describe foraging as the most prominent application and benchmark for swarm robotics [5, 23]. One of the primary areas in which swarm robotics has been thought to be applicable is that of space exploration, with first mention of this application occurring in 1989 [24]. The National Aeronautics and Space Administration (NASA) is particularly interested in using swarms of rovers to explore other planets such as Mars in search of ice [25]. If enough ice is gathered into a central processing area it can be separated into oxygen and hydrogen to be used for fuel. This objective has been translated into a national competition that has run between 2016-2018, eliciting foraging approaches from university teams [25].

One of the issues with comparing performance between central place foraging approaches is the lack of standard means of distributing resources throughout the environment. This is not a trivial problem as the use of different distribution types can have a large impact on the performance of any particular algorithm [6, 9]. Some algorithms are tailored to specific distribution types, while others are intended to be more generalist. This means that testing can be easily imbalanced depending on the types of resource distributions tested upon. In addition to the number of resources distributed throughout the environment, the arrangement of them may also follow various patterns. Resources may be distributed using uniformly random methods, exist in clusters, or follow formulaic patterns similar to that of minerals in a vein. The availability of resources all over the environment that occurs with uniform distributions reduces the effect that the spacial separation of agents has on distribution of work [6]. However, natural resources rarely occur in uniform distributions, often occurring in clumps or veins, as noted in [6]. Furthermore, recruitment behaviors in nature often only occur with the condition of highly concentrated resources. When resources are grouped together, it raises the amount of work that needs to be done in that area over time. This typically raises the spatial density and therefore the characteristic interference that occurs in that area of the environment. This is supported by [6] which notes that "when a cluster of targets is encountered collisions between robots increase near the cluster". In addition to distribution, the quantity of resources in the environment can impact the applicability of approaches.

Typically central place foraging environments feature exhaustible resources, making their availability scarcer as the foraging operation goes on, with this effect compounded by increases in the size of the swarm. In [9] and [6] the growing scarcity of resources is noted as one of the two primary obstacles to scalability in swarm foraging systems. It is understandable then, why studies seeking to focus on the long term emergent behaviors of swarms sometimes remove this constraint on the environment, utilizing infinite regrowing resources [12, 26]. Liu et al. use renewable resources for their approach which attempts to maximize the energy intake of the system, modeling resources as energy input and agent movement as draining energy, by optimizing the number of active foragers within the environment [12]. Pini et al. similarly uses infinite resources to explore the benefit of a task partitioning strategy, attempting to optimize the amount of work spent on a resource each time it is acquired [26]. Pini et al's environment utilizes instant regrowth upon resource pickup, which makes the scarcity of resources static throughout the foraging experiment, while Liu et al's resources grow back over time, which makes resource scarcity dynamic. Both [26] and [12] limit the number of concurrently available resources in the environment to the order of tens or hundreds, but regrowth allows thousands to be collected over the course of the experiment.

Experiments can also involve long term collection without regrowth, simply having a large environment with thousands of exhaustible resources [9]. Such a configuration makes resource scarcity become a serious problem, but this can be a desired consequence of the system to be studied; especially when approaches use recruitment mechanics or return to resource pickup location. Both of these cause agents to travel to sites in which resources are found repeatedly, which would make regrowing resources highly exploitable. The system would probably quickly converge to agents gathering from the same sites again and again, giving an inaccurate representation of the natural systems intended to be replicated. It is therefore inadvisable for systems utilizing memory or recruitment mechanisms to base their performance evaluation on environments with infinite resources, as this would produce unbalanced performance benefits that don't necessarily reflect the true performance of the system on a typical central place foraging formulation. However, this does not mean to say that the combination of purposeful repeated gathering mechanisms and infinite resource sites do not warrant further study. These formulations could provide useful insights into the trade-off between exploitation of known resource sites and exploration for new, better situated sites. Furthermore interference provides a limit on the number of agents that could successfully exploit a singular resource site at the same time, possibly necessitating the need to diversify individual agents to less optimal resource locations, to achieve optimal group performance. These represent some avenues of research to be studied with formulations incorporating infinite resources.

This section has presented different ways of formulating the problem of central place foraging and various associated behaviors that attempt to solve it. The variety of formulations exist as a result of a lack of standard definitions as well as the ability for reduction in the scope of the problem complexity and specialization to the types of interactions intended on being studied. Variations in the problem formulation that modify the environment can produce changes in the emergent swarm behavior as well. The behavior of swarms emerges not only from the behaviors of agents and the interactions between them, but also from agent interactions with the environment. Due to this characteristic, the lack of a standard problem formulation introduces benefits and issues in the study of central place foraging. The primary benefit of central place foraging lacking a strict definition or stringent requirements is the domain is highly adaptable to the needs and interests of the researcher. This has led to a wide variety of hardware systems being applied to central place foraging, as well as a corresponding array of swarm capabilities being explored. However, the same adaptability that benefits individual researchers creates some issues for the study of central place foraging as a whole, particularly in the area of evaluation. It is generally difficult to compare approaches that use different formulations, and applying an approach to a different formulation than originally intended produces different results. This is an open problem for the field of swarm robotics and central place foraging, resulting in difficulty of systematic comparison and evaluation, which extends beyond the scope of the present work.

2.3 Primary Foraging Behaviors

Behavior based swarm design is centered around the development of a layering of behaviors the agent transitions between in order to take it from its initial state to its ending goal state, this usually involves the use of a finite state machine. The behavioral design process typically begins with the creation of the lowest level of behaviors, which usually consist of simple movements and reactions to sensory data, such as turning when encountering an obstacle. High level behaviors incorporate several low level ones in order to accomplish more complex tasks [14]. In this way simple movement behaviors can be built into searching behaviors, which can be incorporated into controllers that can accomplish the overall foraging task.

Typically foraging behavior occurs in the following steps: search until a resource is found, collect it, and bring it to the collection depot [6]. The integration of search and collection within a single process causes knowledge of the environment to be gained gradually as agents collect resources. Agents must begin with dispersal from the place they are initialized in order to search for the resources. The two primary approaches to searching behavior are stochastic and deterministic techniques. Stochastic searches tend to move in a direction for some amount of time, and if nothing useful is perceived, a new random direction is chosen [9, 27]. These types of searches can be very effective at distributing agents all around the environment, reducing spatial density with decent area coverage. However, as resources are depleted from the environment, finding the remaining resources takes more and more time with the stochastic search, because they are likely to be farther out away from the depot, making it less likely for the agents to get there. The behavior of a stochastic algorithm, the Central Place Foraging Algorithm (CPFA) [9], is discussed at length in the next section. Deterministic approaches follow a pattern that is determined before the experiment begins, usually designed to cover the maximal amount of area as fast as possible. The search component of our approach is a deterministic pattern, and so is also described in detail in the next chapter.

When a resource is able to be perceived by the sensors of an agent, that resource has now been detected. Researchers have tended to try to simplify resource recognition as much as possible, to concentrate on the gathering aspects rather than that of image recognition and analysis. Accordingly, in controlled experiments resources are often simple objects that are easily recognizable. For approaches integrating search and collection, detecting a resource while the agent is not already carrying a resource typically means that the agent will then begin the process of picking up the resource. Due to the wide variety of swarm physical configurations, "picking up" can mean different things based on the type of robot and resource being worked with. This pick up work varies across different problem formulations; some require a gripper to physically grasp the object [26,28], and in other instances a picture of the resource is taken by the agent's camera and this is all that is necessary to represent having picked up the resource [9, 29]. Regardless of resource configuration, the resource ceases to be available within the environment, and becomes carried by the agent.

The amount of work required in order to pick up a resource can determine the applicability of certain foraging approaches. Several approaches attempt to distribute the movement of any particular resource between its initial location and its delivery at the depot between multiple agents. This mechanism, known as task partitioning, breaks the task of delivering a resource to the collection depot into several subtasks. Each resource is brought back some distance at a time before being abandoned by that agent and picked up later by another. This type of partitioning helps distribute the amount of work along the way, having a slight staggering delivery effect similar to what was observed in [6] for CPFA, as well as helping prevent error prone agents from holding on to a resource indefinitely away from the depot. Task partitioning is highly dependent on the resource formulation. High amounts of work needed to pick up a resource can increase inter-agent interactions around areas with high numbers of resources, such as clusters. As the cost of picking up a resource increases, such as when needing to use grippers, approaches that require the picking up of each resource multiple times may lose some or all of their performance benefits. Furthermore, with systems utilizing the image based resource, these task partitioning is difficult or impossible to emulate, due to these resources not actually physically moving.

When a resource is detected but an agent cannot pick it up due to currently being at capacity, the resulting state of that resource depends on the implementation of the foraging approach. Some approaches will completely ignore a resource that cannot be picked up [9], making it as if the detection had never occurred. Other approaches may try to notify other agents of the resources existence at that location [30, 31]. In an early work in swarm robotics recruitment, Sugawara and Sano use a light beacon system in order to have agents with an empty resource capacity attracted to those that have found a resource, who are holding still for a set duration before starting delivery [31]. The light beacon mechanism has also been used for attracting agents to the delivery location [26]. Sugawara and Sano find that their recruitment strategy makes the robots perform worse when resources are distributed uniformly, with the attraction increasing collisions and since the resources are distributed uniformly the position of one resource does not in any way indicate the position of another. However, when resources were placed in a singular cluster, the attraction mechanism was able to improve gathering efficiency. This corresponds to the results of both CPFA and real ant behavior, who only utilize recruitment strategies when resources are heavily clustered [9, 32].

Once the resource is in the agent's possession, the agent must return to the collection depot location for delivery. The amount of work that this step requires depends on the problem formulation as well as the agent's localization capabilities. For agents with error-free localization, agents are able to travel directly between the pickup site and the delivery location. When there is high noise in sensory data however, this step can be just as random a search as when looking for a resource. In some setups the depot has a distinctive light beacon visible from any area of the environment [26]. This reduces the complexity of finding the collection depot and limits reliance on localization. Upon finding the depot location, usually represented by something recognizable like a beacon or image, the agents drops off the resource and then resumes searching. One complication of homing behavior is that this is the one place in the environment which all agents must consistently go to in order to perform their task. This makes the interactions that occur near the collection depot one of the primary targets for interference mitigation techniques.

Site fidelity has the agent return to the last harvest location after delivering a resource in order to see if there exist additional resources near that location before resuming search. This behavior was observed in Desert Harvester Ants [33] before being incorporated into DDSA [6] and CPFA [9]. While the usage of memory mechanisms can have a large impact on task performance, its incorporation is not always a straightforward decision [32]. In large swarms, memory requires coordination schemes to prevent agents from going to resources they remember but have been collected by someone else. At the hardware level, localization within swarm systems often has a high degree of noise and error, making remembered resource locations possibly unreliable. The usage of memory based mechanisms is therefore recommended for systems with sufficient localization as well as communication capabilities. Systems not incorporating site fidelity resume search behavior upon delivering a resource.

The totality of these behaviors are what constitute an approach to central place foraging. At the core of every approach are search and collection. Aspects of these two high level behaviors are often accentuated with the types of mechanisms described in this section, such as recruitment, task partitioning, and site fidelity. The emergent group behavior that stems from the operation of these individual behaviors are impacted by the formulation of the problem in which they operate. In the next section, we study the two approaches most related to ours, CPFA [9] and DDSA [6], in detail.

2.4 Related Approaches

Foraging occurs frequently within nature, leading to many aspects of solutions in this problem domain to be biologically inspired. Organisms provide a glimpse into how artificial swarm systems could operate under varied natural conditions. Ant gathering behavior has had much attention from a combination of biologists and swarm roboticists. Ants are able to accomplish complex foraging tasks through a variety of simple but efficient mechanisms. They are able to estimate their relative location to the ant hill using the distance and direction they have traveled [33]. Ants can also communicate resource rich locations through the use of pheromones. Ants initially travel in a random pattern until a resource or pheromone trail is found, leaving a weak trail of their own pheromones behind them. Once a resource is found, ants are able to travel between that resource and the nest repeatedly in a much more direct path than when initially searching, a mechanism known as site fidelity within swarm robotics literature [34]. This causes the pheromone trail to be strengthened as long as there are remaining resources in that area to be collected. The areas with the most dense resources thereby have the strongest pheromone trails, attracting additional ants to help gather there [32]. These ant behaviors provide the inspiration for the Central Place Foraging Algorithm [9].

The Central Place Foraging Algorithm (CPFA) is directly modeled off of Desert Harvester ant behavior, incorporating stochastic search, site fidelity, and pheromone based recruitment behaviors [9]. CPFA agents default to a randomized search when no pheromone trail is found to emulate ants' initial search behavior. Once a resource is found, agents lay pheromones with a frequency and decay rate tuned by a genetic algorithm. In combination with a site fidelity mechanic, virtual pheromone trails leading to heavily clustered resources attract other agents. In the paper establishing CPFA [9], collisions are not modeled in the simulation, allowing swarm sizes up to 700 agents to continue to improve overall group performance. The choice for collisions to be ignored in research seeking to closely emulate ant behavior makes sense, as ants are very small and can climb over each other with little inconvenience. Furthermore, eliminating collisions uncovers emergent behavior typically obfuscated by the powerful effect of interference. The authors note that "other researchers have focused on inter-agent interference as the main cause of sub-linear scaling (in agent performance), but we observe sub-linear scaling even without including collisions in the simulation" [9]. The sub-linear scaling to which the authors are referring to is that adding agents in central place foraging solutions does not produce a corresponding linear increase in performance in relation to the size of the swarm. This is typically primarily thought to be almost entirely due to the effect of interference, but with that eliminated, performance in relation to size of the swarm remains sub-linear. The authors hypothesize that this is due to the increasing distance of the diminishing resources, as close resources are more likely to be stumbled upon by the randomly searching agents [9].

In a problem domain in which the entire point is to gather resources, it is counterintuitive that other agents collecting, interference aside, would be an issue for the individual agent. However, CPFA's results show decreasing usage of recruitment behavior in proportion to the size of the swarm [9]. The authors note that the parameters being evolved by their genetic algorithm has "the probability of laying pheromones decrease" and faster rates of pheromone waypoint decay in larger size swarms [9]. The reduction in pheromone laying rates and increase in rates of pheromone decay in larger size swarms indicates that the recruitment of helper agents in large swarms might do more harm than good. Small swarms may benefit from helper recruitment in order to speed up the gathering of a highly concentrated resource, and the lower total amount of agents in the system prevents more helpers discovering that trail than are actually helpful. If one looks at a single cluster of resources as its own central place foraging subproblem, there is likely a small critical number of agents that would be useful to the task, as the optimal number decreases with number of resources. However in a large swarm, no such prevention occurs, causing resource competition rather than cooperation. When an agent is recruited to a resource that is no longer there, this is an occurrence of "overshoot". This is a problem that occurs with natural recruitment behavior as well, leading to ants rarely using pheromone trails and instead tend to rely far more often on individual memory [32]. These aspects of emergent swarm behavior can be difficult to see with collisions enabled, but collisions are an important issue in foraging that cannot be ignored when comparing approaches.

A great deal of effort has been put into studying the characteristics and mitigation of inter-robot interference within multi-agent systems. In [28] interference is said to follow as a direct result of embodied agents operating within a shared environment. The authors identify interference as “the key stumbling block of efficient group interactions” as well as establishes its close relationship with spacial density [28,35]. Spacial density is the ratio of agents to a set space over a given time. Spatial density can be increased in a variety of ways including adding additional agents to the system or constraining the space the agents must operate in. The relationship of these factors, spacial density, and interference is well documented by the results of approaches to swarm behaviors [6, 11, 35]. In [11] the collection depot is identified as the place where spatial density is the highest and therefore where interference is the most pronounced. The authors describe this problem as so significant for their approach that despite all of the resources being found and picked up by the agents, they were not always able to be delivered back to the collection depot within the time limit due to the extreme congestion occurring at that location. In [6] it is explicitly stated that “crowding at the collection point is the main driver for degradation of performance”, and this sentiment is echoed by the authors of the Dual Agent Algorithm [36].

In [9] it is noted that while in simulation without collision, the approach was able to scale up to 700 agents efficiently, this might not be the case when utilizing physical robots. This data however suggests that if the cost of collisions was negligible, then that many central place foraging algorithms would have much more scalability, but the effect of collision avoidance when using mobile robots is actually quite significant. In [9], up to 6 physical agents were tested in a controlled environment that was essentially a scaled down version of the simulation in number of agents, environment size, and number of contained resources. From doing physical testing, it was apparent to the authors that the simulation "increasingly overestimates swarm efficiency as swarm size decreases" as "a result of the inter-robot interference in the real world that is not captured in the simulation" [9]. In order to measure the performance of the approach as it might work on physical agents, collisions needed to be taken into account. The results of CPFA when incorporating collision detection were provided for comparison to a later approach from the same lab [6], which provide a more reliable means of comparison.

The Distributed Deterministic Spiral Search Algorithm, or DDSA, aims to eliminate the randomness inherent to stochastic approaches by having agents expand outward from the collection depot in a formulaic spiral pattern [6]. DDSA agents bring resources back to the depot as soon as they are found, and then return to the pickup location in order to resume the search. This deterministic algorithm has the potential to outperform a stochastic algorithm given proper congestion mitigation techniques. DDSA is able to outperform CPFA up to about 15 agents due to having a much faster average search and collection time, due to collecting the closest targets first in a deterministic fashion. This trend continues with increases in the number of resources in the environment, with the authors noting that "for each additional target the time for DDSA collection increases by 10.67s compared to 23.4s per additional target with CPFA. [6] This showcases the drawbacks of a stochastic search, which is inefficient at finding of far away resources due to the low probability of an agent reaching any particular point. CPFA still manages to complete the task with linear scaleup, as the authors note that "In the CPFA the time to find uniform targets increases exponentially as the number of remaining targets decreases, however the time to complete collection scales linearly with the number of targets" [6]. As the number of agents in the environment continues to scale, DDSA's performance can no longer keep up with CPFA's, "due to crowding at the collection point" [6]. DDSAs benefit of having multiple agents retrieve the closest targets first and similarly distant resources thereafter is actually causing multiple agents to arrive at the depot at the same time. This causes the critical congestion around the collection location and results in an overall inability to scale to large numbers of agents. CPFAs stochastic nature seems to stagger deliveries and spatially isolate agents enough so that it continues to perform increasingly well up to at least 30 agents [6]. CPFA agents spread around the map randomly, resulting in both a significant increase in time taken to locate and retrieve a resource, but providing the benefit of making it unlikely to have many agents delivering resources at the same time. The authors of [6] suggest that the efficiency of CPFA is likely to also follow a parabola, eventually diminishing in performance due to the effect of interference, although this is not captured within the scope of their results [6].

3 Problem and Multimodal Approach

3.1 Problem Statement

As noted in the Introduction, central place foraging consists of agents finding desired resources within an unexplored environment and returning them to a singular depot location. However, there exists no standard formulation of the problem. The primary constraint differentiating central place foraging from other application areas is the existence of the singular depot location and the need to gather resources from the environment. This leaves the specifics of the problem, including the characteristics of the environment, resources, and the agents largely up to the researchers. Foraging is a complicated behavior that requires the incorporation of many mechanisms such as communication, localization, object manipulation, and obstacle avoidance in order to be performed successfully. Problems within any of these sub-behaviors can obfuscate the interactions the researcher is intending to observe. The plasticity of the problem specification allows the researcher to simplify aspects of these behaviors in order to better concentrate on the specific behaviors which are intended to be studied. This has influenced our problem specification in several aspects, so that interactions of the gathering behaviors can be clearly observed without the problems stemming from localization and object manipulation. These merited the study of a regularly shaped environment, the exclusion of static obstacles, the existence of minimal positional error, as well as resources which are transported virtually through image storage.

We assume that we need to plan paths to coordinate a group of N differential drive skid-steer robots to accomplish the objectives of central place foraging. The state of a skid-steer vehicle x can be represented by the triplet $(x_d, y_d, \theta) \in SE(2)$, where $(x_d, y_d) \in \mathbb{R}^2$ describe the position of the vehicle center of gravity and $\theta \in \mathbb{S}^1$ represents the orientation. Vehicle control input $u \in \mathbb{R}^2$ is modeled as $(v, \omega) \in \mathbb{R}^2$, where v is the forward velocity of the vehicle center of gravity and ω is the rate of change in vehicle orientation. Motion of the vehicle evolves according to the following kinematic equation:

$$\dot{x} = \begin{bmatrix} \dot{x_d} \\ \dot{y_d} \\ \dot{\theta} \end{bmatrix} = f(x,\ u) = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{bmatrix}. \tag{1}$$

For more details on skid-steering kinematics, the authors refer you to [37].

We then assume that a control algorithm exists that, in the absence of obstacles, will guarantee rover traversal to the goal waypoint [37]. This controller is modeled by:

$$u = g(x_c,\ x_g) \tag{2}$$

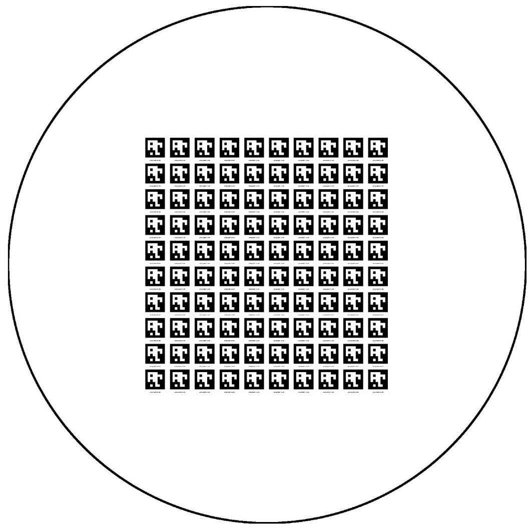
where $x_c \in SE(2)$ describes the rover's current pose, and $x_g \in SE(2)$ describes the rover's goal pose. In order to move from a current position to a goal location, the agent transitions between two motor stages; rotational and skid-steer. Upon reaching any given waypoint, the agent orients itself towards the general direction of the next goal location by only rotating. Once it falls within a tolerable theta threshold, the agent begins moving forward in a skid-steer stage that also allows it to course correct as necessary if it moves out of alignment with the goal. The combination of skid-steer and rotational steps allow the agent to correctly model the sharp corners necessary in a rectangular spiral search.

Our agents detect obstacles via three range sensors forming a cone in front of each agent. The middle range sensor is directly in front of the agent and the other two are situated 45 degrees away in either direction. This allows agents to determine if an obstacle is being detected in the front-left, front-center, or front-right, and react accordingly.

Resources are represented using the image based April Tag system [38], shown in Figure 3. These tags are able to be scanned by the agent's camera, similar to a barcode or QR tag. This means that the resources do not actually physically move, and in order to be "picked up", agents simply send a request to a central server with an image of the tag. This ensures that multiple agents do not try to deliver the same tag, a necessity as they do not actually get removed from the environment upon pickup or delivery. When a valid pickup request is received, the server stores that image as being in the sending agent's inventory, with a maximum capacity of one. This represents the agent carrying the resource as it attempts to find the depot location. The collection depot region is a disk one meter in length at the center of the environment, shown in Figure 4. The edges of the disk is with a unique tag only found at that location. In order to "deliver" the resource the agent must publish an image of this unique depot tag, confirming to the server that it has found the collection depot. At this point the server removes the saved image of the resource from the agent's inventory, making that resource delivered.

Figure 3: *Image Based Resource in simulation provided by the University of New Mexico [1]*

Figure 4: *Collection depot model in simulation provided by the University of New Mexico [1]*



3.2 Multimodal Approach

The implementations of central place foraging discussed previously are both modeled on ant behavior, but perhaps honeybees provide a better source of inspiration. Bees utilize a surprisingly complex system of search and communication in order to efficiently gather nectar [39]. In [40] a hive of bees is described as a system dynamically reacting to a complex and changing environment through a system of simple individuals leading to complex group behavior. This is accomplished through a system of scouts, employed workers, and onlookers, gathering and retrieving nectar information and reacting accordingly. Scout bees first survey the area around the hive, and upon route completion, they are able to communicate to other bees the concentration of nectar found by means of a waggle dance [40]. The collector bees then are allocated appropriately to gather nectar in the most concentrated areas [41, 42].

The Multimodal approach mixes aspects of DDSA and Honeybee behavior to efficiently forage and mitigate traffic. In this approach, a preliminary spiral search phase is separated from the collection behavior. The isolation of a preliminary search phase allows agents to search uninterrupted in a deterministic pattern designed to quickly search an area of interest. Agents use the locations gathered in the search phase to try and make better informed decisions about which resource to collect next. This approach uses the behavioral based design discussed previously in Primary Foraging Behaviors, in order to organize the foraging task of the agents. This means that layers of finite state machines transition between various low level behaviors in order to accomplish high level actions. While the approach is designed from the bottom up, starting from the most basic behaviors such as movement, we will describe the approach from the top down. At the highest level are the two modes that are the namesake of the Multimodal approach; search and collection. All agents begin in the search mode, and once they have traversed the area to which they have been assigned, they transition to collection until the end of the task.

Agents in the search mode follow a predetermined set of waypoints in order to detect as many resources within the agent's field of view as possible. The pattern that the agents follow is therefore encoded a priori, rather than in reaction to the environment. Similarly to DDSA, we utilize a spiral technique, as a spiral configuration is optimal for maximal coverage of the given area in the shortest time [7]. DDSA utilizes an interlocked spiral in which the agents each have a singular lane the width of an agent, in which they traverse. Multimodal differs in this aspect by having the agents do independent spiral searches of ring-like layers of the environment, as seen in Figures 5 and 6. For more details on the formulation of the searching pattern, including the recurrence relation that defines it, we refer to our previous publication [29].

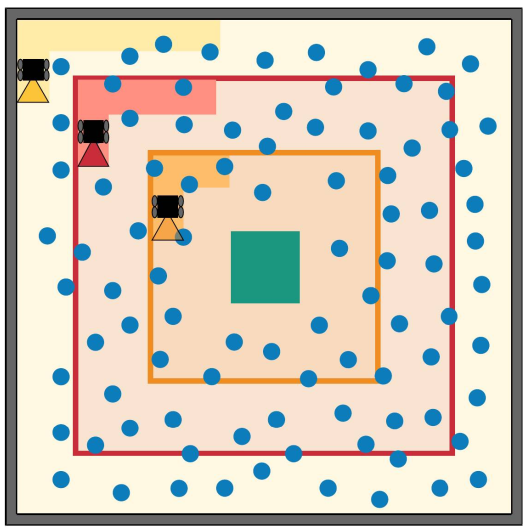


Figure 5: *Visualization of Search Behavior*

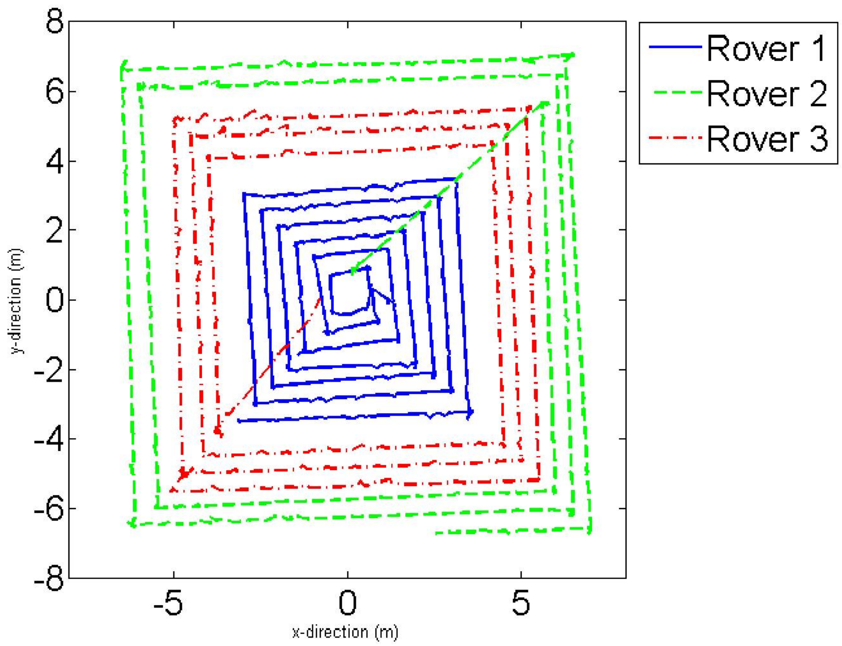


Figure 6: *Plot of agent locations for the first 350 seconds of Multimodal algorithm execution*

The objective of an agent in search mode is to follow the set of waypoints in order to detect as many resources throughout the environment as possible. Detection occurs as soon as a resource falls within the camera's field of view. Upon receipt of each camera frame, image processing provided by the April Tag library [38], occurs to locate all tags within frame, and returns a list of their identifiers and locations relative to the agent's camera. When localization errors are minimal, this combined with the agent's position gives an accurate location for that resource. This location, associated with the resource's identifier, is both stored in the agent's memory and is broadcasted to all other agents. This process is encoded into lines 1-3 of Algorithm 1.

Given that the search regions cover all of the environment and are of the same area, when the search is complete, the location of every resource in the environment should be in the memory of each of the agents. It is at this point that the agents switch to collection mode.

Algorithm 1 *Target Handler*: Determines what actions to take upon seeing target or depot tags.

```
for <target in targets_in_view> do
  if isUnknownTarget(target) then
    reportDetected(target);
  if role == COLLECTOR then
    if capacity == CARRYING then
      if isHomeTag(target) then
        isDepotSeen = true;
        dropOff(claimed);
    else
      if capacity == CLAIMED then
        if claimed! = target then
          unclaim(claimed);
      isTargetSeen = true;
      pickup(target);
```

When all of the resource locations are known and agents have the ability to travel directly to any location, the collection process for any resource is greatly simplified. The problem now switches from that of search, to something more akin to a vehicle routing problem, typically in which agents deliver "packages" to several known customers throughout the environment. The difference between this class of problems is that agents doing informed collection must return to the collection depot after reaching each resource site. Agents travel from the depot, where they just finished dropping off a resource, directly to the next resource site, and then return. This movement pattern of a straight path from the depot to resource and back again, shown in Figure 7, represents the minimal amount of work required to collect and is therefore ideal. At the individual task level, the resource that will be collected the fastest is the one that is currently closest to the agent. However the process of optimally selecting the next resource to collect is not as straightforward as it might seem. When resources are selected that are near each other, this can set two agents on a direct collision course (Figure 8). The cost of these collisions often outweigh the benefits of collect the immediately closest resource, making routing agents on non-intersecting paths more beneficial than putting them on individually optimal routes.

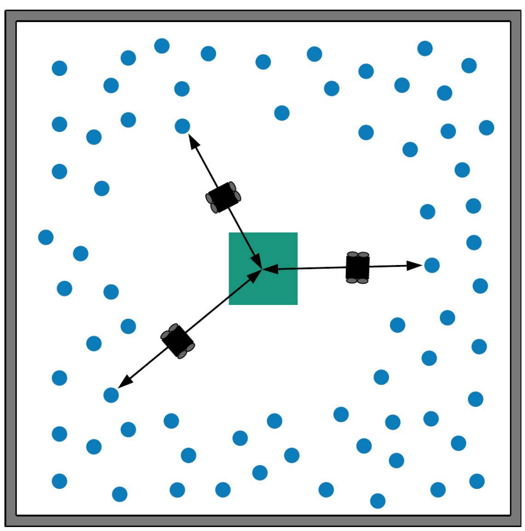


Figure 7: *Ideal Collector Movement*

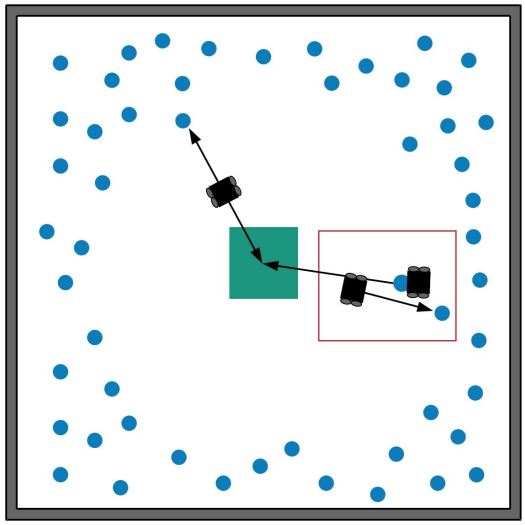
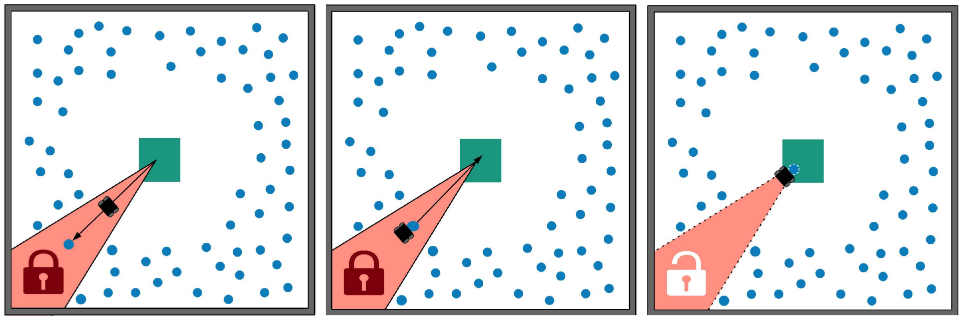


Figure 8: *Traffic Congestion from Target Selection*

In order to prevent this problem, agents utilize a sector locking algorithm visualized in Figure 9. The sector locking algorithm divides the environment into fixed angular sectors similar to slices of a pie. When the agent selects the next resource to collect, it is required to choose from resource locations in unlocked sectors. Once a resource is chosen, the sector it resides in becomes locked (Figure 9a). The lock continues when the agent picks up the resource (Figure 9b) and is only unlocked once the agent delivers the resource back to the depot location (Figure 9c). This algorithm ensures that a sector pathway between the collection depot and a resource is clear, allowing a singular agent to make an uninterrupted collection in that sector (Figure 7). Keeping these pathways free of traffic while allowing collection to occur in multiple sectors at once eliminates a portion of the interference problem, that which occurs between resources and the collection depot.



(a) *Initial Lock* **(b)** *Lock Continues* **(c)** *Unlock*

Figure 9: *Sector Locking Algorithm*

This sector locking algorithm is encoded into various parts of the collection mode state machine, with a simplified version presented in lines 6-17 of Algorithm 2 and lines 4-14 of Algorithm 1. Collectors that need a target traverse their list of resource locations and find the closest one that is not within a currently locked sector. Once this resource is chosen, it becomes claimed, making the sector that it resides in become locked to all other agents. The agents travel in mostly straight paths between the resources and the depot, so there is little opportunity for agents to cross sectors. If any other resource that can be picked up on the way to the claimed resource is encountered, that resource is picked up instead and the claim on the old resource is dropped for the time being. When the collecting agent arrives at the resource location, it will pick it up if seen, otherwise performing a randomized search around that location. This is sometimes necessary due to the camera either not picking up the April Tag in the current frame, or issues with localization. This type of search may also be performed around the depot area at the delivery stage for the same reasons. After pickup, the agent returns towards the depot location until the April Tag indicating the collection depot is seen, upon which point that resource is considered delivered. This relies on lines 5-8 of Target Handling, in Algorithm 1. The collector then repeats this process, starting from target selection.

While the goal of the algorithm is to avoid situations in which agents can collide, encounters in which agents must avoid each other still occur, especially around the depot location. This makes both the detection and capability of handling obstacles a necessity. The authors of [43] describe obstacle avoidance as one of the necessary low level behaviors of any multi-agent control system, and showed a significant performance enhancement just by rotating on detection of an obstacle . Upon any of these detections, the agent stores its current goal location onto a stack, and deviates towards an alternative location away from the obstacle detection. Once obstacles are no longer detected, the agent retrieves the old goal location from the bottom of the obstacle waypoint stack, and resumes its interrupted search or collection behavior. This behavior is visualized in lines 1-5 and 18-21 in Algorithm 2.

Algorithm 2 *Collector State Machine*: Determines which goal and state to transition to.

```
if obstacle_encountered == true then
    goal = getAlternativeLocation();
    previous_state = current_state;
    obstacle_encountered = false;
    current_state = OBSTACLE;
if current_state == GO_TO_TARGET then
    if isTargetSeen then
        goal = getDepotLocation();
        current_state = GO_TO_DEPOT;
    else if isGoalReached() then
        goal = getRandomNearbyGoal();
else if current_state == GO_TO_DEPOT then
    if isDepotSeen() then
        goal = getNextTargetGoal();
        current_state = GO_TO_TARGET;
    else if isGoalReached() then
        goal = getRandomNearbyGoal();
else if current_state == OBSTACLE then
    if isGoalReached() then
        goal = getCurrentLocation();
        current_state = previous_state;
goTowardsGoal();
```

In summary, the Multimodal approach is divided into two modes: search and collection. Agents begin in the search phase and attempt to detect as many resources as possible, recording their locations in what is essentially a shared memory. Agents follow a deterministic set of spiral waypoints during this phase and when these have all been traversed, agents switch to the collection mode. In the second mode, agents use the resource locations in order to more effectively route themselves. They utilize a sector locking technique to claim a clear pathway between depot and resources, reducing the effect of interference. The following section seeks to establish how we measured the performance of our approach.

4 Simulation Experiments

The previous chapters in this thesis have established the issues of systematic evaluation of swarm approaches stemming from a lack of problem standardization and heterogeneity of swarm composition from research group to research group. This makes the selection of an approach for comparison difficult, as one must find an approach with similar agent capabilities and developed for a similar environment. Furthermore the specifics of algorithm details are not always established or publicly available. DDSA was created specifically to address this problem, seeking to "establish itself as a baseline of comparison for other central place foraging algorithms" [6]. DDSA has several properties that make it a viable baseline for our Multimodal approach. First, it is simple to define, with its agents executing an interlocked spiral search and collecting resources immediately upon collection. This is similar to our approach if search and collection were integrated. Secondly, DDSA operates within a simple, well defined environment, already described in the problem statement. No changes are required in the capabilities of the agents when running either approach. These properties are ideal for showcasing any performance differences from the separation of search and collection, the intent of this thesis.

4.1 Data Gathered

In order to compare the performances of DDSA and our Multimodal algorithm, we ran foraging simulations using the ROS/Gazebo environment, a common robotics research platform [44]. The initial starting configurations for the environment in which our approach is applied is encapsulated by Figures 10 and 11. The environment in which the tests take place is a regular square with sides 15 meters in length, and the collection depot placed in the center of the square. At the edges of this square are walls outlining the area being observed, these and other agents are the only obstacles that agents can encounter. Agents are always initialized within 1 meter of the collection depot location. Surrounding the agents are always 256 resources with unknown locations, which must be returned to the collection depot location. There are two types of resource distributions on which our approach was tested; uniform and clustered. The uniform distribution, in Figure 10, is a common pattern for central place foraging environments, representing individual resources being randomly distributed evenly throughout the environment. In our uniform distribution, 256 of these individual resources are placed within the confines of the studied environment. A clustered distribution, in Figure 11, reflects a kind of resource concentration often found in nature, dense groups of resources surrounded by empty space. For tests utilizing the clustered distribution, 4 clusters of 64 resources are scattered randomly, making the total number of resources equal to that of the uniform tests. The upper time limit, 4500 seconds, was chosen based on the average amount of time taken for the Multimodal agents to complete collection, about an hour, plus 15 minutes to account for variations in performance. These tests were run on 25 variants of each distribution type using both DDSA and Multimodal approaches. Upon every execution of the primary state machine for either algorithm, occurring ten times a second, messages encoding portions of each agent's internal state were written to data files. These messages consisted of the current time since the beginning of the approach execution, the agent's current position, and the number of resources known to be collected by the swarm. The positions of the agents were used for overhead movement tracking throughout the execution, producing graphs similar to those seen in Figure 6. The collected resource count was used for performance tracking, used in Figures 12 and 13.

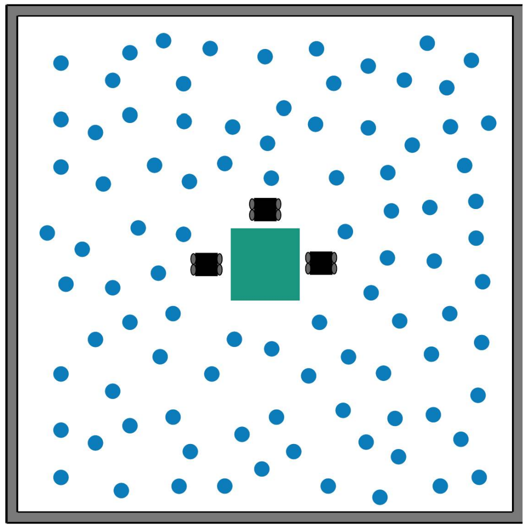


Figure 10: *Uniform Initial Configuration*

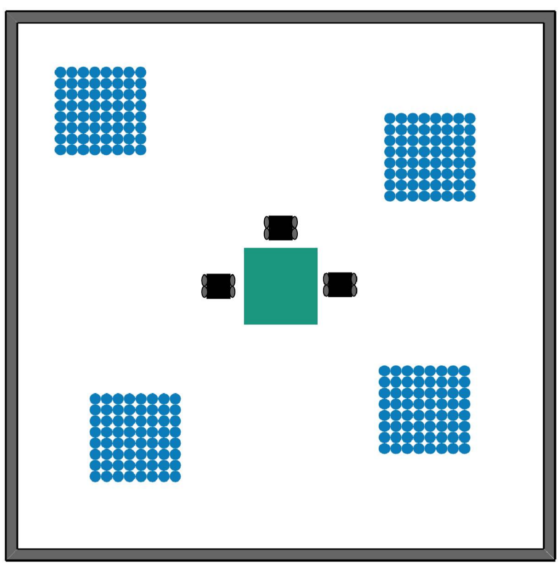


Figure 11: *Clustered Initial Configuration*

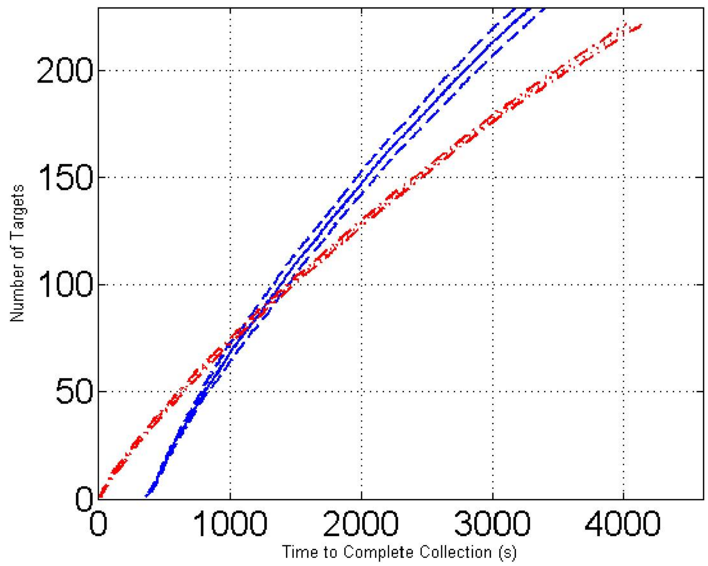


Figure 12: *Performance comparison of uniform distribution testing. (Multimodal in blue, DDSA in red)*

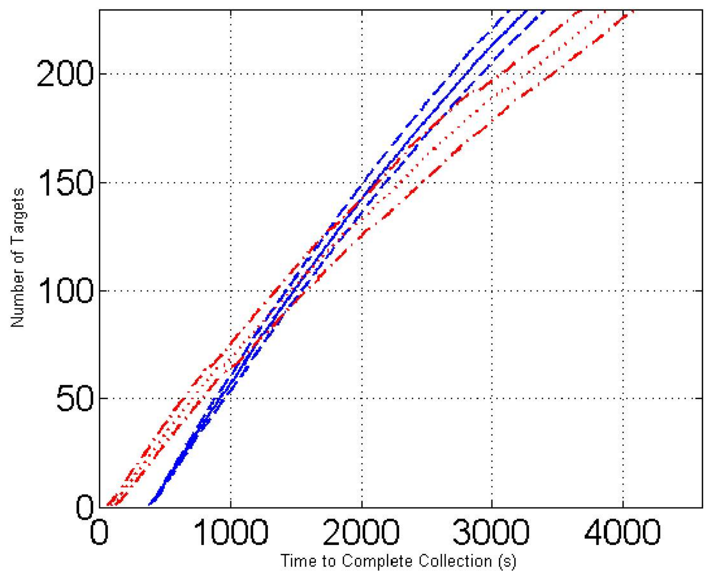


Figure 13: *Performance comparison of clustered distribution testing. (Multimodal in blue, DDSA in red)*

Figures 12 and 13 relay the overall performance of both algorithms throughout the entire task. The y-axis represents the time since the simulation began whereas the x-axis is the number of targets that have been delivered to the collection depot. At any time, having a point higher on the graph represents an increase in performance in comparison to a lower point. Figure 12 contains the average and 95% confidence intervals, as calculated in [45], for the data gathered on 25 different uniformly distributed trials, whereas Figure 13 contains the corresponding information for the same number of clustered trials. In order to measure performance we chose to count the number of resources collected over time, a common metric within this problem domain. Portions of these values at the 15, 30, 45, and 60 minute marks are encoded into Table 1.

		% Collected			
		15 minutes	30 minutes	45 minutes	60 minutes
Uniform	Multimodal	23.1 ± 0.9	51.8 ± 1.9	76.0 ± 2.0	96.3 ± 2.1
	DDSA	26.1 ± 0.6	45.8 ± 0.6	63.7 ± 0.6	78.9 ± 0.7
Clustered	Multimodal	19.0 ± 1.1	49.6 ± 2.3	75.7 ± 2.9	95.2 ± 2.3
	DDSA	25.5 ± 2.4	48.5 ± 3.7	67.4 ± 3.7	84.0 ± 3.6

Table 1: *Percentage of Total Targets Collected after 15, 30, 45, and 60 minutes for both Clustered and Uniform target distributions. Each entry shown with corresponding* 95% *confidence interval.*

4.2 Analysis

One area of investigation for this thesis is the effect of resource distribution on approach performance. The results of [9] and [6] indicate that interference plays a key role in such performance differences. In [6], the follow up results on CPFA note that turning on collision in the simulation results in a reversal of the distribution type CPFA performs best on, from clustered to uniform. The authors state that this is due to the advantage of the recruitment behavior being offset by "the initial time to discover a resource and the increase in collisions at clusters" [6,9]. We believe this is the cause of DDSA's variability in performance on clustered distributions compared to uniform performance shown in Figure 13. Our sector locking algorithm specifically addresses these problems by attempting to limit cluster gathering to one agent by locking the sector of the environment that the cluster is contained in. This treatment of the cluster, one resource and one agent at a time, makes the performance of these distributions closer to that of uniform ones, as seen in Figure 15.

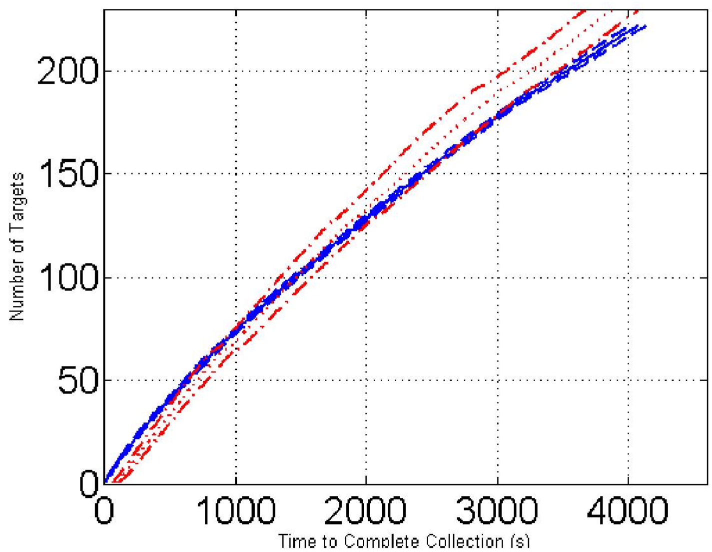


Figure 14: *DDSA performance on the two resource distributions. (Uniform in blue, Clustered in red)*

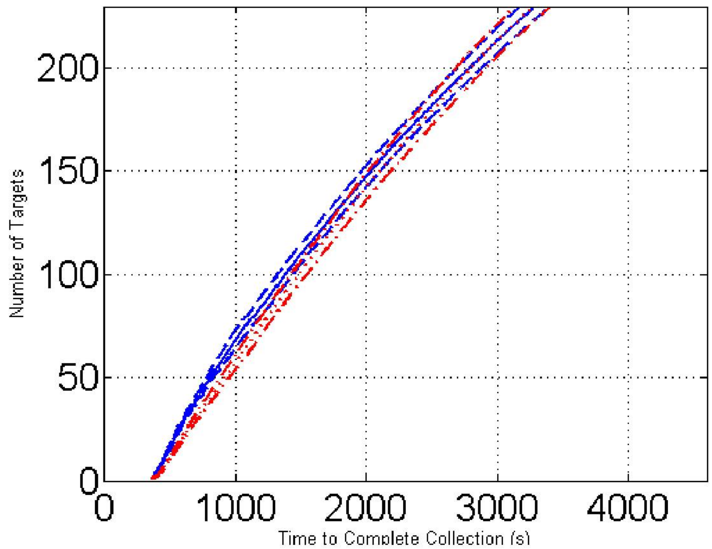


Figure 15: *Multimodal performance on the two resource distributions. (Uniform in blue, Clustered in red)*

One side effect of using the image based resources, means that they are not disturbed during the search phase, allowing agents to drive directly over them without movement. This makes the initial search cost unaffected by resource distribution, as indicated by the shared point in which agents begin collection in Figure 15. This would not be the case if resources were able to be physically manipulated by the agents. This resource type would need to be traversed, otherwise possibly invalidating their recorded location. This traversal cost would be much higher on uniform distributions given the much greater rate of encounter at any given point. This problem is avoided by DDSA due to their property of collecting the closest targets first due to their expanding spiral [6]. The Multimodal approach is therefore best used on types of resources that can be detected without being disturbed such as when search is possible from the air, or if this is not the case, on environments with low resource density, allowing the cost of traversal to be low.

As previously described, the Multimodal approach begins with a search phase in which the locations of resources are recorded but none are collected until the search phase is complete. This differs from the typical integrated approaches that collect and deliver resources as soon as any are found. The Multimodal approach therefore begins at a performance deficit; with DDSA collecting during the time Multimodal agents are searching. In order for the Multimodal approach to be viable, the benefits of informed collection must outweigh this initial debt. It is therefore necessary to examine the degree to which the Multimodal approach falls behind DDSA during the search phase, visualized in Figure 16. DDSA is able to deliver its first resource in an average of 10 seconds while Multimodal takes about 360 seconds to accomplish the same task. This occurs due to DDSA quickly collecting nearby targets first, whereas Multimodal agents are only searching and recording resource locations. While DDSA has collected 33 out of 256 resources before Multimodal agents collect 1, the Multimodal agents now have a shared knowledge of most, if not all, of the surrounding resources. From this point on, Multimodal agents are able to navigate directly to and from resource locations as well as avoid traffic using the sector locking algorithm. While the cost of search is essentially static for this problem formulation, it is by no means constant for any application. In addition to the effect of resource distribution and type discussed previously, characteristics of the environment and that of the swarm play key roles as well.

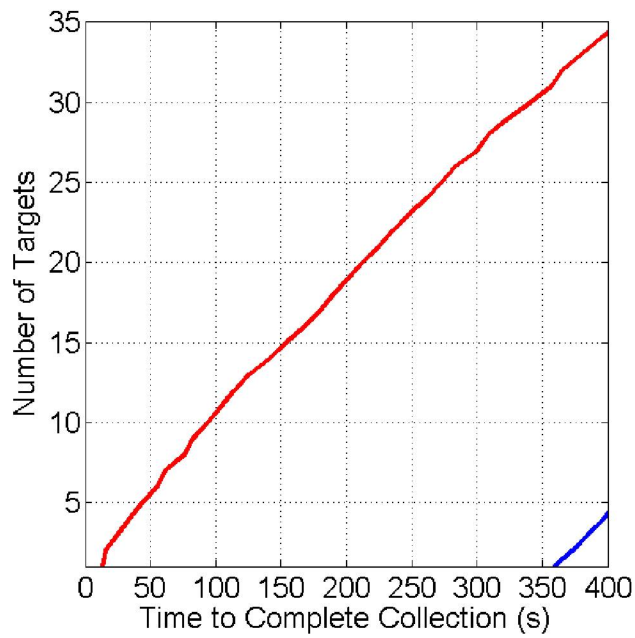


Figure 16: *Search cost indicated by performance during the first 400 seconds (Multimodal in blue, DDSA in red)*

At the most simplified level, the cost of the search phase is determined by the amount of new area all agents can detect per time unit divided by the size of the unknown area at the outset of the task. The size of the environment can require more or less traversal, whereas irregular shapes can require agents to cover the same ground repeatedly to detect the entire environment. The movement speed and detection capabilities of the agents play a role in this ratio as well. The faster an agent travels, the fewer frames the resource is within the agent's detection range, making there usually exist a speed at which resource detection is no longer reliable. This effect can be reduced by slowing down, but this increases the search time. Additional agents reduce the amount of area that each agent has to cover, but there exist limits to this as well. At a certain point adding additional agents does not increase the amount of new ground being covered. These combination of factors make search time only calculable given a very precise definition of the problem formulation, and even then is far more easily and accurately established through simulation and physical testing.

Once search is complete the location of most if not all of the resources are known to all of the agents. This allows for the collection of each resource to be a simple process of traveling to that location, picking it up, driving straight back to the collection depot, and dropping it off. The time to collect a resource, in the absence of interference, therefore scales linearly with the distance of that resource from depot. This is most evident from Figures 17 and 18. Here the times for Multimodal agents to collect each additional resource is shown to be typically lower and with less variability than those of DDSA. The variability in DDSA's next collection times is due to each resource needing to be located before it is collected. This is most striking on clustered distributions, in Figure 18, where the four highest points represent the large amount of time spent before locating the next cluster. However, not shown in these graphs is the average time taken for Multimodal agents to collectively deliver the first resource, which is about 360 seconds. This, shown in Figure 19 for clustered distributions, is far greater than any other measurement. DDSA is able to deliver its first resource in an average of 10 seconds while Multimodal agents take 360 seconds to accomplish the same task (Figure 16). Despite this starting deficit, Multimodal agents are able to catch up to and then surpass the performance of agents using DDSA.

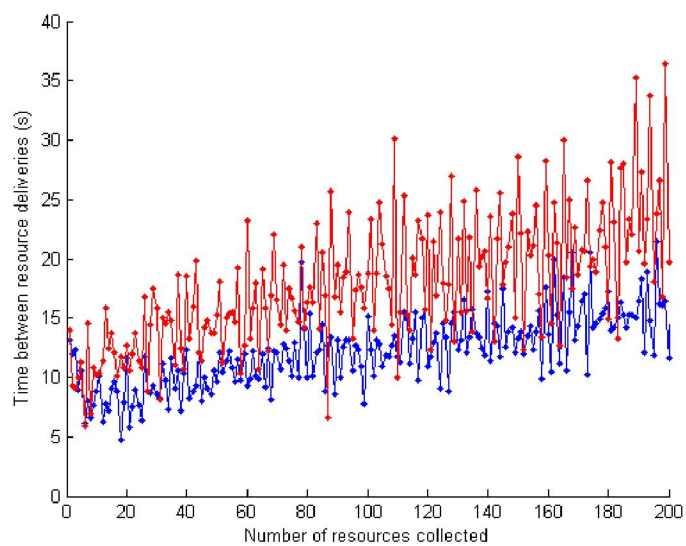


Figure 17: *Time between target collections on uniform distributions. (Multimodal in blue, DDSA in red)*

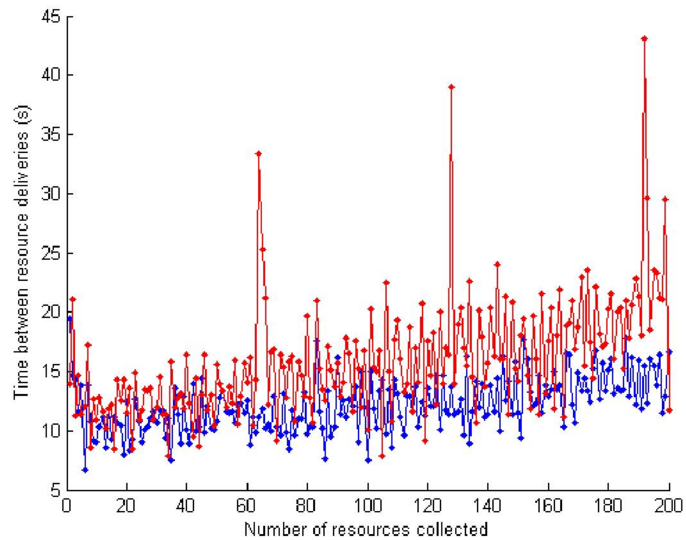


Figure 18: *Time between target collections on clustered distributions. (Multimodal in blue, DDSA in red)*

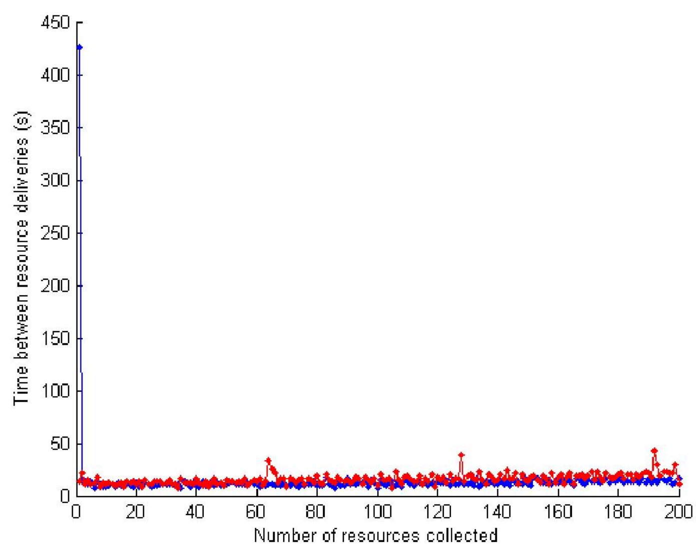


Figure 19: *Time between target collections including Multimodal search phase. (Multimodal in blue, DDSA in red)*

However both algorithms actually share the same base search and collection costs, distributed over different times. DDSA and the Multimodal approach both perform a complete spiral search of the environment meaning that DDSA is paying Multimodal's initial search cost, but over the course of the entire task. Due to DDSA using site fidelity, it travels the same collection distances as Multimodal agents; from the pickup location to the collection depot and then back. This means that in order to collect all of the resources, the total sum of the search and collection times, all other factors excluded, are exactly the same. This hypothesis is supported by the preliminary findings of using the Multimodal approach without sector locking and comparing it to DDSA, seen in Figure 20. Without the traffic mitigation aspect, Multimodal agents would always choose the closest remaining target from the depot. This is essentially the same behavior as DDSA, due to its expanding interlocked spiral causing the collection of closest targets to the collection depot first. As seen in the figure the roughly equal amounts of work distributed over different times causes the performance of the Closest Target First version of the Multimodal approach to converge with that of DDSA's in the long term. The cause of the full Multimodal approach outperforming DDSA is therefore through the selection of its targets, via the sector locking mechanism. This supports our primary hypothesis; that the separation of search and collection, while incurring an initial performance cost, can better inform the collection phase, resulting in overall performance gains.

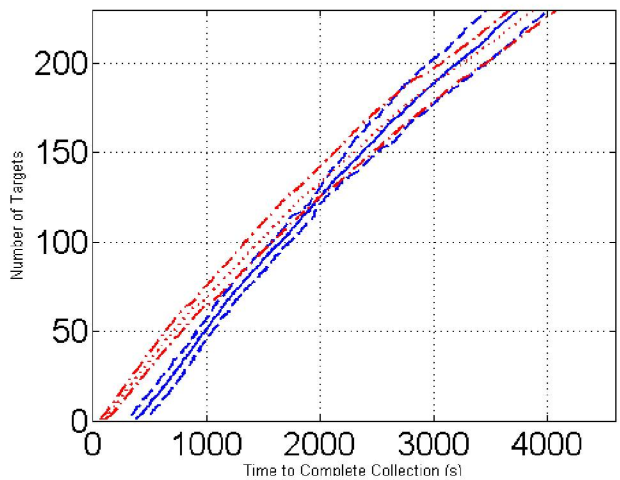


Figure 20: *Performance of Multimodal with no sector locking versus DDSA on 5 clustered environments (Multimodal in blue, DDSA in red)*

Once the search portion is complete, Multimodal agents are able to navigate directly to and from resource locations as well as avoid traffic using the sector locking algorithm. These logistical improvements allow Multimodal to catch up to DDSA in an average of 1250 seconds by collecting resources at a faster average rate. Once intercepting the performance of DDSA agents, Multimodal agents continue to widen the gap until the simulation's end (Figure 21). This is due to Multimodal continuing to collect resource at a near constant rate, while DDSA agents struggle to find the last remaining resources at the edges of the environment, as well as time lost due to traffic congestion. These results indicate that the long term performance benefit of a preliminary search phase has the potential to far outweigh the initial startup cost.

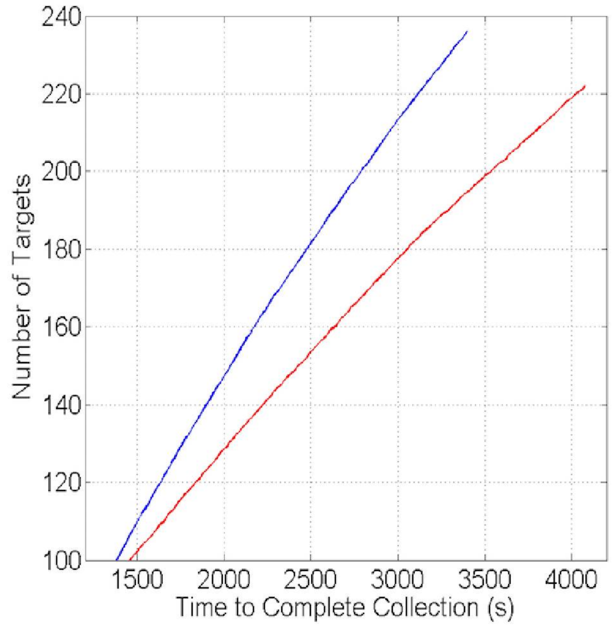


Figure 21: *Long Term Performance (Multimodal in blue, DDSA in red)*

5 Conclusions

The aim of this thesis was to investigate whether performance advantages could be gained in central place foraging from techniques that separate search and collection into two consecutive rather than concurrent tasks. Completing the search before collection begins implies a period in which no resources are being delivered, representing a large deficit in initial performance. However we hypothesized that through the usage of information gathered during the search phase, agents could better select targets during operation in collection mode that resulted in less interference, one of the primary subproblems within this domain. We therefore proposed the Multimodal approach that uses a sector locking algorithm in order to try and create clear paths between individual resources and the collection depot. In order to determine the viability of such an approach, we implemented both the Multimodal approach and a baseline algorithm, DDSA [6]. We next simulated agents utilizing these algorithms within two problem formulations used by other central place foraging algorithms, including DDSA. Both problem variants, uniform and clustered resource environments, had 25 different trials generated and tested on with both algorithms. Experiments consisted of agents executing their foraging algorithm until either all the resources were collected or a time limit of 4500 seconds was reached. Metrics were gathered from the internal states of the agents throughout each experiment, which form the basis of our results.

These results indicate that the Multimodal approach is viable for the application of central place foraging, at least within the specified problem formulations. The initial search performance debt was found to be surmountable by the benefits gained from more informed collection target selection. The Multimodal approach's performance catches up to and then surpasses that of DDSA's in both problem formulation before either is halfway finished collecting the 256 resources. The Multimodal approach represents a proof of concept that the separation of search and collection can lead to the incorporation of more advanced interference mitigation techniques that improve the overall performance of the foraging task.

The sector locking algorithm by no means claims to be the optimal target selection strategy. It is highly likely that there exist other selection mechanisms that outperform the current Multimodal approach. However in order to properly design and evaluate such new mechanisms, a more systematic understanding of the effect of interference is required. Despite interference being commonly referred to as one of the primary barriers to central place foraging scalability, no systematic discussion and analysis of the problem has yet been written, nor does there exist established and efficient techniques for mitigation. This is the main aspect in which primarily theoretical rather than applied work must be done in order to proceed in an effective fashion.

Larger than the development of additional mitigation techniques is the overall study of informed collection. There exist a wide variety of approaches that could be based upon preexisting knowledge of resource locations, independent of how they were found. In order to properly test and evaluate these approaches a variety of baseline approaches as well as standard problem formulations must be developed. The Closest Target First variation of the Multimodal algorithm represents one such appropriate baseline algorithm, and the problem formulations used by this thesis, based upon those used in DDSA and CPFA, are also appropriate starting points. However, as noted throughout this thesis, differences in environment, resources, agents, and swarms all produce changes in emergent system behavior. These differences should be carefully studied using the baselines and controlled for in the development of any standard problem formulations.

Finally, this thesis could be greatly improved by the inclusion of results of testing on physical robotics systems. This is currently lacking due to the high financial and time cost associated with building and maintaining a swarm of physical agents. It is highly worth investigating the minimal characteristics and capabilities necessary for an agent to work specifically only in this problem domain. It is possible that through this minimally viable rather than general purpose agent type and through formulation simplification and control of the operating environment that many of these costs could be mitigated and large physical swarms of foraging agents could be tested upon.

References

[1] UNM Swarmathon code repository, 2016. Retrieved on 06/22/2018 from https://github.com/BCLab-UNM/SwarmBaseCode-ROS.

[2] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer, 2004.

[3] Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Eric Bonabeau, and Guy Theraula. *Self-organization in biological systems*, volume 7. Princeton university press, 2003.

[4] Levent Bayndr. A review of swarm robotics tasks. *Neurocomputing*, 172:292 – 321, 2016.

[5] Alan F. T. Winfield. *Towards an Engineering Science of Robot Foraging*, pages 185–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] G Matthew Fricke, Joshua P Hecker, Antonio D Griego, Linh T Tran, and Melanie E Moses. A distributed deterministic spiral search algorithm for swarms. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4430–4436. IEEE, 2016.

[7] Scott Burlington and Gregory Dudek. Spiral search as an efficient mobile robotic search technique. In *Proceedings of the 16th National Conf. on AI, Orlando Fl*, 1999.

[8] Edward Tunstel, Gary Anderson, and Edmond Wilson. Motion trajectories for wide-area surveying with a rover-based distributed spectrometer. In *Automation Congress, 2006. WAC'06. World*, pages 1–8. IEEE, 2006.

[9] Joshua P. Hecker and Melanie E. Moses. Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1):43–70, 2015.

[10] Maja J Matarić. Designing and understanding adaptive group behavior. *Adaptive Behavior*, 4(1):51–80, 1995.

[11] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141, 2002.

[12] Wenguo Liu, Alan FT Winfield, Jin Sa, Jie Chen, and Lihua Dou. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive behavior*, 15(3):289–305, 2007.

[13] Dylan A Shell and Maja J Mataric. On foraging strategies for large-scale multi-robot systems. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2717–2723. IEEE, 2006.

[14] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, Mar 1986.

[15] Anita Flynn. Gnat robots (and how they will change robotics). 1987.

[16] G. Beni. The concept of cellular robotic system. In *Proceedings IEEE International Symposium on Intelligent Control 1988*, pages 57–62, Aug 1988.

[17] Marco Dorigo and Erol Şahin. Guest editorial. *Autonomous Robots*, 17(2-3):111–113, 2004.

[18] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In Paolo Dario, Giulio Sandini, and Patrick Aebischer, editors, *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[19] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng. Cooperative mobile robotics: antecedents and directions. In *Proceedings 1995*

IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, volume 1, pages 226–234 vol.1, Aug 1995.

[20] Maja J. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the Second International Conference on From Animals to Animats 2 : Simulation of Adaptive Behavior: Simulation of Adaptive Behavior*, pages 432–441, Cambridge, MA, USA, 1993. MIT Press.

[21] Sanza T Kazadi. *Swarm engineering.* PhD thesis, California Institute of Technology, 2000.

[22] Alan F. T. Winfield, Christopher J. Harper, and Julien Nembrini. *Towards Dependable Swarms and a New Discipline of Swarm Engineering*, pages 126–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[23] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[24] Rodney A. Brooks and Anita M. Flynn. Fast, cheap and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, 42:478–485, 1989.

[25] Learn More NASA Swarmathon, 2017. Retrieved on 10/19/2017 from http://nasaswarmathon.com/about.

[26] Giovanni Pini, Arne Brutschy, Carlo Pinciroli, Marco Dorigo, and Mauro Birattari. Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adaptive behavior*, 21(2):118–136, 2013.

[27] Alexandre R. Mesquita, João P. Hespanha, and Karl Åström. Optimotaxis: A stochastic multi-agent optimization procedure with point measurements. In Magnus Egerstedt and Bud Mishra, editors, *Hybrid Systems: Computation and Control*, pages 358–371, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[28] Miguel Schneider-Fontán and Maja J Mataric. A study of territoriality: The role of critical mass in adaptive task division. In *From animals to animats IV*. Citeseer, 1996.

[29] Nicholas Dolan-Stern, Kevin Scrivnor, and Jason Isaacs. Multimodal central place foraging. In *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pages 72–78. IEEE, 2018.

[30] Ronald C. Arkin, Tucker Balch, and Elizabeth Nitz. Communication of behavorial state in multi-agent retrieval tasks. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 588–594. IEEE, 1993.

[31] Ken Sugawara and Masaki Sano. Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D: Nonlinear Phenomena*, 100(3-4):343–354, 1997.

[32] Kenneth Letendre and Melanie E Moses. Synergy in ant foraging strategies: memory and communication alone and in combination. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 41–48. ACM, 2013.

[33] Sandra Wohlgemuth, Bernhard Ronacher, and Rüdiger Wehner. Ant odometry in the third dimension. *Nature*, 411(6839):795, 2001.

[34] TO Crist and JA MacMahon. Individual foraging components of harvester ants: movement patterns and seed patch fidelity. *Insectes Sociaux*, 38(4):379–396, 1991.

[35] Dani Goldberg and Maja J Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Aaai/iaai*, pages 637–642, 1997.

[36] Brian Geuther, Eric Johnsrud, and Yunfeng Wang. A dual agent algorithm for foraging. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 279–284. IEEE, 2012.

[37] Elie Maalouf, Maarouf Saad, and Hamadou Saliah. A higher level path tracking controller for a four-wheel differentially steered mobile robot. *Robotics and Autonomous Systems*, 54(1):23 – 33, 2006.

[38] April Tag Library, 2016. Retrieved on 06/22/2018 from https://april.eecs.umich.edu/software/apriltag.html.

[39] S Cazamine and J Sneyd. A model of collective nectar source selection by honey bees. *Journal of Theoretical Biology*, 149(4):547–571, 1991.

[40] Valery Tereshko and Andreas Loengarov. Collective decision making in honey-bee foraging dynamics. *Computing and Information Systems*, 9(3):1, 2005.

[41] Valery Tereshko and Troy Lee. How information-mapping patterns determine foraging behaviour of a honey bee colony. *Open Systems & Information Dynamics*, 9(2):181–193, Jun 2002.

[42] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[43] Maja J Mataric. Minimizing complexity in controlling a mobile robot population. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 830–835. IEEE, 1992.

[44] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source robot operating system. In *ICRA Workshop on open source software*, volume 3, page 5. Kobe, 2009.

[45] David S Moore. *The basic practice of statistics*. Palgrave Macmillan, 2010.