Summer 7-15-2019

An Efficient Multiple-Place Foraging Algorithm for Scalable Robot Swarms

Qi Lu
University of New Mexico - Main Campus

Qi Lu

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Melanie E. Moses , Chairperson

Stephanie Forrest

Carlo Pinciroli

Joshua P. Hecker

An Efficient Multiple-Place Foraging Algorithm for Scalable Robot Swarms

by

Qi Lu

B.A., Computer Science & Technology, Hubei University, China, 2005
M.S., Computer Science, Aalborg University, Denmark, 2011
M.S., Computer Science, University of New Mexico, NM, 2014

DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Computer Science

The University of New Mexico

Albuquerque, New Mexico

July, 2019

Dedication

To my parents, for their constant devotion; to my wife, Wenhao, for her support, encouragement, and patience; to my children Joseph and Miga, for bringing me joy and hope.

'The greatest truths are the simplest.'
— Lao Tzu

Acknowledgments

My advisor, Dr. Melanie Moses, supported me and gave me valuable guidance. Thank you for entrusting me with responsibility of investigating such a significant project and allowing me to explore all aspects of my research. Thank you for your seemingly endless patience and for pointing me to right directions.

Thank you for rebuilding my self-confidence. Thank you for making my Ph.D. life enjoyable. Thank you for lighting up my future. I can not express my gratitude to you in more words. Without your effort, I can not reach current stage. I learned a lot from you far beyond research.

Thanks to Joshua Hecker and G. Matthew Fricke for guiding me in this project. Without their leading work, along with NASA Swarmathon team (Antonio Griego, Jarett Jones, Jake Nichol, William Vining, Vanessa Surjadidjaja, John Ericksen, Wayne Just, Sarah Ackerman, Kristiana Rendon, Shannon McCoy-Hayes, and etc.), all my work would be confined to a limited scope.

Thanks to my project collaborators. Antonio was the first person who introduced ARGoS simulation environment to me. He also the person who assisted me a lot in physical experiments. Jarett, Jake, and William are helpful in our physical robot system.

Thanks to Dr. Carlo Pinciroli for discussing the implementation issues in ARGoS. His prompt response and thorough knowledge was very impressive. Thanks to Dr. Stephaine Forrest for giving me insight suggestions. She motivated and inspired me think further in my project.

Thanks also to Dr. Ken'ichi Yano and his master students Takaya Tsuno and Tatsuhiro Morimoto from MIE University, Japan. Their hard work and technical expertise has been essential in designing the physical depot we expected.

Finally, thanks to my dissertation committee for volunteering their time to evaluate this work and provide suggestions for improvement.

An Efficient Multiple-Place Foraging Algorithm for Scalable Robot Swarms

by

Qi Lu

B.A., Computer Science & Technology, Hubei University, China, 2005
M.S., Computer Science, Aalborg University, Denmark, 2011
M.S., Computer Science, University of New Mexico, NM, 2014

Ph.D., Computer Science, University of New Mexico, 2019

Abstract

Searching and collecting multiple resources from large unmapped environments is an important challenge. It is particularly difficult given limited time, a large search area and incomplete data about the environment. This search task is an abstraction of many real-world applications such as search and rescue, hazardous material clean-up, and space exploration. The collective foraging behavior of robot swarms is an effective approach for this task. In our work, individual robots have limited sensing and communication range (like ants), but they are organized and work together to complete foraging tasks collectively. An efficient foraging algorithm coordinates robots to search and collect as many resources as possible in the least amount of time. In the foraging algorithms we study, robots act independently with little or no central control.

As the swarm size and arena size increase (e.g., thousands of robots searching over the surface of Mars or ocean), the foraging performance per robot decreases. Generally, larger robot swarms produce more inter-robot collisions, and in swarm robot foraging, larger search arenas result in larger travel distances causing the phenomenon of *diminishing returns*. The foraging performance per robot (measured as a number of collected resources per unit time) is sublinear with the arena size and the swarm size.

Our goal is to design a scale-invariant foraging robot swarm. In other words, the foraging performance per robot should be nearly constant as the arena size and the swarm size increase. We address these problems with the Multiple-Place Foraging Algorithm (MPFA), which uses multiple collection zones distributed throughout the search area. Robots start from randomly assigned home collection zones but always return to the closest collection zones with found resources. We simulate the foraging behavior of robot swarms in the robot simulator ARGoS and employ a Genetic Algorithm (GA) to discover different optimized foraging strategies as swarm sizes and the number of resources is scaled up. In our experiments, the MPFA always produces higher foraging rates, fewer collisions, and lower travel and search time than the Central-Place Foraging Algorithm (CPFA). To make the MPFA more adaptable, we introduce dynamic depots that move to the centroid of recently collected resources, minimizing transport times when resources are clustered in heterogeneous distributions.

Finally, we extend the MPFA with a bio-inspired hierarchical branching transportation network. We demonstrate a scale-invariant swarm foraging algorithm that ensures that each robot finds and delivers resources to a central collection zone at the same rate, regardless of the size of the swarm or the search area. Dispersed mobile depots aggregate locally foraged resources and transport them to a central place via a hierarchical branching transportation network. This approach is inspired by ubiquitous fractal branching networks such as animal cardiovascular networks that deliver resources to cells and determine the scale and pace of life. The transportation of resources through the cardiovascular system from the heart to dispersed cells is the inverse problem of transportation of dispersed resources to a central collection zone through the hierarchical branching transportation network in robot swarms. We demonstrate that biological scaling laws predict how quickly robots forage in simulations of up to thousands of robots searching over thousands of square meters. We then use biological scaling predictions to determine the capacity of depot robots in order to overcome scaling constraints and produce scale-invariant robot swarms. We verify the predictions using ARGoS simulations.

While simulations are useful for initial evaluations of the viability of algorithms, our ultimate goal is predicting how algorithms will perform when physical robots interact in the unpredictable conditions of environments they are placed in. The CPFA and the Distributed Deterministic Spiral Algorithm (DDSA) are compared in physical robots in a large outdoor arena. The physical experiments change our conclusion about which algorithm has the best performance, emphasizing the importance of systematically comparing the performance of swarm robotic algorithms in the real world. We illustrate the feasibility of implementing the MPFA with transportation networks in physical robot swarms. Full implementation of the MPFA in an outdoor environment is the next step to demonstrate truly scalable and robust foraging robot swarms.

Contents

List of Figures **xiii**

List of Tables **xxi**

Glossary **xxii**

1 Introduction **1**
1.1 The Scalability of Foraging Robot Swarms 2
1.2 The Multiple-Place Foraging Algorithm 3
1.3 From Simulation to Physical Robots 7
1.4 Organization and Contributions 9

2 Background **12**
2.1 Stochastic Central-Place Foraging 12
2.2 Distributed Deterministic Spiral Search 15
2.3 Task Partitioning 16
2.4 Existing Simulators and Physical Robot Platforms 17

3 The MPFA: A Multiple-Place Foraging Algorithm for Robot Swarms **20**
3.1 Publication Notes 20
3.2 Abstract 21
3.3 Introduction 21
3.4 Background: The CPFA 24
3.5 Methods 26
3.5.1 The Design of the MPFA 26
3.5.2 Experimental Configuration in ARGoS 30
3.6 Results 32
3.6.1 Foraging Performance 32

3.6.2 Collision Avoidance 34
3.6.3 Search and Travel Efficiency 35
3.6.4 Observed Trends in Parameters 36
3.7 Discussion 38
3.8 Acknowledgments 41

4 The Scalability and Adaptation of MPFA 42
4.1 Publication Notes 42
4.2 Abstract 43
4.3 Introduction 44
4.4 Related Work 45
4.5 The Design of The MPFA 47
4.6 The GA Evolution 49
4.7 Experimental Configuration in ARGoS 52
4.8 Results 53
4.8.1 Foraging Efficiency 53
4.8.2 Collision Efficiency 54
4.8.3 Travel and Search Efficiency 56
4.9 Discussion 59
4.10 Acknowledgements 61

5 Multiple-Place Swarm Foraging with Dynamic Depots 62
5.1 Publication Notes 62
5.2 Abstract 63
5.3 Introduction 64
5.4 Related Work 67
5.4.1 Central-Place Foraging 67
5.4.2 Multiple-Place Foraging 68
5.4.3 Foundations of the MPFA 70
5.5 Methods 76
5.5.1 Implementation of Robot Controllers 78
5.5.2 Evolving Swarm Behavior 82
5.6 Experimental Configuration 84
5.6.1 ARGoS Implementation 87
5.7 Results 89
5.7.1 Foraging Performance 90
5.7.2 Search and Travel Time 91
5.7.3 Collision Time 93
5.7.4 Scalability 94
5.7.5 Transport to A Central Depot 96
5.8 Discussion 98

5.8.1 Online Decision-Making in Response to Local Information . . 100
5.8.2 Broader Implications for Scalable Design 101
5.8.3 The Path to Implementation 103
5.9 Acknowledgements 106

6 A Bio-Inspired Hierarchical Branching Transportation Network 107
6.1 Publication Notes 107
6.2 Abstract 107
6.3 Introduction 108
6.4 Related Work 111
6.5 Similarities between cardiovascular systems and robot swarms 114
6.6 Scaling laws for foraging swarms 115
6.6.1 Assumptions 115
6.6.2 The explosion network 117
6.6.3 The hierarchical branching transportation network 118
6.6.4 Scale-invariant transportation network 119
6.7 Experimental Setup 120
6.8 Results 122
6.8.1 Prediction I 122
6.8.2 Prediction II 125
6.8.3 Prediction III 125
6.9 Discussion 125

7 Comparing Physical and Simulated CPFA and DDSA 136
7.1 Publication Notes 136
7.2 Abstract 137
7.3 Introduction 138
7.4 Related work 140
7.5 Central Place Foraging Algorithm: CPFA 141
7.6 Distributed Deterministic Search Algorithm: DDSA 143
7.7 Description of Simulated & Physical Robots 145
7.8 Experimental Setup 149
7.9 Results 150
7.10 Discussion 154

8 From Simulation to Physical MPFA$_\mathbf{T}$ 157
8.1 Physical Depot Design 158
8.2 Gazebo Simulation 159
8.3 From Physical to Virtual and Back Again 160
8.4 Future work 162

9 Conclusions **163**

9.1 Concluding Remarks 163

9.2 Broader Impact 165

List of Figures

1.1 The MPFA running in ARGoS, front view. Resources are shown as black dots arranged in a partially clustered distribution. Red circles indicate uniform distributed collection zones. Colored rays indicate pheromone waypoints with different strength (green indicates high and red indicates low). 4

1.2 Illustration of the inspiration from cardiovascular networks in biology to hierarchical branching network $\mathrm{MPFA_T}$ in robot swarms (left figure replicated from (Moses et al., 2016)). 6

1.3 A physical Swarmie robot with a cube on its gripper. 8

2.1 The CPFA running in ARGoS, overhead view. The circle in the center indicates the collection zone. The partially clustered distribution of resources are shown as black dots, robots blue larger dots, lines indicate the paths taken by robots during the experiment. 13

2.2 The DDSA running in ARGoS. The robots search on pre-planned spiral search paths beginning at a central collection zone. Resources are shown as black dots arranged in a partially clustered distribution. Robots are marked with blue larger dots. Colored lines are the paths of robots. 16

3.1 Schematics showing individual robot foraging trips in (a) the CPFA and (b) the MPFA. In (a), a robot begins its search at a central nest (red circle) and travels to a search site (step 1). Upon reaching the search site, the robot searches for resources by uninformed random walk (step 2) until a resource (black square) is found and collected. After sensing the local resource density, the robot returns to the nest (step 3). In (b), 4 nests are placed. The foraging behavior is identical to the CPFA, except that the robot returns to the nest closest to the location where it finds a resource. The robot path in the upper left of panel (b) shows the robot returning to the nest that it departed from. The path in the lower right of panel (b) shows a robot that finds a resource closer to a different nest, and so it deposits that resource at the new closer nest. If the robot chose to lay a pheromone waypoint, the waypoint would connect the new nest to the resource location. . 27

3.2 The placement of nests and resources in ARGoS. In all experiments 256 resources (black points) and 24 robots are placed in a 10 × 10m arena, and some number of nests (red circles) are distributed uniformly in the search space. The resources are partially clustered in panel (a), unclustered and spread in a uniform random distribution in (b) and clustered into 4 piles in panel (c). Panel (d) shows a simulation running with 24 robots, the partially clustered resource distribution and four nests. The colored rays indicate pheromone waypoints with different strength. A small area is magnified in each figure to show the resource placement. 31

3.3 Foraging using the CPFA, as well as the 2-nest, 4-nest, and 8-nest MPFA in random, partially clustered, and clustered resource distributions. There is a significant positive trend in the number of resources with the $\log_2$ of the number of nests in all three distributions ($p = 0.02$, $p = 0.017$, and $p = 0.023$, respectively). 33

3.4 The number of collected resources per minute by the CPFA and MPFA. There is a significant positive trend in the number of resources with the $\log_2$ of the number of nests in the first 5 minutes of all three distributions ($p = 0.04$). 34

3.5 Total time spent avoiding collisions for the CPFA and MPFA in three distributions. The p-values of the log-linear regression between the total collision time and the number of nests are $p = 0.05$, $p = 0.85$ and $p = 0.33$ for random, partially clustered, and clustered, respectively. 35

3.6 The search and travel time per resource for the CPFA and MPFA. Search time increases with the number of nests in the random distribution ($p = 0.01$), but has no trend with the number of nests in the partially clustered and clustered distributions ($p = 0.95$ and $p = 0.85$, respectively). Travel time decreases in all three distributions ($p = 0.016$, $p = 0.013$, and $p = 0.045$, respectively). 36

3.7 The evolved probability of laying pheromone when two resources are found in the resource neighborhood. Medians and quartiles for 12 replicates of evolution are shown for each model. A linear regression ($\log_2$ on the number of nests versus the probability of laying pheromone) shows no trend ($p = 0.204$) in the random distribution, but a statistically significant trend for the partially clustered ($p = 0.006$) and clustered ($p = 0.05$) distributions. 37

3.8 The evolved probability of switching to search. Medians and quartiles for 12 replicates of evolution are shown for each model. A linear regression ($\log_2$ on the number of nests versus the probability of switching to search) shows a statistical significant trend for the random distribution ($p = 0.02$). 38

4.1 The flow chart of an individual robot's behavior in MPFA during an experiment. 48

4.2 The placement of nests and targets in ARGoS. 1024 targets (black points) and 16 robots (larger blue points) are placed in a 15×15 m arena, 4 nests (red circles) are distributed uniformly in the arena. The targets are arranged in a partially clustered distribution. Colored lines indicate pheromone trails with different strength. A small area is magnified to show a robot, colored pheromone waypoints, a large cluster of targets, and a single target. 49

4.3 The average efficiency (targets collected per robot, per minute) for the CPFA ($p = 0.08$) and MPFA ($p = 0.04$) decrease as the swarm size increases. The p value is from the average of collected targets and the $\log_2$ of the swarm size. Results are for 100 replicates. The percentage of improvement is labelled. 54

4.4 The average efficiency (targets collected per robot, per minute) for the CPFA ($p = 0.04$) and MPFA ($p = 0.001$) decrease as the number of targets increases. The p value is from the average of collected targets and the $\log_2$ of the number of targets. The efficiency is always higher for the MPFA. 54

4.5 The average efficiency (collision time per robot, per target) for the CPFA ($p = 0.06$) and MPFA ($p = 0.10$) increase as the swarm size increases. 55

4.6 The average efficiency (collision time per robot, per target) for the CPFA and MPFA as the number of targets increases ($p = 0.03$). . . 56
4.7 The average travel time (per robot, per target) for the CPFA and MPFA decrease as the swarm size increases ($p = 0.04$). 57
4.8 The average travel time (per robot, per target) for the CPFA ($p = 0.001$) and MPFA ($p = 0.03$) as the number of targets increases. . . 57
4.9 The average search time (per robot, per target) for the CPFA and MPFA as the swarm size increases ($p = 0.03$). 58
4.10 The average search time (per robot, per target) for the CPFA and MPFA as the number of targets increases ($p = 0.05$). 59

5.1 The flow chart of an individual robot's behavior following the MPFA during an experiment 72
5.2 A single cycle of uninformed search. Four states of a robot in the cycle are shown. A robot departs from a depot (large red circle), travels to a random location, and switches to searching using an uninformed random walk (dark blue circle). If the robot finds a target pile (largest black square), then it collects one target and delivers it to the closest depot. The robot also has a probability of giving up searching (bright green circle) and returning to the closest depot without finding a target 73
5.3 A single cycle of informed search. Five states of a robot are shown. A robot departs from a depot (large gray circle) and travels to the previous location (dark blue circle), and switches to searching using an informed correlated walk. If it finds a target pile (largest black square), then it collects one target and delivers it to the closest depot (red circle in the lower right). The robot also has a probability of giving up searching (light green circle) and returning to the closest depot without finding a target (red circle) 74
5.4 An example of a dynamically allocated depot using the k-means++ clustering algorithm. The resources (black squares) are classified into four clusters (red ellipses). Depots (dark red solid circles) are placed at the centroids of these clusters 77

5.5 Depot movement in MPFA$_{dynamic}$. A depot (gray circle) is at the centroid c_1 of the sensed resources (dark blue squares) at positions p_1, p_2, and p_3, where w_1, w_2, and w_3 are the number of resources sensed by robots at each position, respectively. After some time, if resources at position p_1 are completely collected by robots, then the pheromone waypoints at p_1 will decay. If, at the same time, w_4 resources are sensed at a new location p_4, then the depot will move to the centroid c_2 of the sensed resources (red circle) at positions p_2, p_3, and p_4 79

5.6 The placement of depots and resources in ARGoS. 384 resources (small points) and 24 robots (middle-sized points) are placed in a 10 × 10 m arena, and 4 depots (large points) are distributed. The resources are unclustered and spread in a uniform random distribution in (a), partially clustered in (b), and clustered into 6 equally-sized piles in (c). The colored rays indicate pheromone waypoints with different strength that eventually evaporate and disappear. A small area is magnified in (c) to show resources, robots, and a depot in the center 86

5.7 Foraging times for CPFA and MPFA swarms of 24 robots in a 10 × 10 m arena. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from MPFA$_{dynamic}$ which is emphasized by red ellipses. The performance of each algorithm is represented by a notched box plot in a different shade, ordered left to right, lightest to darkest in the same order indicated in the legend. The notches indicate the 95% confidence interval of the median so that overlapping ranges of the notches indicate statistically indistinguishable results at the $p = 0.05$ level . 90

5.8 Foraging times for CPFA and MPFA swarms of 24 robots with noise $e = 0.4$ in a 10 × 10 m arena 92

5.9 The search and travel time (per swarm) for the CPFA and MPFAs . 93

5.10 Total time spent (per swarm) avoiding collisions for the CPFA and MPFAs. The boxplot of MPFA$_{dynamic}$ is emphasized by blue ellipses 94

5.11 The foraging time for each swarm for increasing arena sizes. Results are for 100 trials and data for each swarm is shown by the box plot. The lines show the best-fit linear regression 95

5.12 The foraging time for each swarm of 96 robots in 50 × 50 m arena. Results are for 100 replicates for each algorithm. Asterisks indicate a statistically significant difference ($p < 0.001$). The boxplot of MPFA$_{dynamic}$ is emphasized by red ellipses 97

5.13 Foraging times for CPFA swarm of 28 robots and MPFA$_{dynamic}$ swarms of 24 robots in a 10 × 10 m arena. Depots deliver collected resources to the central placed depot when they have 24 resources. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from MPFA$_{dynamic}$ 98
5.14 The physical robot on which components of the CPFA have been implemented 104
5.15 A mobile depot with blue cover and four foraging robots simulated in Gazebo 105
5.16 A swarm of 6 robots (3 shown) implementing central place foraging in a 23 x 23 m arena 106

6.1 Paths of depots in explosion (MPFA) and hierarchical (MPFA$_{\mathrm{T}}$) transportation networks. Each small square is a service region that contains 4 searching robots. 118
6.2 Number of collected resources vs. arena size (panel a) and time spent avoiding collisions per robot vs. arena size (panel b) in Set I. The low foraging of the CPFA (panel a) is explained in part by the long time spent avoiding collision. 123
6.3 The number of resources collected in 30 minutes vs arena size is in experiment Sets I, II, and III. Both axes are on a log scale. Each dashed line indicates the $\log_2$ linear regression of the mean of collected resources with the $\log_2$ of arena area. The solid black line indicates the predicated slope in each configuration. The * algorithms (hollow symbols) that lack collisions among depots demonstrate that foraging under ideal conditions is close to theoretical predictions. $p < 0.001$ in all experiments. In panel (a), $r^2 = 0.1$ for the CPFA, $r^2 = 0.95$ for the MPFA, $r^2 = 0.85$ for the MPFA$_{\mathrm{T}}$, and $r^2 = 0.99$ for the MPFA* and MPFA$^*_{\mathrm{T}}$. In panel (b), $r^2 = 0.66$ in the MPFA$_{\mathrm{T}}$ and $r^2 > 0.98$ in other algorithms. In Set III, $r^2 = 0.96$ for the MPFA and $r^2 > 0.99$ for other algorithms. 124
6.4 The collision time per robot, per minute for the CPFA and MPFA with 4 collection zones. The data is from Fig. 4.3 in Section 4. The arena size is 15 × 15 m. Results are for 100 replicates. 127

6.5 The number of depot trips required to deliver resources in the MPFA and the $\mathrm{MPFA_T}$. The colored squares indicate the locations of collection zones. The numbers in collection zones indicate the average depot trips required to deliver resources. Each experiment is replicated 60 times. The data is from the third set of experiments in Prediction I. There are 16 collection zones in the MPFA and 20 collection zones in the $\mathrm{MPFA_T}$. The central collection zone is not shown since no depot trip is required on it. 128

6.6 The scenarios of the MPFA in ARGoS simulation. The configuration is the third one in Set I for Prediction I. 64 searching robots (with green LEDs), 48 depots (with red LEDs), 256 uniformly distributed resources (black dots), and 17 collection zones (green circles) are in a 16 × 16 m arena. Blue lines indicate paths for delivering resources. Yellow dots indicate locations where robots collected resources in their last trip. Robots remember those locations and they may return to those locations using a process called *site fidelity*. 129

6.7 The scenarios of the $\mathrm{MPFA_T}$ in ARGoS simulation. The configuration is the third one in Set I for Prediction I. 21 collection zones are in a 16 × 16 m arena. 130

6.8 The collision time per robot, per minute for the MPFA and $\mathrm{MPFA_T}$ in Prediction III. The robot density is nearly constant (0.03) in all arenas. The data is from Fig. 6.3(c). Results are for 100 replicates. . 132

6.9 The number of depot trips required to deliver resources in the MPFA and the $\mathrm{MPFA_T}$. The configuration is the second one in Set III for Prediction III. 17 collection zones are in the MPFA and 21 collection zones are in the $\mathrm{MPFA_T}$. The arena size is 20 × 20 m. 132

6.10 The scenarios of the MPFA in ARGoS simulation. The configuration is the second one in Set III for Prediction III. 17 collection zones are in a 20 × 20 m arena. 133

6.11 The scenarios of the $\mathrm{MPFA_T}$ in ARGoS simulation. The configuration is the second one in Set III for Prediction III. 21 collection zones are in a 20 × 20 m arena. 134

6.12 The number of depot trips required to deliver resources without collisions in the MPFA^* and the $\mathrm{MPFA}^*_\mathrm{T}$. 135

7.1 Robot states in the CPFA and DDSA. 145

7.2 The architecture of the CPFA and DDSA in ROS. 148

7.3 Simulated and Physical experiments with 4 robots, 128 cubes, 4 obstacles and one central collection zone. Configuration 2 is shown, Target cluster sizes are described in Table 1, obstacles are placed 3 to 5 m from the center, and the exact location of each obstacle, target and target cluster is chosen at random. 150
7.4 Foraging performance of the DDSA and CPFA with and without obstacles, for 30 trials in simulation, and 15 trials in physical experiments using configurations 1 and 2 (shown in Fig. 7.3). 152
7.5 Foraging performance with cubes lined to the edges and clustered in corners. 152
7.6 Overall foraging performance with all experiments. 153
7.7 Odometry traces of 4 robots in simulation (left column) and physical experiments (right column). Each robot path is a different colored line. Obstacles are not shown, but the empty areas in (c) and (d) imply their location. 154

8.1 A physical and simulated depot. (a) A physical depot with 8 cubes on its carriage. The depot is a Swarmie robot equipped with a plexiglass plate. A stainless steel wire connects the shaft of the plate and a servo motor. The motor controls the plate to dump cubes. The robot dimensions are 34 × 25 × 22 cm and the carriage dimensions are 31 × 24 × 3 cm. (b) A simulated depot in Gazebo. Its size is identical to the physical depot. 159
8.2 A scenario of multiple depots in Gazebo. Four depots with blue covers are distributed uniformly in a 14 × 14 meter arena. Four searching robots are searching for cubes in the arena. The collection zone is located in the center. 160

List of Tables

3.1 Experimental configuration in ARGoS 31

4.1 Experimental configuration in ARGoS 52

5.1 Parameters for robot controllers 79
5.2 Experimental configuration 85

6.1 Similarities between cardiovascular systems and robot swarms 115
6.2 Experimental Setup 121

7.1 Experimental Setup and Replicates 150

Glossary

April Tag
These visual codes are used as target markers. April tags provide information to the robot about target distance and orientation in space.

ARGoS
A swarm robot simulator. ARGoS is similar to Gazebo but runs much faster at the expense of realism. In this work we use the 2D physics engine dyn2d.

CDF
Cumulative distribution function. The CDF for a value x is the sum of the PDF for all values less than or equal to x. We fit models to empirical CDFs rather than PDFs.

CPF
Central place foraging. A common task performed by groups of organisms is the discovery and transportation of food items to a central location. Transportation of resources, or other materials, to a depot is also of interest to developers of robot swarms.

CPFA
Central place foraging algorithm. A desert harvester ant inspired algorithm for the collection of resources. (Hecker and Moses, 2015)

DDSA The deterministic spiral algorithm. A square spiral search strategy for swarms of robots.

EKF Extended Kalman filter. A method of integrating sensor data that uses a dynamic noise covariance matrix to weight inputs.

GA Genetic algorithm. An evolutionary optimization technique in which a population of solutions navigate an optimality landscape using biologically inspired mechanisms such as gene crossover and mutation.

Gazebo An environment closely tied to ROS that allows the simulation of robots. In this work we used Gazebo 2.0 and the Open Dynamics Physics Engine (ODE).

GPS Global positioning system.

IMU Inertial measurement unit.

MPFA Multiple-place foraging algorithm. An explosion transportation network is built on it in later version.

MPFA_{static} The MPFA with static nests. It is another name of the MPFA.

$\text{MPFA}_{dynamic}$ The MPFA with dynamic depots.

$\text{MPFA}_{global\_static}$ The MPFA with depots that have global information about target locations using static depots.

$\text{MPFA}_{global\_dynamic}$ The MPFA with depots that have global information about target locations using dynamic depots.

MPFA_{T} The MPFA with the hierarchical branching transportation network.

MPFA* It is the MPFA in the idealized environment where it does not have collisions in transportation.

MPFA$^*_\mathrm{T}$ It is the MPFA$_\mathrm{T}$ in the idealized environment where it does not have collisions in transportation.

Reality gap A common problem in robotics is the development of robot algorithms in simulation that do not translate well to real robots. Simulations enable the evaluation of algorithms many times faster than would be possible with real robots, but cannot completely reproduce the complexity of real environments interacting with embodied robots. A robot interacting with its environment is in itself a complex system with potentially non-linear feedback between actuators, environment, and sensors. This makes modelling difficult, especially in the case of swarms of robots which have the added complication of robot-robot interactions.

ROS The Robot Operating System. This is the software platform produced by the Open Robotics Software Foundation that we use to develop the software for the Swarmathon robots. In this work we use the Indigo Igloo release of ROS.

Chapter 1

Introduction

Searching is a universal process in nature and engineered systems. Ant colonies search for food, immune cells search for pathogens, computer programs search for results, and robots search for targets. In this research, we focused on the foraging behavior of robot swarms. Foraging is the behavior of searching for resources and transporting them to a specific collection zone (or nest). Effective collective foraging requires coordination, navigation, and communication and is therefore a useful abstraction of many complex, real-world applications such as humanitarian de-mining, search and rescue, intrusion tracking, construction, transportation, agricultural harvesting, collection of hazardous materials, and space exploration (Winfield, 2009b; Gro and Dorigo, 2009; Yun and Rus, 2014; Bac et al., 2014; Fink et al., 2005; Gazi and Passino, 2004; Brambilla et al., 2013).

However, searching for multiple resources in a large arena is challenging, particularly the robots we consider here with limited sensing and communication range. Without the global information about the environment, it is not feasible to explore the entire area in a limited time. A more efficient foraging algorithm will allow robots to search and collect resources in the same amount of time. In this research, my goal is to design an efficient and scalable foraging algorithm that is equally effective in swarms ranging from tens to thousands of robots in large arenas. We get the inspiration of the scaling theory in biology and then we design a scale-invariant transportation model for robot swarms. We demonstrate the viability of the biological models and then focus on the design of the bio-inspired engineering model. Our work shows that the bio-inspired transportation model brings an efficient and scalable solution to our foraging robot swarms.

1.1 The Scalability of Foraging Robot Swarms

Scalability of robot swarms has gained recent interest (Bonabeau et al., 1999; Kennedy and Eberhart, 2001; Şahin, 2005; Barca and Sekercioglu, 2013; Khaluf et al., 2017). A scalable foraging system should be effective in swarms ranging from tens to thousands of robots without reducing per robot foraging times. In central place foraging, robots gather dispersed resources from a foraging arena and consolidate them in a single centrally-placed collection zone that robots depart from and return to in order to deposit resources (Liu et al., 2007; Hecker and Moses, 2015; Castello et al., 2016). However, two major problems limit scalability and produce *diminishing returns* (Brue, 1993). First, large swarms with many robots produce more inter-robot collisions both during the search process and during the return of resources to a relatively small collection zone. Second, large foraging arenas require, on average, that robots travel further distances (requiring more time) to find resources and transport them to the central collection zone. When foraging in large areas, for example, collecting resources on the surface of Mars, or an ocean search and rescue operation, the search area can extend many kilometers, necessitating that robots travel very long distances.

1.2 The Multiple-Place Foraging Algorithm

We proposed the Multiple-Place Foraging Algorithm (MPFA) with multiple collection zones that robots depart from and return to. Robots always return to closest collection zones for delivering resources. Therefore, robots have shorter travel distances and fewer collisions.

The MPFA is inspired by behaviour observed in biology. For example, spider monkeys have been characterized as multiple central place foragers (Chapman et al., 1989). The monkeys select a sleeping site close to current feeding areas, a strategy that entails the lowest travel costs. A study by Tindo et al (Tindo et al., 2008) showed that wasps living in multiple nests have greater survival and increased productivity. The polydomous colonies of invasive Argentine ants are comprised of multiple nests spanning hundreds of square meters (Flanagan et al., 2013; Suarez et al., 2001; Carpintero et al., 2005). Multiple-place foraging also resembles global courier and delivery services, which use many distributed stores to collect and deliver packages efficiently.

In the MPFA, multiple collection zones are distributed in a search arena uniformly. Robots are evenly distributed around collection zones. Robots start from a random collection zone, but return to the closest collection zone to their positions after finding a resource (see Fig. 1.1). They search globally – they can travel in the entire arena. Robots have *priori* knowledge of the locations of collection zones.

An individual robot may remember the location of a previously found resource and repeatedly return to the same location using a process called *site fidelity* (Beverly et al., 2009). Robots can also communicate using pheromones (Sumpter and Beekman, 2003; Jackson et al., 2007) which are simulated as artificial waypoints (Campo et al., 2010) to recruit robots to known clusters of resources. Each pheromone trail starts at the collection zone and ends at a waypoint specifying a location in the arena.

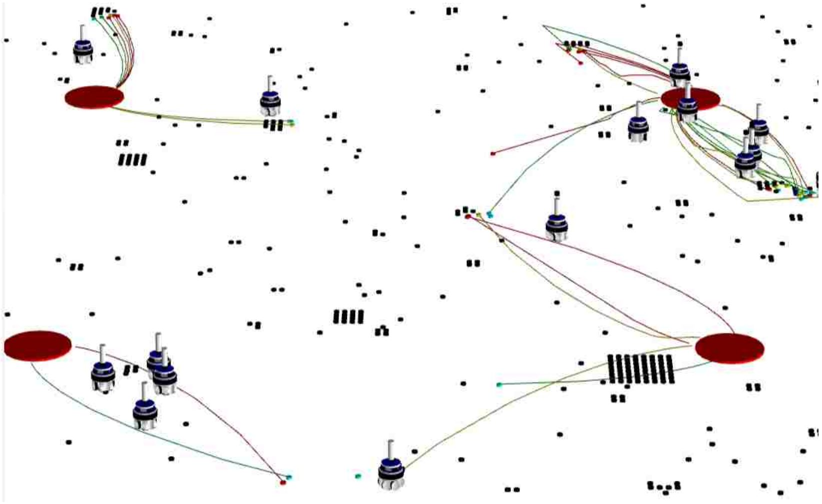


Figure 1.1: The MPFA running in ARGoS, front view. Resources are shown as black dots arranged in a partially clustered distribution. Red circles indicate uniform distributed collection zones. Colored rays indicate pheromone waypoints with different strength (green indicates high and red indicates low).

A set of real-valued parameters specifying the individual robot controllers is evolved by a Genetic Algorithm (GA) to optimize the collective behavior of the swarm in the multi-physics robot simulator Autonomous Robots Go Swarming (ARGoS) (Pinciroli et al., 2012). Every robot in the swarm uses the same controller. We observe how the number of collection zones affects swarm foraging performance (i.e., the number of resources collected). The MPFA produce higher foraging rates and lower average travel time compared to the Central-Place Foraging Algorithm (CPFA).

However, the travel distance is still large if the arena size is large. In other work (Lu et al., 2019b), we introduce the dynamic depots into the MPFA (MPFA$_{dynamic}$) instead of static collection zones. Dynamic depots move to the centroid of recently collected resources, minimizing transport times and making the MPFA$_{dynamic}$ more adaptable to patchy and heterogeneous distributions of resources (Ritchie, 2009). The MPFA$_{dynamic}$ improves scalability from the original MPFA, but the time to transport resources from dispersed collection zones to a single location still results in diminishing returns.

We propose the MPFA$_{\mathrm{T}}$ with a bio-inspired hierarchical transportation network solves this problem. In this scale-invariant design, the per-robot foraging time is invariant with respect to arena size and swarm size. The transportation network draws inspiration from biological scaling theory that describes the scaling consequences of transporting resources from a central heart to dispersed cells via the animal cardiovascular network (West et al., 1997; Banavar et al., 2010) (see Fig. 1.2). The cost of large size is that resources take longer to transport through the system, which ultimately slows the cells of larger animals. Thus, physiological rates (i.e. heart rate, growth rate, and reproductive rate) are systematically slower, and lifespans and gestation times are systematically longer, in large vs. small animals.

We show that transportation costs in foraging robots are constrained by the same principles as the vascular system in plants and animals. We derive scaling relationships for a 2D foraging area (rather than a 3D animal volume). We use this scaling law to predict the transportation infrastructure required to maintain constant

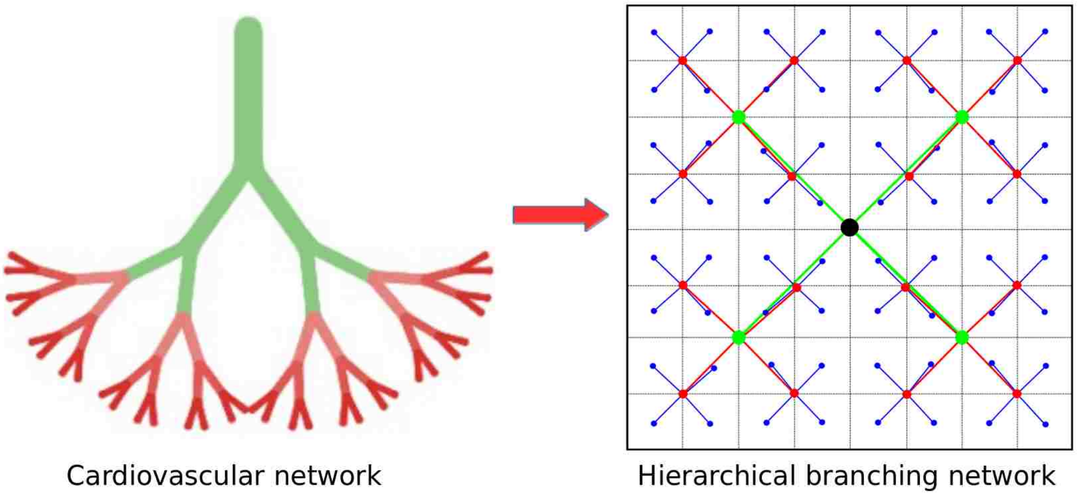


Figure 1.2: Illustration of the inspiration from cardiovascular networks in biology to hierarchical branching network MPFA$_\mathrm{T}$ in robot swarms (left figure replicated from (Moses et al., 2016)).

per-robot foraging rate with increasing swarm and arena size. We demonstrated that the foraging performance per robot decreases with robot density increase in the CPFA and the MPFA. In the work of MPFAT, we demonstrate that even within a constant robot density, collision rates still increase in the CPFA and the MPFA, but it does not increase in the MPFA$_\mathrm{T}$. As fewer collisions in the MPFA$_\mathrm{T}$, robots have less chance to interfere with other robots and they are more robust.

We then simulate foraging using a hierarchical transportation network (MPFA$_\mathrm{T}$) composed of mobile depots with carrying capacities determined by the scaling theory. Our simulations show that this design overcomes scaling constraints resulting in nearly scale-invariant foraging. We scale the swarm size up to thousands of robots in arenas that are thousands of square meters in area. We test all of our algorithms in the ARGoS simulator using foot-bots as a model robot.

1.3 From Simulation to Physical Robots

While simulations are useful for initial evaluations of the viability of algorithms, they are insufficient for the ultimate goal of predicting how algorithms will perform when physical robots interact in the unpredictable conditions of environments they are placed in. It is critical to evaluate collective algorithms in physical robots (Ligot and Birattari, 2018) because there is a "reality gap" between simulation and physical robots. It is not feasible to simulate all aspects of a physical environment (Frigg and Hartmann, 2018), and foraging performance can be altered by variable conditions and by sensor and actuator noise that affect localization, object retrieval, and collision avoidance. All of these components can alter the performance of algorithms real robotic experiments compared to simulations (Jakobi et al., 1995; Mouret et al., 2013).

We have our robot simulation in Gazebo (Koenig and Howard, 2004) which includes many realistic physical features such as localization, navigation, sensing, object pickup and drop off, and collision avoidance. In the CPFA, robots randomly search objects. Eventually, robots detect the distribution of objects in the arena. They remember and share the locations of detected objects with other robots. Then, the robots use the information to guide their search. Some locations are searched many times, but some locations may not be searched at all. In the DDSA, the search paths are pre-planned. Robots search for objects on their own paths. This deterministic search strategy guarantees a complete search coverage of the arena. Each location is searched once.

Currently, we implemented the CPFA and Distributed Deterministic Spiral Algorithm (DDSA) in the Robot Operating System (ROS) which can run in Gazebo simulation and physical robots called “Swarmies” (see Fig. 1.3) directly. The Swarmie robot is designed for the NASA Swarmathon competition (Secor, 2016; Ackerman et al., 2018). Each Swarmie robot is equipped with a front web camera, three pairs of ultrasound range sensors, and a gripper for target pickup and drop off. The Swarmies physically pick up and drop off targets and operate outdoors under variable ground and light conditions. Complete build instructions for the Swarmie robot are publicly available[1].

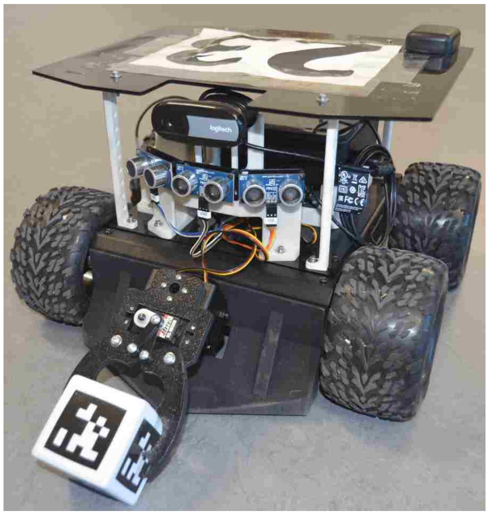


Figure 1.3: A physical Swarmie robot with a cube on its gripper.

Programs are directly loaded onto the Swarmie onboard Linux computer for physical robot experiments. In a physical experiment, we run the same program on a host computer first. The host computer connects to robots through a wireless network. Robots receive messages from the host computer. The GUI of the software acts as a communicator between users and robots. The updates of robots are sent back to the host computer.

[1] https://github.com/BCLab-UNM/Swarmathon-Robot

In physical robots, pheromone waypoints are stored in the host computer. The communication is in a centralized manner which is different to other work in artificial pheromone (Garnier et al., 2007; Campo et al., 2010; Schmickl and Crailsheim, 2008). In the future implementation of the MPFA, we will update the centralized model to be decentralized. In the decentralized model, robots still connect to the host computer initially. Then, the host computer still can monitor robots, but robots can run without the host computer.

We designed a set of experiments that we replicated in a Gazebo simulation and in physical robots with various placements of resources and obstacles. Our conclusion from comparing the two algorithms that the deterministic DDSA is more efficient than the CPFA in the simulation. However, the stochastic CPFA marginally outperforms the DDSA in physical experiments. The performance of the DDSA is more degraded by variable and unexpected conditions in the physical world, suggesting that the CPFA is more tolerant of real-world conditions.

Finally, we demonstrate how we can use our existing Gazebo simulation and Swarmie hardware to implement the MPFA with transportation networks into physical robots. The implementation of of the CPFA and DDSA in dozens of replicated experiments with physical hardware (Lu et al., 2019a) suggests that the approach we outline here is feasible in physical robots.

1.4 Organization and Contributions

Our main contributions are divided into six chapters and summarized below. They are published in peer-reviewed conference proceedings and journals, with the exception of Chapter 6 which is in revision to the resubmitted.

In Chapter 3, we propose the MPFA with multiple collection zones for robot swarms. We use a GA to optimize collective foraging behaviors in ARGoS. We simulate different numbers of collection zones: 1 (replicating the CPFA), 2, 4, and 8. We discovered more collection zones produce better foraging performance. In Chapter 4, we deploy 4 nests (or collection zones) uniformly in the same size of search arena. We compare the foraging performance of the MPFA to the CPFA when increases the number of robots and the number of resources. In all cases the MPFA increases foraging rates compared to the CPFA by reducing travel time, search time, and collision time. These results indicate a new direction of improving foraging performance with distributed multiple collection zones.

In Chapter 5, we propose the multiple-place foraging algorithm with dynamic depots (MPFA$_{dynamic}$). Depots are special robots which are able to carry multiple resources. Depots move to new locations based on the mean positions of the remaining resources sensed by the robots. We compare the performance of the MPFA$_{dynamic}$ with MPFA and CPFA. The MPFA$_{dynamic}$ outperforms the MPFA and CPFA. It is also more scalable than other algorithms in a very large arena (50×50 m) with 96 robots.

In Chapter 6, we propose a bio-inspired hierarchical transportation network to improve the scalability of the MPFA. The transportation network draws inspiration from the cardiovascular network in biology. In this design, the per-robot foraging time is invariant with respect to arena size and swarm size. We derive scaling relationships for a 2D foraging area. We use this scaling law to predict the transportation infrastructure required to maintain constant per-robot foraging rate with increasing swarm and arena size. Our simulations show that this design overcomes scaling constraints resulting in nearly scale-invariant foraging. We scale the swarm size up to thousands of robots in arenas that are thousands of square meters in area.

In Chapter 7, we compare the deterministic DDSA and the stochastic CPFA in simulations and in physical robots. The conclusion we draw from comparing the two algorithms is: the DDSA outperforms the CPFA in the simulation. However, the CPFA marginally outperforms the DDSA in physical experiments. It suggests that the CPFA is more tolerant of real-world conditions. This work emphasizes the importance of evaluating and comparing algorithms in physical experiments. The results presented here also provide benchmarks for comparison for other foraging algorithms (e.g. MPFA) in physical robots.

In Chapter 8, we present the design of physical depots (delivering robots). We demonstrate the feasibility of shifting from simulation to physical robots. Our goal is to have depots delivering multiple objects and dumping them into collection zones autonomously. We will implement the MPFA with the hierarchical transportation network in physical robots. In the future work, we will compare it with the DDSA and CPFA in physical robots.

Chapter 2

Background

2.1 Stochastic Central-Place Foraging

When robots depart from a centrally-placed collection zone to search for and transport resources back to the collection zone, this process is called central-place foraging. Central-place foraging is a canonical collective task commonly studied in swarm robotics (Şahin, 2005; Brambilla et al., 2013).

Prior work (Hecker and Moses, 2015) introduced the CPFA, which was designed to emulate seed-harvester ant behaviors governing memory, communication, and movement (see Fig. 2.1). It mimics a repertoire of foraging behaviors used by desert seed-harvester ants to search for resources that are distributed in a variety of ways (Hecker et al., 2013; Gordon and Kulig, 1996). These ants are restricted to foraging in short time windows during which not all available resources can be collected. Hecker and Moses formalized and implemented the behaviors of the CPFA (Hecker and Moses, 2015; P. Hecker et al., 2012; Hecker and Moses, 2013; Hecker et al., 2015) based on Flanagan and Letendre's ant field studies (Flanagan et al., 2012; Letendre and Moses, 2013). The foraging strategies were evolved by a GA that were tolerant of real-world sensing and navigation error, flexible for a variety of target distributions, and scalable to large swarm sizes.

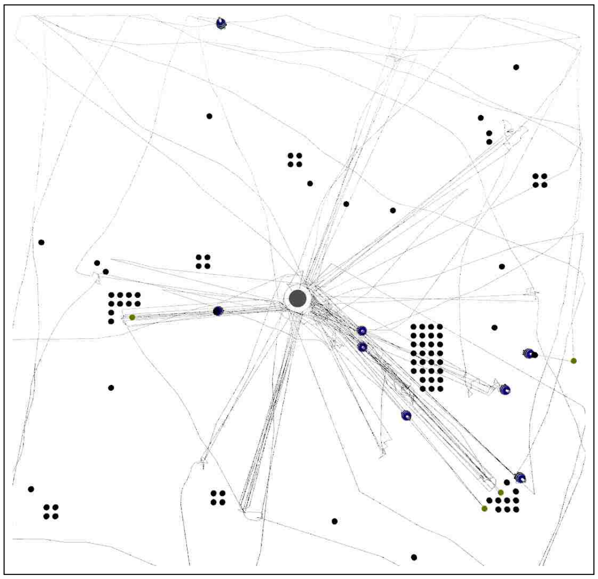


Figure 2.1: The CPFA running in ARGoS, overhead view. The circle in the center indicates the collection zone. The partially clustered distribution of resources are shown as black dots, robots blue larger dots, lines indicate the paths taken by robots during the experiment.

In the CPFA, robots initially disperse from the central collection zone to random locations, followed by a search behavior (Fewell, 1990) in which an uninformed correlated random walk is used to locate resources (Crist and MacMahon, 1991). Robots pick up one resource at a time and return to the collection zone when they either find a resource or give up searching.

When a robot locates and finds a resource, it stores a count c of sensed resources in neighborhood of the found resource. This count c represents the density of resources in the local area. In ARGoS simulation, the count is from the 8-cell neighborhood of the found resource which is the center of 3×3 cells. In physical robots, it is the number of cubes detected in the camera of a robot by spinning in a circle. An individual robot may remember the location of a previously found resource and repeatedly return to the same location using a process called *site fidelity* (Beverly et al., 2009). Robots can also communicate using pheromone which are simulated as artificial waypoints to recruit robots. Waypoints are maintained in lists in which pheromone strength of each waypoint decreases exponentially over time. We simulate the artificial evaporation process of waypoints which are removed once their values drop below a threshold.

Our initial implementation requires that all robots know the location of the collection zone. Robots use the detected resources to decide whether to create a pheromone waypoint which adds the location and the strength to a list, mimicking ant pheromone trails. Waypoints are communicated (and can be followed) only when robots return to the collection zone.

In physical robots, pheromone waypoints are stored in the host computer. The communication is in a centralized manner which is different to other work in artificial pheromone (Garnier et al., 2007; Campo et al., 2010; Schmickl and Crailsheim, 2008). In the future implementation of the MPFA, we will update the centralized model to be decentralized. In the decentralized model, robots still connect to the host computer initially. Then, the host computer still can monitor robots, but robots can run without the host computer.

The robot uses the density count c to decide whether to use site fidelity in the next round of foraging, lay a pheromone waypoint, or follow the pheromone waypoint. If a robot returns to a previously found resource area, it searches using an informed correlated random walk that searches more thoroughly than robots searching randomly selected locations.

2.2 Distributed Deterministic Spiral Search

The Distributed Deterministic Spiral Algorithm (DDSA), a deterministic search for central-place foraging is designed to collect resources in robot swarms (Fricke et al., 2016)). It gets inspiration from many previous studies (Bentley et al., 1980; Baeza-yates et al., 1993; Isbell, 1957). Generally, these studies take a geometric approach which exploits the optimality of spiral search demonstrated for single agents generalized to a swarm of robots by having robots move to a uniform random location before beginning the spiral. Isbell (Isbell, 1957) described a target detection search pattern for individual ships in which it performs a continuous space-filling spiral. A hidden expanded spiral search pattern is discovered in foraging desert ants of the genus *Cataglyphis* (Müller and Wehner, 1994). Burlington and Dudek (Burlington and Dudek, 1999) extend the single searcher spiral search pattern to a complex environment. A square search pattern carried out by a single helicopter is described in (Ryan and Hedrick, 2005).

In the DDSA, an interlocking square spiral paths are computed for robot swarms. Robots start near the central collection zone and search for resources by following the pre-planned paths. This deterministic search strategy is different from the stochastic search CPFA. When operating without error, noise, or collisions, the DDSA guarantees that robots will find the nearest resources first which minimizes transport cost. This provides complete coverage of an area while minimizing repeated searches of the same location (see Fig. 2.2).

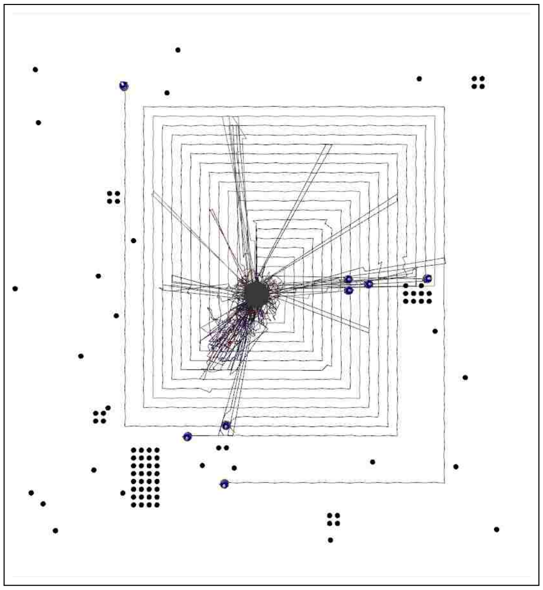


Figure 2.2: The DDSA running in ARGoS. The robots search on pre-planned spiral search paths beginning at a central collection zone. Resources are shown as black dots arranged in a partially clustered distribution. Robots are marked with blue larger dots. Colored lines are the paths of robots.

In each subsequent foraging trip, the robot returns directly to the last location where it found a target (effectively implementing site fidelity for every foraging trip) where it resumes its spiral search. This relatively simple algorithm guarantees that the closest resources are found first, and due to site fidelity a robot will repeatedly return to a location so that it efficiently collects clusters of resources.

2.3 Task Partitioning

Task partitioning can be an approach to improve location accuracy, mitigate collisions, and reduce travel distances to improve scalability implicitly. Pini et al. (Pini et al., 2014) demonstrated that a static partitioning strategy can provide a scalable and robust foraging robot swarm. Buchanan et al. (Buchanan et al., 2016a) improved the scalability of the robot swarms using a dynamic partitioning strategy which mitigates dead reckoning error. The work in (Ferrante et al., 2015) describes a leafcutter ant inspired foraging algorithm. The robot swarm achieves maximum foraging performance by dividing foraging and delivering tasks automatically using a nature-inspired evolutionary method known as Grammatical Evolution (Ferrante et al., 2013).

We get inspiration from the task partitioning to introduce mobile depots for the transportation task, separate from searching robots that search for resources and deliver them to a local depot. When robot swarms in a large arena, robots take the advantage of task partitioning. We can allocated multiple collection zones in the arena. Instead of traveling a long distance to deliver resources to the central collection zone, searching robots can focus on searching and only deliver collected resources to their home collection zones. Mobile depots are designed to travel from their home collection zones to destination collection zones for delivering resources. Mobile depots distributed in those destination collection zones deliver resources to their destination collection zones in next level. Every mobile depot only travel on the path from its home collection zone to its destination collection zone. Eventually, collected resources are delivered to the central collection zone. It produces less collisions and increases foraging performance. Therefore, the task partitioning improves the scalability of foraging robot swarms.

2.4 Existing Simulators and Physical Robot Platforms

Though swarm robot foraging has been studied for decades, replicated experimental analyses that compare different algorithms in simulation and in real robots are rare, particularly in outdoor environments (Winfield, 2009a; Brambilla et al., 2013). For example, Many task partitioning and foraging algorithms have been simulated in the Stage multi-robot simulator (Gerkey et al., 2003; Liu et al., 2007; Castello et al., 2016), the ARGoS multi-robot simulator (Ferrante et al., 2015; Pini et al., 2014; Buchanan et al., 2016b) and Microsoft(R) Robotics Developer Studio (MRDS) (Hoff et al., 2010). Physical foraging experiments have been conducted with "foot-bots" equipped with grippers, IR sensors, and cameras for foraging tasks in (Pini et al., 2014; Buchanan et al., 2016b) and custom platforms like MinDART (Rybski et al., 2008). However, they evaluate robots in laboratory environments.

In practice, many complex physical experiments with swarm robots require human support, e.g. (Rosenfeld et al., 2017). Many studies do not have simulation of some aspect completely. For example, the Robotarium provides a testbed for remotely accessible physical robots (Pickem et al., 2017), but localization is governed by an overhead camera. Other studies simulate simplified physical pickup and drop-off of objects. For example, (Brutschy et al., 2015; Castello et al., 2016) uses a group of e-puck robots and our prior work (Hecker and Moses, 2015) used iAnt robots which detect targets but do not physically pick them up. Kilobots can operate autonomously to push items, but they have relatively limited mobility and only operate in controlled laboratory environments (Rubenstein et al., 2012; Jones et al., 2018). Collaborative warehouse robots may require buried guide-wires or visual markers to navigate (Enright and Wurman, 2011). The Swarmanoid project demonstrates an innovative heterogeneous physical swarm robotics system whose robots collaborate to solve a complex object retrieval task (Dorigo et al., 2013), but it was designed as a demonstration of swarm capabilities, not to test algorithms in a physical environment.

We conduct automated, replicated experiments to test autonomous collective foraging in outdoor environment with variable ground and light conditions. These factors are important sources of error and noise in our experiments. Our physical robots can pick up and drop off resources with any additional support. The limitations of our physical robots are that they use GPS, a global (but still noisy) signal, to mitigate the localization problem. We also occasionally use human intervention to prevent robots from leaving the foraging arena.

Chapter 3

The MPFA: A Multiple-Place Foraging Algorithm for Robot Swarms

3.1 Publication Notes

Citation: Qi Lu, Joshua P. Hecker, Melanie E. Moses. The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3815-3821, December, 2016.

Received: March 1, 2016

Accepted: July 1, 2016

Published: December 1, 2016

3.2 Abstract

Finding and retrieving resources in unmapped environments is an important and difficult challenge for robot swarms. Central-place foraging algorithms can be tuned to produce efficient collective strategies for different resource distributions. However, efficiency decreases as swarm size scales up: larger swarms produce more inter-robot collisions and increase competition for resources. We propose a novel extension to central-place foraging in which multiple nests are distributed in the environment. In this multiple-place foraging algorithm, robots depart from a home nest but always return to the nest closest to them. We simulate robot swarms that mimic foraging ants using the multiple-place strategy, employing a genetic algorithm to optimize their behavior in the robot simulator ARGoS. Experiments show that multiple nests produce higher foraging rates and lower average travel time compared to central-place foraging for three different resource distributions. Time spent avoiding robot-robot collisions is not always reduced as was expected, primarily because the use of pheromone-like waypoints leads to more collisions when robots forage for clustered resources. These results demonstrate the importance of careful design in order to create efficient multiple collection points to mitigate the central-place bottleneck for foraging robot swarms.

3.3 Introduction

Swarm robotics draws inspiration from biology to coordinate large numbers of relatively simple physically embodied agents. Biological studies have revealed self-organized coordination mechanisms in social insects which can be effectively implemented in swarm robotics systems (Camazine et al., 2001; Şahin, 2005; Şahin et al., 2008).

Multiple robots can be organized to collectively accomplish tasks that a single robot cannot easily complete. Swarm robotics researchers aim to design robust, scalable, and flexible collective behaviors for multiple autonomous robots (Şahin, 2005; Brambilla et al., 2013). Simple rules and local interactions among individual robots result in desired collective swarm behavior without centralized control. Such collective behaviors could be combined to tackle complex real-world applications, e.g. collective foraging (Liu, 2008; Liu and Winfield, 2010) and construction (Werfel et al., 2014).

Foraging robots must retrieve objects from an environment and bring them back to a collection point, or nest. Effective collective foraging requires coordination, navigation and communication and is therefore a useful abstraction of many complex, real-world applications such as humanitarian demining, search and rescue, intrusion tracking, and collection of hazardous materials and natural resources (Brambilla et al., 2013; Winfield, 2009a). In particular, foraging is commonly used as a testbed for collective exploration, collective transportation, and collective decision-making (Brambilla et al., 2013; Gazi and Passino, 2004).

The central-place foraging algorithm (CPFA) (Hecker and Moses, 2015) uses a centrally-placed nest which robots depart from and return to as they collect resources. Due to crowding, collision avoidance increases with the number of robots. Therefore, one central nest cannot serve a large number of robots efficiently. Additionally, resources that are located far away from the central nest impose long travel times.

The multiple-place foraging algorithm is inspired by behaviour observed in biology. For example, the polydomous colonies of Argentine ants are comprised of multiple nests spanning hundreds of square meters (Flanagan et al., 2013). Spider monkeys have been characterized as multiple central place foragers (Chapman et al., 1989). The monkeys select a sleeping site close to current feeding areas, a strategy that entails the lowest travel costs. A study by Tindo et al. (Tindo et al., 2008) showed that wasps living in multiple nests have greater survival and increased productivity. Multiple-place foraging also resembles global courier and delivery services, which use many distributed stores to collect and deliver packages efficiently.

In this work, we propose a multiple-place foraging algorithm (MPFA) with multiple nests that robots depart from and return to. We use a genetic algorithm (GA) to evolve collective foraging behaviors in the multi-physics robot simulator Autonomous Robots Go Swarming (ARGoS) (Pinciroli et al., 2012). A set of real-valued parameters specifying the individual robot controllers is evolved to optimize the collective behavior of the swarm. Every robot in the swarm uses the same controller. We simulate different numbers of nests: 1 (replicating the CPFA), 2, 4, and 8. We test how quickly resources are collected from random, partially clustered and fully clustered resource distributions. We observe how the number of nests affects swarm foraging performance (i.e., the number of resources collected), specifically:

- Collision time: The time required to detect and avoid collisions with other robots.

- Search time: The time that a robot spends searching for resources.

- Travel time: The time that a robot spends traveling to and from a nest when collecting resources.

We show that the genetic algorithm tunes the MPFA differently depending on the resource distribution. In all cases the MPFA increases foraging rates compared to the

CPFA by reducing travel time. However, for some resource distributions the MPFA increases search time and in others it increases collisions. These results indicate complex tradeoffs that must be balanced in order to maximize foraging rates given multiple collection points.

The remainder of this chapter is organized as follows. The foundation of the CPFA is introduced in Section 3.4. The design of the MPFA, and the configuration of the MPFA in ARGoS, are provided in Section 3.5. Section 3.6 shows the experimental results and explains them based on trends in MPFA parameters, and Section 3.7 discusses the results.

3.4 Background: The CPFA

The central-place foraging algorithm (CPFA) mimics a repertoire of foraging behaviors used by desert seed-harvester ants to search for resources that are distributed in a variety of ways (Hecker et al., 2013; Gordon and Kulig, 1996). These ants are restricted to foraging in short time windows during which not all available resources can be collected; thus, the CPFA is designed to collect many resources quickly, but not to optimally collect all resources. Hecker and Moses formalized and implemented the behaviors of the CPFA (Hecker and Moses, 2015; P. Hecker et al., 2012; Hecker and Moses, 2013; Hecker et al., 2015) based on Flanagan and Letendre's ant field studies (Flanagan et al., 2012; Letendre and Moses, 2013).

In the CPFA, robots initially disperse using travel behavior from the central nest to random locations, followed by a search behavior (Fewell, 1990) in which an uninformed correlated random walk (Crist and MacMahon, 1991) is used to locate resources (see Fig. 3.1a). Robots pick up one resource at a time and return to the nest when they either find a resource or give up searching.

When a robot locates and finds a resource, it stores a count c of sensed resources in the 8-cell neighborhood of the found resource which is the center of 3×3 cells. This count c represents the density of resources in the local region. An individual robot may remember the location of a previously found resource and repeatedly return to the same location using a process called *site fidelity* (Beverly et al., 2009). Robots can also communicate using pheromones (Sumpter and Beekman, 2003; Jackson et al., 2007) which are simulated as artificial way points (Campo et al., 2010) to recruit robots to known clusters of resources. Each pheromone trail is represented by a starting waypoint and an ending waypoint. Waypoints are maintained in lists in which pheromone strength of each waypoint decreases exponentially over time. Waypoints are removed once their values drop below a threshold of 0.001. The robot uses the density count c to decide whether to use site fidelity in the next round of foraging, lay a pheromone waypoint, or follow the pheromone waypoint. If a robot returns to a previously found resource area, it searches using an informed correlated random walk that searches more thoroughly than robots searching randomly selected locations. The MPFA uses these same behaviors (see Fig. 3.1b).

The CPFA is implemented in real physical iAnts using a central nest illuminated by a beacon that robots can detect. Robots use a combination of ultrasonic distance, magnetic compass headings, time-based odometry and an on-board forward-facing camera to estimate locations of pheromone waypoints and locations to return to via site fidelity (Hecker and Moses, 2013).

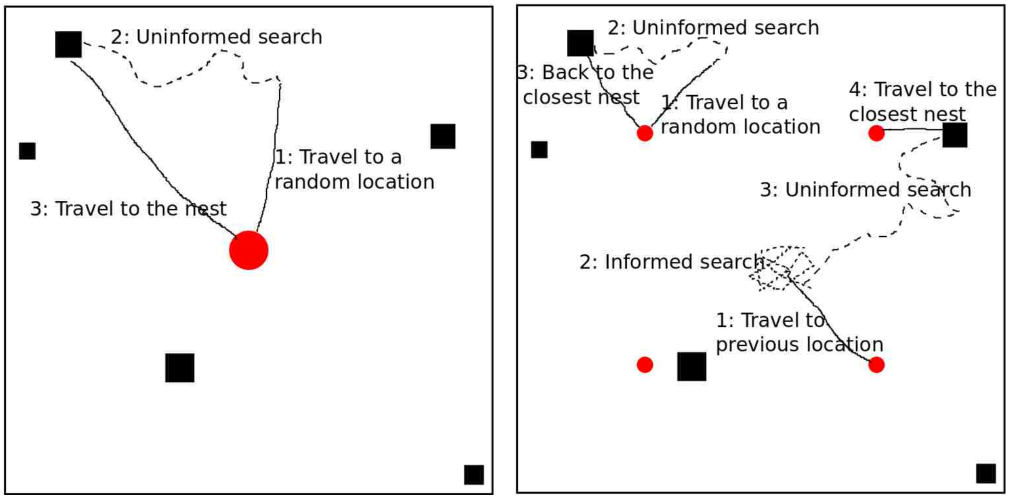
3.5 Methods

We propose the multiple-place foraging algorithm (MPFA), an extension to central-place foraging in which multiple nests are distributed in the environment. In the MPFA, robots always return to the nest closest to them in the area. If a returning robot chooses to communicate resource information using a pheromone-like waypoint, this waypoint will only be shared with other robots that return to the same nest.

3.5.1 The Design of the MPFA

In the MPFA, robots are evenly distributed around nests. Robots start from a random nest, but return to the closest nest to their positions after finding a resource. Robots have *priori* knowledge of the locations of nests. The use of multiple collection points is the fundamental difference between the CPFA and the MPFA; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of multiple nests on swarm foraging efficiency. As in the CPFA, robots use site fidelity or follow pheromone waypoints to exploit resource-rich areas. Our simulations assume that all nests are illuminated by a beacon and that robots can detect the closest beacon.

In the MPFA, the robots search globally as in the CPFA – they can travel in the entire arena (see Fig. 3.1b). The key difference is that robots will always return to the nest closest to the location where they discovered resources. They share pheromone waypoints locally at their current nest. This is in contrast to the CPFA, where pheromone waypoints are associated with the centrally-placed nest and are globally available to all robots. Since robots always return to the closest nest with a found resource in the MPFA, the sensed information relevant to a given resource neighborhood is always associated with the nest closest to the position of the identified neighborhood. Thus, if a robot follows a pheromone waypoint from a nest, then the distance from the nest to the destination of the pheromone is the shortest distance to the resource neighborhood identified by the waypoint.


2: Uninformed search
1: Travel to a random location
3: Travel to the nest
2: Uninformed search
3: Back to the closest nest
1: Travel to a random location
4: Travel to the closest nest
3: Uninformed search
2: Informed search
1: Travel to previous location

(a) CPFA model

(b) MPFA model

Figure 3.1: Schematics showing individual robot foraging trips in (a) the CPFA and (b) the MPFA. In (a), a robot begins its search at a central nest (red circle) and travels to a search site (step 1). Upon reaching the search site, the robot searches for resources by uninformed random walk (step 2) until a resource (black square) is found and collected. After sensing the local resource density, the robot returns to the nest (step 3). In (b), 4 nests are placed. The foraging behavior is identical to the CPFA, except that the robot returns to the nest closest to the location where it finds a resource. The robot path in the upper left of panel (b) shows the robot returning to the nest that it departed from. The path in the lower right of panel (b) shows a robot that finds a resource closer to a different nest, and so it deposits that resource at the new closer nest. If the robot chose to lay a pheromone waypoint, the waypoint would connect the new nest to the resource location.

The set of seven MPFA parameters is identical to the set of CPFA parameters developed by Hecker and Moses (Hecker and Moses, 2015), and is defined below:

- **Probability of switching to search:** When traveling from the nest in a randomly selected direction (step 1 in Fig. 3.1a), at each step robots have a probability of switching from travel to uninformed random search. This governs the time and distance that robots travel in a straight line away from the nest before beginning to search. The probability is initialized by a uniform random distribution, $\mathcal{U}(0, 1)$.

- **Probability of returning to nest:** During search each robots have a probability of giving up search and returning to the nest. It is initialized by a uniform random distribution, $\mathcal{U}(0, 1)$.

- **Uninformed search variation:** During search without *prior* information (not using site fidelity or following pheromones), the turning angle of the correlated random walk θ_t is defined as $\theta_t = \mathcal{N}(\theta_{t-1}, \sigma)$, where θ_{t-1} is the turning angle in the current step, and σ is the uninformed search variation, which determines the turning angle of the next step. σ is initialized by a uniform random distribution, $\mathcal{U}(0, \pi)$.

- **Rate of informed search decay:** Robots searching with *prior* information use a correlated informed random walk that covers area thoroughly using a standard deviation ω of the successive turning angles that decays as a function of time t: $\omega = \sigma + (2\pi - \sigma)e^{-\lambda_{id}t}$, where λ_{id} is the rate of informed search decay, and σ is the uninformed search variation. The degree of turning is initially large and makes the search more local and thorough in the current area. As the search time increases the degree of turning decays to σ and approaches uninformed search (see Fig. 3.1b).

- **Rate of laying pheromone:** Robots are more likely to lay pheromone waypoints when a high density c of resources has been detected. The probability is defined by a Poisson cumulative distribution function (CDF) $POIS(k, \lambda_{lp})$, where λ_{lp} is the evolved parameter.

- **Rate of site fidelity:** Robots that detect a high density of resources are more likely to return to a previously found resource area using site fidelity. The probability is defined by a Poisson CDF $POIS(k, \lambda_{sf})$, where λ_{sf} is the evolved parameter.

- **Rate of pheromone decay:** Rate at which pheromone waypoints decay exponentially over time. It is defined by a decay function $e^{-\lambda_{pd}t}$, where λ_{pd} is the evolved parameter.

The GA selects 7 parameter values for each swarm with fitness defined as foraging performance in experiments implemented in ARGoS. Performance is averaged over experiments on 8 different random resource placements (of a given distribution) to determine the fitness of a parameter set. The GA uses a population size of 50, a 50% uniform crossover rate and a 5% Gaussian mutation rate with a standard deviation of 0.02. The new value v_m of the mutated parameter is equal to $v_c + \alpha D$, where v_c is the current value of the parameter, α is the value generated by the Guassion distribution $\mathcal{N}(0, 0.02)$, and D is the maximum value in the domain of this parameter. We use elitism to keep the parameter set with the highest fitness.

We altered the termination criteria of the GA in order to hasten parameter convergence. We ran the GA for a minimum of 20 generations and a maximum of 100 generations, stopping earlier if termination criteria were met. The GA terminates based on three criteria. The criteria evaluate the number of generations, the convergence of fitness, and the diversity of populations, which is introduced in GAlib (Wall, 1996). The GA will stop if fitness has converged and the diversity of the population is low. Otherwise, the GA will stop after 100 generations.

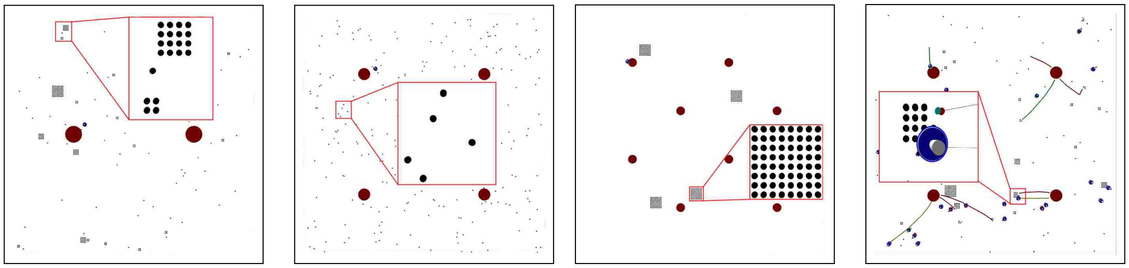
In our GA, 88% of the evolutionary runs terminate due to the convergence of parameter values and fitness values. Across 12 independent evolutionary runs, all evolved parameter sets were nearly equally fit: the standard deviation in fitness was at most 5% of the mean fitness value (Finally, the fitness of the best parameter set is evaluated on 100 additional resource placements).

3.5.2 Experimental Configuration in ARGoS

We implement the CPFA and MPFA in ARGoS. We evaluate both algorithms on a foraging task for 256 resources scattered in random, partially clustered, and clustered resource distributions (see Fig. 3.2).

The random distribution has 256 resources scattered uniformly. The clustered distribution has 4 clusters of 64 resources distributed uniformly, each arranged in an 8×8 grid. The partially clustered distribution has a power law distribution of cluster sizes: 1 cluster of 64 resources, 4 clusters of 16 resources, 16 clusters of 4 resources and 64 resources scattered uniformly.

The configurations of the CPFA and MPFA (listed in Table 3.1) are identical except for the number of nests. We distribute the nests in the arena uniformly as described in Section 3.5.1. The sum of the area of the nests in each MPFA model is equal to the area of the central nest in the CPFA. The nest size reflects the capacity of the nest. We scale the nest radii as 0.5, 0.35, 0.25 and 0.18m in 1, 2, 4 and 8-nest cases, respectively.



(a) 2-nest MPFA and partially clustered distribution (b) 4-nest MPFA and random distribution (c) 8-nest MPFA and clustered distribution (d) A running scenario of 4-nest MPFA model

Figure 3.2: The placement of nests and resources in ARGoS. In all experiments 256 resources (black points) and 24 robots are placed in a 10 × 10m arena, and some number of nests (red circles) are distributed uniformly in the search space. The resources are partially clustered in panel (a), unclustered and spread in a uniform random distribution in (b) and clustered into 4 piles in panel (c). Panel (d) shows a simulation running with 24 robots, the partially clustered resource distribution and four nests. The colored rays indicate pheromone waypoints with different strength. A small area is magnified in each figure to show the resource placement.

Table 3.1: Experimental configuration in ARGoS

Size of the arena (m)	10 × 10
Number of nests	1, 2, 4, or 8
Radius of nests (m)	0.5, 0.35, 0.25, or 0.18
Number of resources	256
Number of robots	24
Foraging time (minutes)	12

The radius of each resource is 0.02 m. Every experiment uses 256 resources and 24 robots. The radius of each robot is 0.085 m. The speed of each robot is 0.16 m/s.

We previously observed an exponential increase in collection time after the majority of resources are collected, and the resource distribution becomes sparse (Hecker et al., 2015). We mitigate this confounding factor by limiting swarm foraging time to 12 minutes, ensuring that swarms do not collect more than 90% of the available resources.

We identify whether performance varies systematically with the number of nests and statistically analyze the trends of evolved parameters by calculating a log-linear regression in which foraging performance is compared to the $\log_2$ of the number of nests.

3.6 Results

We compare the performance of the CPFA and MPFA. The results show the MPFA outperforms the CPFA in foraging performance, is more efficient in collision avoidance, and requires less overall travel time. Source code is available on Github[1]. We then evaluate how parameters changed given 1, 2, 4, or 8 nests. The parameters governing the turning angle of the random walk and the use of site fidelity were qualitatively similar to those previously observed by Hecker and Moses (Hecker and Moses, 2015), and they did not differ systematically with the number of nests. However, two parameters showed interesting patterns. We observe trends in the **probability of laying pheromone** and the **probability of switching to search** which controls the amount of time traveling in a straight line away from the nest before switching to search.

3.6.1 Foraging Performance

The foraging performance of the CPFA and MPFA are shown in Fig. 3.3. Multiple nests produce better foraging performance than the CPFA in all three distributions.

[1] `https://github.com/BCLab-UNM/iAnt-ARGoS/tree/lukey_development`

The number of collected resources increases as the number of nests increases. The foraging of the 8-nest MPFA is 13% higher in the random distribution, 19% higher in the partially clustered distribution, and 27% higher in the clustered distribution. The CPFA has the lowest foraging performance in the clustered distribution.

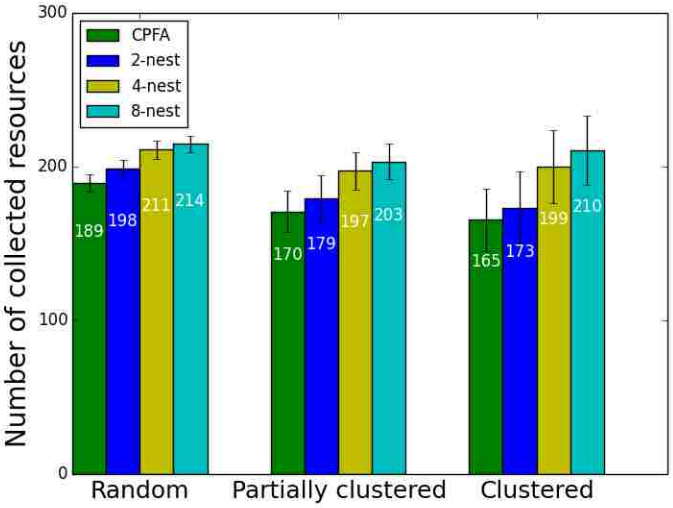


Figure 3.3: Foraging using the CPFA, as well as the 2-nest, 4-nest, and 8-nest MPFA in random, partially clustered, and clustered resource distributions. There is a significant positive trend in the number of resources with the $\log_2$ of the number of nests in all three distributions ($p = 0.02$, $p = 0.017$, and $p = 0.023$, respectively).

The foraging performance per minute for each experiment is shown in Fig. 3.4. Foraging performance significantly increases with the number of nests in the first 5 minutes of the experiments for all three distributions ($p = 0.04$). The foraging performance for the random distribution is initially the highest, while foraging performance for the clustered distribution is initially lowest. Foraging performance decreases over the first several minutes for the random distribution, and increases for the clustered distribution. The partially clustered distribution shows an intermediate pattern. The reasons for these patterns are explored in Section 3.6.4.

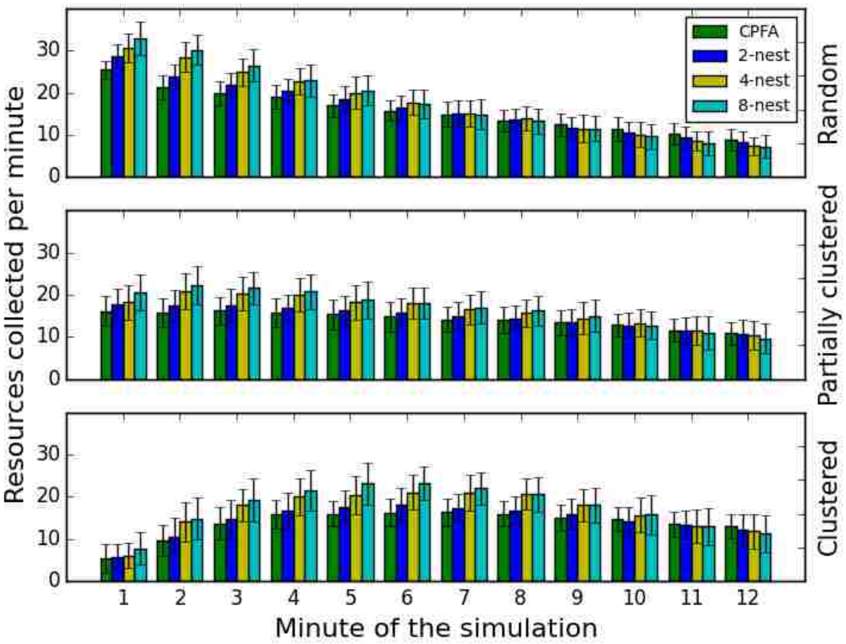


Figure 3.4: The number of collected resources per minute by the CPFA and MPFA. There is a significant positive trend in the number of resources with the $\log_2$ of the number of nests in the first 5 minutes of all three distributions ($p = 0.04$).

3.6.2 Collision Avoidance

In our simulation, if the distance between two robots is less than 0.25 m, each robot will detect a collision. Each robot senses the location of the other and turns left or right in order to avoid the collision, moving approximately 8 cm before resuming traveling.

The total time spent avoiding collisions in each swarm is shown in Fig. 3.5. In the random distribution, the total collision time in collision with multiple nests is less than the total time for the CPFA. In the partially clustered and the clustered distributions, we observe no clear trend.

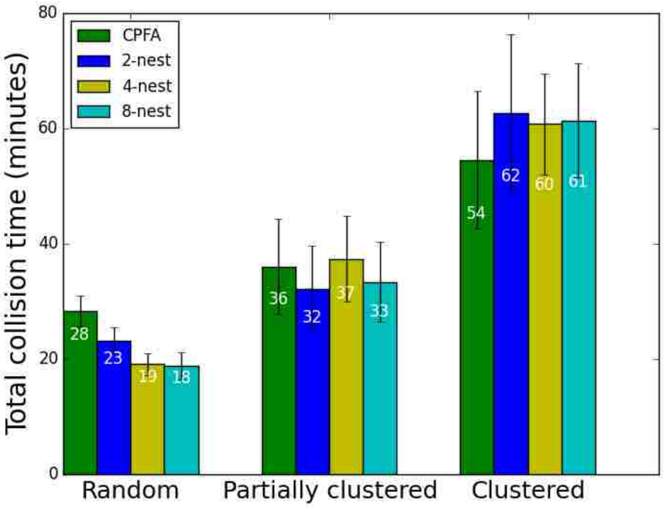


Figure 3.5: Total time spent avoiding collisions for the CPFA and MPFA in three distributions. The p-values of the log-linear regression between the total collision time and the number of nests are $p = 0.05$, $p = 0.85$ and $p = 0.33$ for random, partially clustered, and clustered, respectively.

3.6.3 Search and Travel Efficiency

Foraging time is the composition of two distinct activities. When a robot departs from its nest, it travels to a location. Once at the location, the robot engages in a localized search. Once a resource is discovered, the robot takes approximately the same *travel time* back to the nest. All other robots take approximately the same travel time back to the location of the discovered resource, but their *search time* is reduced by the information communicated through pheromone waypoints.

The average search and travel time per resource in each swarm is shown in Fig. 3.6. Round trip foraging time, i.e., the sum of search and travel time per resource, decreases as the number of nests increases in each distribution. The search time increases as the number of nests increases in the random distribution, while there is no significant trend for the partially clustered and clustered distributions. However, the travel time decreases as the number of nests increases in each distribution. The travel time is lowest in the random distribution and highest in the clustered distribution. The travel time with multiple nests is less than with the CPFA: swarms using the CPFA require up to 50% more travel time in the random distribution, up to 33% more travel time in the partially clustered distribution, and up to 30% more travel time in the clustered distribution.

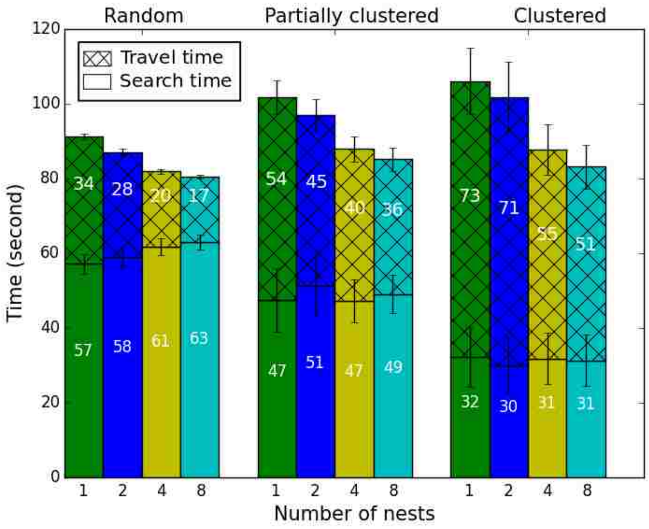


Figure 3.6: The search and travel time per resource for the CPFA and MPFA. Search time increases with the number of nests in the random distribution ($p = 0.01$), but has no trend with the number of nests in the partially clustered and clustered distributions ($p = 0.95$ and $p = 0.85$, respectively). Travel time decreases in all three distributions ($p = 0.016$, $p = 0.013$, and $p = 0.045$, respectively).

3.6.4 Observed Trends in Parameters

Fig. 3.7 illustrates how the probability of laying pheromones changes with the number of nests. The figure shows the probability of laying pheromones (calculated from a Poisson distribution with the evolved parameter λ_p) given that $k = 2$ resources were detected in the local neighborhood of the most recently found resource. Results are shown for the evolved parameter set with the highest fitness for each distribution. The probability of laying pheromones is very low for the random distribution, regardless of how many nests are placed. The probability is higher for partially clustered, and even higher for clustered resources, and in both of those cases the probability increases with the number of nests.

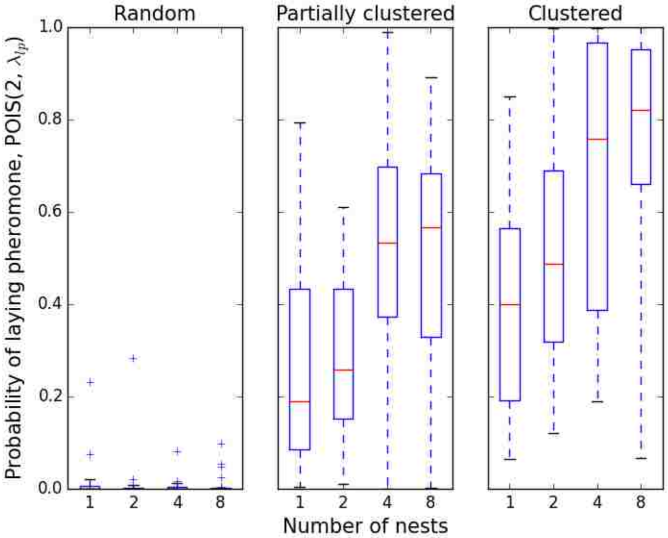


Figure 3.7: The evolved probability of laying pheromone when two resources are found in the resource neighborhood. Medians and quartiles for 12 replicates of evolution are shown for each model. A linear regression ($\log_2$ on the number of nests versus the probability of laying pheromone) shows no trend ($p = 0.204$) in the random distribution, but a statistically significant trend for the partially clustered ($p = 0.006$) and clustered ($p = 0.05$) distributions.

The evolved probability of switching to search is shown in Fig. 3.8. This parameter indicates how long the robot travels in a straight line directly away from the nest before it begins to search for resources. Higher probabilities indicate that robots stay closer to the nest. The probability increases as the number of nests increases in the random distribution, indicating that robots stay closer to their nest when more nests are placed in the arena. There are no significant trends in the partially clustered and clustered distributions.

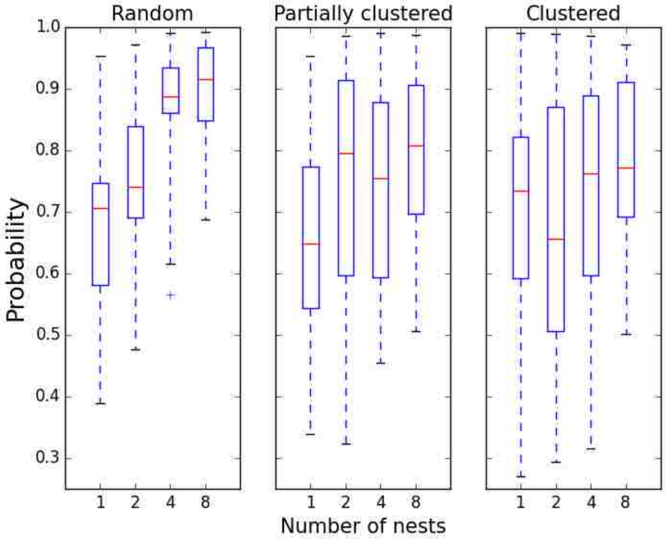


Figure 3.8: The evolved probability of switching to search. Medians and quartiles for 12 replicates of evolution are shown for each model. A linear regression ($\log_2$ on the number of nests versus the probability of switching to search) shows a statistical significant trend for the random distribution ($p = 0.02$).

3.7 Discussion

This chapter explores how the use of multiple nests affects foraging performance. We show that robot swarms using the MPFA exhibit higher foraging performance and spend less time spent on collision avoidance. We examine the time spent actively searching for resources and the time spent traveling from the resource to the nest and back. Not surprisingly, increasing the number of nests (to 2, 4, or 8) increases the foraging rate (see Fig. 3.3), and decreases travel time (see Fig. 3.6). However, the relationship between search time, collision time, and the number of nests depends on how resources are distributed. Given a random resource distribution (see Fig. 5.6b), search time generally increases when robots have access to more nests. Given a clustered distribution (see Fig. 5.6c), more time is spent in collision avoidance, regardless of the number of nests.

Fig. 3.4 shows how foraging rates change with time in each distribution. In the random distribution, the resources are scattered in the entire space, so the robots can discover and collect more resources at the beginning of the experiment, but fewer in the end as resources become more sparse. In the clustered distribution, the resources are initially harder to discover, so foraging rates are low. Once piles are found, robots are recruited, which increases the foraging rate, until the foraging rate declines as the remaining resources become sparse and harder to find (Hecker et al., 2015). The evolved pheromone laying rate (see Fig. 3.7) is the highest for the clustered distribution, and the lowest for the random distribution. In the partially clustered distribution, both processes (discovering random resources and recruiting to large piles) occur, and so there is no clear trend in foraging rates over time. The same trends are seen for the MPFA simulations and the CPFA simulation, indicating that the MPFA does not fundamentally alter the process of finding resources.

Fig. 3.5 shows that the total collision time in the MPFA is slightly reduced compared to the CPFA for the random distribution. However, collision time in the MPFA is higher than the CPFA in the clustered distribution even though the robots are dispersed to more nests. We hypothesize that more collisions occur with more nests because the MPFA evolves greater pheromone use (see Fig. 3.7), and these pheromones concentrate the robots on short routes between the locations where resources are clustered and the nearest nest. This increases the total foraging rate while simultaneously increasing the time spent avoiding collision.

Pheromones have an additional effect on foraging rates, as demonstrated in Fig. 3.6. In the partially clustered and clustered distributions, where pheromones can be used effectively, search times are shorter than in the random distribution. Interestingly, search times increase as more nests are added in the random distribution. One cause of this is shown in Fig. 3.8. The more nests there are, the more likely robots will minimize time traveling away from those nests — they will begin search behaviors immediately upon leaving the nest. This means that the smaller search areas around each nest are depleted more quickly, making subsequent resources more difficult to find. This trend is seen in the top panel of Fig. 4 where the 8 nest MPFA is by far the fastest in the initial minutes, but the slowest at the last minute.

These patterns reveal that the MPFA improves foraging rates, not just because of the simple intuitive reduction in travel distance. There are tradeoffs between the distance travelled from the nest, the time spent searching and the collision avoidance of robots. The GA tunes parameters to balance these tradeoffs and optimizes the performance of each swarm, resulting in systematic changes in parameters governing pheromone laying and distance travelled from the nest as more nests are added.

In future work, we will examine how these tradeoffs can be balanced dynamically, for example, by dynamically allocating and de-allocating nests as resources are found and depleted.

3.8 Acknowledgments

We gratefully acknowledge members of the Moses Biological Computation Lab, and particularly thank Tatiana P. Flanagan for discussing the multiple nest strategies of ants as a framework for the MPFA.

Chapter 4

The Scalability and Adaptation of MPFA

4.1 Publication Notes

Citation: Qi Lu, Melanie E. Moses, and Joshua P. Hecker. A Scalable and Adaptable Multiple-Place Foraging Algorithm for Ant-Inspired Robot Swarms. Workshop on On-line decision-making in multi-robot coordination at the 2016 Robotics Science and Systems Conference, 2016.

Received: May 12, 2016

Accepted: May 25, 2016

Published: June 19, 2016

4.2 Abstract

A swarm of simple robots works together is an effective approach to explore large unmapped areas. The central-place foraging algorithm (CPFA) is effective for coordinating robot swarm search and collection tasks. Robots start at a centrally placed location (nest), explore potential targets in the area without global localization or central control, and return the targets to the nest. The scalability of the CPFA is limited because large numbers of robots produce more inter-robot collisions and large search areas result in substantial travel costs. We address these problems with the multiple-place foraging algorithm (MPFA), which uses multiple nests distributed throughout the search area. Robots start from a randomly assigned home nest but return to the closest nest with found targets. We simulate the foraging behavior of robot swarms in the robot simulator ARGoS and employ a genetic algorithm to discover different optimized foraging strategies as swarm sizes and the number of targets is scaled up. In our experiments, the MPFA always produces higher foraging rates, fewer collisions, and lower travel and search time compared to the CPFA for the distribution of the partially clustered target. The main contribution of this paper is that we systematically quantify the advantages of the MPFA (reduced travel time and collisions), the potential disadvantages (less communication among robots), and the ability of a genetic algorithm to tune MPFA parameters to mitigate search inefficiency due to less communication.

4.3 Introduction

A large number of simple individual robots working together has the potential to be useful for tasks which a traditional single expensive, specialized and complicated robot is not able to handle, such as searching in large unmapped areas (Stormont, 2005), distributed contaminant cleanup, and rescue (Kantor et al., 2006). Robot swarms can also be involved in sophisticated problem solving, including cooperative transportation, de-mining, and space exploration (Brooks and Flynn, 1989; Landis, 2004; Fink et al., 2005; Stolleis et al., 2015).

We focus on developing a scalable, decentralized search-and-collection algorithm based on ant-like foraging (Gordon and Kulig, 1996; Winfield, 2009b; Liu and Winfield, 2010). The swarm can adapt to changes in swarm size and the number of targets through real-time response to conditions without external or off-line intervention. Each robot in the swarm makes real-time in-situ decisions on whether to communicate using pheromones, forego communication but individually return to search a location, or return to the collection zone to gather additional information from other robots. The robot behaviors are modeled after those of a particular genus of desert seed harvester ants that (Flanagan et al., 2011, 2012) are restricted to foraging in short-time windows during which not all available targets can be collected; So they are designed to collect as many targets as possible, but not for optimal complete collection (Hecker et al., 2015).

Here, we present the multiple-place foraging algorithm (MPFA) with multiple nests that robots depart from and return to. The robots make on-line decisions to switch to new collection zones based on proximity to their last-found target. The MPFA was presented in our recent work (Lu et al., 2016a) and it showed that distributing 2, 4, or 8 nests in the MPFA produce higher foraging rates and lower average travel time compared to the central-place foraging algorithm (CPFA) developed by Hecker and Moses (P. Hecker et al., 2012; Hecker and Moses, 2013). Here we compare the scalability and adaptation of the MPFA to the CPFA when the number of robots and the number of targets increase. In the MPFA we deploy 4 nests uniformly in the same size of search arena. A set of real-valued parameters specifying the individual robot controllers is evolved by a Genetic Algorithm (GA) to optimize the foraging strategy in the multi-physics robot simulator Autonomous Robots Go Swarming (ARGoS) (Pinciroli et al., 2012). Every robot in the swarm uses the same controller.

We evolve foraging strategies for different swarm sizes (4, 8, 16, 32 and 64) and number of targets (128, 256, 512, 1024 and 2048). We observe the average foraging rate, collision time, travel and search time change as swarm size and the number of targets increase.

The remainder of this chapter is organized as follows. Section 4.4 introduces related work. The design of the MPFA and the description of evolution are provided in Section 4.5 and Section 4.6. The configuration of the MPFA in ARGoS and the experimental results are in Section 4.7 and Section 4.8. Section 4.9 discusses the conclusions.

4.4 Related Work

Central-place foraging is commonly studied in swarm robotics (Şahin et al., 2008; Brambilla et al., 2013). Hecker and Moses utilized and formalized the behaviors from Flanagan and Letendre's ant field studies (Flanagan et al., 2011, 2012; Letendre and

Moses, 2013) to create the CPFA. The algorithm is well designed and applied to real physical robots, which are designed on the iAnt robots platform (Moses et al., 2014). The error-tolerance, flexibility, and scalability were evaluated on both simulated and real robot swarms (Hecker and Moses, 2015). However, the simulated robots were not physics-based and collisions between robots were not considered.

The studies on task allocation by Hsieh et al (Hsieh et al., 2008; Halász et al., 2007; Berman et al., 2008) showed that a bio-inspired approach to the deployment of a homogeneous swarm of robots to multiple sites. The robots autonomously redistribute themselves among the candidate sites to ensure task completion by optimized stochastic control policies. The studies model the swarm as a hybrid system where agents switch between maximum transfer rates and constant transition rates. In our method, the robots are distributed and initialized to multiple nests evenly. Then, robots transit to other nests autonomously based on the distribution of remainder targets and the evolved search strategy. The search strategy is evolved by GA automatically and all the robots use the same strategy.

There are a few studies on multiple-place foraging in biological systems. The polydomous colonies of Argentine ants are comprised of multiple nests spanning hundreds of square meters (Flanagan et al., 2013; Schmolke, 2009). A study by Chapman et al (Chapman et al., 1989) showed that a community of spider monkeys can be considered as multiple central place foragers (MCPF). The monkeys select a sleeping site close to current feeding areas, and the MCPF strategy entails the lowest travel costs. A study by Tindo et al (Tindo et al., 2008) showed that wasps living in multiple nests have greater survival rate and increased productivity. However, multiple-place foraging has not been systematically compared to central placed foraging in robotic swarms which we do here.

4.5 The Design of The MPFA

In the MPFA, robots are evenly distributed around nests. They start from a nest but return to the closest nest to their position after finding a target or giving up the search. The use of multiple collection points is the fundamental difference between the CPFA and the MPFA; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of multiple nests on swarm foraging efficiency.

The behavior of an individual robot in an MPFA foraging round is shown in Fig. 4.1. Each robot transitions through a series of states as it forages for targets. This differs from the CPFA (Hecker and Moses, 2015) in how the robots return to nests which are in steps 4 and 5. In the MPFA, robots initially disperse from the nests closest to them, followed by random selected travel paths (step 1). An uninformed correlated random walk is used to search targets when robots stop to follow the paths (step 2) (Fewell, 1990). Robots navigate home to nests closest to them when they retrieve targets or give up search (step 4 and 5) (Crist and MacMahon, 1991). Robots that find targets will detect the local target density before return to nests (step 3) (Hölldobler, 1976). Robots that are more likely to return to previously found sites using site fidelity or pheromone recruitment (step 6), then they search the sites thoroughly with informed walk (step 7).

In our design, the robots search globally just as in the CPFA – they can travel in the entire arena. As in the CPFA, pheromone trails are simulated using a list of pheromone-like waypoints to identify target-rich areas. When a robot returns to a nest, it will probabilistically select a waypoint from the nest's list and travel to that location. The robot shares information (pheromone waypoints) locally at its current

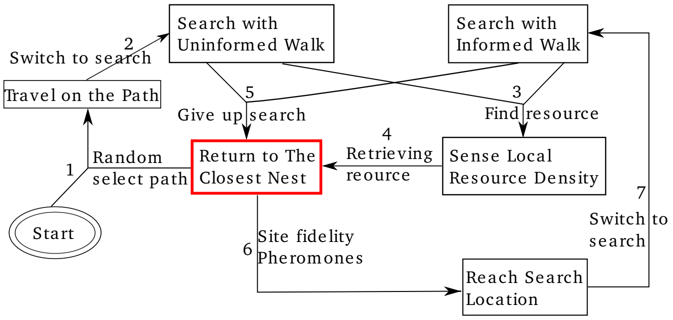


Figure 4.1: The flow chart of an individual robot's behavior in MPFA during an experiment.

nest (see Fig. 4.2). In contrast to the CPFA, pheromone waypoints are globally available to all robots.

Since robots always return to the closest nest with a found target, the sensed information relevant to a given target neighborhood is always associated with the nest closest to the position of the identified neighborhood. Thus, if a robot follows a pheromone waypoint from a nest, then the distance from the nest to the destination of the pheromone must be the shortest distance to the target neighborhood identified by the pheromone.

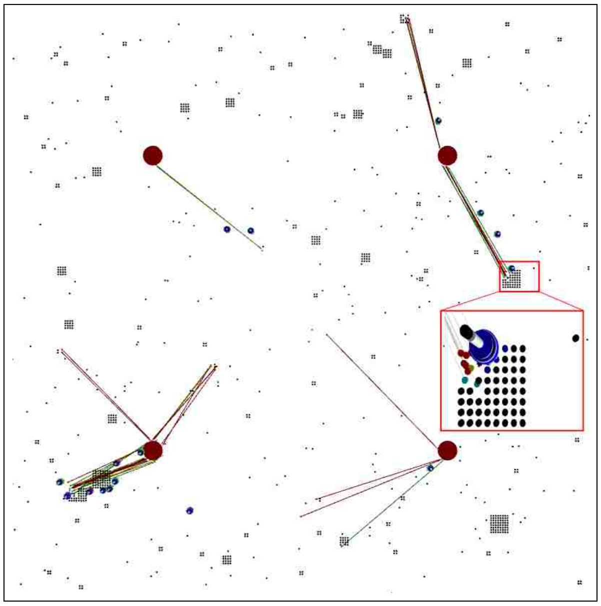


Figure 4.2: The placement of nests and targets in ARGoS. 1024 targets (black points) and 16 robots (larger blue points) are placed in a 15×15 m arena, 4 nests (red circles) are distributed uniformly in the arena. The targets are arranged in a partially clustered distribution. Colored lines indicate pheromone trails with different strength. A small area is magnified to show a robot, colored pheromone waypoints, a large cluster of targets, and a single target.

4.6 The GA Evolution

We implement the CPFA and MPFA on a foraging task for different experiments in ARGoS. Furthermore, we use a GA to identify MPFA parameters that maximize foraging strategy. We implement our GA using GAlib (Wall, 1996) following parameters described by Hecker and Moses (Hecker and Moses, 2015). The set of seven MPFA parameters is identical to the set of CPFA parameters. The movement, sensing, and communication of each single robot are evolved and evaluated. The parameters are described in the following,

- **Probability of switching to search:** The robot has the probability of switching from travel to uninformed random search. The probability is initialized from a uniform random distribution, $\mathcal{U}(0,1)$.

- **Probability of returning to nest:** The robot has the probability of giving up search and returning to nest. It is initialized from a uniform random distribution, $\mathcal{U}(0,1)$.

- **Uninformed search variation:** If the robot searches using a correlated uninformed random walk, the successive turning angles θ_t is defined by $\theta_t = \mathcal{N}(\theta_{t-1}, \sigma)$, where θ_{t-1} is the turning angle in the current step, and σ is the standard deviation or uninformed search variation, which determines the turning angle of the next step. σ is initialized from a uniform random distribution, $\mathcal{U}(0,\pi)$.

- **Rate of informed search decay:** If the robot searches using an informed correlated random walk, the standard deviation of the successive turning angles σ decays as a function of time t, $\sigma = \omega + (2\pi - \omega)e^{-\lambda_{id}t}$, where λ_{id} is the rate of informed search decay. λ_{id} is initialized from an exponential decay function $\exp(5)$.

- **Rate of laying pheromone and rate of site fidelity:** The information decisions are governed by parameterization of a Poisson cumulative distribution function as defined by $\mathcal{POIS}(k, \lambda)$, where k is the likelihood of detecting at least k additional resources, and λ is the rate of laying pheromone or the rate of site fidelity. It is initialized from a uniformed random distribution, $\mathcal{U}(0, 20)$. The robot returns to a previous location if $\mathcal{POIS}(k, \lambda) > \mathcal{U}(0,1)$. If k is large, the robot is likely to return to the same location using information on its next foraging trip.

- **Rate of pheromone decay:** The pheromone decays exponentially over time t as defined by $e^{-\lambda_{pd}t}$, where λ_{pd} is the rate of pheromone decay. It is initialized from an exponential decay function exp(10).

We repeat the evolutionary process 10 times for the CPFA as well as for the MPFA, in order to generate 10 independently evolved foraging strategies for each experimental configuration.

In summary, using a swarm size of 40 robots, we evaluate each swarm 8 times on different random placements of targets in the partially clustered distribution to determine their fitness. We use a 50% uniform crossover rate and a 5% Gaussian mutation rate with a standard deviation of 0.02. We use elitism to keep the individual with the highest fitness.

We altered the termination criteria of the GA in order to hasten parameter convergence and ran the GA for a maximum of 100 generations. The GA terminates based on three criteria: the number of generations, the convergence of fitness, and the diversity of swarm sizes, which are introduced in GAlib (Wall, 1996). The GA will stop if the fitness is convergent and the diversity of the population is low. Otherwise, it will stop after 100 generations. Our code is available on GitHub[1].

In our GA, 89% of the evolution terminates on the convergence of fitness and the diversity of swarm sizes. Across 10 independent evolutionary runs, all evolved parameter sets were nearly equally fit: The standard deviation in fitness was at most

[1] `https://github.com/BCLab-UNM/iAnt-ARGoS/tree/lukey_development`

5% of the mean fitness value. The fitness of the best parameter set, evaluated on 100 target placements, is shown in Fig. 4.3 and Fig. 4.4.

4.7 Experimental Configuration in ARGoS

Table 4.1 shows the experimental configuration in ARGoS. To test scalability, the number of targets is always 1024, and the number of robots is scaled to be 4, 8, 16, 32 or 64. We set different foraging time windows for each swarm, depending on the swarm size. The selected times allow the evolved swarms to collect approximately half of the targets. The foraging time of robots are the same across all experiments: by multiplying the number of robots by the foraging time, we have 480 robot-minutes (or 8 robot-hours) in our experiments (see Table 4.1).

To test adaptation, the number of robots is always 32. The number of targets is 128, 256, 512, 1024 or 2048. The foraging time is set independently for each experiment so that approximately 40% of the targets are collected by the best evolved strategy. All experiments are replicated 100 times. The locations of targets and robots are initialized randomly in the 100 replicates.

Table 4.1: Experimental configuration in ARGoS

Scalability	Robots	4, 8, 16, 32 or 64
	Targets	1024
	Time (minutes)	120, 60, 30, 15 and 7.5
Adaptation	Robots	32
	Targets	128, 256, 512, 1024 or 2048
	Time (minutes)	5, 8, 10, 12 and 30

The targets are placed in a partially clustered distribution. This distribution has various sizes of square clusters. The targets are placed either in a large cluster, a medium cluster or individual targets in a uniform random distribution (see Fig. 4.2). Both algorithms are tested in a simulated arena size of $15 \times 15m^2$. The CPFA has one center nest and the MPFA has 4 uniformly and evenly distributed nests.

4.8 Results

We compare the efficiency of the CPFA and the 4 nest MPFA on foraging rate, collision time, and travel and search time when the swarm sizes and the number of targets are scaled up. We identify statistical differences using a t-test, and we identify whether performance varies systematically by calculating a log-linear regression in which the performance are compared to the $\log_2$ of the swarm sizes or the number of targets.

4.8.1 Foraging Efficiency

The total foraging rate of each swarm is the sum of the total collected targets in the swarm. We measure the average foraging rate, which is the number of targets per robot collected in every minute. Fig. 4.3 shows the average foraging rate as the swarm size increases. The average foraging efficiency of the MPFA exceeds that of the CPFA in all cases, by up to 66% in the case of 64 robots.

Fig. 4.4 shows the average foraging rate as the number of targets increases. The average foraging efficiency of the MPFA exceeds that of the CPFA in all cases, by up to 50% in the case of 2048 targets.

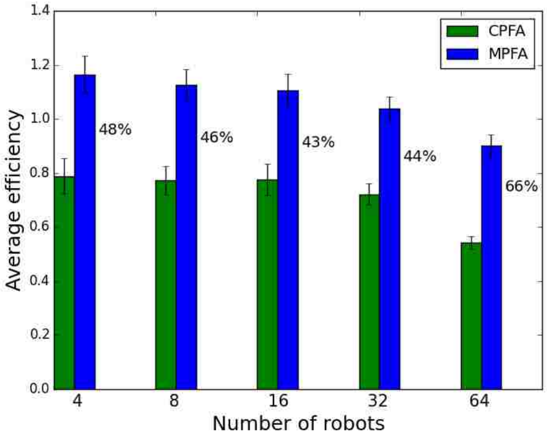


Figure 4.3: The average efficiency (targets collected per robot, per minute) for the CPFA ($p = 0.08$) and MPFA ($p = 0.04$) decrease as the swarm size increases. The p value is from the average of collected targets and the $\log_2$ of the swarm size. Results are for 100 replicates. The percentage of improvement is labelled.

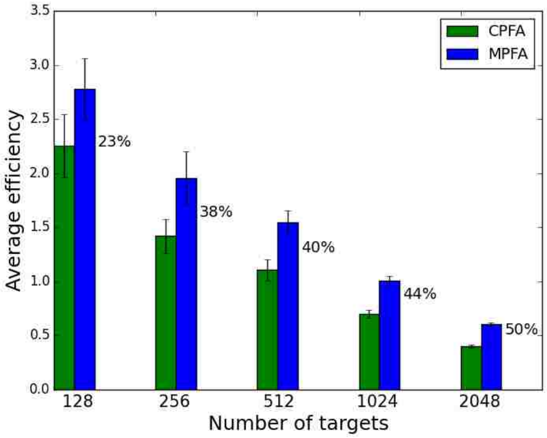


Figure 4.4: The average efficiency (targets collected per robot, per minute) for the CPFA ($p = 0.04$) and MPFA ($p = 0.001$) decrease as the number of targets increases. The p value is from the average of collected targets and the $\log_2$ of the number of targets. The efficiency is always higher for the MPFA.

4.8.2 Collision Efficiency

In our simulation, if the distance between two robots is less than 0.25 m, each robot will detect a collision. Each robot senses the location of the other and turns left or right in order to avoid a collision, moving approximately 8 cm before resuming traveling.

The collision time is the time required to avoid a collision. The total collision time of each swarm is the sum of the total collision time for all robots in the swarm. We measure the average collision time, which is the collision time per robot in collecting a target. The "per robot, per target" collision makes the comparison fairly. For "per robot", it is obvious that a larger swarm results in more collisions, but the rate of increase is not obvious. It is easier to analyze the trend of collision rates on each robot rather than on the swarm when the swarm sizes are different. However, collisions are higher if more targets are collected. We should consider the collision rate based on the foraging rate (in our results, the MPFA always has higher foraging rate). The average collision time as swarm size increases is shown in Fig. 4.5.

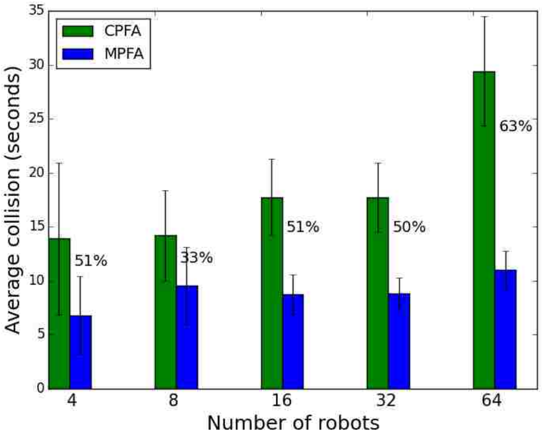


Figure 4.5: The average efficiency (collision time per robot, per target) for the CPFA ($p = 0.06$) and MPFA ($p = 0.10$) increase as the swarm size increases.

The collision time for the MPFA is less than the collision time for the CPFA. We also see that the collision time for the MPFA is reduced as the number of targets increases (see Fig. 4.6).

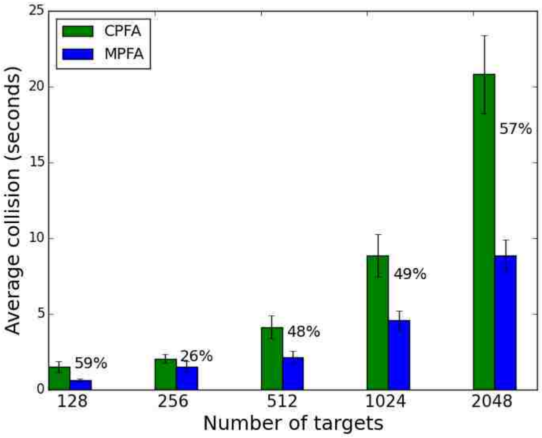


Figure 4.6: The average efficiency (collision time per robot, per target) for the CPFA and MPFA as the number of targets increases ($p = 0.03$).

4.8.3 Travel and Search Efficiency

Foraging time is composed of two distinct activities. When a robot departs from its nest, it travels to a location where it starts to search for targets. Once at the destination, the robot engages in a localized search. Once a target is discovered, the robot takes approximately the same *travel time* back to the nest. Some robots take approximately the same travel time back to the location of the discovered target if they are recruited by pheromones, but their *search time* is reduced.

We measure the average travel time and search time spent to collect one target by a robot. The average travel time for the MPFA (see Fig. 4.7) is less than the CPFA for all swarm sizes.

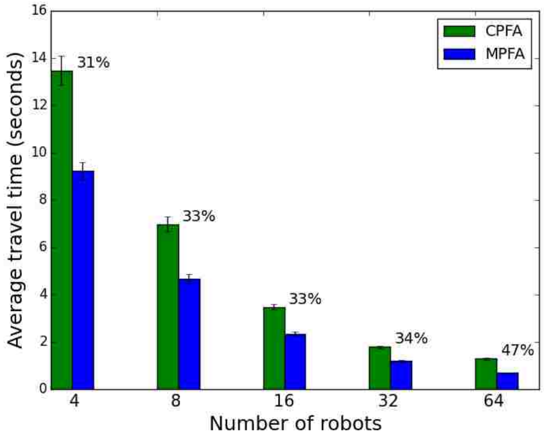


Figure 4.7: The average travel time (per robot, per target) for the CPFA and MPFA decrease as the swarm size increases ($p = 0.04$).

The travel time for the MPFA (see Fig. 4.8) is also less than the CPFA as the number of targets increases.

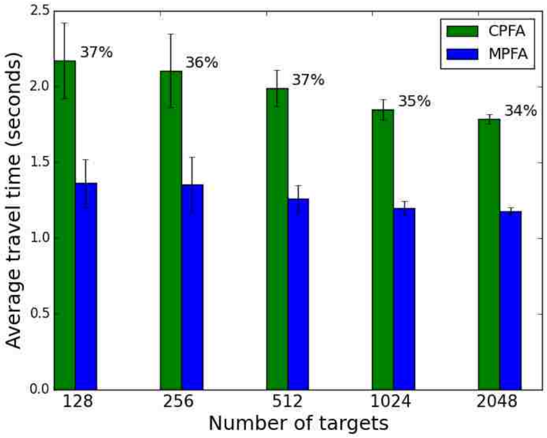


Figure 4.8: The average travel time (per robot, per target) for the CPFA ($p = 0.001$) and MPFA ($p = 0.03$) as the number of targets increases.

Fig. 4.9 shows that the average search time decreases as the number of robots increases. The search time for the MPFA is less than the CPFA. The search time for the CPFA decreases faster than the MPFA. The improvement is up to 34% in the first case for 4 robots and it is down to 19% in the last case for 64 robots.

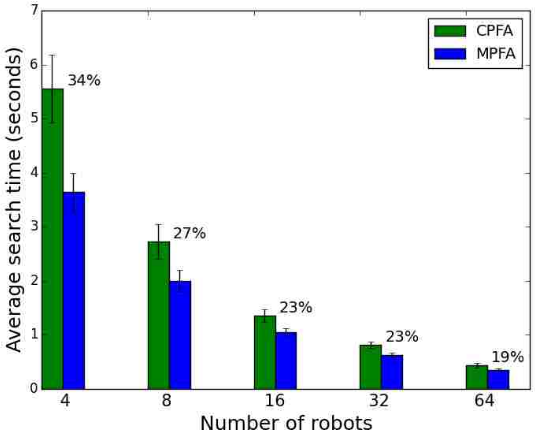


Figure 4.9: The average search time (per robot, per target) for the CPFA and MPFA as the swarm size increases ($p = 0.03$).

The search time decreases as the number of targets increases (see Fig. 4.10). The search time for the MPFA decreases faster than the CPFA. The improvement goes up to 31% in the last case.

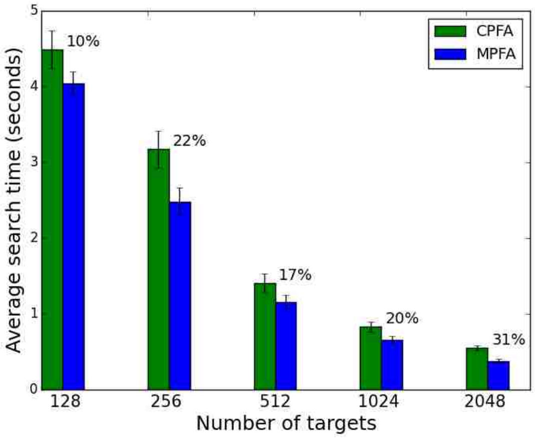


Figure 4.10: The average search time (per robot, per target) for the CPFA and MPFA as the number of targets increases ($p = 0.05$).

4.9 Discussion

This chapter explores how swarm size and the number of targets affect foraging rates, collision time, travel and search time. Not surprisingly, increasing the swarm sizes or the number of targets decreases the average foraging rate (see Fig. 4.3 and Fig. 4.4), but decreases slower for the MPFA. This implies that the MPFA is more efficient in larger swarms or in an environment with more targets.

The average collision time for the MPFA is much lower than the CPFA as the swarm size or the number of targets increases (see Fig. 4.5 and Fig. 4.6). The collision time for the CPFA increases faster as the number of targets increases (see Fig. 4.6). We hypothesize that the more targets there are, the harder robots will spread out in the CPFA. The result demonstrates that the MPFA has the advantage of avoiding collisions in large swarm size or in an environment with a large number of targets.

The increase of swarm sizes makes the average travel time for the MPFA decrease faster than the CPFA (see Fig. 4.7). This shows that the MPFA has the advantage of reducing travel time as the swarm size increases. We hypothesize that the evolved probability of returning to the nest increases faster as the swarm size increases. The more robots there are, the more likely robots will minimize time traveling in the MPFA. On the other hand, search time has smaller difference than travel time (see Fig. 4.9).

The addition of more nests shortens the travel time for the MPFA. However, the information (pheromone waypoints) is distributed to multiple nests. In contrast to the CPFA, pheromone waypoints are globally available to all robots. So, there are tradeoffs among communication (and therefore search time) and travel time and congestion. In addition, the MPFA may get the benefit from all resources are not eventually be moved to one nest. However, we can consider a ”high-speed” delivery (multiple targets can be moved in one round) in the future.

The search time for the MPFA decreases faster than the CPFA with increasing numbers of targets (see Fig. 4.10). This shows that the MPFA has the advantage of reducing search time in an environment with large number of targets. We hypothesize that the evolved probability of laying pheromone increases and it is higher for the MPFA, or the rate of pheromone decay decreases and it is lower for the MPFA as the number of targets increases. The more targets there are, the more likely pheromone will be laid, or slower pheromone decay.

These discoveries reveal that the MPFA improves foraging rates when the swarm size or the number of targets are scaled up. This is not only because of the simple intuitive reduction in travel time, but also because of the significant improvement in avoiding collisions. Overall, the MPFA has better performance as the swarm size or the number of targets increases.

In the future work, we will discover the trends on the evolved seven parameters and confirm the above hypothesis for the random, partially clustered and clustered resource distributions. In addition, we will consider the cost of deploying multiple nests and evolve the optimized number of nests for different resource distributions.

4.10 Acknowledgements

We gratefully acknowledge members of the Moses Biological Computation Lab for their assistance with the multiple-place foraging swarm robotics project. Thanks to Antonio Griego for developing the CPFA algorithm in ARGoS. Thanks to Matthew Fricke for discussing issues in the experiments.

Chapter 5

Multiple-Place Swarm Foraging with Dynamic Depots

5.1 Publication Notes

Citation: Qi Lu, Joshua P. Hecker, and Melanie E. Moses. Multiple-place Swarm Foraging with Dynamic Depots. Journal of Autonomous Robots. 42(4):909-926, January, 2018.

Received: December 21, 2016

Accepted: December 16, 2017

Published: January 09, 2018

5.2 Abstract

Teams of robots can be organized to collectively complete complex real-world applications, for example collective foraging in which robots search for, pick up, and drop off resources in a collection zone. In the previously proposed central-place foraging algorithm (CPFA), foraging performance decreases as swarm size and search areas scale up: more robots produce more inter-robot collisions and larger search areas produce longer travel distances. We propose the multiple-place foraging algorithm with dynamic depots (MPFA$_{dynamic}$) to address these problems. Depots are special robots which are initially distributed in the search area and can carry multiple resources. Depots move to the centroids of the positions of local resources recently detected by robots. The spatially distributed design reduces robot transport time and reduces collisions among robots. We simulate robot swarms that mimic foraging ants using the MPFA$_{dynamic}$ strategy, employing a genetic algorithm to optimize their behavior in the robot simulator ARGoS. Robots using the MPFA$_{dynamic}$ find and collect resources faster than both the CPFA and the static MPFA. MPFA$_{dynamic}$ outperforms the static MPFA even when the static depots are optimally placed using global information, and it outperforms the CPFA even when the dynamic depots deliver resources to a central location. Further, the MPFA$_{dynamic}$ scales up more efficiently, so that the improvement over the CPFA and the static MPFA is even greater in large (50 × 50 m) areas. Including simulated error reduces foraging performance across all algorithms, but the MPFA still outperforms the other approaches. Our work demonstrates that dispersed agents that dynamically adapt to local information in their environment provide more flexible and scalable swarms. In addition, we

illustrate a path to implement the MPFA$_{dynamic}$ in the physical robot swarm of the NASA Swarmathon competition.

5.3 Introduction

One major goal of swarm robotics research is to design robust, scalable, and flexible collective behaviors for multiple autonomous robots (Şahin, 2005; Moses and Banerjee, 2011; Brambilla et al., 2013). Simple rules and local interactions among individual robots result in desired collective swarm behavior by self-organized coordination mechanisms. Biological studies have revealed self-organized coordination mechanisms in social insects which can be effectively implemented in swarm robotics systems (Camazine et al., 2001; Şahin, 2005).

In this research, we focus on the foraging behavior of robot swarms. The challenge is to develop an effective, decentralized search-and-collection foraging algorithm for ant-like robot swarms (Gordon and Kulig, 1996; Winfield, 2009a; Liu and Winfield, 2010). Robots must retrieve objects from an environment and bring them back to a depot (or nest). Effective collective foraging requires coordination, navigation, and communication and is therefore a useful abstraction of many complex, real-world applications such as humanitarian de-mining, search and rescue, intrusion tracking, collection of hazardous materials, and space exploration (Winfield, 2009b; Brambilla et al., 2013). In particular, foraging is commonly used as a testbed for collective exploration, collective transport, and collective decision-making (Gazi and Passino, 2004; Brambilla et al., 2013).

We propose the multiple-place foraging algorithm with dynamic depots (MPFA$_{dynamic}$). Depots are special robots which are able to carry multiple resources.

Targets are objects such as mineral resources, hazardous waste, or any item that needs to be retrieved from the environment and collected at a location. Foraging robots depart from a depot to forage for resources and then return to the closest depot to deliver these resources (the closest depot may be different from the one the robot departed from). Depots move to new locations based on the mean positions of the remaining resources sensed by the robots. The positions of the sensed resources are stored at each depot when each foraging robot returns to that depot. The stored positions are relative to the depot's current location so that no central controller is needed to facilitate information sharing across the swarm.

The final delivery of resources that are collected by the depots depends on the application. Resources may be processed at the dispersed locations where they are collected; they may be collected by another larger robotic agent that empties depots and delivers their contents to a central location; or, as the depots become full, they may drive the resources to the desired location. We explore the latter scenario in a subset of our experiments.

We compare the performance of the MPFA$_{dynamic}$ with our previous MPFA with static nests (MPFA$_{static}$) proposed by (Lu et al., 2016a) with uniformly-distributed static depots, and to the central-place foraging algorithm developed by (Hecker and Moses, 2015).

In order to assess the effectiveness of our approach, we also compare our results to algorithms with access to global information. We compare the MPFA$_{dynamic}$ which uses only local information, to versions of the MPFA with global information describing the initial locations of all resources. These algorithms use the k-means++ clustering algorithm to determine the initial positions of the depots to minimize transport distance. We evaluate the MPFA with depots that have global information about target locations using both static depots (MPFA$_{global\_static}$) and dynamic depots (MPFA$_{global\_dynamic}$).

We test how quickly resources are collected using the five algorithms (CPFA, MPFA$_{static}$, MPFA$_{global\_static}$, MPFA$_{dynamic}$, MPFA$_{global\_dynamic}$) across different distributions of resources. We observe how much the mobile depots improve swarm foraging performance, specifically: *i*) the time required to collect a fixed fraction of the resources (*foraging time*), *ii*) the time required to detect and avoid collisions with other robots (*collision time*), *iii*) the time that a robot spends searching for resources (*search time*), and *iv*) the time that a robot spends traveling to and from a depot when collecting resources (*travel time*). We show that our proposed algorithm, MPFA$_{dynamic}$, outperforms both the CPFA and the MPFA$_{static}$ on all performance criteria. We also show that MPFA$_{dynamic}$ performs approximately as well as MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$ without depending on global communication. This is a significant advantage of MPFA$_{dynamic}$ because global information is costly to obtain, and reliance on centralized communication is a single point of failure and efficiency bottleneck.

We also compare the scalability of the five algorithms by increasing the number of robots in the swarm and the size of the experimental arena. Our results show that MPFA$_{dynamic}$ has better scalability than the other four algorithms: increasing the arena size has a smaller negative effect on the foraging time of swarms using MPFA$_{dynamic}$, and increasing swarm size in a large arena has a larger positive effect on the foraging time of those swarms. In addition, we implement the MPFA$_{dynamic}$ with depots that transport their contents to a central depot, thus completing the central place foraging task. We compare this implementation to the CPFA.

Finally, we demonstrate how we can use our existing ROS/Gazebo simulation and Swarmie hardware for the NASA Swarmathon competition (Secor, 2016; Ackerman et al., 2018) to implement the dynamic MPFA in a physical robot swarm.

5.4 Related Work

5.4.1 Central-Place Foraging

Central-place foraging is a canonical collective task commonly studied in swarm robotics (Şahin, 2005; Brambilla et al., 2013). Robots depart from a centrally-placed depot to search for resources and return to this central place to deliver resources. The central-place foraging task can be instantiated into a number of real-world target collection applications, including crop harvesting (Bac et al., 2014; Sebbane, 2012) and extra-planetary resource collection (Brooks and Flynn, 1989; Landis, 2004; Fink et al., 2005).

In prior work, Hecker and Moses (2015) presented the central-place foraging algorithm (CPFA), which was designed to emulate seed-harvester ant behaviors governing memory, communication, and movement. The error-tolerance, flexibility, and scalability of the CPFA were evaluated on both simulated and real robot swarms. Hecker and Moses used a genetic algorithm (GA) to evolve foraging strategies that were tolerant of real-world sensing and navigation error, flexible for a variety of target distributions, and scalable to large swarm sizes.

The behaviors of the CPFA emulate harvester ant foraging that maximizes the number of resources collected in short foraging time periods (Flanagan et al., 2012; Gordon and Kulig, 1996), but is not designed for complete target collection. The foraging efficiency of the CPFA was recently compared to the distributed deterministic spiral algorithm (DDSA), a deterministic benchmark for central-place foraging (Fricke et al., 2016) that is designed to collect the nearest resources first. Results showed that robot swarms using the DDSA were faster at complete collection tasks than swarms using the CPFA.

However, the CPFA outperformed the DDSA by collecting more resources in fixed time windows for large swarms with more than 20 robots. The deterministic DDSA suffered from more robot collisions in more crowded environments. Since our goal for the MPFA is to increase foraging rates in large swarms, we build upon and compare to the CPFA in this work. We also focus on collecting resources quickly rather than complete target collection.

Although the CPFA is more scalable than the DDSA, CPFA swarms also exhibited diminishing returns as swarm size increased (i.e. sublinear scaling of foraging rate per robot given larger numbers of robots in the swarm). Diminishing returns are expected for central place foraging because robots in larger swarms on average travel farther to collect more resources, and there are more collisions given more robots. As shown in (Lu et al., 2016a), the MPFA$_{static}$ mitigates those effects. We show in this work that adding dynamic depots to the MPFA further mitigates scaling limitations.

5.4.2 Multiple-Place Foraging

Previous work has demonstrated that a single, central depot cannot serve a large number of robots efficiently due to long travel times (Hecker and Moses, 2015) and heavy crowding (Fricke et al., 2016). To mitigate this issue, we proposed the multiple-place foraging algorithm (MPFA) with multiple static depots, where robots are programmed to always return to the depot closest to the position of the target that the robot has found (Lu et al., 2016a,b).

The MPFA was primarily inspired by behaviors observed in groups of insects and primates, as well as the immune system. For example, polydomous colonies of Argentine ants are comprised of multiple nests spanning hundreds of square meters (Flanagan et al., 2013; Schmolke, 2009); additionally, a study (Tindo et al., 2008) showed that wasps living in multiple nests have greater survival rates and increased productivity. (Chapman et al., 1989) showed that communities of spider monkeys can be also considered as multiple central place foragers (MCPF), where monkeys select a sleeping site close to current feeding areas, and the MCPF strategy entails the lowest travel costs. In another biological system, (Banerjee and Moses, 2010b) showed that the decentralized, sub-modular nature of the immune system increases the foraging efficiency of immune cells that aggregate in lymph nodes distributed throughout the body. These dispersed aggregation points (analogous to multiple nests) speed up immune response rates, particularly in large animals that may have trillions of immune cells. Recently dynamic lymph nodes that appear near sites of infection have been discovered (Moyron-Quiroz et al., 2004), motivating the use of depots as dynamic aggregation points for robotic foraging.

The use of dynamic docks is introduced in the related work (Couture-Beil and Vaughan, 2009). That work demonstrates that mobile docks mitigate the spatial interference and improve overall task performance when mobile robots execute a transportation task and periodically recharge from a docking station.

Multiple-place foraging also resembles the task allocation of global courier and delivery services, which use many distributed stores to collect and deliver packages efficiently. Several studies on task allocation in robot swarms have used biologically-inspired approaches in the deployment of homogeneous swarms of robots to multiple sites (Halász et al., 2007; Berman et al., 2008; Hsieh et al., 2008). These robots autonomously redistribute themselves among the candidate sites to ensure task completion by optimized stochastic control policies. In general, each swarm is modeled as a hybrid system where agents switch between maximum transfer rates and constant transition rates.

5.4.3 Foundations of the MPFA

In our original implementation of the MPFA (Lu et al., 2016a,b), robots were initially assigned in equal numbers to static collection points called nests. Nests were evenly placed in the environment, i.e. given 4 nests, each was placed at the center of one quadrant of a foraging arena with 1/4 of the robots assigned to each nest. The robots could autonomously switch to other nests as they foraged. If the location of a found target was closer to another nest, the robot (which had traveled a long distance from its initial depot and discovered this target) delivered this target to the closer depot. The transition from one depot to another one is shown in Fig. 5.3.

The use of multiple collection depots is the fundamental difference between Hecker and Moses' CPFA and the MPFA; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of multiple depots on swarm foraging efficiency.

The CPFA

There are several essential features of the CPFA that make it possible to implement the MPFA$_{dynamic}$. The CPFA implements site fidelity in which a robot remembers and returns to the location where it last found resources. The CPFA implements pheromone waypoints as a list of target-rich locations that have been found by robots. Depots report the list of waypoints to robots when they drop off resources.

When a robot finds a target, it senses the local density of resources and then uses that information to determine whether to use site fidelity to return to the location and whether to communicate that information to other robots by reporting a pheromone waypoint to its depot.

How site fidelity, pheromone waypoints and other details of the CPFA are implemented is described below in Section 5.5 and Algorithm 1 (where in line 7, the closest depot is always the single central depot in the case of the CPFA).

A final important feature of the CPFA in its implementation in real robots is the ability to reliably return to a depot. The CPFA and MPFA rely on the use of beacons that are detectable by any nearby robots. Our experiments with physical iAnt robots running the CPFA demonstrate that a light is an effective beacon that allows robots to reliably return to their nest (Hecker and Moses, 2015). There are alternative beacons that can ensure that robots can reliably locate depots and other important locations. For example, colored LEDs on robots (Nouyan et al., 2009), speaker-induced sound gradients (Nurzaman et al., 2009), and images such as fiducials or roundels (Bezzo et al., 2015) can be used to mark important locations in space to which physical robots can reliably return.

The MPFA$_{static}$

The behavior of an individual robot in an MPFA foraging round is shown in Fig. 5.1. Each robot transitions through a series of states as it forages for resources. The states and transitions emulate foraging behaviors of ants. The MPFA differs from the CPFA in that the robots return to the closest depot in steps 4 and 5.

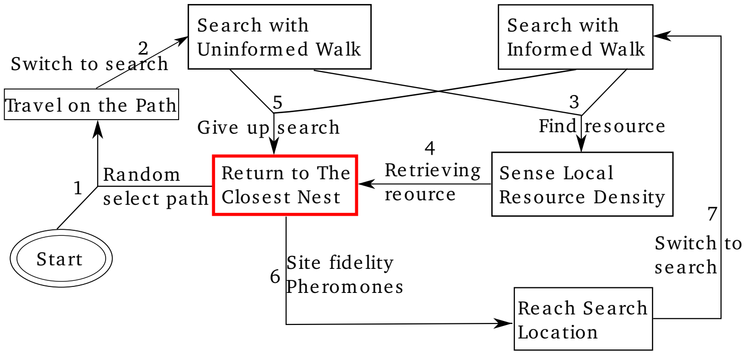


Figure 5.1: The flow chart of an individual robot's behavior following the MPFA during an experiment

Robots initially disperse from depots and follow randomly selected travel paths (step 1). Upon reaching the end of the travel path, robots switch to searching for resources using an uninformed correlated random walk (in which the robot has no knowledge of target locations) observed in ants (step 2) (Fewell, 1990). Robots navigate home to the depot closest to them after they collect a target (step 4) or give up searching (step 5) (as described in ants in (Crist and MacMahon, 1991)). The search cycle for an individual robot foraging using uninformed search is shown in Fig. 5.2.

Robots that discover a target will sense the local target density before returning to their local depot (step 3 and step 4) (Hölldobler, 1976). The density is the number of resources sensed in the local region by robots. The size of the region a robot can detect is described in Subsection 5.6.1. An individual robot may remember the location of a previously found target and repeatedly return to the same location, a process called *site fidelity* in ants (Beverly et al., 2009). Robots can also communicate

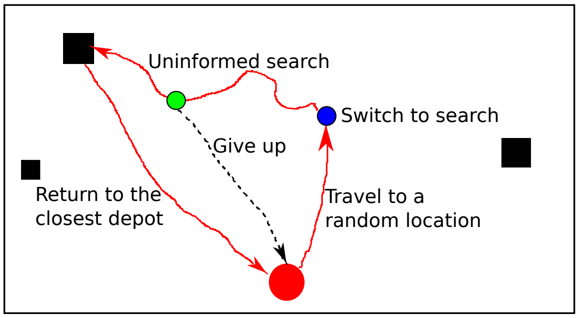


Figure 5.2: A single cycle of uninformed search. Four states of a robot in the cycle are shown. A robot departs from a depot (large red circle), travels to a random location, and switches to searching using an uninformed random walk (dark blue circle). If the robot finds a target pile (largest black square), then it collects one target and delivers it to the closest depot. The robot also has a probability of giving up searching (bright green circle) and returning to the closest depot without finding a target

using pheromones (Sumpter and Beekman, 2003; Jackson et al., 2007) which are simulated as artificial waypoints (Campo et al., 2010) to recruit robots to known clusters of resources. This is also discussed in Subsection 5.5.1. Robots that return to a previously found target site using site fidelity or pheromone recruitment (step 6) will search the target site thoroughly using an informed correlated random walk (step 7). The search behaviors for an individual robot foraging using informed search is shown in Fig. 5.3.

The search strategy is evolved by a genetic algorithm (GA); all robots use the same strategy, but make decisions probabilistically based on the interaction with the environment. Although robots are able to depart from and return to the nearest depot, robots still search globally, meaning that they are able to travel across in the entire arena.

As in the CPFA, pheromone trails are simulated using pheromone waypoints. Different from the CPFA, pheromone waypoints are only reported to the closest

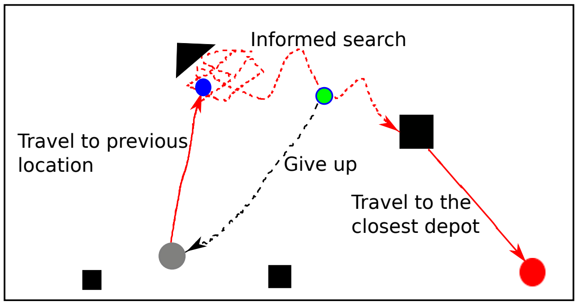


Figure 5.3: A single cycle of informed search. Five states of a robot are shown. A robot departs from a depot (large gray circle) and travels to the previous location (dark blue circle), and switches to searching using an informed correlated walk. If it finds a target pile (largest black square), then it collects one target and delivers it to the closest depot (red circle in the lower right). The robot also has a probability of giving up searching (light green circle) and returning to the closest depot without finding a target (red circle)

depot to the robot when it arrives at the depot. Robots can only send and receive pheromone waypoints when they are returning to a depot.

We use an exponential decay function with a decay rate selected by the GA to simulate the pheromone decay process. After a certain amount of time, the pheromone waypoint will have decayed below a threshold and will be removed from the depot's list. When a robot arrives at the depot, it will probabilistically select a waypoint from that depot's list and travel to the location of the waypoint. The robot may also probabilistically choose to locally share information by sending pheromone waypoints to its current depot. Unlike the CPFA, the pheromone waypoints associated with a given depot are only locally available to robots returning to that depot.

Since robots always return to the closest depot with a found target, the sensed information relevant to a given target neighborhood is always associated with the depot closest to the position of the identified neighborhood. Thus, the robots only travel from the closest depot to any given pheromone waypoint.

In our recent work (Lu et al., 2016a), we conducted simulated experiments with the MPFA using multiple static depots (2, 4, 8, and 16). We ran the experiments with 256 resources and 24 robots in a 10 × 10 m (i.e. 10 meters wide by 10 meters long) arena. The results showed that the MPFA produces higher foraging rates and lower average travel time compared to the CPFA. Increasing the number of depots increases the foraging rate of the swarm and decreases the required travel time per target collected, while the search time per target collected is independent of the number of depots. In most experiments, 4 depots led to significantly faster foraging than the CPFA or 2 depots, but they were indistinguishable from 8 depots, and so we focus on experiments with 4 depots in this paper. We note that determining the optimal number of depots for a given number of robots, resources and arena sizes is itself an interesting question that we leave to future work.

Because pheromone waypoints are distributed across multiple depots, MPFA swarms require less communication among robots, and individual robot spends less time traveling back to the closest depot to make use of the information. In contrast, CPFA swarms use pheromone waypoints that are globally available to the entire swarm; these robots, therefore, have access to more information, but individual robots take longer to travel back to the central depot and use the information. The GA balances these trade-offs automatically by tuning the search strategies and optimizing the performance of each swarm, resulting in systematic changes in parameters governing pheromone laying and distance traveled from the depot as more depots are added.

In other recent work (Lu et al., 2016b), we compared the ability of the MPFA and the CPFA to maintain foraging efficiency as swarm size and target number increase. We increased the size of the swarm (4, 8, 16, 32, and 64 robots given 1024 resources) to test scalability and the number of resources (128, 256, 512, 1024, and 2048 resources given 32 robots) to test adaptability to different target densities.

The MPFA had higher foraging efficiency than the CPFA under increased swarm size and target number. Furthermore, robots using the MPFA spent less time avoiding collisions and required less travel time to collect each target.

5.5 Methods

Previous MPFA experiments (Lu et al., 2016a,b) were conducted using uniformly-spaced static depots, which outperformed central-place foraging swarms, but were not capable of dynamically adapting to different target distributions. In this work, we aim to further improve swarm foraging performance with depots that move to the centroid of known nearby resources in order to minimize the time and distance for foraging robots to transport those resources.

If all of the positions of the resources are known, then we can use this positional information to calculate the optimal location of depots to minimize travel distance to all resources. This problem is analogous to clustering resources based on their distances to the closest depot, where the sum of distances between resources to the center of the cluster is minimum.

Given the locations of all resources in the arena, the k-means++ clustering algorithm (Arthur and Vassilvitskii, 2007) will calculate the locations of depots to minimize the travel distance required to collect all resources. Fig. 5.4 shows an example of a dynamically allocated depot, in which six piles of resources are classified into four clusters and four depots are placed at the centroids of these clusters. This implementation would require global knowledge of all target locations, which violates one of the key features of swarm robotics systems: all sensing and communication must be local (Brambilla et al., 2013).

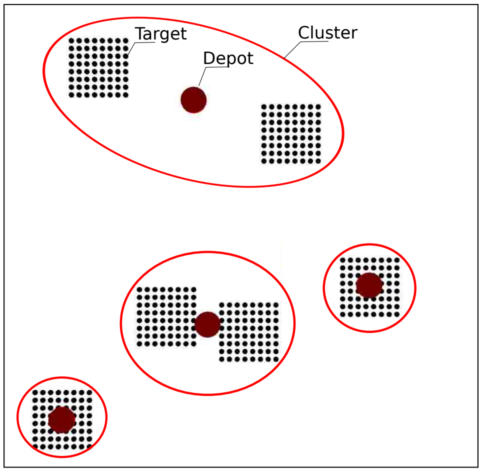


Figure 5.4: An example of a dynamically allocated depot using the k-means++ clustering algorithm. The resources (black squares) are classified into four clusters (red ellipses). Depots (dark red solid circles) are placed at the centroids of these clusters

We use globally informed MPFA algorithms to provide points of comparison for our proposed multiple-place foraging algorithm with dynamic depots (MPFA$_{dynamic}$), an extension to our recent work in which depots move to new locations based on the locations of the resources sensed by robots. Depots always move to the centroid of recently sensed resources, which are maintained in a list and updated whenever site fidelity or pheromone waypoints are used. If site fidelity is not used, or if pheromone waypoints decay, then those sensed resources are removed from the list and no longer contribute to the dynamic calculation of the depot's centroid.

The use of mobile depots is the fundamental difference between MPFA$_{static}$ and MPFA$_{dynamic}$; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of mobile depots on foraging efficiency.

As in MPFA$_{static}$, depots are initially distributed uniformly in MPFA$_{dynamic}$, and robots are evenly distributed to each depot. Depots move to new locations based on the positional information of observed resources sensed by foraging robots. Fig. 5.5 shows how a depot moves based on the sensed positional information of resources reported by foraging robots.

We assume robots can sense resources within camera range, but cannot precisely measure the positions of these resources. Therefore, a robot only reports its current position and the number of resources detected; the robot's current position approximates the centroid of the resources that it has detected. Each depot is allocated to the centroid c_t of the sensed resources at time t, where c_t is defined by Eq. (5.1):

$$c_t = \frac{1}{N}\sum_{i=1}^{N} w_i p_i \tag{5.1}$$

where w_i is the number of sensed resources at location p_i, and N is the total number of different locations where robots have sensed resources.

5.5.1 Implementation of Robot Controllers

Our robots mimic seed-harvester ant behaviors that have evolved over millions of years. We encode these behaviors into a robot controller (see Algorithm 1) using the same set of seven real-valued parameters that define the CPFA (see Table 5.1) specifying movement, sensing, and communication:

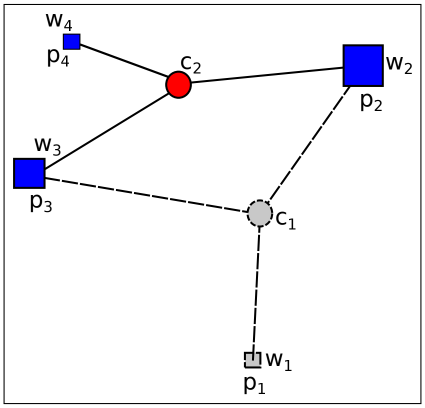


Figure 5.5: Depot movement in MPFA$_{dynamic}$. A depot (gray circle) is at the centroid c_1 of the sensed resources (dark blue squares) at positions p_1, p_2, and p_3, where w_1, w_2, and w_3 are the number of resources sensed by robots at each position, respectively. After some time, if resources at position p_1 are completely collected by robots, then the pheromone waypoints at p_1 will decay. If, at the same time, w_4 resources are sensed at a new location p_4, then the depot will move to the centroid c_2 of the sensed resources (red circle) at positions p_2, p_3, and p_4

Table 5.1: Parameters for robot controllers

Parameter	Description	Initialization
σ	Uniformed search variation	$\mathcal{U}(0, \pi)$
p_s	Prob. of switching to search	$\mathcal{U}(0, 1)$
p_r	Prob. of giving up search	$\mathcal{U}(0, 1)$
λ_{id}	Rate of informed search decay	$Exp(0.2)$
λ_{sf}	Rate of following site fidelity	$\mathcal{U}(0, 20)$
λ_{lp}	Rate of laying pheromone	$\mathcal{U}(0, 20)$
λ_{pd}	Rate of pheromone decay	$Exp(0.1)$

Uninformed search variation: Uninformed robots forage using a correlated random walk with fixed step length and direction $\theta_t = \mathcal{N}(\theta_{t-1}, \sigma)$, where θ_{t-1} is the turning angle from the previous step, and σ is the uninformed search variation (or standard deviation), which determines the turning angle of the next step.

Algorithm 1 Multiple-Place Foraging Algorithm

```
Disperse from depot to random location
while experiment running do
     Conduct uninformed correlated random walk
    if target found then
        Collect it
        Sense resources $c$ near current location $l_f$
        Return to the closest depot to deliver target
        if $\textsc{Pois}(c, \lambda_{lp}) > U(0, 1)$ then
            Lay pheromone to $l_f$
        end if
        if $\textsc{Pois}(c, \lambda_{sf}) > U(0, 1)$ then
            Return to $l_f$
            Conduct informed correlated random walk
        else if pheromone found then
            Travel to pheromone location $l_p$
            Conduct informed correlated random walk
        else
            Choose new random location
        end if
    end if
    end while
```

Probability of switching to search: Robots start at a depot and select a direction θ from a uniform random distribution $\mathcal{U}(0, 1)$, then travel in this direction away from the depot (see Fig. 5.2). Robots have a probability p_s of switching to an uninformed correlated random walk, where higher values of p_s indicate shorter travel distances from the depot.

Probability of giving up search: At each step of the correlated random walk, robots that have not discovered a target may give up searching and return to the closest depot with probability p_r.

Rate of informed search decay: If robots return to a previous location via site fidelity or pheromone waypoint, they search using an informed correlated random walk (see Fig. 5.3), with standard deviation $\hat{\sigma}$ defined by Eq. (5.2):

$$\hat{\sigma} = \sigma + (2\pi - \sigma)e^{-\lambda_{id}t} \tag{5.2}$$

As time t increases, $\hat{\sigma}$ decays to σ, producing an initially undirected and localized search that becomes more correlated over time. This time decay allows robots to search locally where they expect to find a target, but to straighten their path and move to another location if no target is found.

Rate of following site fidelity: The probability of a robot returning to a previous target location via site fidelity is governed by the Poisson cumulative distribution function (CDF) defined by Eq. (5.3):

$$\mathrm{POIS}(k, \lambda_{sf}) = e^{-\lambda_{sf}} \sum_{i=0}^{\lfloor k \rfloor} \frac{\lambda_{sf}^{i}}{i!} \tag{5.3}$$

where k is the number of additional resources detected in a previous location and the parameter λ_{sf} is the average number of detected resources. The Poisson CDF models the probability of following site fidelity given the number of detected resources k appropriately. The probability is highest when $k = \lambda_{sf}$. Robots return to previous locations via site fidelity if the parameterized Poisson CDF exceeds a uniform random value, $\mathrm{POIS}(k, \lambda_{sf}) > \mathcal{U}(0, 1)$, simulating a random sampling process that is weighted by the probability of following site fidelity for a given k. Otherwise, robots follow pheromone waypoints to previous target locations if pheromones are available. If no pheromone exists, robots return to traveling and searching using the uninformed correlated random walk.

Rate of laying pheromone: The probability of creating a pheromone waypoint is also governed by the Poisson CDF in Eq. (5.3). Robots create waypoints for previous target locations if $\mathrm{POIS}(k, \lambda_{lp}) > \mathcal{U}(0, 1)$, where k is also the number of resources detected in a previous location.

Rate of pheromone decay: Pheromone waypoint strength γ decays exponentially over time t as defined by Eq. (5.4):

$$\gamma = e^{-\lambda_{pd} t} \tag{5.4}$$

5.5.2 Evolving Swarm Behavior

The parameters of robot controllers are optimized using a genetic algorithm (GA) to optimize the collective behavior of the entire robot swarm, where every robot in the swarm uses the same controller. The controller is evolved in one set of simulations and evaluated in another set of simulations which are replicated 100 times. We run each foraging algorithm until the robot swarm collects the expected percentage of resources. Fitness is simply defined as the number of resources collected in a specified foraging time. In (Hecker and Moses, 2015) the foraging time was set to 1 hour.

There are an uncountable number of foraging strategies that can be defined by the real-valued parameters of the CPFA and MPFA. Given 100 real values of each parameter, there would be 100^7 possible strategies. Additionally, the online decision making of each robot depends on interactions with environmental conditions. For example, following site fidelity is determined by the condition of $\mathrm{POIS}(k, \lambda_{lp}) > \mathcal{U}(0, 1)$ as described in Subsection 5.5.1. The sampled value from $\mathcal{U}(0, 1)$ is random at each time, and the decision to use site fidelity depends on the value of k and the sampled random value. The GA provides a way to sample both parameter space and the effectiveness of the foraging algorithm evaluated in different environmental conditions.

The parameters in Table 5.1 are independently evolved 16 times in order to generate 16 independent foraging strategies for each of the five foraging algorithms in each target distribution. Thus we have a total of 240 separate evolutionary runs (3 distributions × 5 algorithms × 16 replicates). Each of these evolutionary experiments follows the process described in Experiment 1 in Section 5.6.

In (Hecker and Moses, 2015) we demonstrated that the evolved CPFA controllers could be effectively transferred into physical robots, a process also described in related work (Nelson et al., 2004; Singh and Parhi, 2011). Such controllers could be effectively tuned by the GA to mitigate the real-world error inherent in physical robots (Hecker et al., 2013). We describe steps toward similarly implementing $\mathrm{MPFA}_{dynamic}$ in real robots in Subsection 5.8.3.

We implement our GA using GAlib (Wall, 1996). For each generation of the GA, we evaluate each candidate set of 7 parameters on 10 different random placements of resources (see Fig. 5.6) to determine fitness. We use a 50% uniform crossover rate and a 5% Gaussian mutation rate with a standard deviation of 0.02, and elitism to keep the fittest parameter set.

We set termination criteria of the GA in order to hasten parameter convergence, running for a maximum of 100 generations. The GA terminates based on three criteria: the convergence of fitness values, the diversity of parameter sets, and the number of generations. The GA will stop if the fitness has converged and the diversity is low; otherwise, it will terminate after 100 generations.

In our GA, 89% of the evolutionary runs terminate based on the convergence of fitness and low diversity. Across 16 independent evolutionary runs, all evolved parameter sets were nearly equally fit: the standard deviation in fitness was at most 5% of the mean fitness value of these 16 independently evolved parameter sets. We chose the fittest parameter set to evaluate foraging performance.

5.6 Experimental Configuration

We conducted four sets of experiments using the swarm robot simulator Autonomous Robots Go Swarming (ARGoS) (Pinciroli et al., 2012) to evolve parameters and then test foraging performance. In the first set of experiments, we compared the foraging times of MPFA$_{dynamic}$ to the CPFA and MPFA$_{static}$, as well as to the two idealized versions of the MPFA that rely upon global knowledge of target locations to determine depot locations, MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$. These experiments were conducted with 24 robots in a 10 × 10 m arena.

In the second set of experiments, we tested the scalability of these algorithms to larger arena sizes. We examined the rate of increase in foraging times with increasing arena size (24 robots in arenas that increase from 10 × 10 m to 16 × 16 m). In the third set of experiments, we tested the performance of each algorithm in a very large arena (50 × 50 m) with 96 robots.

In the fourth set of experiments, we account for transportation by the mobile depots to a single central collection point. In these experiments, each of the four mobile depots is a modified robot that carries resources to a central collection point; thus, we also add 4 robots to the CPFA experiments, so foraging performance is evaluated with each having 28 robots that ultimately deliver resources to a central place.

For the first set of experiments, the parameters for the CPFA and MPFAs were each evolved separately as described in Section 5.5. We select the set of evolved parameters which has the shortest foraging time from the 16 sets of evolved parameters for the experiment. These sets of evolved parameters are subsequently used for the corresponding CPFA and MPFAs in the second, third and fourth experiments.

The configuration of the four sets of experiments is summarized in Table 5.2. Each experiment has one central depot in the CPFA, and four depots for each of the four MPFAs. In the fourth experiment, we include a central depot and four dynamic depots in the MPFA$_{dynamic}$ simulations.

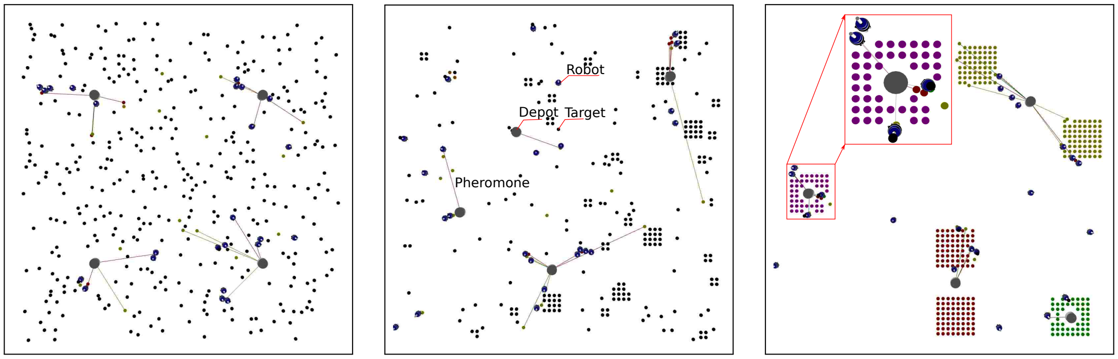
Table 5.2: Experimental configuration

Experiments	Arena width (m)	Number of robots
1	10	24
2	10, 12, 14, 16	24
3	50	96
4	10	28

Foraging time is measured as the time for the entire swarm to collect 88% of the 384 placed resources. This percentage was chosen because it is the inflection point in CPFA foraging performance (Hecker et al., 2015) after which there is an exponential increase in collection time and very high variance in performance due to the sparsity of remaining resources.

In the first set of experiments, we additionally measure the times for different components of the foraging time: travel time, search time, and collision time, each of which is described in Section 5.7.

Each of the five algorithms is tested on three different classes of target distribution: resources placed in a uniform random distribution, resources placed in a partially clustered distribution, and resources placed in a highly clustered distribution. Examples of resources placed in each distribution are shown in Figure 5.6.



(a) MPFA$_{static}$ with 24 robots, 4 uniformly distributed static depots and 384 randomly distributed resources

(b) MPFA$_{dynamic}$ with 24 robots, 4 mobile depots and 384 partially clustered resources

(c) MPFA$_{global\_static}$ with 24 robots, 4 globally distributed static depots by k-means++ clustering and 384 clustered resources

Figure 5.6: The placement of depots and resources in ARGoS. 384 resources (small points) and 24 robots (middle-sized points) are placed in a 10 × 10 m arena, and 4 depots (large points) are distributed. The resources are unclustered and spread in a uniform random distribution in (a), partially clustered in (b), and clustered into 6 equally-sized piles in (c). The colored rays indicate pheromone waypoints with different strength that eventually evaporate and disappear. A small area is magnified in (c) to show resources, robots, and a depot in the center

The partially clustered distribution uses a power law distribution of cluster sizes: 128 clusters that contain a single target, 32 clusters with 4 resources each, and 8 clusters with 16 resources each, for a total of 384 resources. This power law distribution of cluster sizes emulates that of many natural resource distributions in real-world environments (Ritchie, 2009). The fully clustered distribution has 6 clusters of 64 resources each.

Each experiment is replicated 100 times. For each replicate, the individual resources, or centers of target clusters, are chosen at random so that each replicate has a different target placement consistent with the distribution for that experiment. Thus, there are 1500 experimental runs (3 distributions × 5 algorithms × 100 replicates) for the first set of experiments, 6000 experimental runs (one for each of 4 arena sizes) for the second set of experiments, 1500 runs for the third set of experiments, and 600 runs for the fourth set of experiments, for a total of 9600 separate experimental runs.

5.6.1 ARGoS Implementation

Our implementation includes a C++-based robot controller library, and an XML configuration file. The C++ controller specifies the robot's functionality and interaction with the ARGoS environment, while the XML file contains all of the information to set up the size of arena, the type of robots, the physics engines, the parameters of robot controllers, the simulation accuracy, and the distributions of resources, depots, and robots. Source code is available on GitHub[1], and demonstration videos are available on our YouTube playlist[2].

[1] https://github.com/BCLab-UNM/MPFA

[2] https://tinyurl.com/y3kb3e6w

We use the ARGoS 8.5 cm radius foot-bots to represent our robots with a movement speed of 16 cm/s, while the movement speed of a depot is set to be the same. The step size of the simulation is 32 ticks per second, which was chosen to balance simulation accuracy and speed. Depots have a 15 cm radius and resources are cylinders with a 5 cm radius. The distance robots can sense resources is $2\sqrt{2}$ times the target radius.

Each pheromone trail is represented by a starting waypoint and an ending waypoint at a depot. Waypoints provide positional information maintained in lists in which pheromone strength of each waypoint decreases exponentially over time, as described by Eq. (5.4) above. Waypoints are removed once their values drop below a threshold of 0.001.

In the simulation, robots are able to identify and remember the exact locations of depots and the locations of sites visited in last foraging round, but this is not realistic for physical robot hardware. To test potential pitfalls of transferring the behavior of simulated robots to physical robots, we simulate sensor errors that reflect those of iAnt robots.

Following the method used by (Fricke et al., 2016), we simulate sensor error by applying Gaussian noise when robots attempt to return to a previous location via pheromones or site fidelity, mimicking that of the iAnt robots as described in (Hecker et al., 2013). The standard deviation around the intended location increases with the distance the robot travels to its intended destination position, p. This reflects the greater accumulation of odometry errors over longer distances. The distance p is multiplied by a noise coefficient, e, in order to change noise severity. Noise is generated by: $noise \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = e \times p$. For example, given the maximum travel distance of CPFA swarms to the corner of an arena, $p = 7.0$ m, and a noise coefficient of $e = 0.4$, returning robots will arrive within approximately 3 m of their goal destination 68% of the time. MPFA swarms, which have shorter average travel distances (as shown below), and therefore lower modeled error, will return to previous locations with higher accuracy.

5.7 Results

We compare MPFA$_{dynamic}$ to the CPFA, MPFA$_{static}$, MPFA$_{global\_static}$, and MPFA$_{global\_dynamic}$. We replicate each experiment in 100 trials and report the median time for the swarm to collect resources in each experiment. We also examine several components of foraging time: travel time, search time, and collision time. We test the scalability of the algorithms by increasing the arena size and swarm size and examining the trends in foraging time. We demonstrate that MPFA$_{dynamic}$ is faster than the CPFA and MPFA$_{static}$, and similar in performance to MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$. We present our results in notched box plots to show which results are statistically different. We used the Mann-Whitney U test to compare the results of the MPFA$_{dynamic}$ to each of the four other algorithms. The statistical significance is explicitly indicated by asterisks in figures ($p < 0.001$). Additionally, the notch on each plot indicates the 95% confidence interval of the medians. If the notches of two boxes do not overlap, this indicates a statistically significant difference between the medians.

5.7.1 Foraging Performance

Foraging Time

In our simulation, the foraging time of each swarm is the time required to collect 88% (as described above in Section 5.6) of the resources. The configuration of each experiment is shown in Table 5.2. Fig. 5.7 shows the time for each algorithm to collect 88% of the resources for three different classes of distributions of resources.

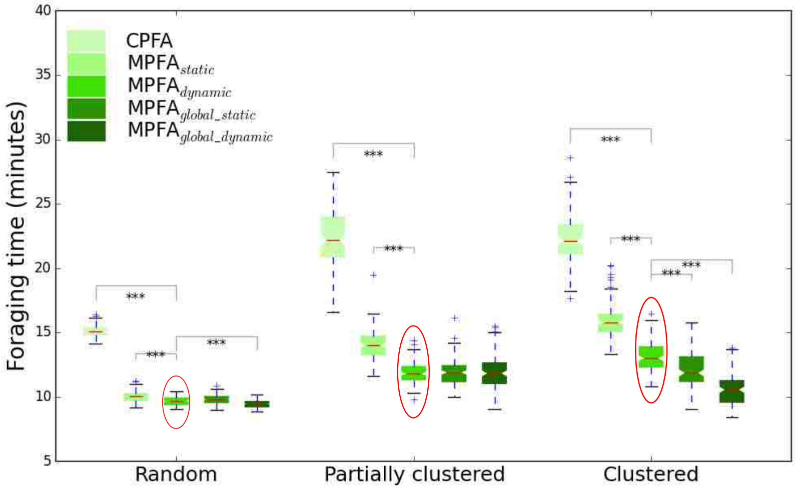


Figure 5.7: Foraging times for CPFA and MPFA swarms of 24 robots in a 10 × 10 m arena. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from MPFA$_{dynamic}$ which is emphasized by red ellipses. The performance of each algorithm is represented by a notched box plot in a different shade, ordered left to right, lightest to darkest in the same order indicated in the legend. The notches indicate the 95% confidence interval of the median so that overlapping ranges of the notches indicate statistically indistinguishable results at the $p = 0.05$ level

Our experiments show that MPFA$_{dynamic}$ outperforms the CPFA and the MPFA$_{static}$ in all three distributions. The MPFA$_{dynamic}$ is 47% faster than the CPFA in the partially clustered distribution and 18% faster than the MPFA$_{static}$ in the clustered distribution. Surprisingly, the MPFA$_{dynamic}$ is either faster than or statistically indistinguishable from both globally informed algorithms in the partially clustered and clustered distributions. It is slightly slower than MPFA$_{global\_dynamic}$ in the random distribution.

Robustness to Error

We examine the effect of localization error on foraging performance. Fig. 5.8) shows foraging time for swarms given simulated error with a noise coefficient 0.4. This error results in robots returning to pheromone or site fidelity waypoints at the far corner of a 10 × 10 m arena being normally distributed around the intended destination, with 68% of the robots within 3 m of the intended destination, a substantial amount of error when searching for resources that are 5 cm in radius. Our experiments show that the foraging times of all algorithms increase moderately (on average by 16%) with this level of error. However, MPFA$_{dynamic}$ still outperforms the CPFA and MPFA$_{static}$ in all three distributions with statistical significance levels similar to the error-free evaluations.

5.7.2 Search and Travel Time

Foraging time is composed of two distinct activities. When a robot departs from a depot, it travels to a location where it starts a localized search for resources. Once a target is discovered, the robot takes approximately the same *travel time* back to the depot as it took to travel to the search location.

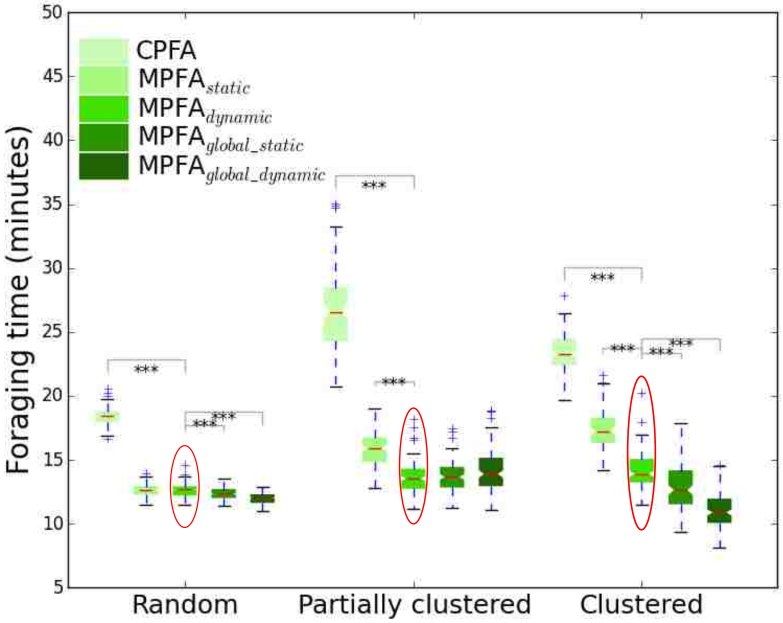


Figure 5.8: Foraging times for CPFA and MPFA swarms of 24 robots with noise $e = 0.4$ in a 10×10 m arena

We measure the total travel time and search time spent by all robots in the swarm. The summed travel and search time of all robots in each swarm are shown in Fig. 5.9. In the MPFA$_{dynamic}$, travel time is reduced in all cases. Compared to the CPFA, the MPFA$_{dynamic}$ is up to 62% faster (in the clustered distribution); compared to the MPFA$_{static}$ it is up to 30% faster (in the clustered distribution). Robots using the MPFA$_{dynamic}$ also search faster in all cases. Compared to the CPFA it is up to 51% faster (in the partially clustered distribution), and compared to the MPFA$_{static}$ (up to 13.6% faster in the partially clustered distribution). It is also faster than the globally informed MPFAs in the partially clustered distribution. It is slightly slower than MPFA$_{global\_dynamic}$ in the clustered distribution.

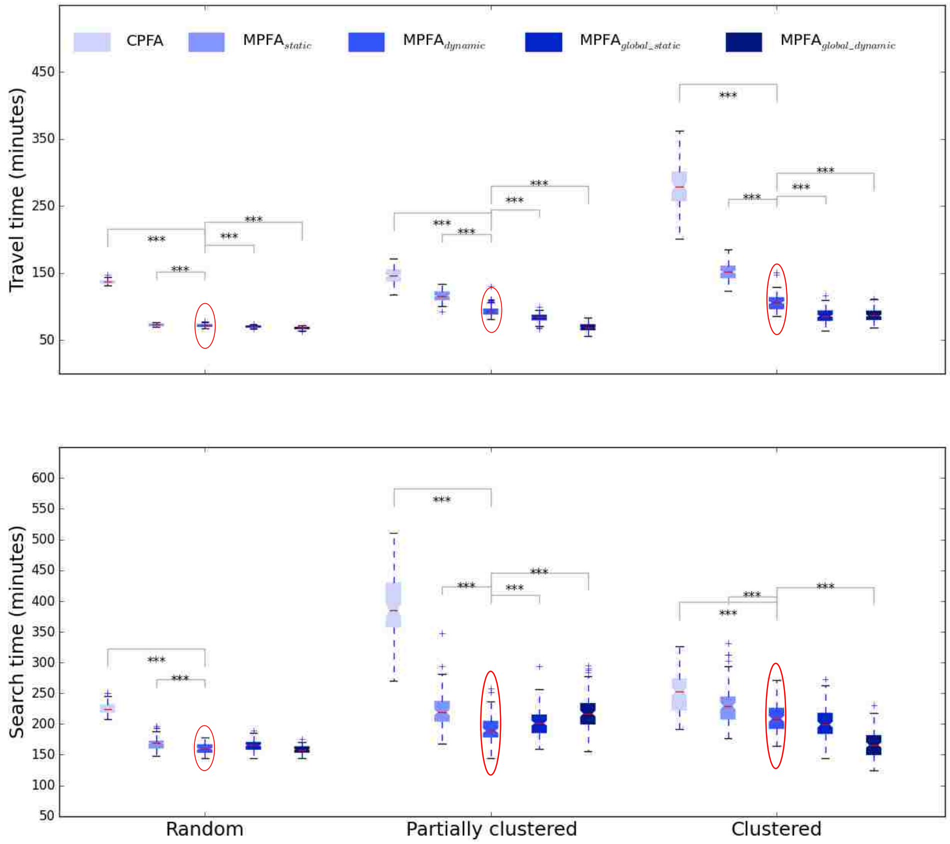


Figure 5.9: The search and travel time (per swarm) for the CPFA and MPFAs

5.7.3 Collision Time

In our simulation, if the distance between two robots is less than 25 cm, each robot will implement collision avoidance. Each robot senses the location of the other and turns left or right in order to avoid a collision, moving approximately 8 cm before resuming traveling. The collision avoidance takes time and will increase foraging times, particularly when the swarm size is large.

Collision time is the time spent to avoid a collision. The total collision time of each swarm is the sum of the total collision avoidance times for all robots in the swarm (shown in Fig. 5.10). The collision time for MPFA$_{dynamic}$ is less than the collision time for the CPFA in all cases, but it is not significantly different from the other variants of the MPFA. Not surprisingly, collision time is lowest in the random distribution where resources and robots are most dispersed, and highest in the clustered distribution where robots crowd around clustered target locations.

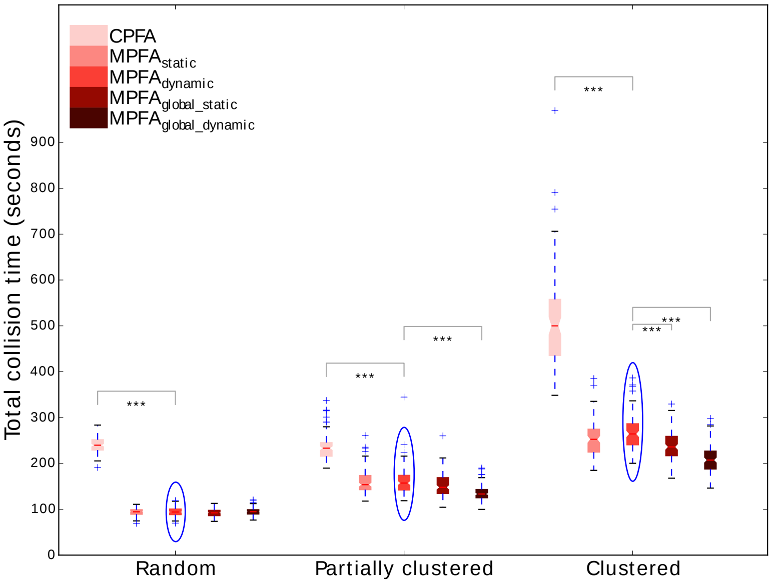


Figure 5.10: Total time spent (per swarm) avoiding collisions for the CPFA and MPFAs. The boxplot of MPFA$_{dynamic}$ is emphasized by blue ellipses

5.7.4 Scalability

We tested the foraging performance of MPFA$_{dynamic}$ with increased arena sizes and swarm sizes. Fig. 5.11 shows the foraging performance in different arena sizes. Not surprisingly, foraging time increases as the arena size increases. MPFA$_{dynamic}$ outperforms the CPFA and MPFA$_{static}$ in all arena sizes and all three distributions. Its performance is similar to MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$.

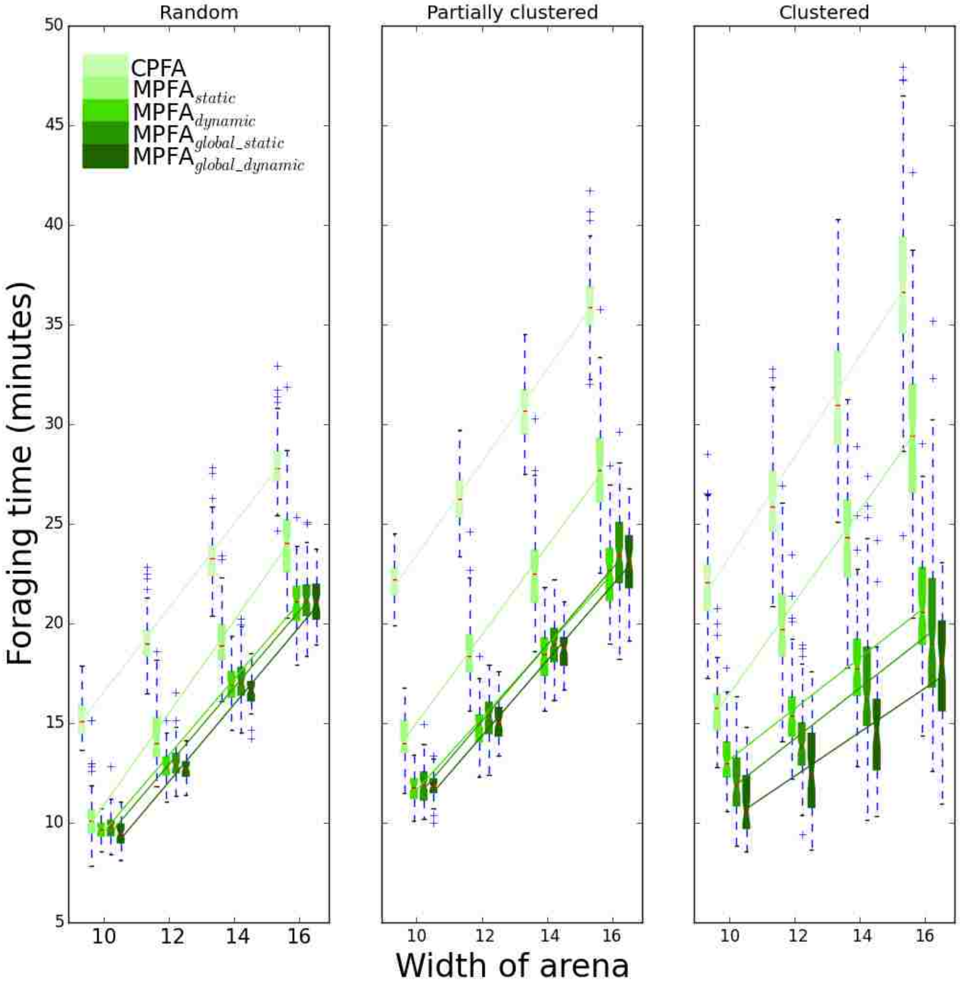


Figure 5.11: The foraging time for each swarm for increasing arena sizes. Results are for 100 trials and data for each swarm is shown by the box plot. The lines show the best-fit linear regression

The increase in foraging time appears to be linear with the length of the foraging arena. However, in the clustered target environment, MPFA$_{dynamic}$ ($slope = 2.55$), MPFA$_{global\_static}$ ($slope = 2.56$), and MPFA$_{global\_dynamic}$ ($slope = 2.21$) have improved scalability compared to the CPFA ($slope = 5.04$) and MPFA$_{static}$ ($slope = 4.61$) as evidenced by the more shallow increase in per-robot foraging time with arena size. The slope of the regression for MPFA$_{dynamic}$ is not significantly different from that of MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$.

To further test scalability, we create an arena 25 times larger (50×50 m) than the basic (10×10 m) arena and we measure foraging times for swarms of 96 robots.

Fig. 5.12 shows foraging performance in this larger arena. MPFA$_{dynamic}$ still outperforms the CPFA (up to 30% in the clustered distribution) and MPFA$_{static}$ (up to 13% in the clustered distribution) in most cases. The MPFA$_{dynamic}$ is either better than or statistically indistinguishable from the MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$ in all cases. These results suggest that the MPFA$_{dynamic}$ is particularly effective for very large swarms and foraging areas.

5.7.5 Transport to A Central Depot

Two caveats should be considered in interpreting the above comparisons of the MPFA algorithms to the CPFA. First, because we consider the mobile depots to be robotic agents, this means that the MPFA swarms have four more robots than the CPFA swarms. Second, in cases where the MPFA is used, but resources must ultimately be collected at a central location, the mobile depots would need to transport resources to a single central depot (as is done in the CPFA).

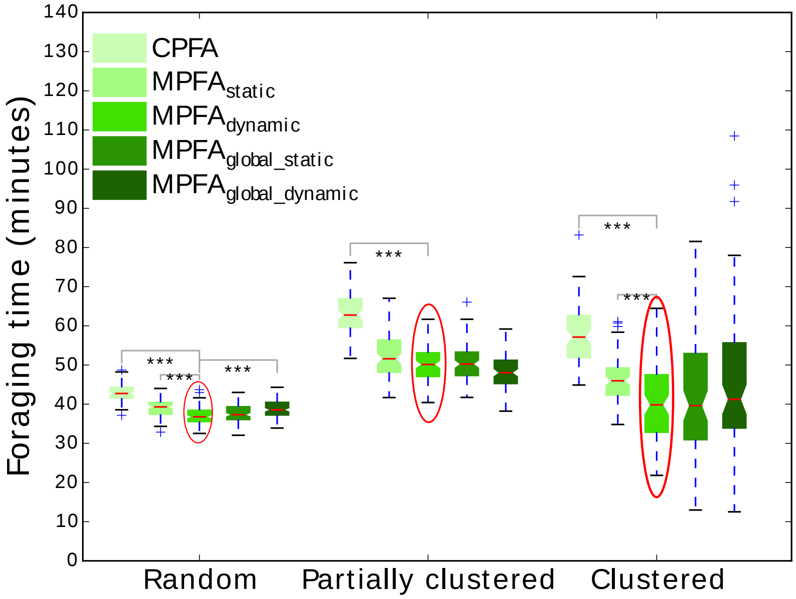


Figure 5.12: The foraging time for each swarm of 96 robots in 50 × 50 m arena. Results are for 100 replicates for each algorithm. Asterisks indicate a statistically significant difference ($p < 0.001$). The boxplot of MPFA$_{dynamic}$ is emphasized by red ellipses

To make a more fair experimental comparison, we added four robots to the CPFA swarm, and we modified MPFA$_{dynamic}$ so that when mobile depots are full (in this case containing 24 resources), they deliver those resources to a single central depot. Foraging robots carrying resources to that depot pause their motion while the depot is traveling to and from the central depot. A demonstration video is available on YouTube[3].

Fig. 5.13 compares the MPFA$_{dynamic}$ with central delivery to the CPFA. Central delivery increases the foraging time of the MPFA$_{dynamic}$ by 5.5% and adding 4 additional robots to the CPFA decreased foraging time by 11%.

[3] `https://tinyurl.com/yyzpkmy2`

However, the MPFA$_{dynamic}$ with central delivery is still significantly faster than the CPFA: 22% faster in the random distribution, 36% faster in the partially clustered, and 32% faster in the clustered distribution. Thus, even with central delivery, the MPFA is on average 30% faster than the CPFA.

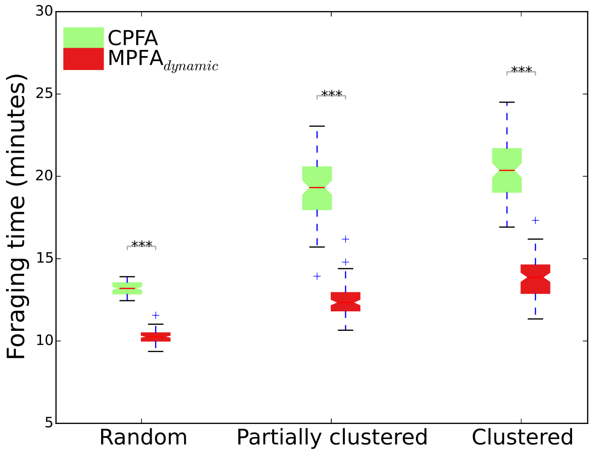


Figure 5.13: Foraging times for CPFA swarm of 28 robots and MPFA$_{dynamic}$ swarms of 24 robots in a 10 × 10 m arena. Depots deliver collected resources to the central placed depot when they have 24 resources. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from MPFA$_{dynamic}$

5.8 Discussion

This paper examines the foraging performance of swarms using the multiple-place foraging algorithm with dynamic depots (MPFA$_{dynamic}$). We test 4 variants of the multiple-place foraging algorithm and a central place foraging algorithm (the CPFA). Because these ant-inspired algorithms are designed for collecting resources quickly rather than for the complete collection of all resources, we report the time required to collect 88% of the available resources in each experiment.

In the first set of experiments with 24 robots in a 10 × 10 m arena (Fig. 5.7), the average foraging time of MPFA$_{dynamic}$ across the three target distributions is 41% faster than the centralized CPFA, and 13% faster than MPFA$_{static}$. Its foraging times are similar to MPFA$_{global\_static}$ and MPFA$_{global\_dynamic}$, illustrating that dynamic depots that respond only to local information are as effective as global methods that require more information to be collected and communicated.

Foraging times are reduced in all versions of the MPFA compared to the CPFA, primarily because travel times are dramatically reduced by an average of 49% over all three distributions. Travel times are reduced the most in partially clustered and clustered distributions, and in those distributions MPFA$_{dynamic}$ also has reduced travel times relative to MPFA$_{static}$ (see Fig. 5.9). The same comparisons are true for search time, but the differences are smaller: MPFA$_{dynamic}$ is 33% faster than the CPFA and 9% faster than MPFA$_{static}$ on average. Collision avoidance times are on average 47% lower for all versions of the MPFA compared to the CPFA (see Fig. 5.10). Since larger swarms produce more inter-robot collisions and reduce foraging performance, a more efficient collision avoidance strategy for reducing collision time will be included in future work, informed by the adaptive bucket-brigade foraging method introduced in (Lein and Vaughan, 2009).

In addition to having faster foraging times for all arena sizes and all target distributions, MPFA$_{dynamic}$ is also more scalable than the CPFA and MPFA$_{static}$. Fig. 5.11 shows that the increase in foraging times with arena size is smaller on the clustered distribution for MPFA$_{dynamic}$ ($slope = 2.55$) compared to the CPFA ($slope = 5.04$) and MPFA$_{static}$ ($slope = 4.61$). MPFA$_{dynamic}$ foraging times are particularly faster for large arenas and clustered resources (e.g., 21% faster than MPFA$_{static}$ in a 16×16 m arena, and 30% faster in the 50×50 m arena in Fig. 5.12).

We also demonstrated how the MPFA$_{dynamic}$ can be used to complete the central place foraging task faster than the CPFA. For these experiments, the mobile depots deliver their contents to a central depot when they are full, and the CPFA is given 4 additional robots for a more fair comparison. The transport time of a small number of trips to the central depot is minimal and has little effect on the total foraging time. Fig. 5.13 shows that the MPFA$_{dynamic}$ is still 30% faster than the CPFA.

Thus, by using mobile depots that adapt to local conditions, MPFA$_{dynamic}$ is an efficient and scalable solution that minimizes the central-place bottleneck of the CPFA and improves foraging times compared to MPFA$_{static}$ without requiring any global information.

5.8.1 Online Decision-Making in Response to Local Information

Real-time adaptive response is a key component of MPFA$_{dynamic}$. Foraging robots adaptively respond to the resources they detect in the environment by making a real-time decision to communicate pheromones or to return to a previous search location using site fidelity. Depots make real-time adjustments each time a foraging robot drops off a target in order to move toward the centroid of the known target locations. The CPFA and MPFA$_{static}$ are both effective algorithms (Hecker and Moses, 2015; Lu et al., 2016a); however, the additional real-time decision-making of mobile depots decreases foraging times in all of our experiments, and the decrease is greatest in the largest arenas and for clustered target distributions (Fig. 5.12).

MPFA$_{dynamic}$ is particularly effective compared to MPFA$_{static}$ for highly clustered resources. Foraging robots adaptively respond to clusters by using pheromones and site fidelity; in turn, depots respond to the observations of the foraging robots by moving closer to clusters of resources. Thus, both foragers and depots respond to the environment to reduce the time to collect resources. The adaptive communication of foragers reduces search time, and the adaptive movement of depots reduces travel time. Real-time adaptation to communicated information about target locations is particularly valuable when resources are highly clustered because each target found in a cluster confers more information about the location of other resources in that cluster (Flanagan et al., 2011).

The benefits of dynamic depot movement are likely to be even greater when resources are ephemeral, i.e. appearing and disappearing over time, and when the resources themselves are mobile because depots can move to new locations where resources appear so that they can be collected quickly (Levin, 2016).

In addition to real-time decision-making, robots also respond adaptively to their environments over evolutionary time. Our previous work showed that robots adjust dispersal parameters and the rate of communication to avoid overcrowding between depots and nearby piles when they are tested in environments with clustered resources (Lu et al., 2016a). This results in scalable algorithms, and scalability is improved further with MPFA$_{dynamic}$.

5.8.2 Broader Implications for Scalable Design

A fundamental problem in computer science is the design of scalable solutions that perform well as the problem size increases. As computational systems interact more with the environment in which they are situated, particularly if they navigate physical space using stochastic movement, they become increasingly analogous to biological systems (Kleinberg, 2007). In biology, scaling theory investigates how efficiently resources can be moved through spatial networks (West et al., 1997; Banavar et al., 2010; Savage et al., 2008). Scaling theory makes predictions beyond individual organisms, to explain the efficiency of ant colonies (Hou et al., 2010), societies (Moses and Brown, 2003; Brown et al., 2011), and even computer chip design (Moses et al., 2016).

MPFA$_{dynamic}$ offers a new perspective on the scaling problem. The use of multiple depots in the MPFA improves scaling compared to the CPFA, and having adaptive and dynamic mobile depots increases scalability even further. This advantage is particularly apparent when the resources to be transported are grouped into clusters, rather than randomly scattered, and when transport distances are very large (i.e., MPFA$_{dynamic}$ is nearly twice as fast as the CPFA and MPFA$_{static}$ for clustered resources in the largest 50×50 m arena as shown in Fig. 5.12). This suggests that adaptive mobile agents in robotic swarms can mitigate the inherent scaling inefficiencies of central-place transport. The experiments in Fig. 5.13 show that this holds even when the dispersed depots transport resources to a central nest.

The success of MPFA$_{dynamic}$ also provides insight into biological mechanisms that improve scalability. While most biological scaling theory focuses on fixed, centralized transport networks, there are biological systems that have features similar to the depots of the MPFA. For example, the immune system, with multiple lymph nodes distributed throughout the search space of an organism, results in a highly scalable immune response with trillions of cells (Banerjee and Moses, 2010a). Our prior works suggest that the partially distributed architecture of the immune system (one in which lymph nodes act as depots) is critical for overcoming the inherent scaling limitations of transporting resources (Moses and Banerjee, 2011).

There is also evidence of mobile depots in the largest colonies of ants: invasive Argentine ant colonies are composed of a network of mobile nests connected by trails, and the dynamic patterns of recruitment and allocation of foragers to nests increases foraging efficiency (Flanagan et al., 2013; Lanan, 2014). These examples suggest that in biological systems, as well as in robotic swarms, adaptive, decentralized and mobile aggregation points increase search efficiency. Thus, biological systems have evolved architectures with the same advantages of MPFA$_{dynamic}$: faster search and foraging, fewer collisions, and reduced travel time.

5.8.3 The Path to Implementation

Our simulations suggest that the MPFA$_{dynamic}$ is robust to the errors that we previously identified as important in our iAnt physical robots, namely error in returning to locations indicated by site fidelity or pheromone waypoints. When we included substantial error in our simulations (leading to robots being up to 3m away from intended destinations), it reduced foraging performance by an average of 16% (see Fig. 5.8) across all of the MPFA and CPFA experiments, but the MPFA continued to be faster than the CPFA.

However, we do not expect that foraging performance in real robots will be as fast as it is in the simulation. In order to implement multiple-place foraging with dynamic depots in a physical robot swarm, we will use our existing robot platform, designed by our lab for the NASA Swarmathon competition (Secor, 2016; Ackerman et al., 2018). The release code for the competition is available on GitHub[4]. Swarmathon robots are outfitted with a grasping apparatus that facilitates the pick up and drop off of target cubes (see Fig. 5.14). Structural modifications will be required to convert four Swarmathon robots into mobile depots capable of holding collected resources inside of a container.

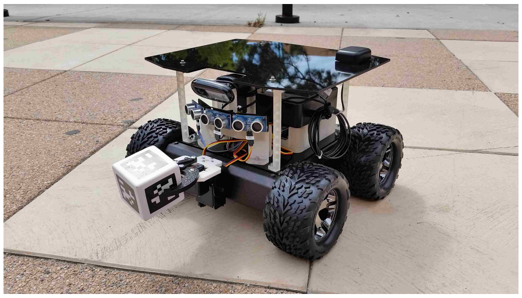


Figure 5.14: The physical robot on which components of the CPFA have been implemented

Swarmathon robots are considerably larger and more powerful than the foot-bots modeled in ARGoS. Swarmathon robots run the Robot Operating System (ROS), a distributed message-passing framework with an extensive, user-supported package that helps streamline algorithm implementation (Quigley et al., 2009). Other swarm algorithms, including the DDSA and components of the CPFA, have been implemented in ROS and subsequently tested in the multi-robot simulator Gazebo (Koenig and Howard, 2004). Based on our experience with these existing foraging algorithms, we implemented a dynamic depot with Swarmathon robots in Gazebo (Fig. 5.15). A demonstration video showing central place foraging in Gazebo and physical Swarmathon robots, as well as a mobile depot simulated in Gazebo is available[5]. The next step is making a straightforward extension to the simulation to include multiple

[4] https://github.com/BCLab-UNM/SwarmBaseCode-ROS

[5] https://tinyurl.com/y47j3hrc

depots implementing pheromone waypoints associated with each depot and centroid estimation by each depot in order to fully implement MPFA$_{dynamic}$.

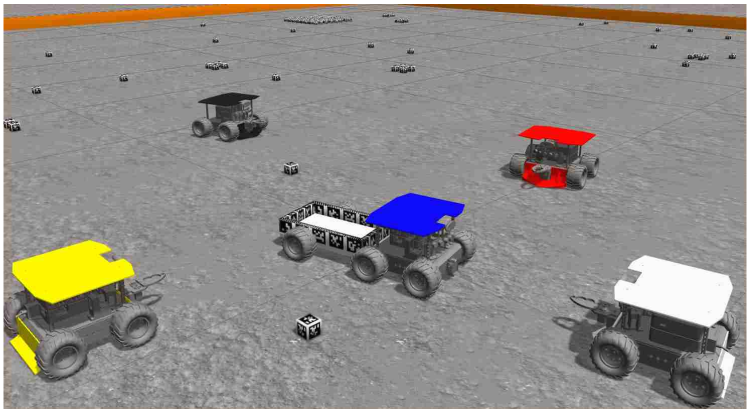


Figure 5.15: A mobile depot with blue cover and four foraging robots simulated in Gazebo

The biggest benefit of implementing the MPFA on the ROS and Gazebo system developed for the NASA Swarmathon is that code is very easily transferred from Gazebo onto the onboard Linux computer on the Swarmathon robots. The ease of this transfer is evidenced by the 19 college teams that successfully transferred their Gazebo code to up to 6 Swarmathon robots that operated in outdoor arenas up to 23 x 23 m for the Swarmathon competition (see Fig. 5.16). These teams showed that Swarmathon robots can reliably return to collection points, and implement site fidelity and recruitment to waypoints. Full implementation of the MPFA in 24 physical robots in an outdoor environment is the next step to demonstrate truly scalable foraging swarms of robots.

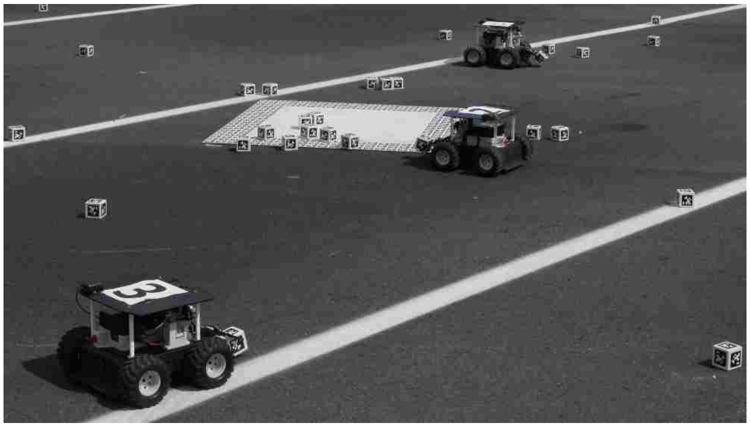


Figure 5.16: A swarm of 6 robots (3 shown) implementing central place foraging in a 23 x 23 m arena

5.9 Acknowledgements

This work was supported by a James S. McDonnell Foundation Complex Systems Scholar Award and NASA MUREP #NNX15AM14A funding for the Swarmathon. We thank the UNM Center for Advanced Research Computing for computational resources used in this work. We gratefully acknowledge members of the Moses Biological Computation Lab for their assistance with the dynamic multiple-place foraging swarm robotics project: thanks to G. Mathew Fricke for discussing the statistical analysis of our results and implementing MPFA in Gazebo, and to Antonio Griego for developing the CPFA algorithm in ARGoS. Thanks also to Carlo Pinciroli for discussing the implementation in ARGoS.

Chapter 6

A Bio-Inspired Hierarchical Branching Transportation Network

6.1 Publication Notes

Citation: Qi Lu and Melanie E. Moses. A Bio-Inspired Transportation Network for Scalable Swarm Foraging. International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 2019.

Received: April 8, 2019

Accepted as an extended abstract: June 21, 2019

6.2 Abstract

Scalability is a significant challenge for robot swarms. Generally, larger groups of cooperating robots produce more inter-robot collisions, and in swarm robot foraging, larger search arenas result in larger travel costs. This paper demonstrates a

scale-invariant swarm foraging algorithm that ensures that each robot finds and delivers resources to a central collection zone at the same rate, regardless of the size of the swarm or the search area. Dispersed mobile depots aggregate locally foraged resources and transport them to a central place via a hierarchical branching transportation network. This approach is inspired by ubiquitous fractal branching networks such as tree branches and mammal cardiovascular networks that deliver resources to cells and determine the scale and pace of life. We demonstrate that biological scaling laws predict how quickly robots forage in simulations of up to thousands of robots searching over thousands of square meters. We then use biological scaling predictions to determine the capacity of depot robots in order to overcome scaling constraints and produce scale-invariant robot swarms. We verify the predictions using ARGoS simulations.

6.3 Introduction

Natural swarms such as colonies of social insects and flocks of birds and fish have long served as inspiration for swarm robotics. Natural swarms have suggested strategies for generating collective behavior from individual actions. Foraging is the behavior of searching for resources (sometimes called food or targets) and transporting them to a specific collection zone. Foraging is a canonical swarm robotics task that is used in search and rescue, construction, transportation, agricultural harvesting, and planetary exploration (Winfield, 2009b; Gro and Dorigo, 2009; Yun and Rus, 2014; Bac et al., 2014; Fink et al., 2005). Biological systems also illustrate how collective systems can be adaptable, robust to individual failures, and scalable, particularly in swarm foraging where multiple robots are advantageous for collecting spatially dispersed resources. (Brambilla et al., 2013; Bonabeau et al., 1999; Kennedy and Eberhart, 2001; Şahin, 2005; Hecker and Moses, 2015).

Scalability of robot swarms has gained recent interest (Bonabeau et al., 1999; Kennedy and Eberhart, 2001; Şahin, 2005; Barca and Sekercioglu, 2013; Khaluf et al., 2017). A scalable foraging system should be effective in swarms ranging from tens to thousands of robots without reducing per robot foraging times. In this paper, we use scaling theory from biology to understand scaling constraints and then design a robot foraging system that overcomes these constraints.

In central place foraging, robots gather dispersed resources from a foraging arena and consolidate them in a single centrally-placed collection zone that robots depart from and return to in order to deposit resources (Liu et al., 2007; Hecker and Moses, 2015; Castello et al., 2016). We focus on central place foraging here, but our results generalize to other foraging tasks.

Two major problems limit scalability. First, large swarms with many robots produce more inter-robot collisions both during the search process and during the return of resources to a relatively small collection zone. The collisions lead to the phenomenon of *diminishing returns* as proposed by economists (Brue, 1993). Second, large foraging arenas require, on average, that robots travel further distances (requiring more time) to find resources and transport them to the central collection zone. When foraging in large areas, for example, collecting resources on the surface of Mars, or an ocean search and rescue operation, the search area can extend many kilometers, necessitating that robots travel very long distances.

Our goal is to design a scale-invariant foraging system in which this per-robot foraging rate is the same for all swarm sizes and arena sizes. We proposed the Multiple-Place Foraging Algorithm (MPFA) to address the problems of increased collisions and transport times as the foraging problem scales to larger sizes. The MPFA uses multiple collection zones dispersed in a foraging arena rather than one central collection zone. In the simplest implementation, collection zones are distributed uniformly across the arena and each robot returns to the closest collection zone to the place where it finds a resource. (Lu et al., 2019b) introduces dynamic depots that move to the centroid of recently collected resources, minimizing transport times and making the MPFA more adaptable to patchy and heterogeneous distributions of resources (Ritchie, 2009).

The MPFA improves scaling, but the time to transport resources from dispersed collection zones to a single location still results in diminishing returns. The bio-inspired hierarchical transportation network we propose here solves this problem. In this scale-invariant design, the per-robot foraging time is invariant with respect to arena size and swarm size. The transportation network draws inspiration from biological scaling theory that describes the scaling consequences of transporting resources from a central heart to dispersed cells via the mammal cardiovascular network (West et al., 1997; Banavar et al., 2010). The cost of large size is that resources take longer to transport through the system, which ultimately slows the cells of larger mammals. Thus, physiological rates (i.e., heart rate, growth rate, and reproductive rate) are systematically slower, and lifespan and gestation times are systematically longer, in large vs. small mammals.

We show that foraging robots are constrained by the same principles as plants and mammals. We derive scaling relationships for a 2D foraging area (rather than a 3D mammal volume). We use this scaling law to predict the transportation infrastructure required to maintain constant per-robot foraging rate with increasing swarm and arena size. We then simulate foraging using a hierarchical transportation network ($\mathrm{MPFA_T}$) composed of mobile depots with carrying capacities determined by the scaling theory. Our simulations show that this design overcomes scaling constraints resulting in nearly scale-invariant foraging. We scale the swarm size up to thousands of robots in arenas that are thousands of square meters in area. We test all of our algorithms in the Autonomous Robots Go Swarming (ARGoS) simulator using foot bots as a model robot. In prior work, we have implemented similar ARGoS experiments of the CPFA and MPFA in dozens of replicated experiments with physical hardware (Lu et al., 2019a), suggesting that the approach we outline here is feasible in physical robots.

The remainder of this chapter is organized as follows. The related work is introduced in Section 6.4. The similarities between cardiovascular systems and robot swarms in shown in Section 6.5. The scaling laws for foraging swarms are derived in Section 6.6. The experimental setup is described in Section 6.7. Section 6.8 shows the experimental results and Section 6.9 discusses the results.

6.4 Related Work

Though swarm robot foraging has been studied for decades (Winfield, 2009b; Barca and Sekercioglu, 2013; Brambilla et al., 2013), analysis of the scalability of large swarms is limited. Where scalability has been analyzed, most studies find that large swarms are less efficient. For example, (Font Llenas et al., 2018) shows dramatic reductions in per robot foraging rates for even modest increases in swarm size. Our prior work (Hecker and Moses, 2015) evaluates the scalability of the Central-Place Foraging Algorithm (CPFA) (Hecker and Moses, 2015) with up to 768 simulated iAnt robots. We found that foraging performance per robot decreased by 70% going from 1 to 768 robots. We also evaluated scalability of the Distributed Deterministic Spiral Algorithm (DDSA) (Fricke et al., 2016) with up to 30 robots and the MPFA with 96 robots in ARGoS (Lu et al., 2019b). For swarms with between 20 and 30 robots, the DDSA performance drops below that of the CPFA due to crowding. The use of adaptive and dynamic mobile depots increases the scalability of MPFA up to 30% with 96 robots in a 50 × 50 m arena.

Task partitioning allows the physical separation of individuals working on different subtasks, and therefore it can be beneficial in the reduction of physical interference. In addition, it can reduce travel distances to improve location accuracy. Therefore, it can improve the performance to increase scalability. Pini et al. (Pini et al., 2014) demonstrated that a partitioning strategy can improve the performance of transferring objects with real robot swarms directly. Buchanan et al. (Buchanan et al., 2016a) improved the scalability of the robot swarms further on Pini's work using a dynamic partitioning strategy which optimizes the number of subtasks and swarm sizes. In (Ferrante et al., 2015), the leafcutter ant inspired foraging robot swarm achieves maximum foraging performance by dividing foraging and delivering tasks automatically using a nature-inspired evolutionary method known as Grammatical Evolution (Ferrante et al., 2013). We build on these earlier work by introducing mobile depots for the transportation task, separate from searching robots that search for resources. In this scenario, there is much less interference between searching and delivering. Therefore, the scalability of our foraging swarms will be improved further.

The recent work in (Schroeder et al., 2019) analyzed the basis for swarm performance with robot size, robot density, and delays incurred due to collisions between robots. It demonstrated that swarm size and individual robot size affect the swarm performance. When the individual robot size is constant, the result shows diminishing returns when adding additional robots. When the total swarm capacity is constant, there is no significant difference in performance between a few large robots and many small robots. The result allows a swarm designer to weigh the design trade-offs of varying the number of robots, varying swarm capacity, or setting different target swarm costs. In our work, we analyzed the delivery rate and the foraging rate in the hierarchical branching transportation networks. When the delivery rate matches the foraging rate, we can predict the optimized swarm size and robot capacity of each delivery robot.

Biological scaling theory provides a quantitative framework for understanding how transportation of resources affects overall system performance (Savage et al., 2004). Scaling theory predicts that the resources delivery rate is proportional to mammal volume (or mass) to the 3/4 power (and more generally to the power $D/(D+1)$ where D is the dimension of the system). The constraint arises from needing to transport resources over larger distances in larger mammals and limitations on the space available for transportation. In essence, in large mammals, a greater fraction of resources are in transport rather than in active use by cells. The empirically supported prediction is that scaling causes per cell rates and times to slow by a $-1/4$ power of body volume. For example, biological rates are thirty times slower in an elephant which is a million times larger than a mouse. Below we show how scaling produces these diminishing returns with a derivation of scaling predictions for the case of foraging on a 2D surface. We have previously used biological scaling theory to predict how power and performance change with chip size, and how decentralized information flow can alter scaling relationships (Moses et al., 2016). Here, we extend this approach from computer architecture to scalable robotics.

6.5 Similarities between cardiovascular systems and robot swarms

The transportation of resources through the cardiovascular system from the heart to dispersed cells (filling the 3D space of an mammal body) is the inverse problem of (2D) transportation of dispersed resources to a central collection zone in robot foraging. Scaling laws in biology explain how the 3/4 power scaling of delivery rates, and −1/4 power scaling of per-cell biological rates with body size result from a hierarchical transportation network which minimizes energy dissipation (West et al., 1997) and resource delivery time (Banavar et al., 2010). Banavar's derivation is based on 'matching supply and demand' so that there is no waiting or delay in pickup or delivery of resources.

To translate the biological scaling theory into a model of scalable robot foraging, we first identify similarities between the delivery of blood (which carries resources to cells) through cardiovascular networks and the robot foraging task (which carries resources to collection zones). In biology, scaling theory considers delivery of resources in 3-dimensional bodies divided into 'service volumes' which are the volume of tissue supplied by one capillary that delivers blood. Our ground-based robot foraging take places in a 2D area divided into search regions surrounding each depot that serves as a local collection zone. Both require transportation between a central site (heart or collection zone) and a service region. The analogies between these systems are listed in Table 6.1.

Table 6.1: Similarities between cardiovascular systems and robot swarms

Organisms	Robot swarms
3D bodies	2D arenas
Blood cells	Robots
Heart	Central collection zone
Service volumes	Search regions
Resources	Resources
Metabolic rate	Foraging rate
Resource delivery	Resource collection

6.6 Scaling laws for foraging swarms

We derive scaling predictions for foraging robot swarms using the following definitions and simplifying assumptions, translated from (Banavar et al., 2010). For simplicity, we omit the constants of proportionality.

6.6.1 Assumptions

1. Each search region (R) is a specified area A_r (which may vary with total arena size) with a collection zone in its center. Each robot is assigned to a search region in which to forage and a collection zone to which to deliver resources. The number of search regions is N_r so that total arena area $A = N_r A_r$.

2. Resource density (D_t) is the number of resources (N_t) in arena A; $D_t = N_t/A$. Simulation experiments have N_t resources distributed uniformly in A. To maintain constant resource density, resources are replenished as they are collected. When a resource is collected, another resource is placed in a location drawn from a uniform random distribution.

3. Foraging rate (F) is the number of resources collected in the central collection zone per unit time. The foraging rate in a region i is (F_r^i), the number of resources collected and transported to the regional collection zone per unit time. Thus $F = \sum_{i=1}^{N_r} F_r^i$.

4. Geometric similarity states that geometric shapes can be characterized by length, surface area, and volume. Since the arena A is a square with edge length l, then $A \propto l^2$ by the Euclidean geometrical scaling law.

5. Robot foraging velocity (v_f) is constant across all experiments. The delivery velocity of depots (v_d) can vary for each foraging model and experimental setup.

6. The capacity of searching robots is always one resource. The capacity of depots (C) can vary for each foraging model and experimental setup.

7. The number of resources in transit is in steady state and proportional to arena area: $N_t \propto A$. This assumption is analogous to the biological scaling theory assertion that the fraction of blood (that transports resources) is constant across mammal sizes. Here, this assumption means that the density of resources in transit is the same across arena sizes.

8. Delivery rate matches the foraging rate. This means that the system minimizes the time that collected resources are stored in regional collection zones waiting for a depot to pick them up; and no depot arrives at a collection zone and has to wait for a searching robot to drop off resources (i.e., if a depot has capacity 4, then there should be exactly 4 resources ready for pickup when it arrives at its collection zone). This design minimizes the delay in delivery. Ideally, this means that resources do not unnecessarily wait to be picked up, and depots do

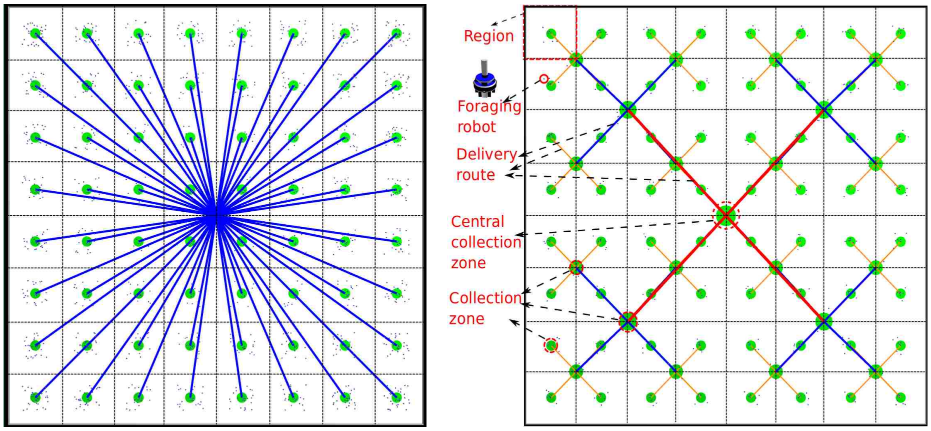
not unnecessarily wait for resources to appear in collection zones to be picked up. Thus, for each collection zone (j) the rate of dropoff equals the rate of pickup $D_j = F_j$ and therefore the delivery rate to the central collection zone (D) equals the total foraging rate (F): $D = F$.

6.6.2 The explosion network

We begin by relating foraging rate (F) to arena area (A) in robot version of the "explosion network" (Fig. 6.1a) as described in (Banavar et al., 2010). The explosion network is a simple model of both the CPFA and the MPFA. In the CPFA each robot leaves a central collection zone and searches for resources in a small region of the foraging arena. Once a resource is found, it is returned directly to the central collection zone. In the MPFA each searching robot delivers resources to a collection zone at the center of its search region. Depots carry a set of resources from the collection zones directly to the central collection zone.

In this model, $N_t \propto nN_{rt}$, where N_{rt} is the number of routes from collection zones to the central collection and n is the average number of resources in transit per route. So, $N_{rt} \propto N_r \propto F$ and $n = \bar{l}_{rt}/v_d$, where $\bar{l}_{rt}$ is the average length of a route and $\bar{l}_{rt} \propto A^{1/2}$ by Assumption 4. So, we have $N_t \propto \bar{l}_{rt}F/v_d$. If v_d is kept constant, we have $N_t \propto FA^{1/2}$. Since from Assumption 7, $N_t \propto A$, then we have **Prediction I:** $F \propto A^{1/2}$. Thus, region length $l \propto (A/N_r)^{1/2}$. Since $N_r \propto F$, then $l \propto A^{1/4}$.

To improve scaling, following (Banavar et al., 2010), we allow v_d to increase with arena size by setting v_d to the maximum value that allows consistently matching collection rate to delivery rate at all collection zones within an arena. As (Banavar et al., 2010) show, this maximum velocity is proportional to the length of



(a) MPFA

(b) $\mathrm{MPFA_T}$

Figure 6.1: Paths of depots in explosion (MPFA) and hierarchical ($\mathrm{MPFA_T}$) transportation networks. Each small square is a service region that contains 4 searching robots.

the shortest route (or region length) l so that $v_d \propto l \propto (A/F)^{1/2}$. This results in $N_t \propto FA^{1/2}/(A/F)^{1/2} \propto F^{3/2}$. Thus, by maximizing velocity, we have **Prediction II:** $F \propto A^{2/3}$, and consequently, $l \propto A^{1/6}$.

6.6.3 The hierarchical branching transportation network

The hierarchical network aggregates the transportation of resources onto paths of increasing length and capacity. Fig. 6.1b shows a network composed of $b = 4$ branches at each level from the central collection zone (level 0, red) to 4 regional hubs (level 1, blue), to 16 sub-regional hubs (level 2, yellow) and finally to 64 search regions with collection zones (green dots) at their center.

The required number of levels L to connect all regions is $\log_b^{N_r}$. The minimum number of depots N_d is calculated such that the delivery rate to the center collection zone equals the rate that resources are collected in the search regions (such that

$D = F$, by Assumption 8 in Subsection 6.6.1). The time of a depot at level i to make a round trip from its source collection zone to its destination collection zone is $T_d^i = 2d_i/v_d$, where d_i is the distance from a collection zone on level i to its destination collection zone on level $i-1$. The number of depots N_d^i is equal to the number of collected resources in T_d^i on level i divided by the capacity C: $N_d^i = 2b^{i+1}F_r b^{L-i-1} d_i/(v_d C)$, where $F_r b^{L-i-1}$ is the rate resources are collected in zones at level i and b^{i+1} is the number of branches at level i.

Summing over all levels gives:

$$\begin{aligned} N_d &= \sum_{i=0}^{L-1} \frac{2F_r b^L d_i}{v_d C} \\ &= \frac{2F_r A^{1/4}(N_r^{3/2} - N_r)}{v_d C} \end{aligned}$$

This leads to the following prediction for the required number of depots:

$$N_d = \begin{cases} \frac{2F_r}{v_d C}(A - A^{3/4}) & \text{if } v_d \text{ is constant} \\ \frac{2F_r}{C}(A - A^{2/3}) & \text{if } v_d \propto A^{\frac{1}{6}} \end{cases} \tag{6.1}$$

6.6.4 Scale-invariant transportation network

Whether biological systems use an explosion network or a fractal branching network, they are limited to sublinear scaling ($F \propto A^{1/2}$ when v_d is constant and $F \propto A^{2/3}$ when v_d scales at its maximum value). However, we can use biological scaling to design a scale-invariant foraging swarm in which the total foraging rate is linear with arena size and swarm size (and per capita foraging rates are constant.) In a collective transport task (Rubenstein et al., 2013), the transport capability of an individual robot would be expected to increase with its size. If we increase the capacity C on each level by ab from level $L-1$ to 0, then Eqn. Eq. (6.1) will have a constant N_{d}^{i} in each level i collection zone. Furthermore, we can increase F by having a constant region size. Then we have **Prediction III:** $F \propto N_r \propto A$.

The number of connections to the center is b in the $\mathrm{MPFA_T}$. We have 4 searching robots in each region and 4 depots in each collection zone. So, the total number of searching robots is $4N_r$ and the total number of depots is $4N_c$, where N_c is the number of collection zones excluding the central collection zone. Thus:

$$N_c = \sum_{i=0}^{L-1} \frac{N_r}{b^i} = \frac{4}{3}(N_r - 1) \tag{6.2}$$

So, the total number of robots is $\frac{28}{3}N_r - \frac{16}{3}$, therefore linear with N_r and A. Note that this means a constant density of searching robots in the regions and a constant density of transport robots with arena size.

6.7 Experimental Setup

We conducted experiments to test Predictions I, II, and III for scalability of the CPFA, MPFA, and $\mathrm{MPFA_T}$, as summarized in Table. 6.2. Each of the three experimental configurations tests one of the three predictions. In Set I, we test the 1/2 scaling in Prediction I. The depot velocity and capacity are constant ($v_d = 0.16m/s$ and $C = 4$), but the region length $l \propto A^{1/4}$ as derived in Section 6.6. In Set II, we test the 2/3 scaling Prediction II. $C = 4$, but $v_d \propto A^{1/6}$ and $l \propto A^{1/6}$. The number of

Table 6.2: Experimental Setup

	Shared configuration									MPFA	MPFA_T	
Set	Delivery velocity (v_d)	Capacity (C)	Region length (l)	Region (N_r)	Arena size (A)	resource (T)	Searching (N_s)	Depot (N_d)	Total robots	Collection zone (N_c)	Collection zone	Level (L)
I	0.16	4	1	1	1× 1	1	4	0	4	1	1	0
			2	4	4× 4	16	16	4	20	5	5	1
			4	16	16× 16	256	64	48	112	17	21	2
			8	64	64× 64	4096	256	896	1152	65	85	3
II	$\propto A^{1/6}$	4	1	1	1× 1	1	4	0	4	1	1	0
			2	16	8× 8	64	64	48	112	17	21	2
			4	256	64× 64	4096	1024	3840	4864	257	341	4
III	0.16	vary	5	4	10× 10	100	16	16	32	5	5	1
				16	20× 20	400	64	80	144	17	21	2
				64	40× 40	1600	256	336	592	65	85	3
				256	80× 80	6400	1024	1360	2384	257	341	4

depots (N_d) in Set I and II is set to meet the constraint that the foraging rate matches the delivery rate ($D = F$) as explained in Assumption 8 and predicted in Eq. (6.1). In set III, we test the linear scaling Prediction III. $l = 5m$, v_d is constant, and the depot capacity C is scaled by ab on each level as described in Subsection 6.6.4. The number of depots (N_d) is $4N_c$ as predicted in Eq. (6.2).

In order to provide a fair comparison among algorithms, we use the same number of total robots in each set. In addition, we use the same number of depots (N_d) for the MPFA and the MPFA_T. In the MPFA, we distributed depots on each delivery route proportional to the length of that route. Thus, the number of depots for collection zone j is $N_j = (d_j/D)N_d$, where d_j is the distance from the collection zone j to the central collection zone and D is the total distance of all collection zone to the central collection zone. The number of searching robots (N_s) is 4 robots per region ($N_s = 4N_r$). In the CPFA, the depots are converted to searching robots that directly deliver resources to the central collection zone. The CPFA is only included in Set I where it is demonstrated not to be scalable (consistent with results from prior studies) due to dramatically increasing collisions with swarm size.

Each experiment runs for 30 simulated minutes, and each configuration is replicated 60 times. The density of resources is constant across all experiments ($D_t = 1/m^2$). Resources are distributed at uniform random and collected resources are replenished in a new random location in order to keep D_t constant. The source code of all three models is available on Github[1].

We include two additional foraging experiments which allow depots to pass through one another without colliding (MPFA* and MPFA$_\mathrm{T}^*$) in MPFA and MPFA$_\mathrm{T}$. These allow us to test whether predictions would hold under the idealized condition that collisions among depots were ignored.

6.8 Results

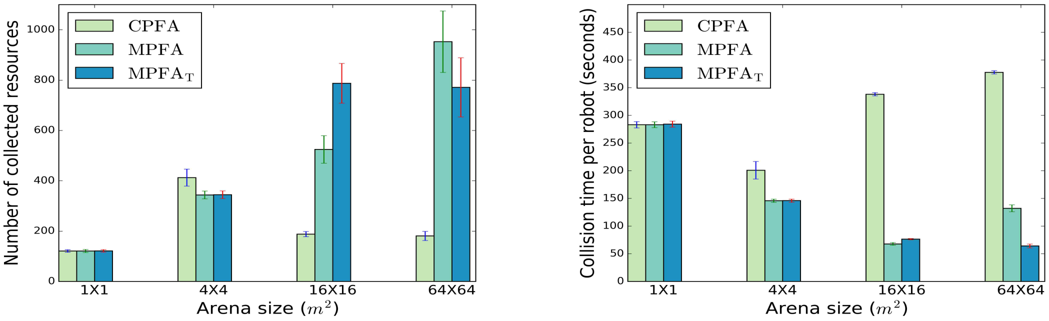
Generally, scaling relationships are analyzed as $Y = aX^b$ where a is a conversion factor, and b is an exponent which indicates the scaling relationship between X and Y. We display and analyze log-transformed data so that $\log_2 Y = b \log_2 X + \log_2 a$. The slope of the regression between the $\log_2 Y$ and the $\log_2 X$ gives an estimate of the scaling exponent b.

6.8.1 Prediction I

The foraging performance in Set I (see Table 6.2) of the CPFA, MPFA, and MPFA$_\mathrm{T}$ is shown in Fig. 6.2a. Foraging performance of the MPFA and MPFA$_\mathrm{T}$ increase with arena and swarm size. However, the CPFA sometimes collects fewer total resources even with hundreds of times more robots. This is in large part due to collisions as shown in panel b. Fig. 6.3a shows the scaling of the MPFA and MPFA$_\mathrm{T}$ are better

[1] `https://github.com/BCLab-UNM/MPFA-ARGoS`

than the CPFA, but neither reaches the predicted 0.5 scaling exponent. All three algorithms suffer from collisions. The slope of the MPFA* and that of the MPFA$^*_{\mathrm{T}}$ (which have no collisions) approach the predicted slope of 0.5, with slopes of (0.46 and 0.45).



(a)

(b)

Figure 6.2: Number of collected resources vs. arena size (panel a) and time spent avoiding collisions per robot vs. arena size (panel b) in Set I. The low foraging of the CPFA (panel a) is explained in part by the long time spent avoiding collision.

In the ARGoS simulation experiments, if the distance between two robots is less than 0.25, each robot will detect a potential collision. Each robot senses the location of the other and turns left or right and moves in order to avoid the collision and resume their previous behavior. This simple collision avoidance consumes time. The total collision time for each experiment is the sum of the total time spent avoiding collision for all robots. The collision time per robot is shown in Fig. 6.2b. The collision avoidance time is much higher for the CPFA than for the other algorithms. Collisions are lowest in the MPFA$_{\mathrm{T}}$ at large swarm size. Collision time explains much (but not all) of the difference between experiments (shown in Fig. 6.2a and the theoretical prediction).

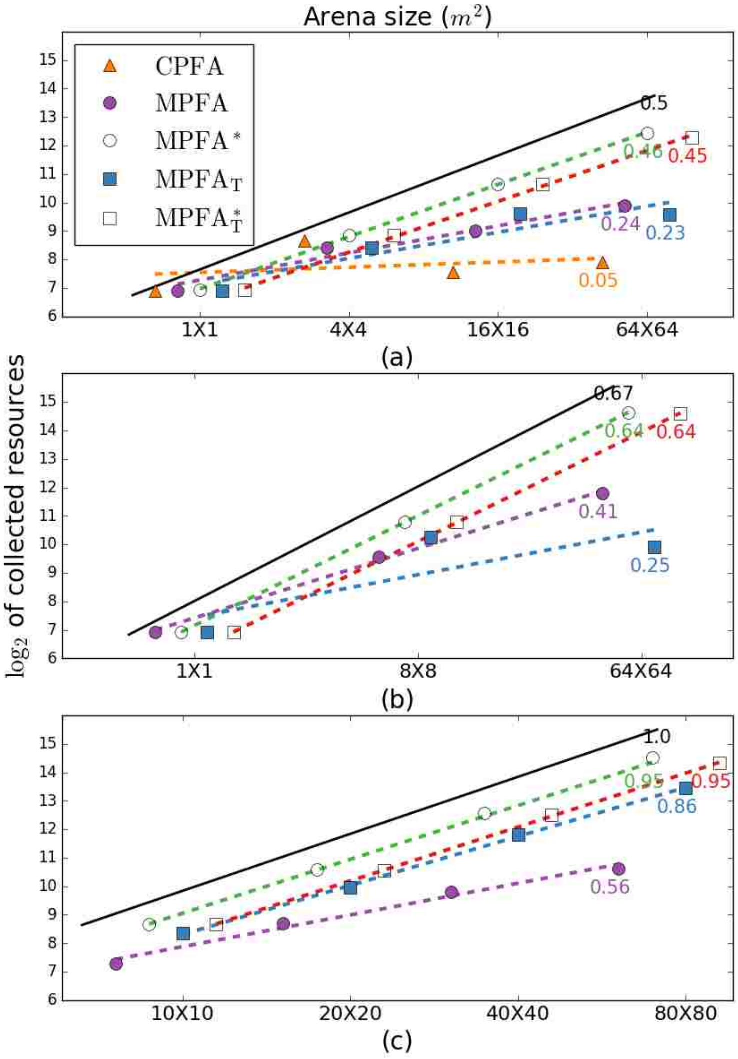


Figure 6.3: The number of resources collected in 30 minutes vs arena size is in experiment Sets I, II, and III. Both axes are on a log scale. Each dashed line indicates the $\log_2$ linear regression of the mean of collected resources with the $\log_2$ of arena area. The solid black line indicates the predicated slope in each configuration. The * algorithms (hollow symbols) that lack collisions among depots demonstrate that foraging under ideal conditions is close to theoretical predictions. $p < 0.001$ in all experiments. In panel (a), $r^2 = 0.1$ for the CPFA, $r^2 = 0.95$ for the MPFA, $r^2 = 0.85$ for the MPFA$_\text{T}$, and $r^2 = 0.99$ for the MPFA* and MPFA$^*_\text{T}$. In panel (b), $r^2 = 0.66$ in the MPFA$_\text{T}$ and $r^2 > 0.98$ in other algorithms. In Set III, $r^2 = 0.96$ for the MPFA and $r^2 > 0.99$ for other algorithms.

6.8.2 Prediction II

Fig. 6.3b shows the log-log plot of foraging performance vs. arena size using constant capacity but variable region size and velocity described in Set II. The slope of the MPFA_T is the lowest (0.25), and the slope of the MPFA is (0.41), both well below the predicted 2/3 slope. However, the slopes of the MPFA^* and the MPFA^*_T (0.64) are close to the predicated 2/3 exponent.

6.8.3 Prediction III

Fig. 6.3c shows foraging performance vs. arena size using constant region size and velocity, but variable depot capacity in Set III. The slope of the MPFA_T (0.86) is higher than the MPFA (0.56) and is the highest slope of all algorithms that consider collisions. The slopes of the collision-free MPFA^* and MPFA^*_T are both (0.95) which is very close to the predicted exponent 1.0.

6.9 Discussion

Many real-world applications require that robots find and collect as many objects as possible in the least amount of time. We have proposed an efficient approach for swarms ranging from a few robots to thousands of robots foraging in arenas that are many square kilometers in area. Scalability is achieved with a hierarchical branching transportation network inspired by mammal cardiovascular networks.

First, in Set I if we ignore collisions in the transportation network and each depot has a constant delivery velocity and capacity, then the foraging follows the predicted 1/2 power scaling (Fig. 6.3a). The algorithms with the * illustrate the maximum scaling exponent if there were no collisions in transportation. However, when we include collisions, the CPFA barely increases the foraging rate as the arena and swarm size increase. The scaling exponents of the MPFA and MPFA_T are closer to but still lower than the expected 1/2 scaling.

Visual inspection of simulations shows extreme congestion at the collection zone for larger swarms. The collision is the major difference between biological systems and robot systems. The unpredictable collisions cause a delay in transportation. The robot density (the area occupied by all robots divided by the arena size. The robot radius is 0.085 m) affects the collision rate and the distribution of collisions also affects the collision rate. Congestion on critical transportation paths produces significant amount of collisions.

Fig. 6.4 shows the average collision time per robot, per minute for the CPFA and MPFA with 4 collection zones. The data is the same as the data in Fig. 4.3 in Section 4, but the x-axis indicates the robot density. The result shows that collisions increase in the CPFA with the increase of robot density. As the robot density increases, the robots spend more time on avoiding collisions rather than performing foraging tasks. In contrast, collisions in the MPFA increase slowly since robots are distributed to multiple collection zones.

The robot density decreases from 0.09 to 0.006 in Set I. So, the collision rate decreases in the MPFA and MPFA_T when robot density decreases as shown in Fig. 6.2b. The collision rate in the CPFA increases since all robots travel to the center which results in high congestion around the center. However, the foraging performance in the MPFA is higher than the performance in the MPFA_T as shown in Fig. 6.3.

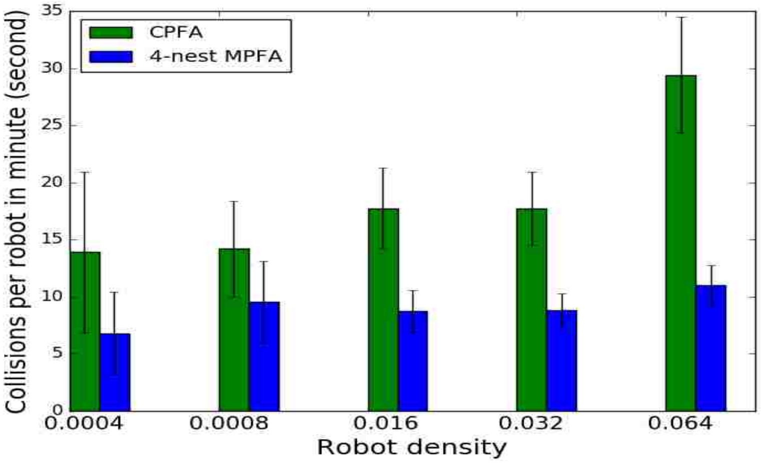


Figure 6.4: The collision time per robot, per minute for the CPFA and MPFA with 4 collection zones. The data is from Fig. 4.3 in Section 4. The arena size is 15 × 15 m. Results are for 100 replicates.

Fig. 6.5 shows the average number of depot trips required to deliver deposited resources from collection zones in the MPFA and $\mathrm{MPFA_T}$. As resources are delivered to the central collection zone from regions directly, collisions are produced around collection zones in the MPFA (see Fig. 6.6). As resources are delivered to collection zones in the next levels and delivered to the central collection zones gradually, collisions are produced more and more around collection zones at higher levels in the $\mathrm{MPFA_T}$. Even the total collision rate is lower in the $\mathrm{MPFA_T}$ than in the MPFA, the congestion on critical transportation paths produces more delay on the delivery (see Fig. 6.7). Ideally, there is no delay in transportation without collisions in the proposed transportation networks. Collected resources should be transported immediately.

Similarly, in Set II, which allows depot delivery velocity to vary according to biological scaling predictions, the scaling exponents with collisions are lower than

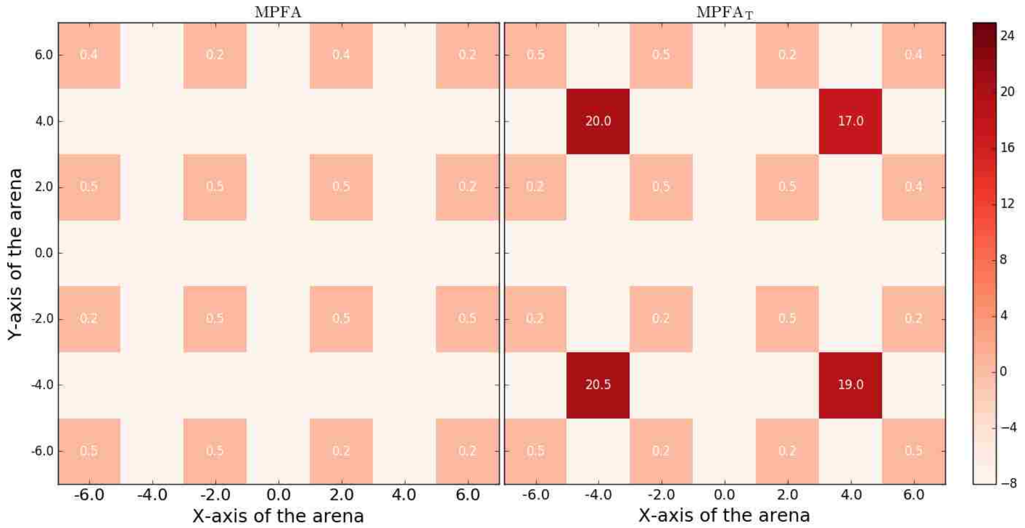
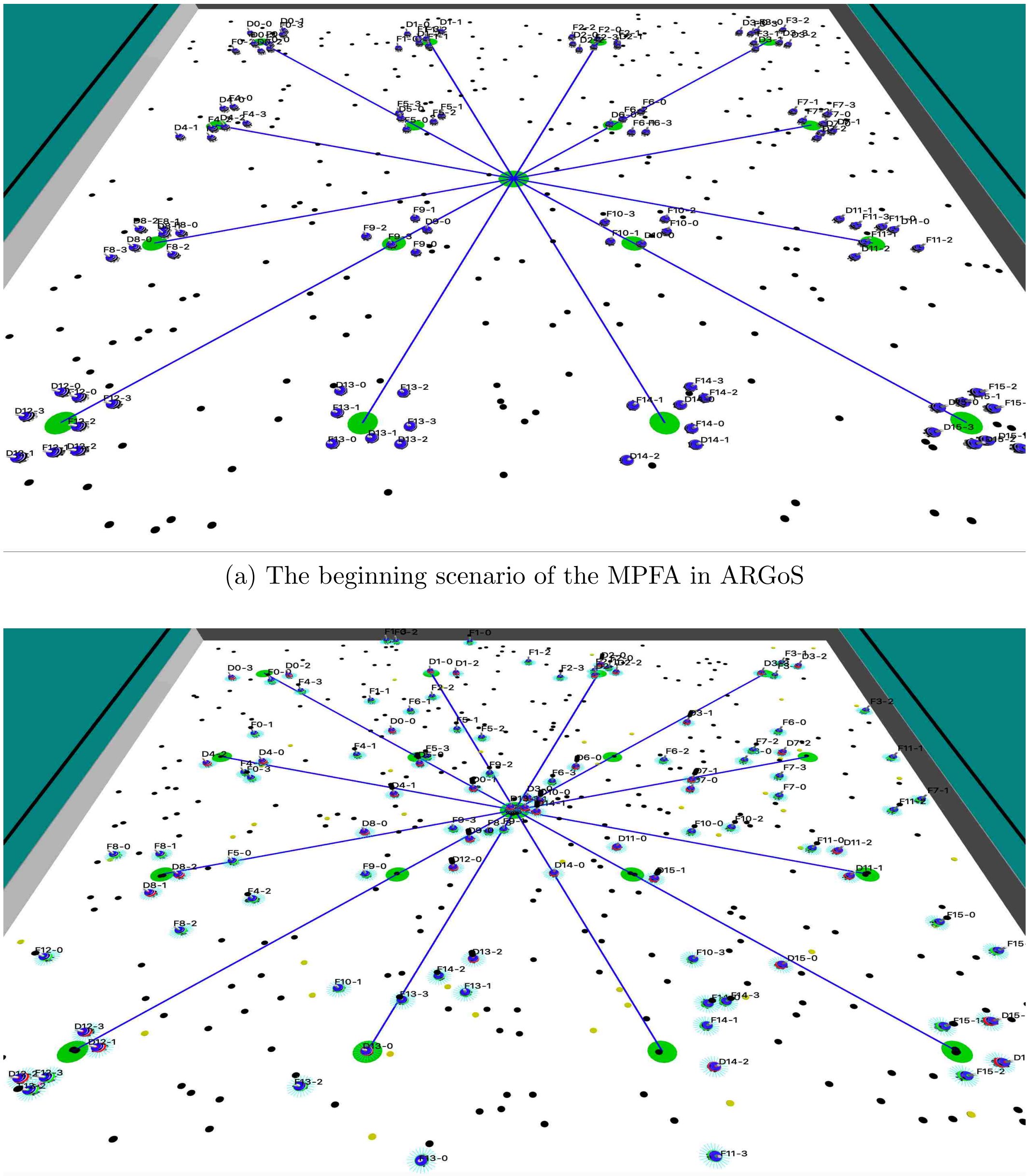


Figure 6.5: The number of depot trips required to deliver resources in the MPFA and the MPFA$_\mathrm{T}$. The colored squares indicate the locations of collection zones. The numbers in collection zones indicate the average depot trips required to deliver resources. Each experiment is replicated 60 times. The data is from the third set of experiments in Prediction I. There are 16 collection zones in the MPFA and 20 collection zones in the MPFA$_\mathrm{T}$. The central collection zone is not shown since no depot trip is required on it.

predicted, but the collision-free transportation scaling exponents are close to the 2/3 prediction (see Fig. 6.3b). These results again validate that biological scaling theory predicts simulated robot foraging rates, at least when we ignore collisions in transportation.

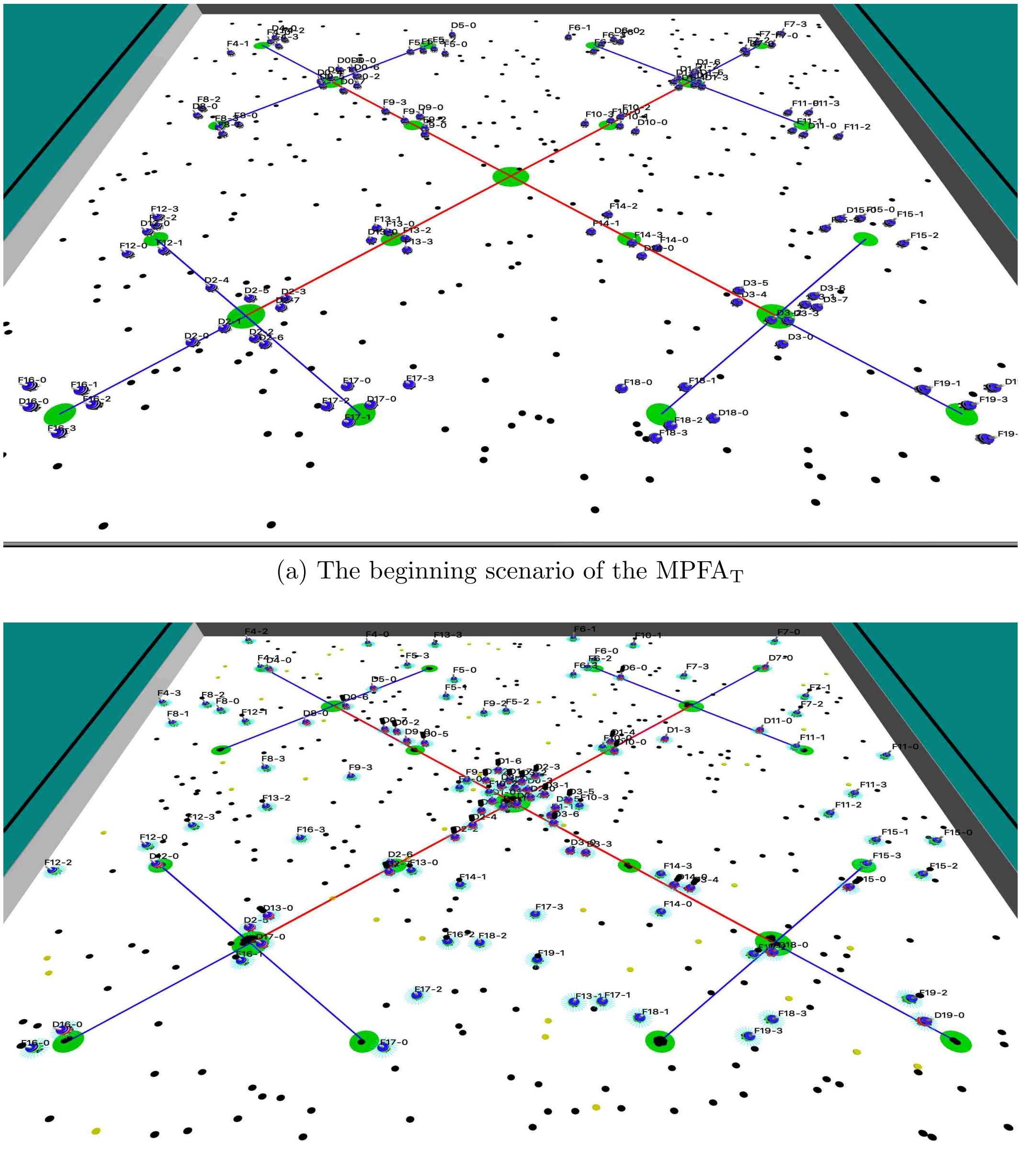
However, in Set III, where we set a fixed foraging region size, fixed velocity, but set the carrying capacity of robots to match supply and demand (as stipulated in Assumption 8), then the hierarchical transportation network implemented in the MPFA$_\mathrm{T}$ results in a much higher scaling exponent 0.95 without collisions and 0.86 with collisions (see Fig. 6.3c).



(a) The beginning scenario of the MPFA in ARGoS

(b) The scenario of the MPFA at 10 minutes in ARGoS

Figure 6.6: The scenarios of the MPFA in ARGoS simulation. The configuration is the third one in Set I for Prediction I. 64 searching robots (with green LEDs), 48 depots (with red LEDs), 256 uniformly distributed resources (black dots), and 17 collection zones (green circles) are in a 16×16 m arena. Blue lines indicate paths for delivering resources. Yellow dots indicate locations where robots collected resources in their last trip. Robots remember those locations and they may return to those locations using a process called *site fidelity*.



(a) The beginning scenario of the $\mathrm{MPFA_T}$

(b) The scenario of the $\mathrm{MPFA_T}$ at 10 minutes

Figure 6.7: The scenarios of the $\mathrm{MPFA_T}$ in ARGoS simulation. The configuration is the third one in Set I for Prediction I. 21 collection zones are in a 16 × 16 m arena.

Fig. 6.8 shows the collision rate increases slowly in the MPFA and it does not increase in the $\mathrm{MPFA}_{\mathrm{T}}$. The collision rates are also shown implicitly by the delivery delay shown in Fig. 6.9. The number of depot trips required is higher in collection zones closer to the center in the MPFA. More robots traveling to the center cause more collisions around the center (see Fig. 6.10). In the $\mathrm{MPFA}_{\mathrm{T}}$, robot depots carry large numbers of resources, reducing the number of robots required to transport resources over long distances. There is less crowding in the $\mathrm{MPFA}_{\mathrm{T}}$ (compared to the MPFA) because the collision rate within a constant robot density (0.0008) and the number of robots to each collection zone, including the largest at the center, equals the branching factor b regardless of swarm and arena size (see Fig. 6.11). This reduces collisions in the $\mathrm{MPFA}_{\mathrm{T}}$, making it more robust. Therefore, the performance per robot in the $\mathrm{MPFA}_{\mathrm{T}}$ is nearly constant (0.86). If there are no collisions in transportation, the delivery delay in the MPFA^* is slightly high in the collection zones closer to the center and there is almost no delivery delay in the $\mathrm{MPFA}^*_{\mathrm{T}}$) (see Fig. 6.12). Their performance per robot is nearly constant (0.95) as shown in Fig. 6.3.

This approach essentially aggregates collected resources in larger depots (where depot capacity is set according to scaling theory), similar to the way in which blood is aggregated in larger vessels like the aorta in cardiovascular networks. However, biological networks face a constraint on the size of blood vessels (their total volume must be equal to a constant fraction of the mammal volume (West et al., 1997). In contrast, the robot foraging transportation network can increase capacity (up to the largest possible depot) to accommodate the increase in transport required in larger

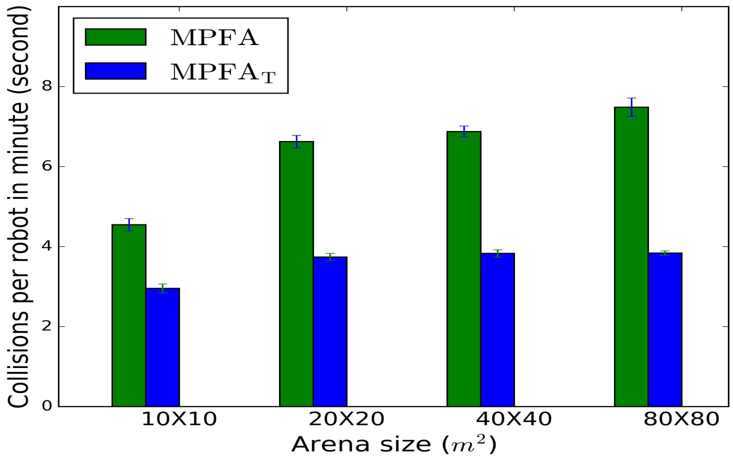


Figure 6.8: The collision time per robot, per minute for the MPFA and $\mathrm{MPFA_T}$ in Prediction III. The robot density is nearly constant (0.03) in all arenas. The data is from Fig. 6.3(c). Results are for 100 replicates.

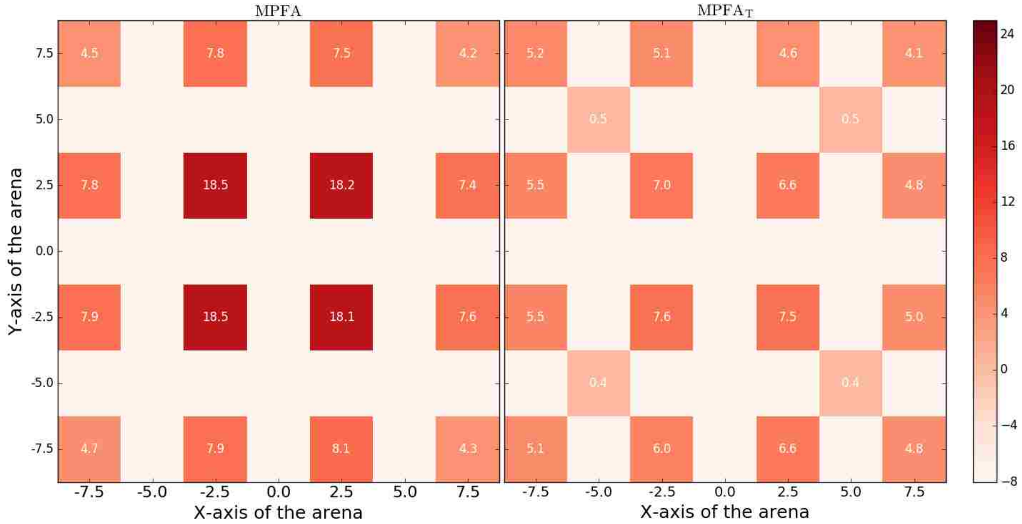
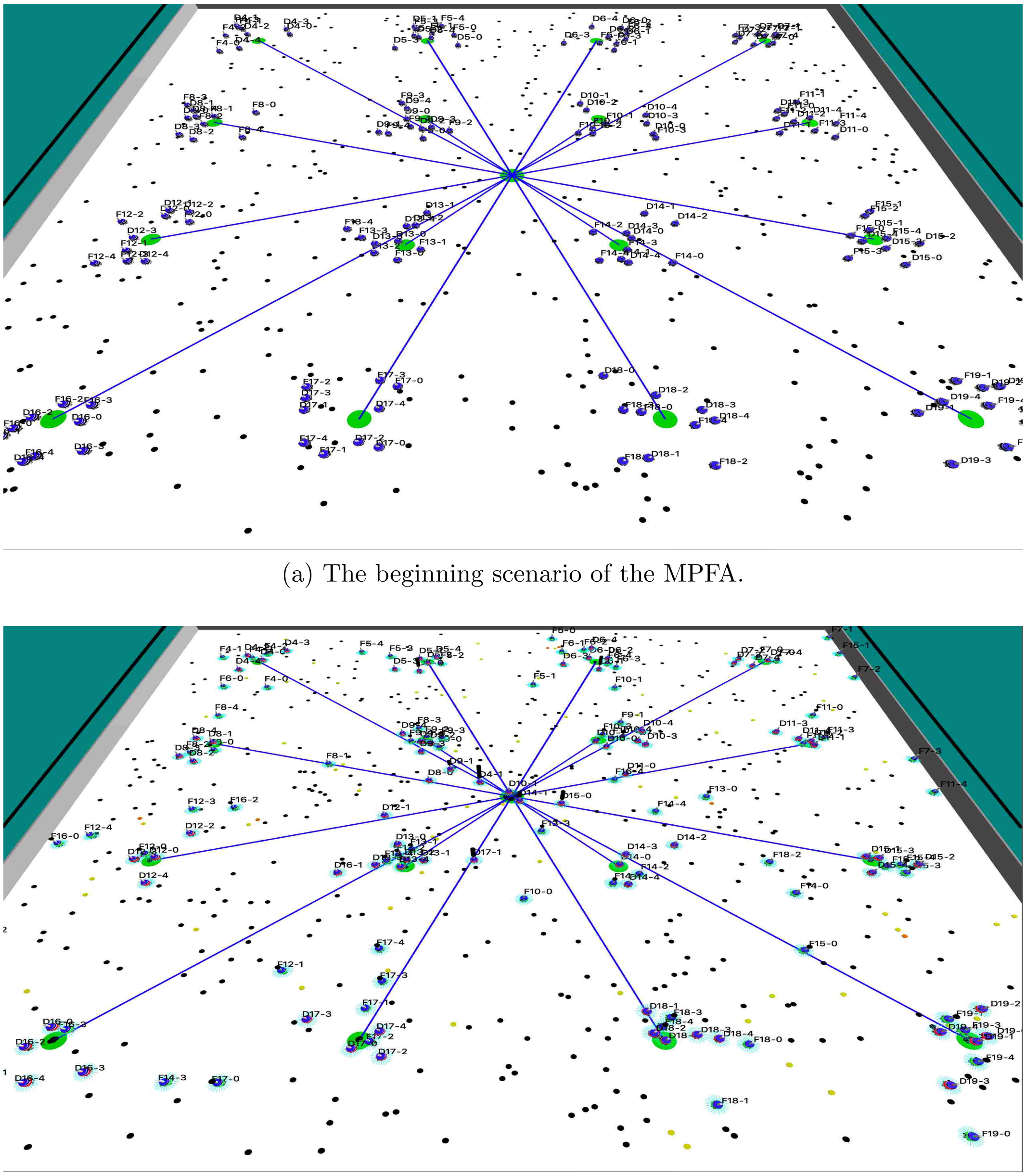


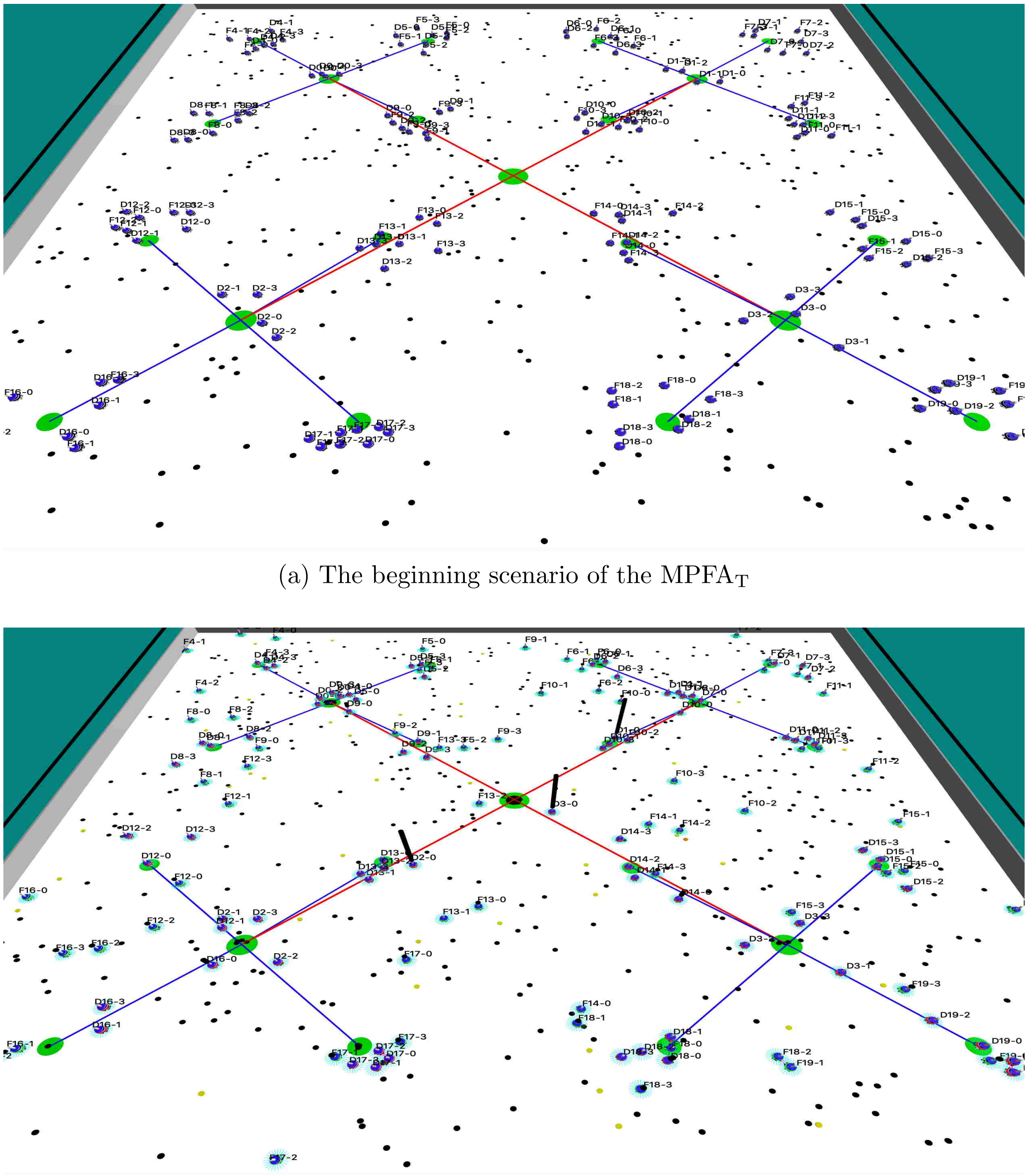
Figure 6.9: The number of depot trips required to deliver resources in the MPFA and the $\mathrm{MPFA_T}$. The configuration is the second one in Set III for Prediction III. 17 collection zones are in the MPFA and 21 collection zones are in the $\mathrm{MPFA_T}$. The arena size is 20 × 20 m.



(a) The beginning scenario of the MPFA.

(b) The scenario of the MPFA at 10 minutes

Figure 6.10: The scenarios of the MPFA in ARGoS simulation. The configuration is the second one in Set III for Prediction III. 17 collection zones are in a 20 × 20 m arena.



(a) The beginning scenario of the $\mathrm{MPFA_T}$

(b) The scenario of the $\mathrm{MPFA_T}$ at 10 minutes

Figure 6.11: The scenarios of the $\mathrm{MPFA_T}$ in ARGoS simulation. The configuration is the second one in Set III for Prediction III. 21 collection zones are in a 20×20 m arena.

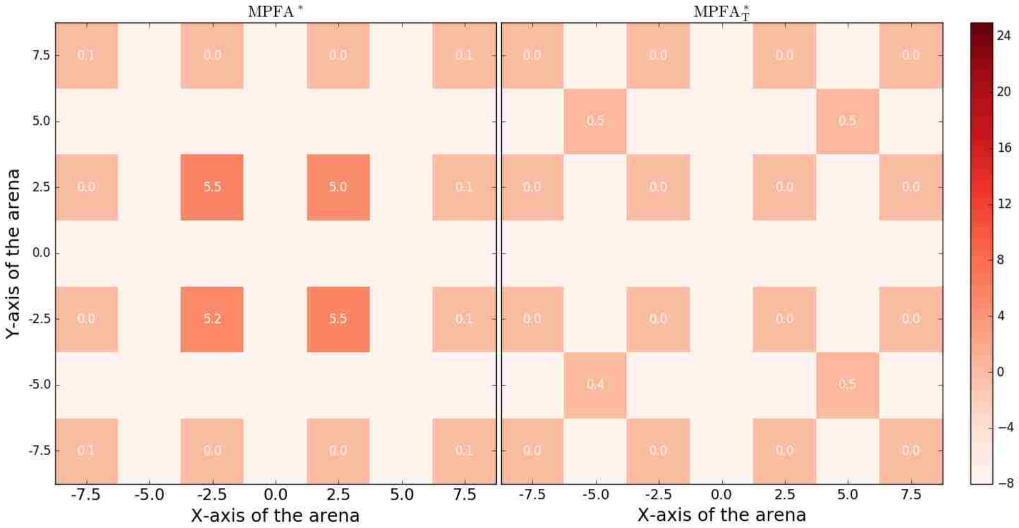


Figure 6.12: The number of depot trips required to deliver resources without collisions in the MPFA* and the MPFA$^*_T$.

arenas and mitigate the increase of collisions to ensure that delivery capacity keeps up with a constant per-forager collection rate.

This work demonstrates the viability of an artificial bio-inspired transportation network in robot swarms. We predict the required number of robots, and the number and size of depots for a given size arena to achieve scale invariant foraging. We are building a prototype depot so that we can test MPFA$_T$ in real robots, following the experimental protocol we developed to go from the ARGoS simulation to a ROS implementation (Lu et al., 2019a).

Chapter 7

Comparing Physical and Simulated Performance of a Deterministic and a Bio-inspired Stochastic Foraging Strategy

7.1 Publication Notes

Citation: Qi Lu, Antonio D. Griego, G. Matthew Fricke, and Melanie E. Moses. Comparing Physical and Simulated CPFA and DDSA]Comparing Physical and Simulated Performance of a Deterministic and a Bio-inspired Stochastic Foraging Strategy, 2019.

Received: September 15, 2018

Accepted: February 15, 2019

mance of a Deterministic and a Bio-inspired Stochastic Foraging Strategy, IEEE/RSJ International Conference on Robotics and Automation (ICRA), May, 2019.

7.2 Abstract

Designing resource-collection algorithms for relatively simple robots that are effective given the noise and uncertainty of the real world is a challenge in swarm robotics. This paper describes the performance of two algorithms for collective robot foraging: the stochastic central-place foraging algorithm (CPFA) and the distributed deterministic spiral algorithm (DDSA). With the CPFA, robots mimic the foraging behaviors of ants; they stochastically search for targets and share information to recruit other robots to locations where they detect multiple targets. With the DDSA, robots travel along pre-planned spiral paths; robots detect the nearest targets first and, in theory, guarantee eventual complete coverage of the arena with minimal overlap. We implemented both algorithms and compared their performance in a Gazebo simulation and in physical robots in a large outdoor arena. In the Gazebo simulation, the DDSA outperforms the CPFA. However, in real-world experiments with obstacles, collisions, and errors, the movement patterns of robots implementing the DDSA become visually indistinguishable from the CPFA. The CPFA is less affected by noise and error, and it performs as well as, or better than, the DDSA. Physical experiments change our conclusion about which algorithm has the best performance, emphasizing the importance of systematically comparing the performance of swarm robotic algorithms in the real world.

7.3 Introduction

Robot swarms are particularly useful in spatially distributed tasks such as central place foraging, in which robots search for targets and transport them to a collection zone (Winfield, 2009b; Brambilla et al., 2013). Swarm foraging algorithms often mimic the stochastic behaviors of social animals, particularly social insects such as ant colonies (Şahin, 2005; Hecker and Moses, 2015; Ferrante et al., 2015).

In this work, we conduct physical and simulated experiments to compare two collective robot foraging algorithms: the Central-Place Foraging Algorithm (CPFA) and the Distributed Deterministic Spiral Algorithm (DDSA). In previous work, these algorithms were compared using the Autonomous Robots Go Swarming (ARGoS) simulator, and it was found that simple robot swarms operating in obstacle-free environments collected resources faster using the DDSA compared to the CPFA (Fricke et al., 2016), at least for swarm sizes of up to 20 robots.

The foraging performance of robots can be measured by the number of targets retrieved in a fixed time. It is important to evaluate collective algorithms in physical robots (Brambilla et al., 2013) because it is not feasible to simulate all aspects of a physical environment (Frigg and Hartmann, 2018), and foraging performance can be altered by variable conditions and by sensor and actuator noise that affect localization, object retrieval, and collision avoidance. All of these components of the "reality gap" can alter the performance of algorithms real robotic experiments compared to simulations (Jakobi et al., 1995; Mouret et al., 2013; Ligot and Birattari, 2018).

Predicting the performance of swarm algorithms in real robots is especially challenging because interactions among robots are inherently difficult to predict. Deterministic algorithms may become effectively random when operating in environments with unexpected interactions. Thus, while simulations are useful for initial evaluations of the viability of algorithms, they are insufficient for the ultimate goal of predicting how algorithms will perform when physical robots interact in the unpredictable conditions of environments they are placed in.

This work implements two swarm foraging algorithms in Robot Operating System (ROS). We systematically compare the foraging performance of the DDSA and the CPFA by measuring how quickly targets are collected in fixed time. We designed a set of experiments that we replicated in a Gazebo simulation (Koenig and Howard, 2004) and in physical robots called "Swarmies" that search for, pick up, and collect physical objects (which we call targets) in outdoor arenas with various placements of targets and obstacles.

In previous work we compared the DDSA and the CPFA in intentionally simple simulations implemented in the foot-bot robot in ARGoS (Fricke et al., 2016). In contrast, in this paper, we describe simulations implemented in Gazebo that include more realistic physical processes that represent the localization, navigation, sensing, object pickup and drop off, and collision avoidance of the Swarmie robots that we implement in physical experiments. Still, our physical experiments include variability inherent to outdoor environments and sensor and actuator noise that is not fully simulated in our Gazebo simulations.

The major contribution of this work is to compare a deterministic and a stochastic swarm foraging algorithm in simulations and in physical robots. Our goal is to test whether the most efficient algorithm in the simulation is also the most efficient in physical experiments. We implement both algorithms in physical hardware and show that the performance of each algorithm is impacted in different ways by the noise and error of the physical world. The conclusion we draw from comparing the two algorithms is: the deterministic DDSA is more efficient than the CPFA in the simulation. However, the stochastic CPFA marginally outperforms the DDSA in physical experiments. The performance of the DDSA is more degraded by conditions in the physical world, suggesting that the CPFA is more tolerant of real-world conditions.

This chapter is organized as follows. The related work is presented in Section 7.4. Sections 7.5 and 7.6 summarize the CPFA and DDSA algorithms. Section 7.7 describes the physical robots and physical and simulated environments. Section 7.8 describes the experiments, with results reported in Section 7.9. Section 7.10 discusses the strengths and limitations of stochastic and deterministic search strategies.

7.4 Related work

Though swarm robot foraging has been studied for decades, replicated experimental analyses that compare different algorithms in simulation and in real robots are rare, particularly in outdoor environments (Winfield, 2009a; Brambilla et al., 2013). Many task partitioning and foraging algorithms have been simulated in the STAGE (Gerkey et al., 2003; Liu et al., 2007; Castello et al., 2016), the ARGoS (Ferrante et al., 2015; Pini et al., 2014; Buchanan et al., 2016b) and Microsoft(R) Robotics Developer Studio (MRDS) (Hoff et al., 2010). Physical foraging experiments have been conducted with foot-bots equipped with grippers, IR sensors, and cameras for foraging tasks in (Pini et al., 2014; Buchanan et al., 2016b) and custom platforms like MinDART (Rybski et al., 2008).

In practice, many complex physical experiments with swarm robots require human support (Rosenfeld et al., 2017) or simulation of some aspect of the foraging task. For example, the Robotarium provides a testbed for remotely accessible physical robots (Pickem et al., 2017), but localization is governed by an overhead camera. Other studies simulate physical pickup and drop-off of objects. For example, (Brutschy et al., 2015; Castello et al., 2016) uses a group of e-puck robots and our prior work (Hecker and Moses, 2015) used iAnt robots which detect targets but do not physically pick them up. Kilobots can operate autonomously to push items, but they have relatively limited mobility and only operate in controlled laboratory environments (Rubenstein et al., 2012; Jones et al., 2018). Collaborative warehouse robots may require buried guide-wires or visual markers to navigate (Enright and Wurman, 2011). Swarmanoid demonstrates an innovative heterogeneous physical swarm robotics system whose robots collaborate to solve a complex object retrieval task (Dorigo et al., 2013).

Swarmie robots allow us to conduct automated, replicated experiments to test autonomous collective foraging. The Swarmies physically pick up and drop off targets and operate outdoors under variable ground and light conditions. These factors are important sources of error and noise in our experiments. However, Swarmies have some limitations as a swarm robotics platform. They use GPS, a global (but still noisy) signal, to mitigate the localization problem. We also occasionally use human intervention to prevent robots from leaving the foraging arena. Finally, while Swarmies can operate in larger swarms, the experiments here are with 4 robots.

7.5 Central Place Foraging Algorithm: CPFA

With the CPFA, robots mimic the foraging behaviors of *Pogonomyrmex* desert seed-harvester ants, social insects which have evolved to cooperate without centralized control (Gordon and Kulig, 1996; Flanagan et al., 2012). Fig. 7.1a shows how individual robots transition between states in the CPFA based on various conditions (further detailed in (Hecker and Moses, 2015)). Robots start from the central collection zone and travel towards a randomly selected location (State A) until they switch to searching using an uninformed correlated random walk (Fewell, 1990) (State B). If a robot detects targets (Condition 3), it collects the closest one (State D) and measures the number of additional targets within its camera view by rotating 360° (State E). The robot uses the measured targets to decide whether to create a "pheromone waypoint" which adds the location and the strength to a list, mimicking ant pheromone trails (Sumpter and Beekman, 2003; Jackson et al., 2007; Lu et al., 2016a). The strength of waypoints decreases over time and waypoints can be added to the list by other robots. Robots communicate pheromone waypoints and may select waypoints probabilistically ranked by strength at the nest.

The robot carries its collected target to the collection zone and drops it off (State F). If a robot does not find a target, it can, give up its search (Condition 6) and return to the collection zone (State F). A robot at the collection zone can share pheromone waypoints with other robots at the nest. Then, the robot takes its next foraging trip. It either selects a random location (Condition 1) or selects a previously visited location (Condition 7, State G) accomplished by either following a pheromone waypoint or by returning to the last location it found a target, a process called *site fidelity*. The probabilities of creating a pheromone waypoint and of using site fidelity are drawn from a Poisson distribution dependent on the number of targets observed at that location. When robots return to locations via either site fidelity or pheromone-following (Condition 8), they search the area thoroughly with an informed walk (state

C) which is characterized by moving more randomly, and therefore searching more thoroughly, than in an uninformed walk.

CPFA robots make real-time decisions based on a set of 7 real-valued parameters specifying the probabilities that govern the transitions in Fig. 1a. The CPFA parameters were selected by a genetic algorithm (GA) to maximize the number of collected targets in (Hecker and Moses, 2015). This is not feasible given the slow run-time of the Gazebo simulations or physical robots. Instead, we hand-tuned the parameters (included in our Github repository) based on the previously evolved results. With the perfect evolved parameters, the performance of the CPFA should be a slightly better than the CPFA with hand-tuned parameters. However, this does not affect the conclusion in this work.

7.6 Distributed Deterministic Search Algorithm: DDSA

In contrast to the CPFA, the DDSA takes a geometric approach which exploits the optimality of spiral search demonstrated for single agents (Bentley et al., 1980; Baeza-yates et al., 1993) generalized to a swarm of robots. Robots using the DDSA start near the central collection zone and search for targets by following a pre-planned pattern of interlocking square spirals. When operating without error, noise, or collisions, the DDSA guarantees that robots will find the nearest targets first which minimizes transport cost. This provides complete coverage of an area while minimizing repeated searches of the same location.

Each robot's path is calculated based on the number of robots r, the c^{th} circuit (where a circuit is one revolution of the spiral), and the distance g between the interlocking spirals. The distance g depends on the target detection range of the robot.

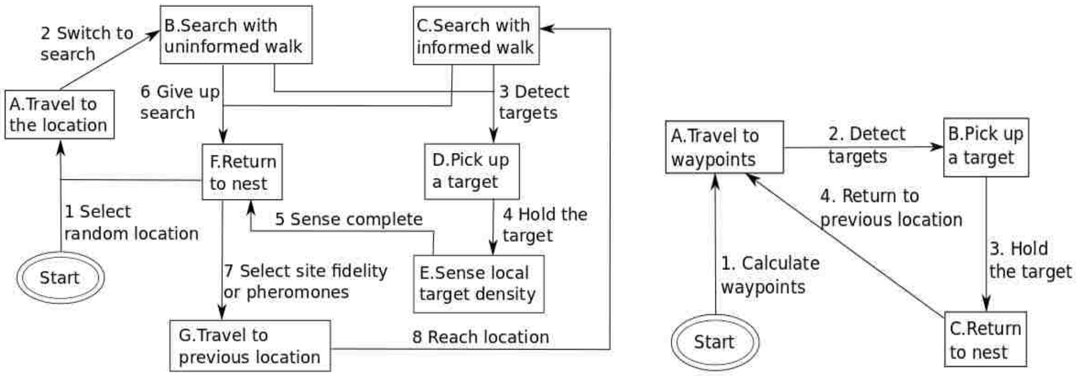
For the Swarmie robots in these experiments, $g = 0.41\,\mathrm{m}$. Eq. (7.1) and Eq. (7.2) calculate the number of steps (F) of each spiral path to the north (N) and south (S) directions by each of i robots on circuit c. c increases by one if robots complete their current circuit. Distances travelled east (E) and west (W) are similarly calculated. By solving the recurrence relation given in (Fricke et al., 2016) we can simplify the DDSA formulation to the following two equations:

$$F^N_{c,i} = F^E_{c,i} = \begin{cases} i & \text{if } c = 1 \\ (2c-3)r + 2i & \text{if } c > 1 \end{cases} \tag{7.1}$$

$$F^S_{c,i} = F^W_{c,i} = \begin{cases} 2i & \text{if } c = 1 \\ F^N_{c,i} + r & \text{if } c > 1 \end{cases} \tag{7.2}$$

Fig. 7.1(b) shows how each individual robot transitions through a series of states as it forages for targets. The robots are initially distributed around the collection zone. Each robot calculates waypoints along its arm of the distributed spiral path (Condition 1). Once it is complete, each robot travels along its planned path and searches for targets (State A). Once a robot finds a target (Condition 2), it picks it up (State B). The robot carries the target directly back to the collection zone (State C).

In each subsequent foraging trip, the robot returns directly to the last location where it found a target (effectively implementing site fidelity for every foraging trip) where it resumes its spiral search. This relatively simple algorithm guarantees that the closest targets are found first, and due to site fidelity a robot will repeatedly return to a location so that it efficiently collects clusters of targets.



(a) States in the CPFA

(b) States in the DDSA

Figure 7.1: Robot states in the CPFA and DDSA.

7.7 Description of Simulated & Physical Robots

Our experiments run in a Gazebo (Koenig and Howard, 2004) simulation and in outdoor arenas using the Swarmie robot platform, all of which were custom designed and built for the NASA Swarmathon swarm foraging competition (Ackerman et al., 2018).

Gazebo simulates physical interactions among robots, targets, obstacles, and the arena. In Gazebo, we carefully construct models of real robots, obstacles, and targets in an arena size scaled to match the 14 m × 14 m arenas used for our physical experiments.

Each Swarmie robot is equipped with a front web camera, three pairs of ultrasound range sensors, and a gripper for target pickup and drop off (see Fig. 7.3a). The camera has a field of view with a 1 rad arc and range of 1 m. Objects detected within 0.6 m by ultrasounds trigger a simple obstacle avoidance routine. The sensors detect collisions every millisecond. The robot senses the location of the object and turns left or right in order to avoid the collision. A diagnostic package monitors hardware components and gives alerts to users. Complete build instructions for the Swarmie robot are publicly available[1], and source code of the CPFA[2] and DDSA[3] are available on Github.

The targets collected by the robots are soft cubes with an AprilTag (2D barcode fiducials developed for robotics applications) (Olson, 2011) on each face. For the first two sets of experiments, targets were distributed in the arena in clusters of various sizes with locations determined at random. The collection zone in the center of the arena was a square area with AprilTags on its boundary. The camera detected the AprilTags which were translated into a location in space relative to the robot's position. This allows the robot to pick up targets in the arena and drop them off in the collection zone. Physical robots used a gripper with an actuated wrist for grabbing and dropping targets, which was also simulated in Gazebo. Target collection is an error-prone complex task. The average number of attempts to pick up a target is 1.85 ± 1.2 in simulation and is 1.96 ± 1.2 in physical experiments. In physical experiments, although robots attempted to visually confirm that a target was successfully picked up, robots sometimes drop targets or detect that a target was collected when it was not. On rare occasions targets were dropped after a collision or robots would steal targets from each other. More commonly, once targets were deposited in the collection zone, robots could accidentally push them out again. We manually counted and then removed collected targets from the collection zone to

[1] `https://github.com/BCLab-UNM/Swarmathon-Robot`

[2] `https://github.com/BCLab-UNM/CPFA-ROS`

[3] `https://github.com/BCLab-UNM/DDSA-ROS`

avoid these accidents. Recognition of targets and the collection zone was impacted by light conditions, particularly the apparent contrast between shadows and lighted areas. While the Gazebo simulation was quite faithful to the rigid body dynamics of the robot and targets, it could not capture subtle effects of lighting and the full range of physical interactions between robots, targets, and the environment.

Localization is a challenge in swarm robotics, particularly with low-cost robots using error-prone sensors and actuators. The robots in these experiments use an extended Kalman filter (EKF) to fuse Global Positioning System (GPS) information and odometry from an Inertial Measurement Unit (IMU) and wheel encoders to determine position, orientation, and the locations of the collection zone and pheromone waypoints. We estimated the average accuracy of GPS localization to be 0.5 m, and we estimated how the IMU and encoders accumulate drift over time by collecting the from our experiments.

Robots use their front web cameras to detect targets, and when close to the collection zone, their cameras detect the AprilTags on the boundary of the collection zone. When robots arrive at the collection zone, they update their locations. To approximate the physical experiments, the magnitudes of the noise are generated by Gaussian distributions $\mathcal{N}(0.4, 0.5)$ on GPS receivers, $\mathcal{N}(0, 0.005)$ on ultrasound sensors, $\mathcal{N}(0, 0.007)$ on cameras, and ($\mathcal{N}(0.35, 0.35)$ on accelerators, $\mathcal{N}(0.5, 0.5)$ on angular rate, and $\mathcal{N}(0, 0.01)$ on heading) of IMUs.

The architecture of the CPFA and DDSA implementation in ROS is shown in Fig. 7.2. Our *Rover GUI* either ran on a computer hosting Gazebo for a simulated swarm or connected to the physical robots in the swarm through a wireless network. The *GUI* acted as a communicator between users and robots. The results of the interaction between robots and objects in the simulation were sent to the ROS adapter.

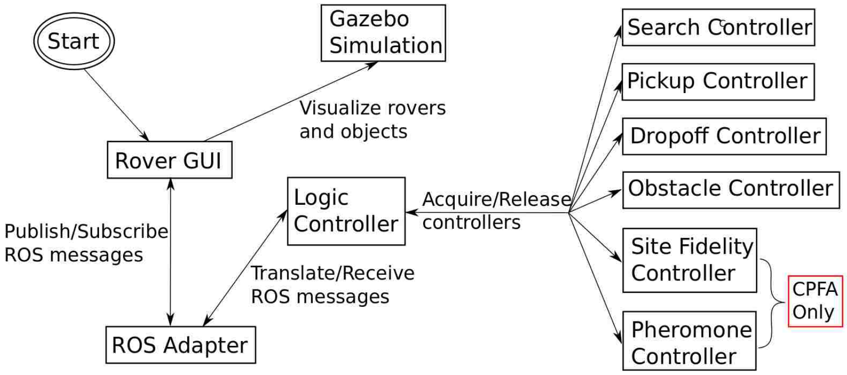


Figure 7.2: The architecture of the CPFA and DDSA in ROS.

We deconstructed the robot control system into a series of functional controllers. Each controller was assigned a specific behavior. To fairly compare the CPFA and DDSA, we used the same implementation of the pickup, dropoff, and obstacle avoidance controllers. The deterministic spiral search and the ant-like stochastic search were implemented in the search controllers of the CPFA and DDSA, respectively. Site fidelity and pheromone controllers were only defined in the CPFA. Site fidelity in the DDSA was incorporated into the search controller as a feature. Different priorities were assigned to controllers in different states. Higher priority controllers subsumed the roles of lower ones by suppressing their outputs. Robots switched controllers when they change their states. The logic controller handled transitions among all the controllers.

The ROS implementations of the CPFA and DDSA were directly loaded onto the Swarmie onboard Linux computer for physical robot experiments (see Fig. 7.3b). A demonstration video showing the CPFA and DDSA in the simulation and physical experiments are available on our YouTube playlist[4].

7.8 Experimental Setup

We evaluated four experimental configurations to measure the performance of the foraging algorithms. In the first two configurations, 128 targets were placed uniformly in a power-law distribution of cluster sizes which emulated the distribution of many resources in natural environments (Ritchie, 2009). The targets were placed in 1 cluster of 4×8 cubes, 2 clusters of 4×4 cubes, 8 clusters of 2×2 cubes, and 32 single cubes with each cluster location chosen at random. The first configuration had no obstacles, while the second configuration had the same distribution of targets and 4 obstacles ($1\,\mathrm{m} \times 0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ synthetic rocks). In the third configuration, 128 targets were placed in lines along the edges of the four arena walls (2 m). In the fourth configuration, four 4×8 clusters of targets were placed in the four corners far (7.43 m) from the collection zone. The third and fourth configurations were designed to be more challenging with targets far from the center with no obstacles. In every experiment we placed 4 robots and a collection zone in the center of a $14\,\mathrm{m} \times 14\,\mathrm{m}$ arena. Robots foraged for 20 min in each experiment.

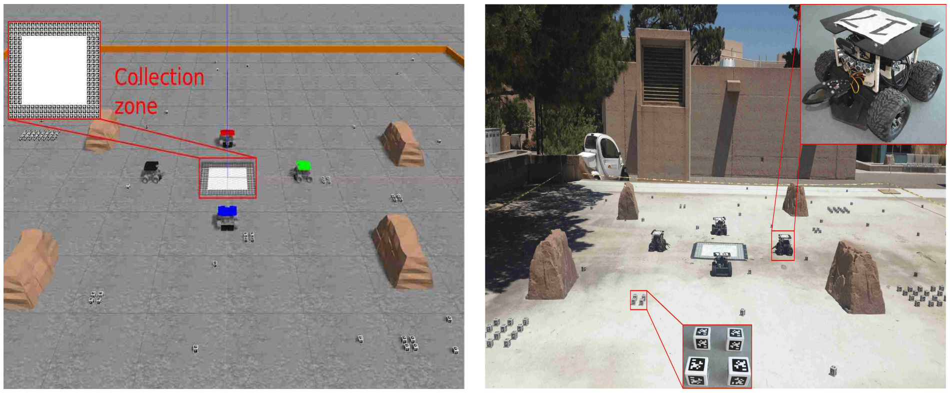
Table 7.1 summarizes the configurations and replicates of the experiments. Fig. 7.3a illustrates an example setup with obstacles in simulation while Fig. 7.3b shows the same setup in a physical experiment. We replicated all experiments 30 times in simulation. Physical experiments were repeated 15 times for the first two configurations (in which the locations of targets were chosen at random) and 5 times

[4] https://tinyurl.com/yceu6p9b

for the third and fourth configurations (in which targets were placed in fixed locations).

Table 7.1: Experimental Setup and Replicates

	Target		Simulation	Physical
Config.	Distribution	Obst.	replicates	replicates
1	Power law	No	30	15
2	Power law	Yes	30	15
3	Edges	No	30	5
4	Corners	No	30	5



(a) Simulated Experiment

(b) Physical Experiment

Figure 7.3: Simulated and Physical experiments with 4 robots, 128 cubes, 4 obstacles and one central collection zone. Configuration 2 is shown, Target cluster sizes are described in Table 1, obstacles are placed 3 to 5 m from the center, and the exact location of each obstacle, target and target cluster is chosen at random.

7.9 Results

We used interquartile-range notched box plots to visualize the statistical relationships between experiments (Mcgill et al., 1978). Non-overlapping notches indicate the measurements were drawn from different distributions at the 95% confidence level. Results were compared using the Mann-Whitney U test for simulated experiments and the student's t-test for physical experimental results (see (De Winter, 2013), the t-test for small sample sizes). The statistical significance is indicated by asterisks in figures (*** indicates $p < 0.001$, ** indicates $p < 0.01$m, * indicates $p < 0.05$, and 'NS' indicates no statistical difference). The notches in the boxes indicate the 95% confidence interval of the medians.

The foraging performance of the DDSA and CPFA in simulated and physical experiments, with and without obstacles, is shown in Fig. 7.4. In simulations, the median number collected by the DDSA was 18% higher than the CPFA without obstacles (the U test two-tailed p-value was $p = 0.0002$), and it was 26% higher than the CPFA with obstacles ($p = 0.01$). In physical experiments, the CPFA was 25% higher than the DDSA ($p = 0.04$) without obstacles and there was no significant difference with obstacles.

In the simulation, when resources were placed in the corners (configuration 4) the DDSA collected 26% more than the CPFA, with no difference between the two algorithms with resources placed along the edges of the arena. There was no significant difference between algorithms in either configuration 3 or 4 in physical experiments (shown in Fig. 7.5).

Fig. 7.6 summarizes the foraging performance across all experiments (240 simulations and 80 physical experiments). The DDSA collects 20% more targets than the CPFA in simulation, but there is no significant difference in physical environments. DDSA performance decreases more dramatically in physical experiments: the DDSA

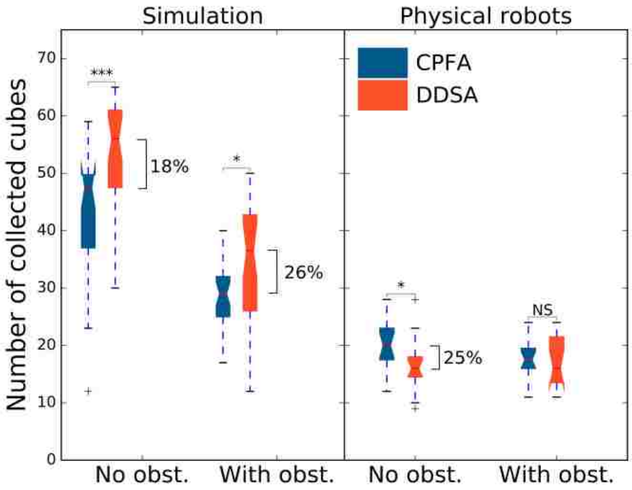


Figure 7.4: Foraging performance of the DDSA and CPFA with and without obstacles, for 30 trials in simulation, and 15 trials in physical experiments using configurations 1 and 2 (shown in Fig. 7.3).

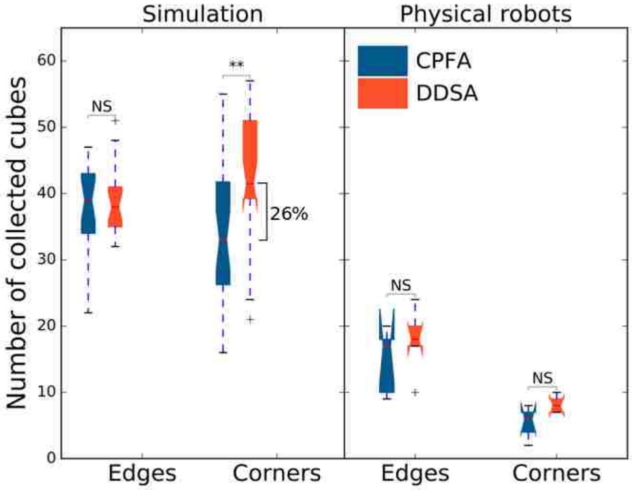


Figure 7.5: Foraging performance with cubes lined to the edges and clustered in corners.

is 163% better in simulated vs. physical experiments, while the CPFA is only 95% better in simulated vs. physical experiments.

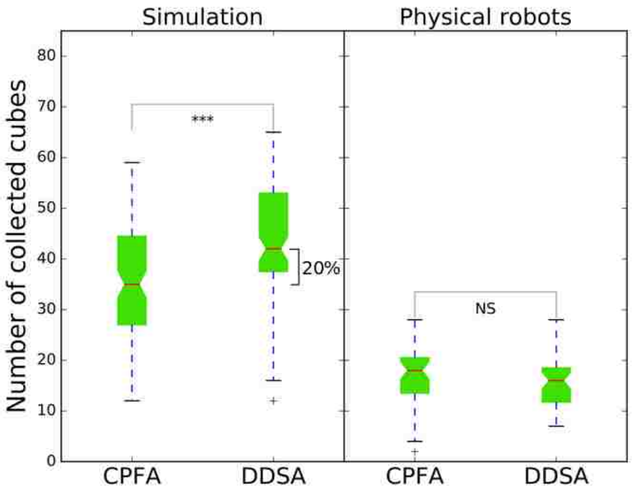
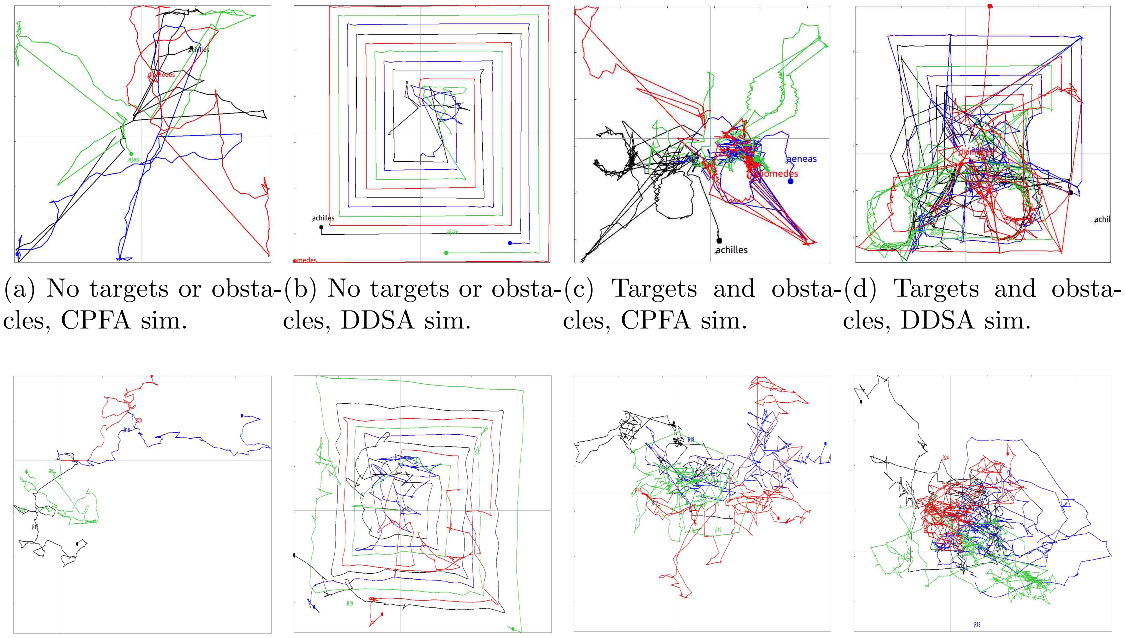


Figure 7.6: Overall foraging performance with all experiments.

Fig. 7.7 shows traces of robots executing the CPFA and DDSA in different configurations. Panels (a) to (f) demonstrate baselines where each algorithm ran for 5 min without targets and obstacles. The traces in panels (a) and (b) illustrate the stochastic search pattern of the CPFA and the interlocking spiral pattern of the DDSA in simulation. The search patterns are still clear even with some drift in physical experiments without targets or obstacles (see (e) and (f)). Panels (c) to (h) demonstrate traces given targets and obstacles. In the simulations (see (c) and (d)), the characteristic search patterns of the CPFA and DDSA are still evident, even though they are disrupted by direct paths to and from targets and empty regions outlining where targets are placed. However, in panel (h), the deterministic spiral is no longer visible because it is disrupted by robots interacting with each other, targets, and obstacles. In real environments with obstacles, the traces from the DDSA appear as random as those from the CPFA.



(a) No targets or obstacles, CPFA sim.
(b) No targets or obstacles, DDSA sim.
(c) Targets and obstacles, CPFA sim.
(d) Targets and obstacles, DDSA sim.

(e) No targets or obstacles, CPFA phy.
(f) No targets or obstacles, DDSA phy.
(g) Targets and obstacles, CPFA phy.
(h) Targets and obstacles, DDSA phy.

Figure 7.7: Odometry traces of 4 robots in simulation (left column) and physical experiments (right column). Each robot path is a different colored line. Obstacles are not shown, but the empty areas in (c) and (d) imply their location.

7.10 Discussion

In a perfect environment without error or noise the DDSA outperforms the CPFA by collecting more targets in a fixed time period. In the CPFA, the movement is random and some locations are visited multiple times while others are never visited at all. The DDSA guarantees complete coverage of the entire arena given sufficient time, and each location is visited only once. Our experiments in simulation confirm this expectation: the DDSA outperforms the CPFA (Fig. 7.4), even when the simulation includes physical interactions, collision avoidance, and some sensing and localization error. Even with resource placements far from the arena center, specifically designed to be difficult for the DDSA, it performs as well as or better than the CPFA in simulations (Fig. 5)

Not surprisingly, foraging performance was higher in both algorithms in simulations compared to physical experiments (Fig. 7.4 and Fig. 7.5). In prior ARGoS simulations (Fricke et al., 2016), the DDSA collected targets 34% faster than the CPFA. Here our simulations include more realistic object pickup and dropoff and the complex physics of the Swarmie platform. In these simulations, the DDSA performed only 20% better than the CPFA (Fig. 6). The DDSA was no better than the CPFA in physical experiments, and in fact, the CPFA outperformed the DDSA in physical experiments without obstacles (Fig. 7.4). As more realism is included, the CPFA becomes as good as, or better than, the DDSA.

To understand why stochasticity affects performance, we recorded odometry traces of robots in simulation and physical robots. Without targets or obstacles in the arena, the essence of the stochastic CPFA and the DDSA spirals were evident in the odometry traces of physical robots (Fig. 7.7(e) and (f)). However, in contrast to the simulated traces, once targets and obstacles were placed in physical arenas, the DDSA spirals were disrupted so much that they were no longer distinguishable in robot paths (Fig. 7.7(g) and (h)).

There are multiple factors that can cause the deterministic movement of the DDSA to appear as stochastic as that of the CPFA: noisy sensors, actuator drift, positional noise from odometry and GPS. The search pattern of the CPFA is also altered by these factors, but it is less relevant since the CPFA search pattern is random by design. The advantage of the CPFA is that it is designed for effective foraging under the assumption that robot movement is random. Thus, it is less impacted by real-world factors that degrade the DDSA, to the point of making it appear effectively random.

Interestingly, the noise in physical experiments generated stochastic robot movement even when the underlying algorithm was deterministic. This suggests that when robotic algorithms are inspired by biological observations, care should be taken to understand whether the biological behaviors are inherently stochastic or if they only appear so because they are observed in noisy natural environments.

Acknowledgments

We are grateful to NASA Swarmathon teams for improved implementations of foraging algorithms in Swarmies, and we thank Jarett Jones, Manuel Meraz, Kelsey Geiger, Tobi Ogunyale, William Vining, Vanessa Surjadidjaja, and John Ericksen with development, testing and editing.

Chapter 8

From Simulation to Physical MPFA$_T$

Our ARGoS simulations demonstrate that the MPFA with the hierarchical branching transportation network can reach a scale-invariant foraging performance per robot. Our ultimate goal is to evaluate the foraging performance of MPFA$_T$ in real robots. Instead of evaluating a large swarm with thousands of physical robots, we will demonstrate the viability of evaluating a few robots foraging resources in a region and a depot delivering resources to the central collection zone. If this proof of concept works in a region, our predictions of scale invariance give us confidence that a large swarm with multiple depots will scale up linearly in a large real world environment.

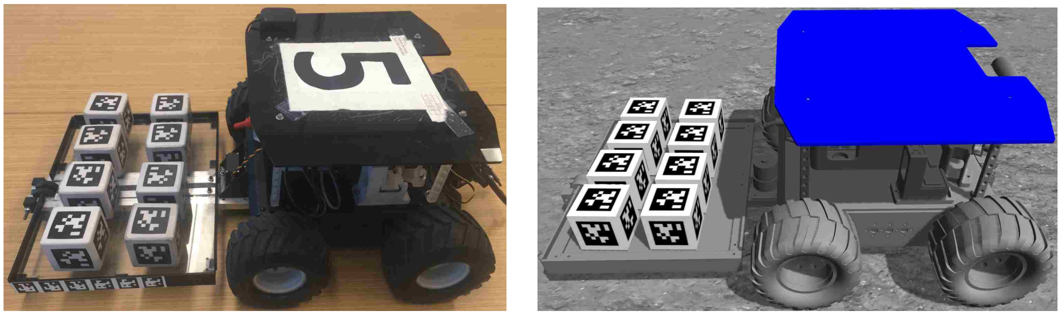
In order to implement the MPFA$_T$, searching robots need to be updated to be able to find depots and drop cubes on depots successfully. In addition, searching robots need to detect whether depots are fully loaded or not. If so, they communicate with depots and depots start to deliver resources. Based on our experience, all these can be simulated and implemented with physical robots.

8.1 Physical Depot Design

We designed a depot which initially sketched in Simulink, then tested in a cardboard prototype, and finally built from metal and plexiglases as shown in Fig. 8.1a. This work was done in collaboration with Tatsuhiro and Takaya from MIE university, Japan.

We use our existing robot platform, designed by our lab for the NASA Swarmathon competition (Secor 2016; Ackerman et al. 2018). The complete build instructions for the Swarmie robot are available on GitHub[1]. Swarmie robots have a gripper that facilitates the pick up and drop off of cubes. Minor structural modifications were required in order to convert Swarmie robots into mobile depots capable of holding multiple (up to 8) collected resources. We have designed depot hardware and built a prototype which we control with a joystick, but eventually will dump resources in the collection zone autonomously.

[1] https://github.com/BCLab-UNM/Swarmathon-Robot



(a) A physical depot

(b) A simulated depot in Gazebo

Figure 8.1: A physical and simulated depot. (a) A physical depot with 8 cubes on its carriage. The depot is a Swarmie robot equipped with a plexiglass plate. A stainless steel wire connects the shaft of the plate and a servo motor. The motor controls the plate to dump cubes. The robot dimensions are $34 \times 25 \times 22$ cm and the carriage dimensions are $31 \times 24 \times 3$ cm. (b) A simulated depot in Gazebo. Its size is identical to the physical depot.

8.2 Gazebo Simulation

Other swarm algorithms, including the DDSA and the CPFA, have been implemented in ROS and subsequently tested in Gazebo. Based on our experience with these existing foraging algorithms, we simulated a MPFA depot with Swarmie robots in Gazebo (see Fig. 8.1b). We import 3D models of real depots in Simulink to Gazebo. Gazebo simulates physical interactions among robots, targets, obstacles, and the arena.

The biggest benefit of implementing the MPFA on Gazebo is that testing is faster in simulation and code is very easily transferred onto the onboard Linux computer on the Swarmie robots. The ease of this transfer is evidenced by the successfully transferred CPFA and DDSA code to Swarmie robots that operated in outdoor arenas up to 14 x 14 m as described in Chapter 7. The physical experiments of the CPFA and DDSA showed that Swarmie robots can detect, pick up and drop off cubes, avoid collision, navigate, and return to collection zones reliably in a real world environment with sensors and actuator errors.

Currently, we imported the 3D model of the physical depot into Gazebo successfully. The carriage can load multiple cubes and it can dump cubes into the collection zone. A scenario of the simulation with multiple depots is shown in Fig. 8.2.

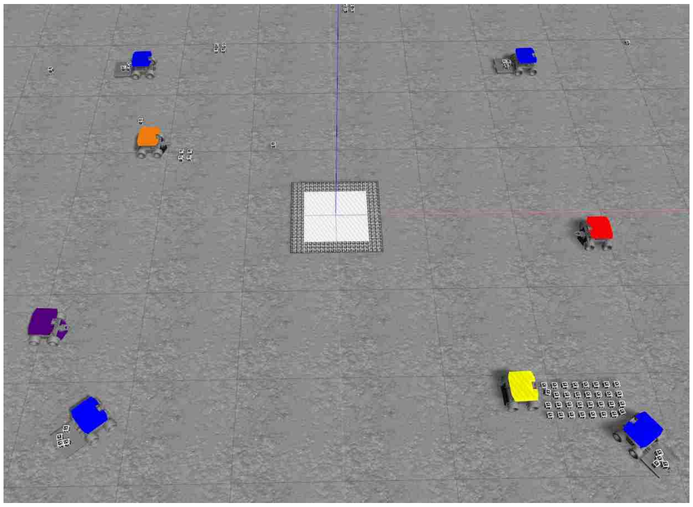


Figure 8.2: A scenario of multiple depots in Gazebo. Four depots with blue covers are distributed uniformly in a 14 × 14 meter arena. Four searching robots are searching for cubes in the arena. The collection zone is located in the center.

8.3 From Physical to Virtual and Back Again

To demonstrate truly scalable foraging swarms, full implementation of the autonomous dump behaviors of physical robots and depots in an outdoor environment is the next step. In the initial stage, physical design provides prototypes and models for simulation. Then, simulating the depot behaviors in Gazebo makes the development process faster since programming and testing on hardware directly is time consuming. The software can transfer to physical depots after the testing in simulation. Then, the feedback of testing physical depots will refine the simulation again. A well designed autonomous physical depot will be produced from the development from the physical robots to simulation back and forth. The physical and simulated implementation of emergent coordination observed in natural systems and robotic swarms is also described in Hecker's work (P. Hecker et al., 2012).

Comparing the performance in physical robots (see Chapter 7) demonstrates the *reality gap* between the simulation and physical robots. We expect the results in a simulation should match those in physical robots. Otherwise, the simulation does not provide sufficient models. George E. P. Box quotes that all models are wrong, but some models are useful.

Swarm simulations tend to be *minimalistic*, so that they include only few relevant features of robots. Other features are modeled as abstract. When a robotic hardware is complicated, simulations need to consider more components, including sensors, actuators, grippers, and etc. These make simulations to tradeoff between the accuracy and speed. Even with accurate simulations, the reality gap can not be completely filled (Jakobi et al., 1995). Once physical environments or robot hardware have minor changes, the accuracy will be affected. However, we have to consider to reduce the reality gap as much as possible. The solution is ranking features by relevance and then simulating higher relevant features with more accuracy.

The simulation accuracy can be improved by using collected errors sampled from real robots as described in the modeling of ARGoS (Pini et al., 2013; Pinciroli et al., 2018). The $\mathrm{MPFA_T}$ also demonstrates the possibility of designing a hierarchical branching transportation network for robot swarms in the 3D world. Instead of using ground robots, UAVs are used to complete foraging tasks in this scenario. The number of branches, the path length, and the region size are updated correspondingly.

However, we design transportation networks in a regular and convex arena. In the real world, the arena is always irregular. In the future, designing transportation networks in irregular and non-convex arenas will be challenging.

8.4 Future work

The planned experiments of simulated and physical robots and depots includes the following steps:

- Searching robots find depots by detecting AprilTags on the boundary of depots.
- Searching robots drop cubes on depot successfully.
- Depots know they are fully loaded (e.g. 8 cubes) by communicating with searching robots which drop cubes on them.
- Depots drive to the central collection zone and dump cubes into it.
- Depots drive back to their original locations.

The physical experiments will be completed in September and the work will be submitted to ICRA 2020 conference.

Chapter 9

Conclusions

9.1 Concluding Remarks

A foraging algorithm should be efficient not only in a small group of robots, but also in a large swarm. However, the foraging performance decreases when the swarm increases. More collisions are produced with a larger swarm, and longer travel distances are required in a larger arena. The foraging performance per robot decreases when swarm and arena size increases. This phenomenon is called *diminishing returns* which is universal in robot swarms. This research presents a scale-invariant foraging robot swarm in which the foraging performance per robot is linear with swarm and arena size.

We presented the MPFA to mitigate the diminishing returns. In the MPFA, multiple collection zones are distributed uniformly in a foraging arena rather than one central collection zone in the CPFA. Robots always return to their closest collection zones for delivering resources. Collisions are distributed to multiple collection zones rather than aggregated around a central collection zone in the CPFA. Our results demonstrate that the MPFA improves foraging performance, mitigates congestion, and reduces travel distances. However, the foraging performance per robot of the MPFA decreases when the swarm size is very large. Ideally, the foraging performance per robot is linear with the swarm size. Inter-robot collisions and long travel distances are a major challenge in scaling robot swarms.

Our solution is introducing the $\mathrm{MPFA_T}$ with a bio-inspired hierarchical transportation network upon the MPFA. Its scalability is achieved with the transportation network inspired by mammal cardiovascular networks. In the $\mathrm{MPFA_T}$, depots travel between two collection zones to deliver resources. A constant number of depots travel to the central collection zone which reduces the local robot density in the center. The hierarchical transportation network minimizes travel distances of depots and mitigates collisions. Therefore, it improves the foraging performance and scalability.

We predict the scaling exponents in robot swarms with a simplified model (a constant delivery velocity) and a biological inspired model (the velocity increases with the arena size). However, biological systems are limited to sublinear scaling. We can build upon biological scaling principles to design a scale-invariant foraging swarm. Biological transportation networks have constraints on the velocity of blood cells and the size of blood vessels. Their total volume must be equal to a constant fraction of the mammal volume (West et al., 1997). In contrast, the capacity of delivering robots can increase to ensure that delivery capacity matches foraging rate in the robot foraging transportation network. This approach essentially aggregates collected resources in larger depots where capacity is set according to the structure of the hierarchical branching transportation network, much the way blood is aggregated in larger vessels like the aorta in cardiovascular networks.

This work demonstrates the viability of an artificial bio-inspired transportation network in robot swarms. The transportation network is optimized when the delivery rate is equal to the foraging rate. Therefore, we can predict the required number of robots, depots, collection zones, and the delivery capacity for a given size arena to achieve scale invariant foraging.

The depots with larger capacities is the analogy to the larger vessels in organisms. However, there are fewer large depots than small depots. Consequently, there will be fewer single depot failures in the new model. Though the failure of a single large depot will cause a major problem, we can have more monitoring on larger depots. It is more efficient to monitor on a small number of large depots than on a large number of small depots.

9.2 Broader Impact

In biology, scaling theory investigates how efficiently targets can be moved through spatial networks (West et al., 1997; Banavar et al., 2010; Savage et al., 2008). Scaling theory makes predictions beyond individual organisms, to explain the efficiency of ant colonies (Hou et al., 2010), societies (Moses and Brown, 2003; Brown et al., 2011), and even computer chip design (Moses et al., 2016). On one hand, our $\mathrm{MPFA_T}$ validates the generalized formula of scaling exponent $D/(D+1)$ in 2D robotic systems. This advantage is particularly apparent when swarm size and arena size are very large. On the other hand, our work impacts other research. Here we list some work impacted by our scalable robot swarms.

Autonomous robot taxis: Autonomous self-driving cars attract more and more attention in recent years. They are supposed to eliminate some of the risks of human error, particularly because their sensors are always paying attention to their surroundings. Uber, Lyft, and Waymo are prime ride-hailing competitors in the U.S. In 2018, Waymo received permits to begin offering robot taxi service in and around Phoenix, Arizona. Our MPFA$_{dynamic}$ brings a solution to the allocation of robot taxis efficiently. We expect robot taxis travel shorter distances to serve passengers. Robot taxis are allocated to areas which are computed by the number and locations of taxi service requests in those areas. They are allocated dynamically to match the service requests.

Autonomous supply chain systems: The success of this work also provides a distributed approach rather than a centralized approach to the supply chain of the future (Akanle and Zhang, 2008). It is likely to see the continued growth of autonomous robots in these areas. Autonomous robots have a strong presence already in manufacturing, final assembly, and warehousing. With the hierarchical transportation network, the locations of warehouses are determined. The transportation performance and the number of warehouses and delivering robots will be predicted. The supply chain will have the following features.

- Scalability: Every robot does its jobs just as fast no matter how big the problem is.

- Minimize idle time: No robot is ever waiting for another robots.

- Minimize storage space and delivery delay: No extended cargo stay in warehouse and is not delayed in transportation from factories to destinations.

Charging in robots: An automatic recharging mechanism is a crucial component in robotics (Pickem et al., 2017). Our decentralized model is an efficient solution to the charging problem in robot swarms. In the centralized model, all robots have to travel long distances back to the central station for charging. Robots fail on the way if there is a large amount of congestion around the central station. Failed robots block the way to the central charging station and cause more failures. In the decentralized model, multiple charging stations are distributed into collection zones. Robots travel a short distance back to the closest charging station. If there is a failure, it does not affect the entire swarm. Robots still can find other charging stations. Even if a charging station fails, it only causes cascading failures on that branch. The charging is more efficient in the decentralized model and it is more robust relative to robot failures.

Communication manners in robots: As the swarm size grows, it is inefficient if every robot connects with the central server. Based on the distributed hierarchical transportation network, the communication manner of robot swarms will transit from the centralized manner to the distributed manner. In the centralized communication, robots connect with the central server. The central server has a high communication workload and it has a high risk of failure. In the distributed manner, robots communicate with distributed servers in their communication range locally. Distributed servers communicate with servers in next hierarchical levels. The communication workload is distributed to multiple servers. If a server experiences failure from a cyber attack, mobile robots have a chance to communicate with other servers close to them. The communication network is more scalable and robust with respect to communication workload.

Taken as a whole, this dissertation presents a comprehensive scalable foraging swarm robotics system. Although it demonstrates the viability of scalable swarm robotics in an academic research laboratory, this work provides a foundation for designing and implementing scalable robot swarms that can work in some real world applications. As humans explore more dangerous and distant new worlds, there will be several projects underway that are sending robots where humans dare not tread. Thousands of robots are deployed into the ocean depths and extra-planetary surfaces. Successful exploration requires large robot swarms that work efficiently in the real world.

Bibliography

Ackerman, S. M., Fricke, G. M., Hecker, J. P., Hamed, K. M., Fowler, S. R., Griego, A. D., Jones, J. C., Nichol, J. J., Leucht, K. W., and Moses, M. E. (2018). The swarmathon: An autonomous swarm robotics competition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*.

Akanle, O. and Zhang, D. (2008). Agent-based model for optimising supply-chain configurations. *International Journal of Production Economics*, 115(2):444 – 460. Institutional Perspectives on Supply Chain Management.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Bac, C. W., Henten, E. J., Hemming, J., and Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6):888–911.

Baezayates, R., Culberson, J., and Rawlins, G. (1993). Searching in the plane. *Information and Computation*, 106(2):234 – 252.

Banavar, J. R., Moses, M. E., Brown, J. H., Damuth, J., Rinaldo, A., Sibly, R. M., and Maritan, A. (2010). A general basis for quarter-power scaling in animals. *Proceedings of the National Academy of Sciences*, 107(36):15816–15820.

Banerjee, S. and Moses, M. (2010a). *Modular RADAR: An Immune System Inspired Search and Response Strategy for Distributed Systems*, pages 116–129. Springer Berlin Heidelberg, Berlin, Heidelberg.

Banerjee, S. and Moses, M. (2010b). Scale invariance of immune system response rates and times: Perspectives on immune system architecture and implications for artificial immune systems. *Swarm Intelligence*, 4(4):301–318.

Barca, J. C. and Sekercioglu, Y. A. (2013). Swarm robotics reviewed. *Robotica*, 31(3):345359.

Bentley, J. L., Weide, B. W., and Yao, A. C. (1980). Optimal expected-time algorithms for closest point problems. *ACM Trans. Math. Softw.*, 6(4):563–580.

Berman, S., Halász, Á., Hsieh, M. A., and Kumar, V. (2008). Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4376–4381.

Beverly, B. D., McLendon, H., Nacu, S., Holmes, S., and Gordon, D. M. (2009). How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behavioral Ecology*, 20(3):633.

Bezzo, N., Hecker, J. P., Stolleis, K., Moses, M. E., and Fierro, R. (2015). Exploiting heterogeneous robotic systems in cooperative missions. pages 1–23.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *From Natural to Artificial Swarm Intelligence*. Oxford University Press, Inc., New York, NY, USA.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Brooks, R. A. and Flynn, A. M. (1989). Fast, cheap and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, 42:478–485.

Brown, J. H., Burnside, W. R., Davidson, A. D., DeLong, J. P., Dunn, W. C., Hamilton, M. J., Mercado-Silva, N., Nekola, J. C., Okie, J. G., Woodruff, W. H., and Zuo, W. (2011). Energetic limits to economic growth. *BioScience*, 61(1):19–26.

Brue, S. L. (1993). Retrospectives: The law of diminishing returns. *Journal of Economic Perspectives*, 7(3):185–192.

Brutschy, A., Garattoni, L., Brambilla, M., Francesca, G., Pini, G., Dorigo, M., and Birattari, M. (2015). The tam: abstracting complex tasks in swarm robotics research. *Swarm Intelligence*, 9(1):1–22.

Buchanan, E., Pomfret, A., and Timmis, J. (2016a). Dynamic task partitioning for foraging robot swarms. In Dorigo, M., Birattari, M., Li, X., López-Ibáñez, M., Ohkura, K., Pinciroli, C., and Stützle, T., editors, *Swarm Intelligence*, pages 113–124, Cham. Springer International Publishing.

Buchanan, E., Pomfret, A., and Timmis, J. (2016b). Dynamic task partitioning for foraging robot swarms. In Dorigo, M., Birattari, M., Li, X., López-Ibáñez, M., Ohkura, K., Pinciroli, C., and Stützle, T., editors, *Swarm Intelligence*, pages 113–124, Cham. Springer International Publishing.

Burlington, S. and Dudek, G. (1999). Spiral search as an efficient mobile robotic search technique.

Camazine, S., Franks, N. R., Sneyd, J., Bonabeau, E., Deneubourg, J. L., and Theraula, G. (2001). *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA.

Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., and Dorigo, M. (2010). Artificial pheromone for path selection by a foraging swarm of robots. *Biological Cybernetics*, 103(5):339–352.

Carpintero, S., Reyes-Lpez, J.-L., and Martinez, L. (2005). Impact of argentine ants (linepithema humile) on an arboreal ant community in doana national park, spain. *Biodiversity and Conservation*, 14:151–163.

Castello, E., Yamamoto, T., Libera, F. D., Liu, W., Winfield, A. F. T., Nakamura, Y., and Ishiguro, H. (2016). Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence*, 10(1):1–31. Electronic supplementary material The online version of this article (doi: 10.1007/s11721-015-0117-7) contains supplementary material, which is available to authorized users.

Chapman, C. A., Chapman, L. J., and McLaughlin, R. (1989). Multiple central place foraging by spider monkeys: Travel consequences of using many sleeping sites. *Oecologia*, 79(4):506–511.

Couture-Beil, A. and Vaughan, R. T. (2009). Adaptive mobile charging stations for multi-robot systems. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1363–1368.

Crist, T. O. and MacMahon, J. A. (1991). Individual foraging components of harvester ants: movement patterns and seed patch fidelity. *Insectes Sociaux*, 38(4):379–396.

De Winter, J. (2013). Using the student's t-test with extremely small sample sizes. *Practical Assessment, Research and Evaluation*, 18(10):1 – 12.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., Burnier, D., Campo, A., Christensen, A. L., Decugniere, A., Di Caro, G., Ducatelle, F., Ferrante, E., Forster, A., Gonzales, J. M., Guzzi, J., Longchamp, V., Magnenat, S., Mathews, N., Montes de Oca, M., O'Grady, R., Pinciroli, C., Pini, G., Retornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stutzle, T., Trianni, V., Tuci, E., Turgut, A. E., and Vaussard, F. (2013). Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics Automation Magazine*, 20(4):60–71.

Enright, J. J. and Wurman, P. R. (2011). Optimization and coordinated autonomy in mobile fulfillment systems. In *Proceedings of the 9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots*, AAAIWS'11-09, pages 33–38. AAAI Press.

Ferrante, E., Duéñez Guzmán, E., Turgut, A. E., and Wenseleers, T. (2013). Geswarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 17–24, New York, NY, USA. ACM.

Ferrante, E., Turgut, A. E., Duez-Guzmn, E., Dorigo, M., and Wenseleers, T. (2015). Evolution of self-organized task specialization in robot swarms. *PLOS Computational Biology*, 11(8):1–21.

Fewell, J. H. (1990). Directional fidelity as a foraging constraint in the western harvester ant, pogonomyrmex occidentalis. *Oecologia*, 82(1):45–51.

Fink, W., Dohm, J. M., Tarbell, M. A., Hare, T. M., and Baker, V. R. (2005). Next-generation robotic planetary reconnaissance missions: A paradigm shift. *Planetary and Space Science*, 53(1415):1419 – 1426.

Flanagan, T. P., Letendre, K., Burnside, W., Fricke, G. M., and Moses, M. E. (2011). How ants turn information into food. In *Artificial Life (ALIFE), 2011 IEEE Symposium*, pages 178–185. IEEE.

Flanagan, T. P., Letendre, K., Burnside, W. R., Fricke, G. M., and Moses, M. E. (2012). Quantifying the effect of colony size and food distribution on harvester ant foraging. *PLOS ONE*, 7(7):1–9.

Flanagan, T. P., Pinter-Wollman, N. M., Moses, M. E., and Gordon, D. M. (2013). Fast and flexible: Argentine ants recruit from nearby trails. *PLOS ONE*, 8(8):1–7.

Font Llenas, A., Talamali, M. S., Xu, X., Marshall, J. A. R., and Reina, A. (2018). Quality-sensitive foraging by a robot swarm through virtual pheromone trails. In Dorigo, M., Birattari, M., Blum, C., Christensen, A. L., Reina, A., and Trianni, V., editors, *Swarm Intelligence*, pages 135–149, Cham. Springer International Publishing.

Fricke, G. M., Hecker, J. P., Griego, A. D., Tran, L. T., and Moses, E. M. (2016). A distributed deterministic spiral search algorithm for swarms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016).*

Frigg, R. and Hartmann, S. (2018). Models in science. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition.

Garnier, S., Tache, F., Combe, M., Grimal, A., and Theraulaz, G. (2007). Alice in pheromone land: An experimental setup for the study of ant-like robots. In *2007 IEEE Swarm Intelligence Symposium*, pages 37–44.

Gazi, V. and Passino, K. M. (2004). Stability analysis of social foraging swarms. *Trans. Sys. Man Cyber. Part B*, 34(1):539–557.

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

Gordon, D. M. and Kulig, A. W. (1996). Founding, foraging, and fighting: Colony size and the spatial distribution of harvester ant nests. *Ecological Society of America*, 77(8):2393–2409.

Gro, R. and Dorigo, M. (2009). Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, 1(1-2):1–13.

Halász, Á., Hsieh, M. A., Berman, S., and Kumar, V. (2007). Dynamic redistribution of a swarm of robots among multiple sites. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2320–2325.

Hecker, J. P., Carmichael, J. C., and Moses, M. E. (2015). Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 434–440.

Hecker, J. P. and Moses, M. E. (2013). An evolutionary approach for robust adaptation of robot behavior to sensor error. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1437–1444, New York, NY, USA. ACM.

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

Hecker, J. P., Stolleis, K., Swenson, B., Letendre, K., and Moses, M. E. (2013). Evolving error tolerance in biologically inspired iant robots. In *In ECAL 2013*.

Hoff, N. R., Sagoff, A., Wood, R. J., and Nagpal, R. (2010). Two foraging algorithms for robot swarms using only local communication. In *2010 IEEE International Conference on Robotics and Biomimetics*, pages 123–130.

Hölldobler, B. (1976). Recruitment behavior, home range orientation and territoriality in harvester ants, pogonomyrmex. *Behavioral Ecology and Sociobiology*, 1(1):3–44.

Hou, C., Kaspari, M., Zanden, H. B. V., and Gillooly, J. F. (2010). Energetic basis of colonial living in social insects. *Proceedings of the National Academy of Sciences of the United States of America*, 107(8):3634–3638.

Hsieh, M. A., Halász, Á., Berman, S., and Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2):121–141.

Isbell, J. R. (1957). An optimal search pattern. *Naval Research Logistics Quarterly*, 4(4):357–359.

Jackson, D. E., Martin, S. J., Ratnieks, F. L. W., and Holcombe, M. (2007). Spatial and temporal variation in pheromone composition of ant foraging trails. *Behavioral Ecology*, 18(2):444–450.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In Morán, F., Moreno, A., Merelo, J. J., and Chacón, P., editors, *Advances in Artificial Life*, pages 704–720, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jones, S., Studley, M., Hauert, S., and Winfield, A. (2018). *Evolving Behaviour Trees for Swarm Robotics*, pages 487–501. Springer International Publishing, Cham.

Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V., Pereira, G., and Spletzer, J. (2006). *Distributed Search and Rescue with Robot and Sensor Teams*, pages 529–538. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kennedy, J. and Eberhart, R. C. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Khaluf, Y., Ferrante, E., Simoens, P., and Huepe, C. (2017). Scale invariance in natural and artificial collective systems : a review. *Journal of the royal society interface*, 14(136):20.

Kleinberg, J. (2007). Computing: The wireless epidemic. *Nature*, 449(7160):287–288.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE.

Lanan, M. (2014). Spatiotemporal resource distribution and foraging strategies of ants (hymenoptera: Formicidae). *Myrmecological news/Osterreichische Gesellschaft fur Entomofaunistik*, 20:53–70.

Landis, G. A. (2004). Robots and humans: synergy in planetary exploration. *Acta Astronautica*, 55(12):985 – 990.

Lein, A. and Vaughan, R. T. (2009). Adapting to non-uniform resource distributions in robotic swarm foraging through work-site relocation. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 601–606.

Letendre, K. and Moses, M. E. (2013). Synergy in ant foraging strategies: Memory and communication alone and in combination. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 41–48, New York, NY, USA. ACM.

Levin, D. F. (2016). *The Environment Constrains Successful Search Strategies in Natural Distributed Systems*. PhD thesis, University of New Mexico.

Ligot, A. and Birattari, M. (2018). On mimicking the effects of the reality gap with simulation-only experiments. In Dorigo, M., Birattari, M., Blum, C., Christensen, A. L., Reina, A., and Trianni, V., editors, *Swarm Intelligence*, pages 109–122, Cham. Springer International Publishing.

Liu, W. (2008). *Design and modelling of adaptive foraging in swarm robotic systems*. PhD thesis, Faculty of Environment and Technology, University of the West of England, Bristol.

Liu, W. and Winfield, A. F. T. (2010). Modeling and optimization of adaptive foraging in swarm robotic systems. *International Journal of Robotics Research*, 29(14):1743–1760.

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 - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 15(3):289–305.

Lu, Q., Griego, A. D., Fricke, G. M., and Moses, E. M. (2019a). Comparing physical and simulated performance of a deterministic and a bio-inspired stochastic foraging strategy for robot swarms. In *IEEE/RSJ International Conference on Robotics and Automation (ICRA).*

Lu, Q., Hecker, J. P., and Moses, M. E. (2016a). The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016).*

Lu, Q., Hecker, J. P., and Moses, M. E. (2019b). Multiple-place swarm foraging with dynamic depots. *Autonomous Robots*, 42(4):909–926.

Lu, Q., Moses, M., and Hecker, J. (2016b). A scalable and adaptable multiple-place foraging algorithm for ant-inspired robot swarms. `https://arxiv.org/abs/1612.00480`.

Mcgill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, 32(1):12–16.

Moses, M. and Banerjee, S. (2011). Biologically inspired design principles for scalable, robust, adaptive, decentralized search and automated response (radar). In *Artificial Life (ALIFE), 2011 IEEE Symposium*, pages 30–37. IEEE.

Moses, M., Bezerra, G., Edwards, B., Brown, J., and Forrest, S. (2016). Energy and time determine scaling in biological and computer designs. *Phil. Trans. R. Soc. B*, 371(1701):20150446.

Moses, M. E. and Brown, J. H. (2003). Allometry of human fertility and energy use. *Ecology Letters*, 6(4):295–300.

Moses, M. E., Hecker, J. P., and Stolleis, K. (2014). The iant project.

Mouret, J.-B., Koos, S., and Doncieux, S. (2013). Crossing the reality gap: a short introduction to the transferability approach. In *In Proceedings of the workshop Evolution in Physical Systems, ALIFE.*

Moyron-Quiroz, J. E., Rangel-Moreno, J., Kusser, K., Hartson, L., Sprague, F., Goodrich, S., Woodland, D. L., Lund, F. E., and Randall, T. D. (2004). Role of inducible bronchus associated lymphoid tissue (ibalt) in respiratory immunity. *Nature medicine*, 10(9):927–934.

Müller, M. and Wehner, R. (1994). The hidden spiral: systematic search and path integration in desert ants, cataglyphis fortis. *Journal of Comparative Physiology A*, 175(5):525–530.

Nelson, A., Grant, E., and Henderson, T. (2004). Evolution of neural controllers for competitive game playing with teams of mobile robots. *Robotics and Autonomous Systems*, 46(3):135 – 150.

Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.

Nurzaman, S. G., Matsumoto, Y., Nakamura, Y., Koizumi, S., and Ishiguro, H. (2009). Biologically inspired adaptive mobile robot search with and without gradient sensing. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407.

P. Hecker, J., Letendre, K., Stolleis, K., Washington, D., and Moses, M. (2012). Formica ex machina: Ant swarm foraging from physical to virtual and back again. In *Swarm Intelligence*, volume 7461, pages 252–259.

Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A. D., Feron, E., and Egerstedt, M. (2017). The robotarium: A remotely accessible swarm robotics research testbed. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1699–1706.

Pinciroli, C., Talamali, M., Reina, A., Marshall, J., and Trianni, V. (2018). Simulating kilobots within argos: models and experimental validation. In Dorigo, M., Birattari, M., Blum, C., Christensen, A., Reina, A., and Trianni, V., editors, *11th International Conference on Swarm Intelligence (ANTS 2018)*, volume 11172 of *Lecture Notes in Computer Science*, pages 176–187. Springer. © 2018 Springer. This is an author produced version of a paper subsequently published in Swarm Intelligence (LNCS 11172). Uploaded in accordance with the publisher's self-archiving policy.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G. A. D., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6:271–295.

Pini, G., Brutschy, A., Pinciroli, C., Dorigo, M., and Birattari, M. (2013). Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adaptive Behavior*, 21(2):118–136.

Pini, G., Brutschy, A., Scheidler, A., Dorigo, M., and Birattari, M. (2014). Task partitioning in a robot swarm: Object retrieval as a sequence of subtasks with direct object transfer. *Artificial Life*, 20(3):291–317.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Ritchie, M. E. (2009). Scale, heterogeneity, and the structure and diversity of ecological communities. *Princeton University Press, Berlin, Boston*.

Rosenfeld, A., Agmon, N., Maksimov, O., and Kraus, S. (2017). Intelligent agent supporting humanmulti-robot team collaboration. *Artificial Intelligence*, 252:211 – 231.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3293–3298.

Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J., and Nagpal, R. (2013). Collective transport of complex objects by simple robots: Theory and experiments. In *In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS '13, pages 47–54, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Ryan, A. and Hedrick, J. K. (2005). A mode-switching path planner for uav-assisted search and rescue. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1471–1476.

Rybski, P. E., Larson, A., Veeraraghavan, H., Anderson, M., and Gini, M. (2008). Performance evaluation of a multi-robot search & retrieval system: Experiences with mindart. *Journal of Intelligent and Robotic Systems*, 52(3):363–387.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics: SAB 2004 International Workshop*, 3342:10–20.

Şahin, E., Girgin, S., Bayindir, L., and Turgut, A. E. (2008). *Swarm Robotics*, pages 87–100. Springer Berlin Heidelberg, Berlin, Heidelberg.

Savage, V., Gillooly, J., Woodruff, W., West, G., Allen, A., Enquist, B., and Brown, J. (2004). The predominance of quarter-power scaling in biology. *Functional Ecology*, 18(2):257–282.

Savage, V. M., Deeds, E. J., and Fontana, W. (2008). Sizing up allometric scaling theory. *PLOS Computational Biology*, 4(9):1–17.

Schmickl, T. and Crailsheim, K. (2008). Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25(1):171–188.

Schmolke, A. (2009). Benefits of dispersed centralplace foraging: An individualbased model of a polydomous ant colony. *The American Naturalist*, 173(6):772–778.

Schroeder, A., Trease, B., and Arsie, A. (2019). Balancing robot swarm cost and interference effects by varying robot quantity and size. *Swarm Intelligence*, 13(1):1–19.

Sebbane, Y. B. (2012). Lighter than air robots: Guidance and control of autonomous airships. *Springer Netherlands*, 58.

Secor, P. (2016). Nasa swarmathon. `http://www.nasaswarmathon.com/`.

Singh, M. K. and Parhi, D. R. (2011). Path optimisation of a mobile robot using an artificial neural network controller. *International Journal Systems Science*, 42(1):107–120.

Stolleis, K. A., Hecker, J. P., and Moses, M. E. (2015). The ant and the trap: Evolution of ant-inspired obstacle avoidance in a multi-agent robotic system. In *Proceedings of Earth & Space 2016 Engineering for Extreme Environments*.

Stormont, D. (2005). Autonomous rescue robot swarms for first responders. In *Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, pages 151 – 157.

Suarez, A. V., Holway, D. A., and Case, T. J. (2001). Patterns of spread in biological invasions dominated by long-distance jump dispersal: Insights from argentine ants. *Proceedings of the National Academy of Sciences*, 98(3):1095–1100.

Sumpter, D. J. and Beekman, M. (2003). From nonlinearity to optimality: pheromone trail foraging by ants. *Animal behaviour*, 66(2):273–280.

Tindo, M., Kenne, M., and Dejean, A. (2008). Advantages of multiple foundress colonies in belonogaster juncea juncea l.: greater survival and increased productivity. *Ecological Entomology*, 33(2):293–297.

Wall, M. (1996). *GAlib: A C++ library of genetic algorithm components*, volume 87. Mechanical Engineering Department, Massachusetts Institute of Technology.

Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science (New York, N.Y.)*, 343:754–8.

West, G. B., Brown, J. H., and Enquist, B. J. (1997). A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309):122–126.

Winfield, A. F. T. (2009a). *Foraging Robots*, pages 3682–3700. Springer New York, New York, NY.

Winfield, A. F. T. (2009b). *Towards an Engineering Science of Robot Foraging*, pages 185–192. Springer Berlin Heidelberg, Berlin, Heidelberg.

Yun, S.-k. and Rus, D. (2014). Adaptive coordinating construction of truss structures using distributed equal-mass partitioning. *Trans. Rob.*, 30(1):188–202.