ISOLATING SUB-POPULATIONS TO EXPLOIT LOCALITY IN DISCOUNTED ROBOT FORAGING

by

Pante a Jabbari
B.Sc., Sharif University of Technology, 2008

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2011

APPROVAL

Name: Pante a Jabbari

Degree: Master of Science

Title of Thesis: Isolating sub-populations to exploit locality in discounted robot foraging

Examining Committee:

Dr. Brian Funt,
Professor, Computing Science
Simon Fraser University
Chair

Dr. Richard Vaughan,
Associate Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Alexandra Fedorova,
Assistant Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Anoop Sarkar,
Associate Professor, Computing Science
Simon Fraser University
SFU Examiner

Date Approved: _____

Abstract

We examine a canonical multi-robot foraging task, in which multiple objects must be located, collected and delivered. Each type of object must go to a unique delivery location. The value of each delivery is discounted over time. We describe a system in which a population of robots are effectively allocated to local (and thus high-reward-rate) foraging tasks, by keeping them ignorant of distant (thus poor-reward-rate) tasks. Robots learn about available tasks by local communication, with a fixed communication range that controls the rate at which task knowledge propagates. Our empirical data suggests that there is an optimal communication radius for our setting. Our system is effective at allocating robots to tasks, performing better than fully-informed robots. Interesting emergent group behaviour dynamics are described.

Keywords: Multi-Robot, Task Allocation, Discounted Reward, Foraging

To my parents.

Acknowledgments

I owe my deepest gratitude to my senior supervisor Richard Vaughan without whose guidance and support this work could not have been possible.

I also would like to thank all my colleagues in the Autonomy Lab who have helped and supported me during my studies.

Finally, I would like to thank my family and friends who have always been there for me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis we study a discounted-reward multi-robot foraging task in which multiple objects must be located, collected and delivered. We introduce an heuristic method for achieving a high reward, with no global or a priori knowledge about the tasks or robot locations.

1.1 Motivation and Goal

In our foraging task, we have different object types, where each type of object must be delivered to a unique delivery location. Since this is a discount-reward task, the value of each delivery is decreased over time so that early deliveries are much more valuable than later deliveries. The search for and delivery of the objects can be mapped to a job sequencing task: when the goal is to minimize the average wait time for the jobs and all of the jobs have the same weight, the optimal solution is to sequence the jobs based on increasing completion time [Smith, 1956]. For our settings, if the robots' speed is constant, the total gained reward will be maximized if robots collect the objects that are closer to the delivery point first, and go to the objects that are located further away later.

When the exact location of all objects and robots and the reward for deliveries is known, finding the optimal joint action plan is NP-hard. However, this information is usually not available in practice. This robotic problem is similar to provisioning in animals [Stephens et al., 2007].

In our thesis, we study the problem where the initial number, distribution, and delivery locations are unknown to the foraging robots.

Since the globally optimal solution is usually impossible or impractical to achieve, numerous studies have been performed on dynamic task allocation, where the system does not rely on global knowledge and instead seeks an approximation of the global optimal solution that is less costly to obtain, and preferably using only locally-available information. [Lerman et al., 2006] and [Gerkey and Mataric, 2004] each introduce mathematical and formal analysis tools for dynamic task allocation in multi robot systems.

There are also numerous dynamic methods suggested for specific tasks. Many of these methods are introduced for foraging tasks ([Lein and Vaughan, 2008], [Vaughan et al., 2001], and [Ulam and Balch, 2003]). This is the sample task that we selected for our implementation.

1.2 Contribution

Here we describe a method whereby a population of robots is dynamically allocated to local (and thus high-reward-rate) foraging tasks, concentrating on object types whose drop-off location is closer, and not being distracted by distant (thus poor-reward-rate) tasks. Robots learn about available tasks by local communication, with a fixed communication range. This communication range controls the rate at which task knowledge is propagated, and we show that by restricting this range we can achieve a higher reward in total compared to what is achieved if the communication range was not restricted.

Our data suggest that there is an optimal communication radius for our setting and that our system is indeed effective at allocating robots to the tasks. This system shows interesting group behavior dynamics that are described throughout the thesis.

As we will discuss in the next chapter, various studies have been done on foraging some of which study discounted reward scenarios. However, the effect of locality in discounted reward foraging has not been studied before. Also the "migration" behaviour that we will introduce in Chapter 5 has not been observed or discussed before.

1.3 Thesis Outline

The purpose of this thesis is to study the problem of task allocation in discounted-reward settings. We first review some of the previous work in this field, and then propose a method for achieving a high reward in a specific discounted-reward multi-robot foraging task, by

restricting the robots' communication range. The rest of this thesis is structured as follows:

- Chapter 2, **Study of Task Allocation** reviews the current status of research in multi-robot task allocation
- Chapter 3, **Simulation** presents the simulation settings and environment we used for our experiments
- Chapter 4, **Method** introduces our method for achieving a high reward when we have a discounted-reward situation
- Chapter 5, **Forgetting Method** describes an undesirable issue with the method and suggests a slight change to solve it and then presents our final experiment results
- Chapter 6, **Discussion** discusses the effects of our method, what it contributes to the world of robotics and the issues we are facing
- Chapter 7, **Conclusion and Future Work** concludes our work and reviews some of the possible future work

Chapter 2

Study of Task Allocation

2.1 Task Allocation in Robotics

Multi-robot systems allow robots to execute tasks in parallel. The Task Allocation problem can be specified as "Which robot should perform which task, and in what order" in order to achieve the desired goal. This problem can be described in terms of tasks and workers.

Typically robots can perform one task at a time. So the problem is often specified as, having n workers that each are looking to perform a task and m tasks, which each need one worker to be performed, assign the tasks to the workers (robots) [Gerkey and Mataric, 2003].

The solution to this is an allocation of tasks to robots, such that a specific objective function is maximized. This function can be anything like the minimum distance traveled by robots, the least power consumed by robots, the least time for all tasks to be delivered, the least time for the first tasks to be completed, etc.

In [Gerkey and Mataric, 2004], the authors present a formal analysis and a domain independent taxonomy for studying Multi Robot Task Allocation problems. They show how their taxonomy can be applied to analyze and classify MRTA problems and to evaluate the proposed solutions to them. They also present algorithms to solve some of the simpler classes.

They present their classification of the MRTA problems not based on the *architectures* of the systems, but on the *problem specifications*:

- Whether they have Single-task Robots (SR), capable of performing at most one task at a time, or Multi-task Robots (MR), capable of executing more tasks at a time.

- Whether they have Single-robot Tasks (ST), which need exactly one robot to be performed, or Multi-robot Tasks (MT), requiring more robots.
- Whether they have Instantaneous Assignment (IA), meaning the available information about the robots, tasks, and environment allows only for an instantaneous assignment, or they have Time-extended Assignment (TA), meaning the information allows for a planning for future allocations.

Based on these specifications 8 classes of MRTA problems are then introduced, using triples indicating each feature from the above categories:

- ST-SR-IA: This is the simplest problem among all classes. The authors show that it is an instance of the Optimal Assignment Problem and it can be solved optimally in $O(mn^2)$ time where m is the number of robots and n is the number of prioritized tasks. A common alternative approach is the family of "auction-based" heuristic methods, which run slower than the centralized approach but need fewer messages to reach the equilibrium.

  Iterated assignment: The authors also introduce this variant, which is an iterated instance of the ST-SR-IA and refer to the greedy algorithm "Broadcast of Local Eligibility" introduced in [Werger and Mataric, 2000], a 2-competitive for the OAP, as a solution. The Optimal Assignment algorithm takes $O(n^3)$ time to perform.

  Online assignment: In this other variant, the set of tasks is not revealed at once, but rather the tasks are revealed one at a time. Another greedy algorithm is introduced by the authors to solve this problem, which is 3-competitive to the optimal post hoc offline solution.

- ST-SR-TA: This problem needs a schedule of tasks for each robot to perform over the time. Since this is an instance of the class of scheduling problems, it is strongly NP-hard. They introduce an approximation algorithm for this problem, which treats it as an instance of ST-SR-IA followed by an instance of the online assignment problem. The performance of this algorithm is at least 3-competitive for online assignment.
- ST-MR-IA: This problem is cast to an instance of Set Partitioning Problem which is NP-hard and needs heuristic methods to be solved.

- ST-MR-TA: This problem includes both the previous problem, and the scheduling problem and so it is also NP-hard.
- MT-SR-IA: Solving this problem is equivalent to solving ST-MR-IA, with the role of robots and tasks switched.
- MT-SR-TA: Solving this problem is equivalent to solving ST-MR-TA, with the role of robots and tasks switched.
- MT-MR-IA: They cast this problem to an instance of the Set Covering Problem, which is NP-hard and needs heuristic methods to be solved.
- MT-MR-TA: This is an instance of a scheduling problem, with multiprocessor tasks and multipurpose machines and is strongly NP-hard.

After classifying the MRTA problems, the authors use the taxonomy and proposed algorithms to analyze the existing approaches toward solving different MRTA problems, using 3 characteristics:

- Computation Requirements: The number of times that some dominant operation is repeated
- Communication Requirements: The total number of inter-robot messages sent over the network
- Solution Quality: A competitive factor which bounds an algorithm's performance as a function of the optimal solution

Although their proposed taxonomy is powerful enough to be able to cover and analyze many of the MRTA problems, the 2 assumptions they use which are "independent utilities" and "independent tasks", restricts the scope of their taxonomy's applicability and it will not suffice for many other important problems such as multi-robot exploration problems, which is the type of problem we are studying in this thesis.

An interesting study by [Lerman et al., 2006] , introduces a mathematical model for a dynamic task allocation. Given two tasks, the goal is for the robots to achieve an assignment with no communication or global knowledge. In their work, robots use repeated local observations to estimate the state of the environment and decide which task to choose.

These studies give a formal analysis of MRTA. Gerkey proves that the optimal solution for ST-SR-IA problem in a **known** environment is $O(m^2n)$ where m is the number of robots and n is the number of tasks. And if we consider that the robots can perform the computations in parallel then it is $O(mn)$, for each robot.

Although this time complexity may not seem very crucial for a small number of robots and tasks, it will certainly become a great overhead as these numbers increase. Also since this overhead happens in the very beginning of the system's action, it is of essential importance in the discount-reward conditions.

On the other hand, the solution to MRTA is not only time consuming, but it also needs complete and perfect knowledge about the tasks and robot locations, which in many practical settings is not available.

Another important issue with achieving the optimal solution to MRTA, is that the robots' estimates of their task fitness are assumed to be perfect, which can hardly be true in practice. For all these reasons, usually domain-specific heuristics are used for solving the MRTA problem.

2.2 Dynamic Methods for MRTA

When global and a priori knowledge about the tasks that need performing is not present, online adaptation in the robots is necessary. There are numerous dynamic, adaptive, and self-organizing methods suggested for specific tasks.

2.2.1 Animal Inspired Methods

Many of the heuristic methods for MRTA, have been inspired by animals. It has been argued (e.g. [Brooks, 1986]) that their level of intelligence is probably more easily achievable artificially than humans, but also animals that live and work in colonies such as birds, ants and bees are interesting to study, because of their "swarm intelligence" (a.k.a "collective intelligence").

The term "swarm intelligence"' was first introduced in [Beni and Wang, 1989], where it is defined as

> "Systems of non-intelligent robots exhibiting collectively intelligent behaviour evident in the ability to unpredictably produce specific ([i.e.] not in a statistical

sense) ordered patterns of matter in the external environment."

An example of these behaviours can be performing a sorting task only by random movements and simple decision rules ([Holland and Melhuish, 1999] and [Deneubourg et al., 1990]). Examples in animals include the shape of a flock of birds flying or food clustering done by ants. In [Garnier et al., 2007] authors review the biological principles that cause the "swarm intelligence" effect.

An interesting general work in animal intelligence by [Bonabeau et al., 1999] studies the intelligent behaviours of animals and discusses how to use them towards artificial intelligence. The authors review several behaviours observed in animals such as foraging and division of labor, and then present models of the phenomena and engineering applications of them.

In [Liu et al., 2007] the authors use local sensing and communication cues to automatically adjust the number of robots to perform the task. They study a swarm foraging task where robots can only use local sensing and communication and cannot get global information. In this work robots follow the "I forage when you forage" and "I rest when you rest" rules in order to automatically change the ratio of foragers and resters and improve efficiency.

A similar concept is studied by [Wawerla and Vaughan, 2010a]. The authors study a method in which each robot uses local information and then makes rule based decisions to either allocate itself to a task or rest, and show how this method can outperform centralized task allocation. In this method robots decide whether to continue doing the task, to wait, or to send out recruiting messages based on availability of pucks and existence of a queue of robots waiting to pick up the pucks.

There are numerous work in task allocation inspired by ants and bees. [Vaughan et al., 2001], [Low et al., 2004] and [Lein and Vaughan, 2008] are only a few examples. Many of these dynamic methods are introduced for foraging tasks. As we will explain in Chapter 3, this is the sample task that we selected for our implementation.

2.2.2 Learning-based Methods

Many of the dynamic methods for MRTA use learning. These methods vary from reinforcement learning to decision trees, for the robots to know which task they have to perform.

In a work by [Strens and Windelinckx, 2005] reinforcement learning as well as a lookahead value are used to allocate robots to the tasks, and also have them ready for the new

tasks. [Tangamchit et al., 2000] combines reinforcement learning with heuristics for the robots to learn the proper allocation.

MRTA also has many issues and restrictions. One of the issues arises when the robots' density is high in an area, and they interfere with each other rather than cooperate. This problem is studied in [Mataric, 1998] where the author uses local undirected broadcast communication between robots to decrease the effect of interference and achieve a higher performance.

2.3 Discounted Reward MRTA

The purpose of robots' existence is to perform work [McFarland and Spier, 1997].When work is performed by robots, some reward is earned to benefit the owner and justify the robots' existence. This reward may be independent of how long it has taken for a job to be performed. This condition is called a "constant reward".

All the methods of MRTA that were discussed before assume a constant reward condition. However, in many situations such as food collecting or rescue operations, the reward that can be gained by delivering a task is decreased over the time. This condition is called "diminishing reward", or technically "discounted reward". [Wawerla and Vaughan, 2007] presents the optimal time a robot should choose to go for recharge under discounted rewards. [Wawerla and Vaughan, 2010b] introduces a method to choose from available tasks to earn a higher long-term average reward under discounted reward scenarios.

2.4 Restricted vs. Complete Knowledge

It may seem reasonable that the more knowledge we have about the environment, including knowledge about the robots and tasks, the easier it would be to achieve an optimal task allocation and therefore a higher performance. This is true when we are planning the allocation thoroughly. However, when we cannot obtain information about the entire environment and instead plan a dynamic task allocation, this may not always be true.

Sadat and Vaughan [Sadat and Vaughan, 2010] show that, in a related foraging task, artificially constricting the robots' sensor field of view, thereby reducing the amount and quality of information available, can reduce mutual spatial interference and thus increase performance.

Using ignorance as the mechanism for action selection can be thought of as an example of the "worse is better" phenomena, described recently by Freedman and Adams [Freedman and Adams, 2009], showing that appropriate forgetting may enable robots to improve their performance by filtering irrelevant data.

Chapter 3

Problem Statement

3.1 Discounted Reward

Discounted reward situations are common scenarios in real life robotic problems. An example of a discount-reward foraging task can be the rescue operation after an earthquake, in which we want the victims to be found and transported to the designated zones as soon as possible. The goal is to minimize suffering, so the task allocation in this scenario would be finding the most critical victims and transporting them to the closest safe zone.

$$R = \alpha t^{\beta} \tag{3.1}$$

In a discount reward situation, the reward that can be gained by delivering a task, is diminished over the time. Typically, this reward is modeled by Eq. 3.1 where R is the reward, α and β are constants, and t is the time passed. β is called the discount rate. To maximize the total gained reward in such situations, we want more tasks to be delivered in earlier times than in later times. In other words, although we may have delivered fewer tasks overall, but if we deliver them earlier than what it would have taken for more tasks to be delivered, we gain a higher reward.

3.2 Task

The foraging task is defined as a task in which one or multiple agents have to search for and collect a set of target objects over an environment, and consume them or deliver them

to a designated zone [Goldberg and Mataric, 2001]. Multi-foraging, which is a variation of the typical foraging, is a task in which there are multiple types of objects that need to be collected and each consumed or delivered properly [Balch, 1999].

The foraging task is one of the canonical test-beds for cooperative robotics [Cao et al., 1997]. The real life applications of foraging can be in rescue operations, food collection, clean-up and so on. As Cao et al. put it, this task is interesting in many ways, including the fact that it can be performed independently by each robot and because of its relation to cooperative robot systems with biologically inspired approaches.

This is the sample task that we selected for our implementation. In our multi-foraging task, we have two types of objects: red pucks and blue pucks which are randomly distributed over our world and are to be transported to different zones. These pucks can be distinguished by real robots using a camera, or as in our simulations, with a fiducial detector using different fiducials for different colors. Each robot is capable of delivering pucks of both colors, but can pick only one puck at a time. At any time, the robots can either be searching or delivering a puck. The robots can switch between the puck types they are assigned to when searching, but they cannot drop a puck in the middle of delivering and do something else.

There is no a priori knowledge about these tasks in the robots. But at each of the delivery locations ("sinks") there is a human (or substitute any suitable agent), who desires a certain color of puck delivered to that location.

3.3 Foraging

In a scenario like the one illustrated in (Fig 3.1(a)), we want the robots to transport the red pucks to the upper right corner and the blue pucks to the lower left corner. If the locations and the colors of the pucks, and the locations of the robots are known to us at the beginning, and all the robots are within our communication range, we can plan the optimized path and announce it to robots so they can begin with the closer pucks and reach the highest possible discounted reward.

Aside from the time complexity that we discussed in the previous chapters, the necessary information (locations of the pucks or even robots) is likely not to be available. In the earthquake rescue example, we are most likely not to have complete information about the locations of the victims, so it is impossible to find the optimal path at the beginning and then start the rescue mission.

On the other hand, since we are dealing with discounted reward over the time, we want the robots to start their mission as soon as possible. Although in most arrangements we start with the robots by our side (and hence within our communication range) and then instruct them on the task and leave them to distribute over the world, we will probably face the disadvantage of robots reaching the tasks (pucks, victims, etc.) located further away much later than is affordable in the situation. However, if we already have the robots distributed over the world, they can learn about the tasks from other robots located close to them instead of learning from the human, and they can perform the tasks earlier and increase the total reward.

3.4 Apparatus

The goal of this thesis was to introduce a method by which we can increase the gained discounted-reward. In this method, we wanted to study the effect of "hiding" some information, on the robots' performance. For doing this, we were interested in computing the effect of information broadcast radius, on the total gained reward by the robots. This needs a large number of experiments to be done, for a large number of parameters. It also needs a fairly large group of robots, with ability to perform the assigned task. Since each experiment on real robots takes a lot of time and equipment, we decided to perform our experiments only in simulation.

Stage is now widely used for doing simulation in robotics. It is a C++ library suitable for multi-robot simulations and it can run up to 1000 times faster than real time [Vaughan, 2008]. Because of its many useful, easy-to-use features and also easy-to-track simulations, Stage has become one of the most commonly used robotic simulators in research, and is the environment that we chose for our simulations.

In our simulations, we arbitrarily decrease the reward for delivering a puck over the time to simulate the discount-reward condition.

3.5 Implementation

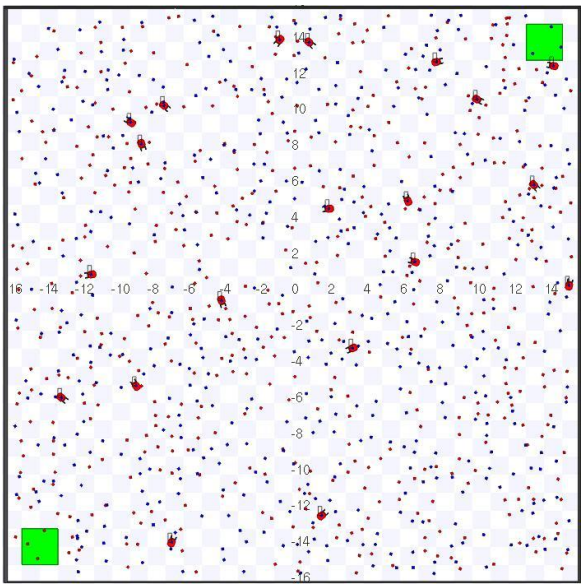
In our Stage foraging simulation we have a 32m by 32m square world map with no obstacles (Fig 3.1(b)). There are 500 red and 500 blue pucks uniformly distributed over the world. There are two "sink" regions designated for delivery of the pucks. We suppose there are two

humans standing at these regions. These humans are auxiliary and just serve as an origin for the knowledge broadcast. Each human is interested in a different color: the human at (-15,-15) is interested in the blue pucks and the one standing at (15,15) is interested in the red pucks. Humans desire pucks to be delivered to them as early as possible. They respond truthfully to all task inquiries they receive.

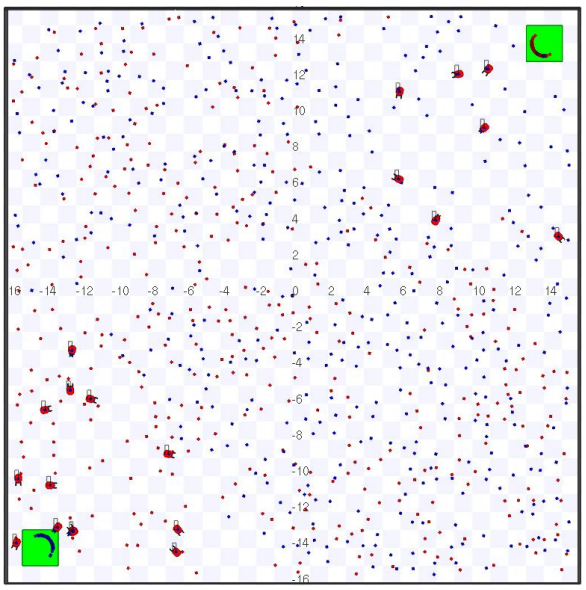
Also there are 20 robots initially distributed uniformly at random over the map. Each robot is equipped with scanning laser range finder, fiducial finder which detects the special ID assigned to the different colors of pucks, and a gripper, and is able to communicate with other robots and/or humans within a constant range. The range is the same for all robots.

Although interference is a really important factor in robotic studies, but to obtain more clear results, we neglected the interference between the robots, i.e. the robots can co-exist in one location without causing a crash. We will present the results that we obtained by having the interference between the robots turned on in the next section.

The robots start their local random search from their initial state, as described before. As we mentioned before, discount rewards are formulated by Eq. 3.1. In our experiments, we set α to be 10^8 and t to be time in microseconds. We conducted the experiments for the discount rates (β) of $-\frac{1}{2}$, -1, -2 and -3. These numbers could be any other constants, and although their choice of magnitude may affect the numerical results, they do not affect the quality of the outcome. In each experiment, the simulation is done with a different communication radius. Each experiment lasted 10,000 simulated seconds and has been repeated 20 times.



(a) Initial state



(b) After 1000 simulated seconds. Most of the red pucks near the red sink and the blue pucks near the blue sink have been delivered, and the robots are working in distinct localities.

Figure 3.1: Stage Simulation. The green squares show the destination zones for puck deliveries. The green zone on upper right is the destination for red pucks and the zone on the bottom left is destination for the blue pucks. The little dots show the pucks and the little squares with two grippers show the robots. Axes are labeled in meters.

Chapter 4

Method

In discounted reward scenarios, the reward that can be gained by completing a task is decreased over time. So if we complete a task at time t, we gain a higher reward than we will gain by delivering the same task at any time t' later than t.

In these situations, we want to complete as many tasks, as soon as possible. That is, although in the end we may have completed a smaller overall number of tasks than what could have been possible with a different task allocation, but we have completed more tasks at earlier times and thus have increased our total discounted reward.

It is known that when all of the tasks have the same value or "weight", for minimizing the average wait time we have to schedule the tasks that take a shorter time to finish, first [Smith, 1956]. This insight can be mapped to completing of tasks by robots, where for minimizing the average wait time and hence maximizing the discounted-reward, we have to complete the tasks that need less time to be completed first. In the content of our foraging scenario, we consider a "task" to be delivery of a puck with a certain color to the proper destination. In these settings where completing a task consists of searching and transporting, or in general "traveling", this can mean that the robots need to deliver the pucks that need less travel time first.

4.1 Hypothesis

Our motivating hypothesis is that by having a relatively small communication range between robots, we can exploit locality by recruiting to a task the robots located close by. If these robots remain ignorant of other available tasks, they have effectively been allocated to the

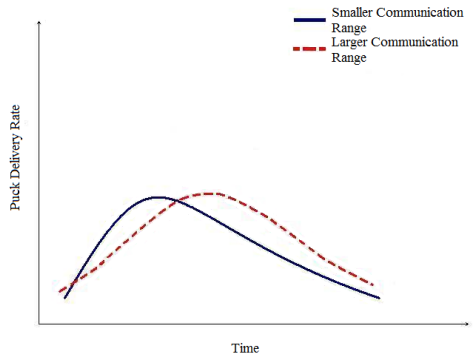


Figure 4.1: A sketch showing our hypothesis: The system with smaller communication range has a higher delivery rate at earlier times.

local task alone.

This should result in more objects being delivered at earlier times, and a higher system reward being earned. The idea is sketched in Fig 4.1, where although at the end both systems may have delivered the same number of objects, the system with the smaller communication radius delivers more tasks at earlier times.

In this thesis we present a novel method for recruiting robots to perform different discounted reward tasks. We consider the condition in which robots have no a priori knowledge about the tasks. In this case, we want the robots to search around, and if they find a puck, ask if someone needs it to be delivered. If they receive an answer indicating that puck is needed somewhere, they will deliver it.

The idea is based on the fact that in a discount reward situation where we want tasks to be delivered as soon as possible, it makes sense to recruit the resources that are initially located closer to the task location. By doing so, we can reduce the time between earlier puck deliveries, while the available reward rate is highest, and hence increase the gained reward.

4.2 Algorithm

As we mentioned in Chapter 3, we chose a foraging task for implementing our method. The details of this foraging task are presented in the Task and Foraging sections of that chapter.

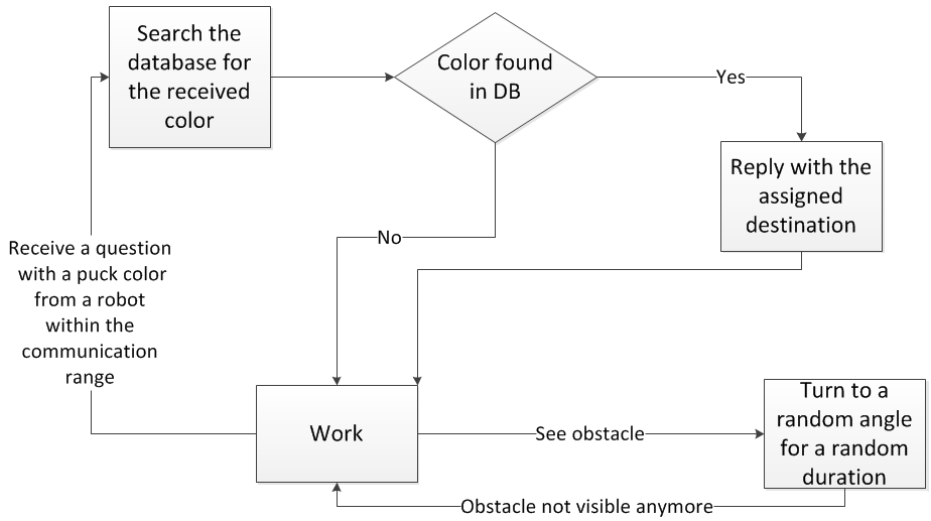


Figure 4.2: Flow diagram of the first level of the controller.

The robots have a certain communication radius that they can communicate to other robots or humans within. The robots start by doing a local random walk which is described in the next section. Whenever the robot senses a puck using its fiducial detector, the robot will change its path towards the puck. When it is close enough to the puck that it can reliably detect its color and feel it with the grippers, it will broadcast a message asking about that color of puck. If anyone (humans or other robots) within the communication range knows where to deliver that color of puck it will reply with the destination. If the robot does not receive an answer to the query after a short time, it will leave the puck and start the random walk again. If it does receive an answer, meaning the puck is needed at some destination and there is some robot or a human within the communication range that knows about it, the robot will pick up the puck and take it to the destination announced in the reply. It will also remember the destination for this puck color in its database. It will immediately collect pucks of that color in future, and it may reply to received questions about that color.

The general flow diagram of the controller is shown in Fig 4.2 and the flow diagram of the “Work” process is shown in Fig 4.3 for better understanding of the algorithm.

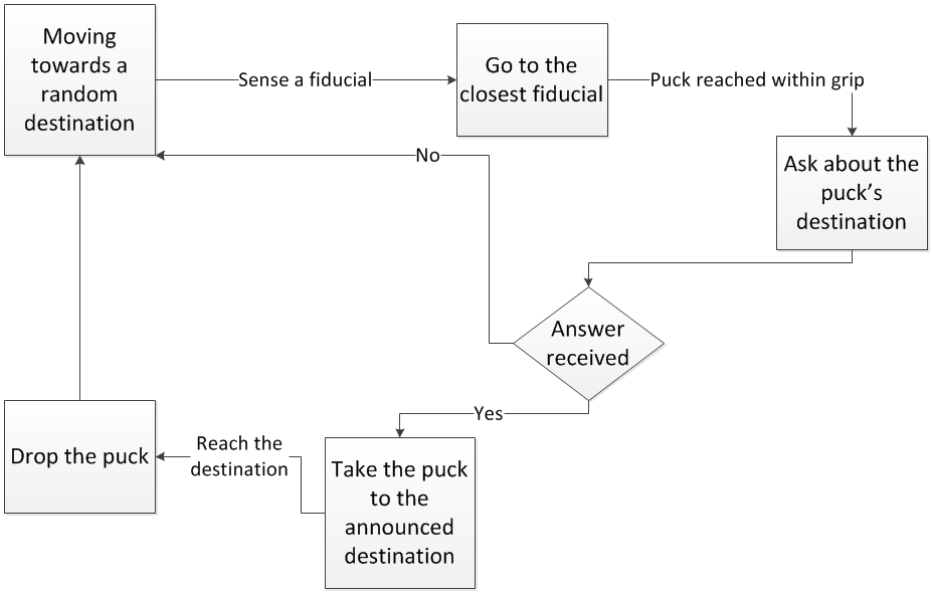


Figure 4.3: Flow diagram of the "Work" process from the previous diagram.

4.3 Locality

We use a "local walk" for the robots' exploration method. The robots have to randomly search around the environment to find tasks to perform. By keeping this search "local" in addition to "random", we can keep the information about the tasks local, which is an asset for our algorithm: closer robots are informed about the task, and also the robots who are informed about the task will not travel far from its sink.

We also need the randomness of the search to allow the robots to explore the entire world to find tasks. The method we use for the local random movement of the robots is similar to the method introduced by [Gonzalez et al., 2008]. In this random movement method, robots select a goal to move to, by selecting an angle and a travel distance.

The angle is selected randomly between 0 and 360 degrees with a uniform probability distribution. The distance is selected between 0 and the diameter of the world (or any suitably large distance) with probability inversely proportional to the squared magnitude of the distance. That is, the probability of each distance of magnitude d chosen, is equal to $\frac{c}{d^2}$, where c is a constant. This causes the robots to perform a locally-biased random walk, i.e. selecting closer distances with a higher probability. The random walk guarantees that

the entire world will be explored eventually. Parameter c controls the degree of locality. We used $c = 1$ in our experiments.

We do not restrict the robots to choose the distance from a short range, but we assign a higher probability to the closer distances; because we want the robots to be able to explore further distances too to find all the tasks and not to be trapped in a restricted area with no pucks for too long.

4.4 Dynamics: Intuition

Suppose we have a world (Fig 3.1(a)), where pucks of two colors which need transportation, are uniformly distributed over it. Assume that we have two sinks for delivery of the pucks. Assume that there are two humans standing at these two sinks, each interested in a different color. Each of these humans is trying to recruit the closest robots to them to deliver the closest pucks of their own interest to their sinks, causing the discount-reward to be as high as possible.

Assuming the communication range is small compared to the distance between delivery locations and the robot population density is small enough to prevent a totally connected network, the first robot that will learn about the task should be within that range of the original sink for the task. Similarly, all the robots that learn about that task (red, for instance) before the other (blue) either can form a path to the proper sink consisting of robot nodes within the communication range of each other who have just learned about the task, or have some robot within the range that learned about the task and delivered the puck and is now back searching for more. As a result, the robots that are allocated to a task at first are probably closer to the sink for that task rather than the other one.

Because the robots perform a local search, they will (more probably) not travel far away from their initial zone, which will get assigned to the closer sink as explained, or from the sink after delivering a puck. This will keep the knowledge about one task local, preventing the robots located further to learn about it. Thus those further robots cannot be distracted by a task that needs more traveling time and can continue their local search to find tasks in their zone.

This will result in the closest pucks to the goal sink to be transferred first, and the total discount reward earned by delivering the pucks be increased (Fig 3.1(b)).

4.5 Dynamics: Discussion

The only information our robots need for delivering a puck in this system, is the bearing from the location they have picked up a puck, to the destination for that puck color. The robots do not measure or use the distance to any location in our settings.

But if they were to use this extra information, it could not help them decide on whether or not they should deliver a puck. In a scenario where a robot finds a puck from a certain color, it cannot decide whether to deliver that puck to its destination or not, only based on the distance from its current location to the destination; but it will also need some information about the distributions of the pucks from different colors.

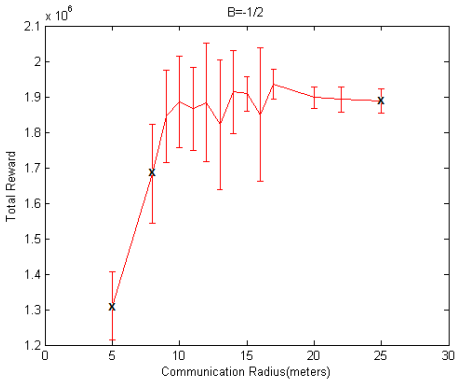
The robot can only decide not to pick up a puck from a certain color (blue, for instance) based on its distance from the goal, only if it **knows** that there will be a puck from the different color (red) which it can both **find** and **deliver** in a shorter time. This requires knowledge about the puck distributions, which is not available in the system.

4.6 Performance

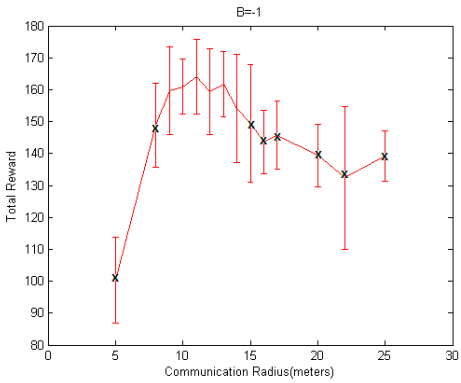
Our experiments were done for the discount rates of $-\frac{1}{2}$, -1, -2 and -3. Although in real world applications the reward usually degrades slower than this (e.g. delivering a victim to the hospital 2 seconds later, probably would not decrease the reward by $\frac{1}{8}$), but we chose these numbers for convenience regarding to the fact that our purpose was to study the dynamics and behaviour of the system, and not the actual numerical results. In each experiment, the simulation is done for different communication radii, ranging from a small distance (slightly higher than the robots fiducial-sensor range) to the diameter of the world.

Each experiment lasted 10,000 simulated seconds, at which point we stopped the simulations because due to the discount rate, after that time the gained reward would be less than $10,000^{\beta}$ times smaller than the initial rate, and thus insignificant to the results. Every experiment was repeated 20 times. Fig 4.4 summarizes the average results for the experiments.

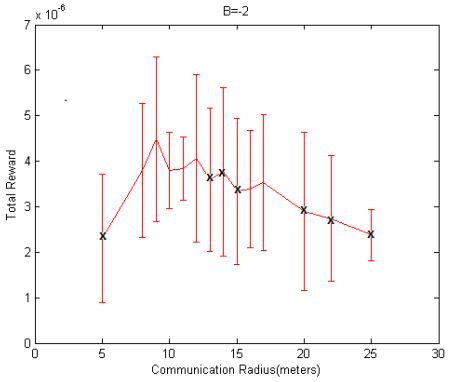
It is seen by the charts that the radius producing the highest reward is in the range 9m to 13m for the different discount rates. If the radius is too short, the robots may never get within each other's communication range and they would never have the chance to learn about the tasks, resulting in no work being done. If the radius is too high and covers all



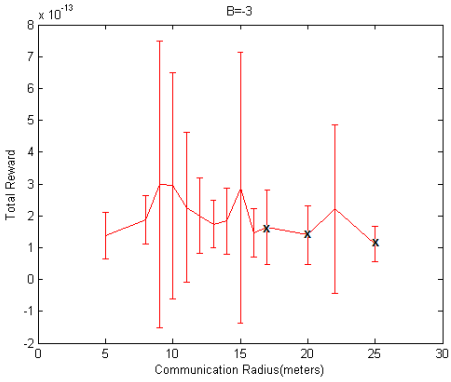
(a) Discount Rate= -1/2. The radii whose reward distributions are different from the reward distribution for R=10 are marked by an "x".



(b) Discount Rate= -1. The radii whose reward distributions are different from the reward distribution for R=11 are marked by an "x".



(c) Discount Rate= -2. The radii whose reward distributions are different from the reward distribution for R=9 are marked by an "x".



(d) Discount Rate= -3. The radii whose reward distributions are different from the reward distribution for R=9 are marked by an "x".

Figure 4.4: Total Reward gained by different radii.

the world, all of the robots will know about all of the tasks, and may choose a task which needs a lot of traveling, rather than a closer task. This will waste some time and results in a lower reward than could have been gained if the closer task had been chosen.

Since the discount rates are high in these experiments, a small difference in delivery time causes a high difference in the reward gained. This will cause the variances in the data to be large. To verify whether these results show different distributions for different radii we performed the Kolmogorov-Smirnov test on them.

For each discount rate, except $\beta = -\frac{1}{2}$ we picked the radius resulting in the best performance[1] and compared the rewards against the rewards for all other radii using the K-S test with significance value of 0.05. For $\beta = -\frac{1}{2}$ there is no evident peak so we picked $R = 10$ which is about $\frac{1}{4}$ of the diagonal of the world, which is in the same range that the peak for other discount rates happens.

For $\beta = -\frac{1}{2}$, the K-S test does not show an evident increase in the performance by limiting the communication radius but rather just a difference in the variance.

For $\beta = -1$ the test shows a significant difference between the selected radius and the radii larger than 15, which supports our hypothesis from Section 4.1, that limiting the communication radius can improve performance in a discount-reward foraging task. Also for $\beta = -2$ the test shows a reasonable difference between the selected radius and the larger radii.

For $\beta = -3$ the K-S test does not find a significant difference between the rewards for different radii since the large discount rate is causing large variances in the rewards as mentioned before.

We also performed the experiment for a discount rate of 0. In other words, in this scenario time does not have an effect on the gained reward and it is only affected by the total number of delivered pucks. In this experiment, although we had the same stages of training phase and then the fast deliveries of pucks, but in the long run all of the communication radii would result in relatively close total rewards, where if we run the experiment long enough for all of the pucks to be delivered, the total gained rewards will all be the same. This experiment was run only to validate our procedure and simulation and no results are presented.

[1] $R = 11$ for $\beta = -1$, $R = 9$ for $\beta = -2$, and $R = 9$ for $\beta = -3$

4.6.1 Interference

In the physical world, we are faced by interference. Robots face obstacles, either features of the environment or other robots, on their path and have to find their way around them. In this part of our simulation, we turned the interference between the robots on. That is, the robots can not pass through each other without causing a crash which would stop them permanently. They have to find their ways around each other so they can all go on their paths and continue working efficiently. Every other detail of the simulation are similar to the ones in the simulation described in the previous section.

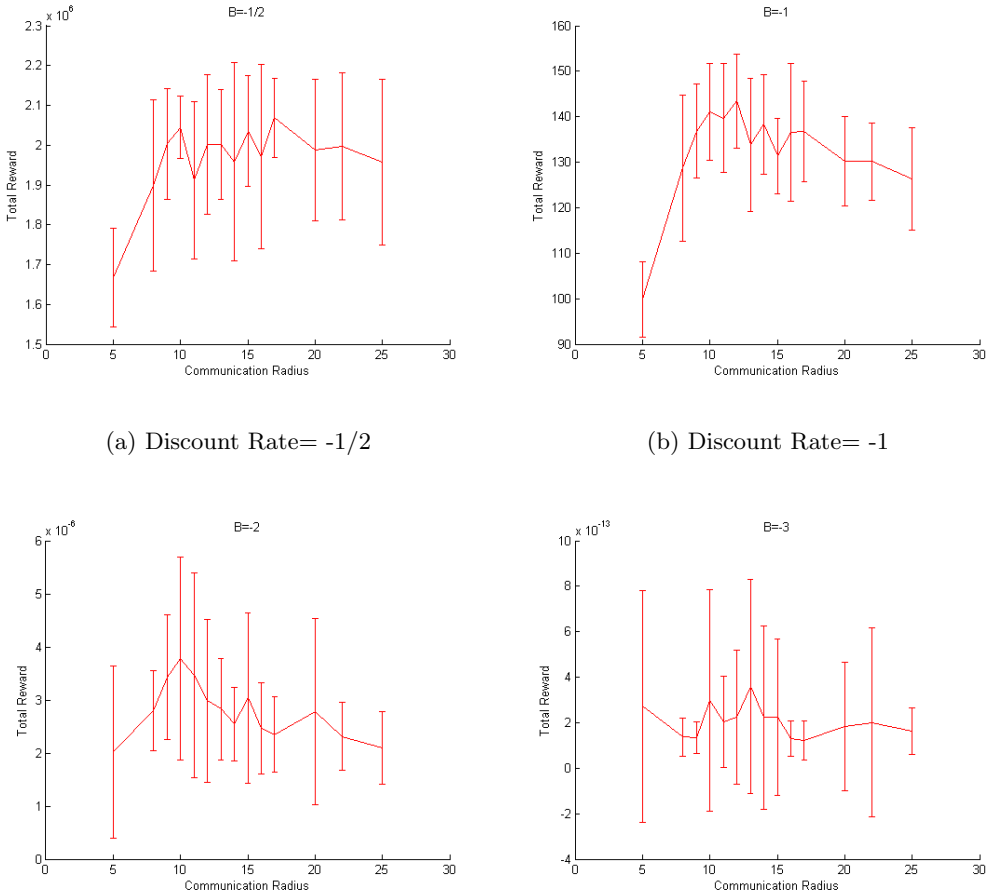
Although dealing with interference is widely studied in robotics, the methods that are introduced are mostly domain based. They also need a whole controller implemented for them, which is a lot of overhead for a simulation whose primary purpose is not studying interference.

This is why we chose a preliminary method for obstacle avoidance in our simulations. In this method, the robots continue their path towards their goal until they detect an obstacle (using IR sensors and laser). When they reach a certain distance from the obstacle, they randomly choose an angle and turn away from that obstacle. They repeat this approach until the obstacle is out of their way and they can continue on their path towards their goal point.

Again, our experiments are done for the discount rates of $-\frac{1}{2}$, -1, -2 and -3. In each experiment, the simulation is done for the same communication radii as the ones in the previous section. Each experiment lasted 10,000 simulated seconds and every experiment has been repeated for 20 times. Fig 4.5 shows the average results for these experiments.

These results show that although the results almost follow the same trends as the ones in the last section, but the variances observed are really large when interference between the robots is present. That is because even the shortest time that it takes for a robot to change its path, can change the overall gained reward drastically when we are dealing with discounted rewards. Therefore, different interference scenarios in different trials, can lead to different results with a large variance.

This is the reason we decided to turn the interference off for the rest of the experiments in this thesis, and leave it as future work to apply a more practical obstacle avoidance algorithm in our methods.



(a) Discount Rate= -1/2

(b) Discount Rate= -1

(c) Discount Rate= -2

(d) Discount Rate= -3

Figure 4.5: Total Reward gained by different radii, with interference between the robots turned on.

Chapter 5

Forgetting Method

We observed an interesting behaviour when we implemented the described method. In this chapter we introduce the problem caused by this behaviour, and introduce a new method to solve this problem. In this context, we refer to this behaviour as the "migration problem".

5.1 Migration Problem

When some time has passed from the start of a simulation and the robots have learned about the tasks and are delivering the assigned pucks to the sinks, a robot may randomly wander far from its emergent "work-site" and closer to the opposite working site.

When this happens, the robots from the other work site (red for instance) will learn about the other task (blue). Since none of the local robots are foraging those (blue) pucks in that site, local blue puck density is very high. So almost all the robots in that site will find a puck from the opposite color (blue) and will forage them to the far corner of the world. When they reach the opposite sink, the robots working at that station will in turn learn about that other task (red). Since the local density of the red pucks is now relatively high and most of the robots have learned about both of the tasks, each robot may pick any of the puck colors and deliver it to its destination. This will cause all the robots to travel, or "migrate" back and forth between the 2 work sites, traveling a long path for each puck to be delivered, hence taking a lot of time and decreasing the reward a lot.

This effect can be seen in Fig 5.1 where the trajectories of the robots are shown. These trajectories are collected from the first 1200 seconds of a simulation with the communication radius of 10 meters. Each path color represents the trajectory for a different robot and the

environment is similar to the one discussed in Chapter 4 and shown in Fig 3.1. The time intervals for different charts are chosen in a manner to best reflect the discussed effect. It can be seen from Fig 5.1 that after being assigned to different work zones and working locally for a while, robots start migrating back and forth, increasing the travel time and hence decreasing the gained reward.

5.2 Candidate Solution 1

A possible solution to this problem would be to make the robots forget about what they have learned after some time and make them start again. But doing this may leave no robots who know about a task, with the danger that no pucks remain close to the corresponding desirous humans. In this case the task can never be relearned.

5.3 Candidate Solution 2

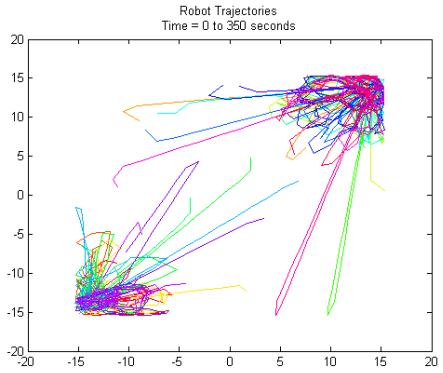
In another attempt, we tried keeping the knowledge local to the drop-off zone by allowing the robots to start teaching others only after their first drop off. This way the robots that are located far away from a task's drop-off zone, will not learn about that task at least for a while. However, during the search the robots will eventually cover the whole world and getting close to the robots from the other task and hence teaching them would be inevitable. So although this may delay the migration problem, it will still occur.

Other issues of this method were uneven allocation of the robots to the tasks, and the long time it takes for the robots to learn about the tasks.

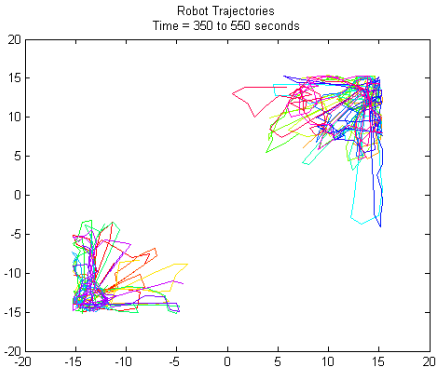
5.4 Candidate Solution 3

The other solution we tried was for the robots to forget about the task they were performing when they learn about a new task. This way although the robots may have traveled to the other work site to perform the other task, but since they have forgotten about the last task, they will not travel back at once; instead they will have to learn about the former task by another robot again in order to travel back.

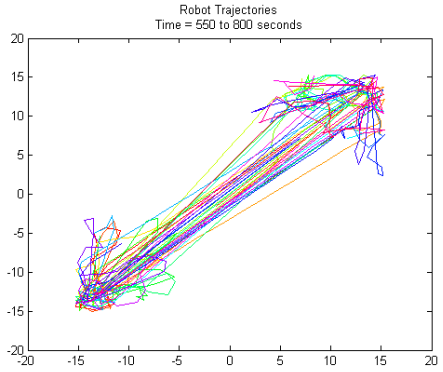
Although this method causes the traveling back and forth to occur less frequently, it will still exist because the probability of a robot meeting a robot from the previous task and



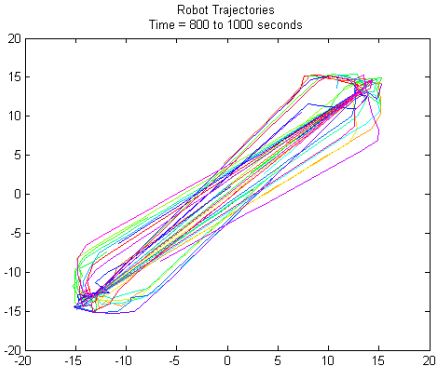
(a) Robots starting to be recruited to different zones



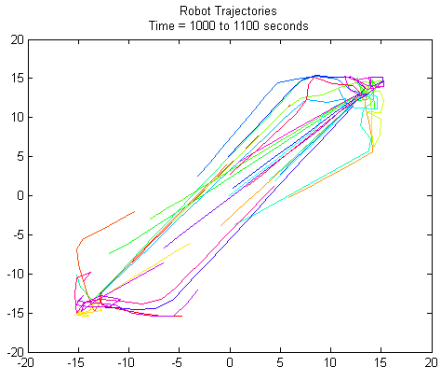
(b) Working locally



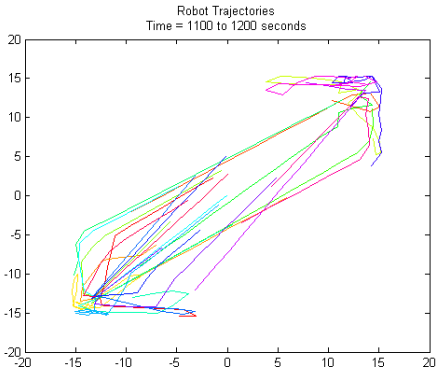
(c) Migrating



(d) Migrating



(e) Migrating



(f) Migrating

Figure 5.1: Robots' trajectories for the times 0 to 1200 seconds in the simulation with the communication radius of 10 meters. Each color represents a different robot.

switching back to that task is high.

5.5 Candidate Solution 4

We proposed another alternative for solving this problem that proved to be effective. In this method, when a robot first learns about a task, it cannot learn about a new task for a certain amount of time τ . After this time has passed, the robot is able to learn a new task and when it does, it has to switch to performing that task for a τ amount of time. After that, when the robot knows about both tasks, there is always a chance of $\alpha = 0.9$ to continue the task it is already performing, and a $1 - \alpha = 0.1$ chance to switch to the other task. In other words, when a robot encounters a puck of the same color as the pucks it was delivering before, the robot picks the puck and delivers it. But when it encounters a puck of the other color, there is only a 0.1 chance that it chooses to deliver it. However, when the switching happens there cannot be a switching back for τ amount of time.

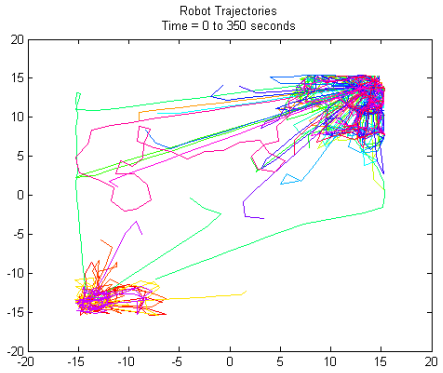
The ability to switch back to a task is necessary in this method, so we can prevent an uneven task allocation. If there is no switching back to a task, a large fraction of robots may be allocated to one task, leaving only a few robots to perform the other. While giving the robots the chance to switch back to a task will prevent this from happening and will ensure that the robots are almost evenly assigned to the tasks.

We performed similar simulations to those from the previous chapter for implementing this method. The results obtained by these new simulations are reported later in this chapter.

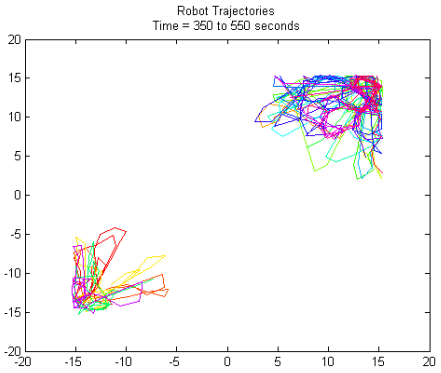
5.6 Implementations

We chose Solution 4 to implement in our experiments. Finding the optimal not-learn-another-task time τ, was not trivial. So we decided to pick a few numbers which are neither too small nor too large and try them out in the experiments. We performed our simulations for τ= 300, 600, and 1200 seconds. We found empirically the time that produced the best effect in "delivering more pucks in earlier times" and hence increasing the discounted reward among these numbers, was τ=600 seconds. This is the time for which we report the results in this chapter.

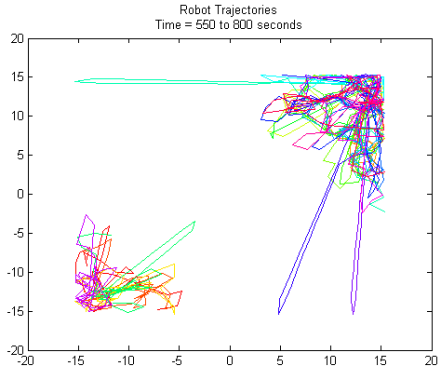
The effect this solution has on the dynamics of the system can be seen in Fig 5.2 where



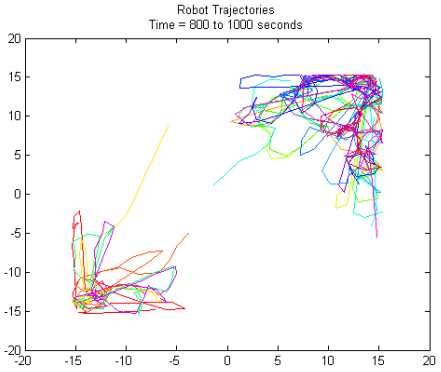
(a) Robots starting to be recruited to different zones



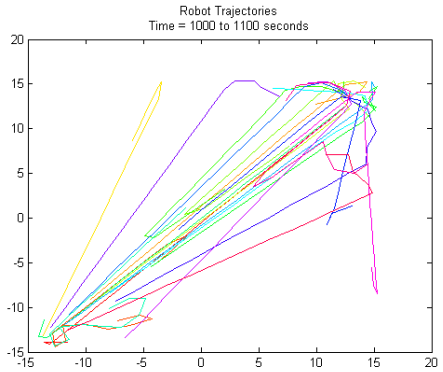
(b) Working locally



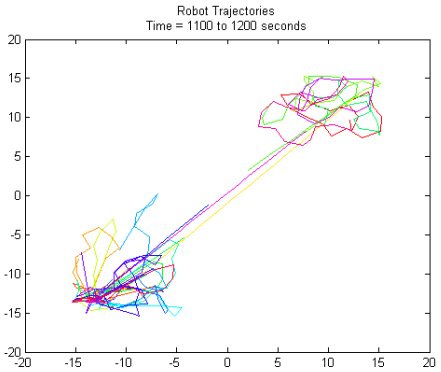
(c) Working locally



(d) Working locally

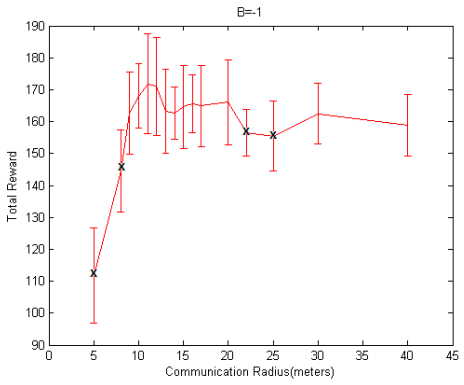


(e) Migrating

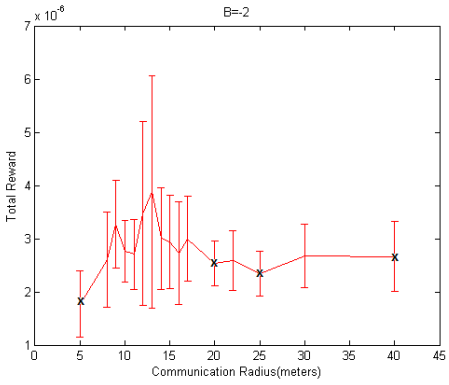


(f) Working locally

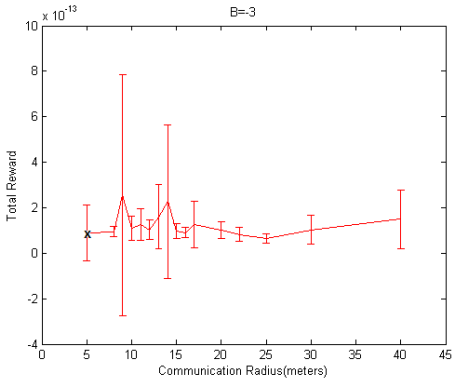
Figure 5.2: Robots' trajectories for the times 0 to 1200 seconds in the simulation with the communication radius of 10 meters. Each color represents a different robot.



(a) Discount Rate= -1



(b) Discount Rate= -2



(c) Discount Rate= -3

Figure 5.3: Total Reward gained by different radii, using Solution 4. The radii whose performance are significantly different from the best performance using the K-S test, are shown by an "x". Each experiment has been repeated 10 times for each radius.

the trajectories of the robots are shown. Again these trajectories are collected from the first 1200 seconds of a simulation with the communication radius of 10 meters which is equal to the radius for which Fig 5.1 was generated. Each path color represents the trajectory for a different robot and the environment is similar to the one discussed in Chapter 3 and shown in Fig 3.1. The time slots for different charts are chosen in a manner to best reflect the discussed effect. It can be seen from Fig 5.2 that after being assigned to different work zones and working locally for a while, robots may migrate to a different zone, but will work locally for a while after that and will not migrate back immediately.

5.7 Results

We repeated the previous experiments from Chapter 3, using the modified algorithm Solution 4. Fig 5.3 shows the results for these experiments. To verify that the results for different radii do come from different distributions, we performed the Kolmogorov-Smirnov test on them. For each discount rate, we compared the results for the radius leading to the highest performance[1] against all the other radii, using K-S test with significance value of 0.05. The radii whose performance are significantly different from the best performance using this test, are shown by an "x" in Fig 5.3.

Here again because of the large discount rates, a small change in the delivery time will have a large effect on the total discounted reward. This will cause large variances in the results and failure of the K-S test to reject the null hypothesis that results for different radii come from different distributions.

Having these facts in mind and for clarity, we present another way of representing these data: the histograms of puck delivery times. The puck delivery times are not affected by the magnitude of the discount rate and since they do not vary greatly by a small change in time, they are easier to study. In Fig 5.4, the histograms for 5 communication radii *r =5, 10, 16, 25* and *40* are shown. Each histogram shows the time of delivery of a puck, either red or blue, to its proper destination.

The times are collected by running 30 experiments for each radius (10 times for each discount-rate, which does not have any effects on the delivery time, but on the gained reward). The times later than 100 simulated seconds are not shown in the histograms, since

[1] $R = 11$ for $\beta = -1$, $R = 13$ for $\beta = -2$, and $R = 9$ for $\beta = -3$

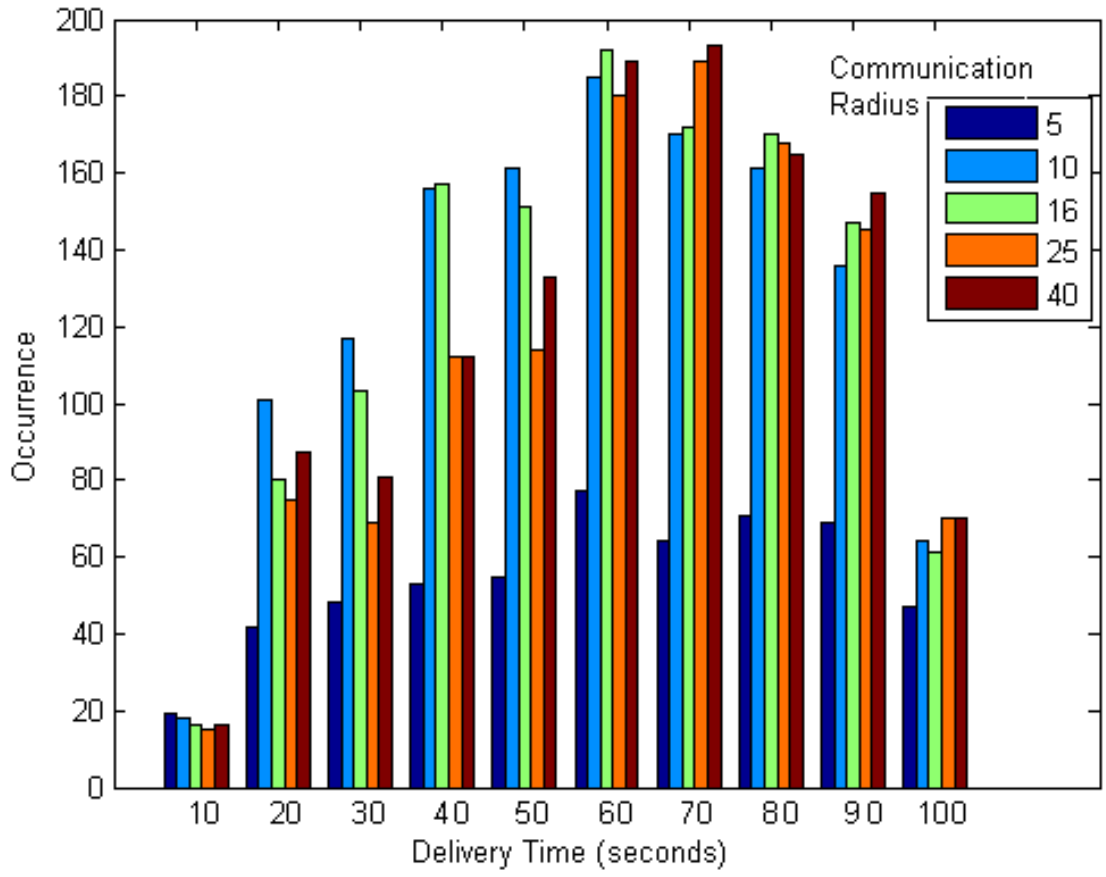


Figure 5.4: Histograms of Puck Deliveries

based on the size of the world and speed of the robots, they are considered to be very late and have negligible effect on the total reward. Since the system is working under a discounted reward situation, we want high delivery frequencies in earlier times. As the discount rate increases, getting high frequencies in early times gets more important. The total reward gained will then be the weighted (based on the discount rate) sum of the histograms.

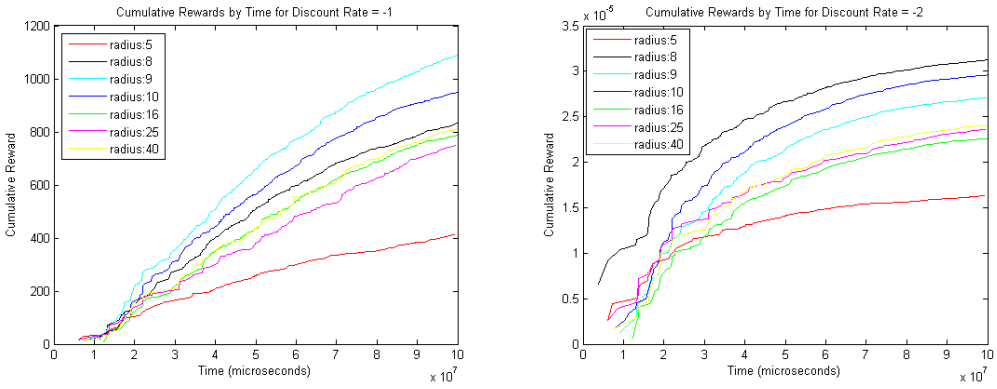
These histograms clearly show that the system in which the communication radius is 10m is outperforming the other systems in earlier times. That is, except rather similar performances in all 5 systems at the very beginning which is due to the "learning time" discussed in the next chapter, the system with communication radius of 10m gets more pucks delivered in earlier times and earns a higher discount-reward.

This is further evidence supporting our hypothesis (Fig 4.1) that the best communication radius for robot recruitment in this system is not the largest radius, and that by reducing the radius down to some point and hiding some information from the robots and hence preventing them from becoming distracted with tasks that are far away from the robots, we get a better overall performance.

There is still another good way of showing our results: By examining the logs of puck delivery times the delivery times, we show the cumulative reward at each delivery time; after all, it is the total reward we are interested in increasing. For each discount rate and radius, we calculate the overall earned reward after each delivery. This helps studying the trend different radii earn the reward under each discount rate. (Fig 5.5) shows these results up to the same time as (Fig 5.4), 100 seconds, which is technically late enough for these discount rates that the deliveries after that do not affect the overall reward much.

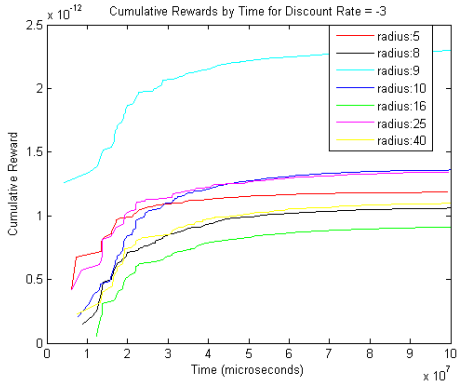
From these diagrams, it can be seen that with discount rates of -1 and -3, the system with communication radius of 9m, and with the discount rate of -2, the systems with communication radius of 8m outperform the other systems. This again shows that the best communication radius is not the largest one, for the reasons we explained before.

Remember that by decreasing the communication radius, we are increasing the time it takes for the first robots to discover the tasks and teach others about them. This is what we called the "learning time", and the overall reward will be the trade-off between this time, and the efficiency we gain by recruiting the closest robots by our method. The learning time is distinguishable to some extent in the beginning part of each curve in the charts in (Fig 5.5). We will explain more about the learning time in the next chapter.



(a) Discount Rate= -1

(b) Discount Rate= -2



(c) Discount Rate= -3

Figure 5.5: Cumulative rewards based on each delivery time earned by different radii.

Chapter 6

Discussion

In this chapter we review how our method and results contribute to fill a gap in previous work. We also introduce and discuss some problems that exist in evaluating the performance of our method.

6.1 How to relate the results back into previous studies

The problem we addressed in this thesis is an instance of ST-SR problem, similar to ALLIANCE Efficiency Problem (Parker 1995) discussed in [Gerkey and Mataric, 2004] which is NP-hard. The objective in that problem is to allocate a subset of the tasks to each robot so as to minimize the maximum time it takes for the robot to serially perform all of its allocated tasks. But since this is an exploration problem in an unknown environment, the taxonomy introduced in [Gerkey and Mataric, 2003] fails to cover it.

In order to have an upper bound to compare our results with, we calculated the maximum discounted reward that the robots could gain in our settings. With the same starting points for the pucks as in our previous experiments, the best result is gained by having our robots start at the n pucks located closest to the destination where n is the number of robots, and then go to the next puck located closest to the destination after each drop off.

Using this algorithm, we calculated the discounted reward based on the previous discount rates. The results are shown in Table 6.1.

Comparing the results shown in this Table 6.1. may show a vast difference, but we shall remember again that the results change exponentially with time and even a small change in the delivery time can change the results drastically.

Discount Rate	-1	-2	-3
Optimal Total Reward	514.702	6.61053e-005	1.6758e-011
Average Total Reward	171.8043 (Radius:11)	3.8777e-006 (Radius:13)	2.5579e-013 (Radius:9)

Table 6.1: The total discounted rewards for different discount rates. The first row shows the optimal rewards. Second row shows the average total reward for the radius resulting in best average reward for each discount rate for the "Forgetting Method" introduced in Chapter 5.

6.2 Issues in Performance Evaluation

A problem in analyzing the efficiency of the system is that it takes some time for the first robot in each experiment to find a puck close to the source, learn about the task, and start working and teaching other robots. Since we have a discount reward situation, this highly affects the total reward that is gained by the robots. However, if we can afford a "learning time" in our system and start discounting after that time, this method can gain a higher performance. For instance, if we have some time to train one or two robots about our desired tasks, and then we want quick results, the performance of the system increases.

Another issue is that in real systems, we have to deal with spatial interference. We have performed simulations with interference between the robots turned on, but the additional stochastic effects of our simplistic obstacle avoidance the results had very large variances and they are not reported here. We are aware that spatial interference is a serious problem for multi-robot systems, and that encouraging robots to work closely together could adversely affect the performance. Disabling interference makes these results less persuasive. However, we hope that in future work with a careful controller and experiment design we can see locality benefits even in the presence of interference.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

We introduced a method for emergent multi-robot task allocation in a discounted reward situation, with no global knowledge about the tasks or robots. Apart from the method, as can be seen from the results, we show that there is an optimal radius for human-robot and robot-robot communications in the method to maximize the total reward gained, and this radius is not the highest possible one. Instead, the radius causing the highest performance is roughly $\frac{1}{4}$ of the diagonal of the world.

We also described an interesting phase-shift in behaviour when isolated populations are invaded by knowledge of different tasks, leading to a sub-optimal oscillatory "migration" behaviour. Then we offered a method to mitigate this problem, and showed experimentally that this improves performance.

7.2 Future Work

The work introduced in this thesis is merely a primary idea on designing adaptive behaviour in discounted foraging situations in order to improve the performance. In this section we introduce some of the future work that interests us.

The most important future work in order to be able to apply our introduced idea in practical settings, is to embed a more intelligent obstacle avoidance method in our system to deal with spatial interference. Also of interest is seeing if the effect can be observed in different world settings, including fixed obstacles, various different sinks and sources,

weighted rewards for object types.

The other important work in terms of application is to try to use our method for different discounted reward situations and different types of tasks. Although as it is, the system is applicable to discount reward foraging tasks such as for rescue robots in an earthquake site, in which we want the injured people to be delivered to the safe site as soon as possible, but we want to expand the system to be able of performing more general tasks besides only foraging.

Another interesting work is to study the mathematical relation between the discount rate and the communication radius resulting in the highest performance, based on the size of the world.

And finally, it is worth noting again that a simple idea such as hiding information instead of revealing it, restricting the communication range instead of using the full possible range, can have an improving effect on the outcome. Maybe we should look for more instances of this idea, where re-examining the traditional beliefs, even the ones that make total sense, could benefit us in the end.

Bibliography

[Balch, 1999] Balch, T. (1999). The impact of diversity on performance in multi-robot foraging. In *Proceedings of the Third International Conference on Autonomous Agents(Agents'99)*, pages 92–99, Seattle, WA.

[Beni and Wang, 1989] Beni, G. and Wang, J. (1989). Swarm intelligence in cellular robotic systems. In *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, volume 102.

[Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *From Natural to Artificial Swarm Intelligence*. Oxford University Press.

[Brooks, 1986] Brooks, R. (1986). Achieving artificial intelligence through building robots. Technical report, Cambridge, MA, USA.

[Cao et al., 1997] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:226–234.

[Deneubourg et al., 1990] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. (1990). The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on from Animals to Animats*, pages 356–363, Cambridge, MA, USA. MIT Press.

[Freedman and Adams, 2009] Freedman, S. T. and Adams, J. A. (2009). Human inspired robotic forgetting: Filtering to improve estimation accuracy. In *Proceedings of the 14th IASTED International Conference on Robotics and Applications (RA 2009)*.

[Garnier et al., 2007] Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31.

[Gerkey and Mataric, 2003] Gerkey, B. P. and Mataric, M. J. (2003). Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2003)*, pages 3862–3867.

[Gerkey and Mataric, 2004] Gerkey, B. P. and Mataric, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *Intl. Journal of Robotics Research*, 9(23):939–954.

[Goldberg and Mataric, 2001] Goldberg, D. and Mataric, M. J. (2001). Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. In *Robot Teams: From Diversity to Polymorphism*, pages 315–344. A K Peters Ltd.

[Gonzalez et al., 2008] Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. (2008). Understanding individual human mobility patterns. *Nature*, 453(7196):779 – 782.

[Holland and Melhuish, 1999] Holland, O. and Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2):173–202.

[Lein and Vaughan, 2008] Lein, A. and Vaughan, R. T. (2008). Adaptive multi-robot bucket brigade foraging. In *Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*.

[Lerman et al., 2006] Lerman, K., Jones, C., Galstyan, A., and Mataric, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 3(25):225–242.

[Liu et al., 2007] Liu, W., Winfield, A. F. T., Sa, J., Chen, J., and Dou, L. (2007). Towards Energy Optimization: Emergent Task Allocation in a Swarm of Foraging Robots. *Adaptive Behavior*, 15(3):289–305.

[Low et al., 2004] Low, K. H., Leow, W. K., and Ang, M. H. (2004). Task allocation via self-organizing swarm coalitions in distributed mobile sensor network. In *AAAI'04: Proceedings of the 19th national conference on Artifical intelligence*, pages 28–33. AAAI Press.

[Mataric, 1998] Mataric, M. J. (1998). Using communication to reduce locality in distributed multi-agent learning. *ournal of Experimental and Theoretical Artificial Intelligence, special issue on Learning in DAI Systems*, 3(10):357–369.

[McFarland and Spier, 1997] McFarland, D. J. and Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*, (20):179190.

[Sadat and Vaughan, 2010] Sadat, A. and Vaughan, R. T. (2010). Blinkered lost: Restricting sensor field of view can improve scalability in emergent multi-robot trail following. In *Proceedings of the IEEE International Conference on Robotics and Automation*. (to appear).

[Smith, 1956] Smith, W. E. (1956). Various optimizers for single-stage production. In *Naval Research and Logistics Quarterly*, volume 3, pages 59–66.

[Stephens et al., 2007] Stephens, D. W., Brown, J. S., and Ydenberg, R. C. (2007). *Foraging - Behavior and Ecology*. University of Chicago Press, Chicago.

[Strens and Windelinckx, 2005] Strens, M. J. A. and Windelinckx, N. (2005). Combining planning with reinforcement learning for multi-robot task allocation. In *Adaptive Agents and Multi-Agent Systems*, pages 260–274.

[Tangamchit et al., 2000] Tangamchit, P., Dolan, J. M., and Khosla, P. K. (2000). Learning-based task allocation in decentralized multirobot systems. In *Distributed Autonomous Robotic Systems*, volume 4, pages 381–390.

[Ulam and Balch, 2003] Ulam, P. and Balch, T. (2003). Niche selection for foraging tasks in multi-robot teams using reinforcement learning. In *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects*.

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

[Vaughan et al., 2001] Vaughan, R. T., Stoy, K., Howard, A., Sukhatme, G., and Mataric, M. J. (2001). Lost: Localization-space trails for robot teams. Technical report, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California. A later version of this paper was published in the IEEE Transactions on Robotics and Autonomous Systems 18:5, 2002.

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

[Wawerla and Vaughan, 2010a] Wawerla, J. and Vaughan, R. T. (2010a). A fast and frugal method for team-task allocation in a multi-robot transportation system. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[Wawerla and Vaughan, 2010b] Wawerla, J. and Vaughan, R. T. (2010b). Online robot task switching under diminishing returns. In *Proceedings of the Twelfth International Conference on Artificial Life (ALife XII)*. (to appear).

[Werger and Mataric, 2000] Werger, B. and Mataric, M. J. (2000). Broadcast of local eligibility for multi-target observation. In *Proceedings of 5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 347–356.