12-1-2015

Evolving Efficient Foraging Behavior in Biologically-Inspired Robot Swarms

Joshua Peter Hecker

Joshua Peter Hecker

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

Lydia Tapia

Rafael Fierro

Alan F.T. Winfield

Evolving Efficient Foraging Behavior in Biologically-Inspired Robot Swarms

by

Joshua Peter Hecker

B.S., Computer Engineering, University of Central Florida, 2005
M.S., Computer Engineering, University of Central Florida, 2008

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

December, 2015

Dedication

To my wife, Ashley, for her constant devotion and unwavering support, and to my daughter, Penelope, who brings me joy beyond measure. You are my everything.

"For small creatures such as we the vastness is bearable only through love" – Carl Sagan

Acknowledgments

My advisor, Dr. Melanie Moses, supported me every step of the way, gave me the freedom to pursue my own research questions, and entrusted me with the responsibility of overseeing such a significant project. Thank you for your seemingly endless patience, and for guiding me through this incredible journey to become the independent researcher I am today.

My comrade-in-robotics, Karl Stolleis, toiled alongside me and dedicated countless waking hours to making our robots work correctly. Thanks for putting up with my incessant questions about how things work, and for all the great life anecdotes.

My drinking and traveling companions, Matthew Fricke and Kenneth Letendre, laughed at my jokes, listened to my complaints, and helped to keep me sane. Thanks for partaking in the good food, great beer, and excellent conversation. You are both truly outstanding friends.

My long-time lab mates, Soumya Banerjee, Kimberly Kanigel-Winner, Neal Holtschulte, and Tatiana Flanagan, provided a myriad of insightful conversations on a wide variety of topics, and gave me critical feedback on many manuscript drafts.

All of the students who participated in the iAnt Research Project, particularly Daniel Washington, Bjorn Swenson, Justin Carmichael, Antonio Griego, and J. Jake Nichol, taught me how to be a better mentor, not to mention a better student.

"If I have seen further, it is by standing on the shoulders of giants" – Isaac Newton

Evolving Efficient Foraging Behavior in Biologically-Inspired Robot Swarms

by

Joshua Peter Hecker

B.S., Computer Engineering, University of Central Florida, 2005
M.S., Computer Engineering, University of Central Florida, 2008

Ph.D., Computer Science, University of New Mexico, 2015

Abstract

Human beings are driven to explore distant new worlds as we seek to better understand our place in the Universe. Because of the inherent dangers of human spaceflight, we often send robots as surrogate explorers, controlled from millions of miles away by teams of capable rover drivers here on Earth. As technology continues to advance, scientists and engineers aspire to build low-cost, durable, fully autonomous rovers to succeed today's tele-operated extraplanetary explorers.

Here we aim to advance this goal by designing and programming robots that can successfully navigate unknown and variable environments. We present a swarm robotics system that mimics the foraging behaviors of seed-harvester ants, employing evolutionary computation and machine learning to mitigate the adverse effects of unreliable information, variable environments, congestion bottlenecks, and sparse resources. We describe a central-place foraging algorithm (CPFA) whose parameters are evolved by a genetic algorithm (GA) to maximize foraging performance under different experimental conditions.

We find that foraging for resources in heterogeneous clusters requires more complex communication, memory, and environmental sensing than strategies evolved in previous work. Additionally, we observe sub-linear scaling in resources collected per robot as swarm size increases, which we attribute to the "bottleneck" constraint imposed by central-place foraging. Finally, we augment our foraging robot swarm with machine learning and statistical models, demonstrating that combining our existing biologically-inspired CPFA with a cluster exploitation algorithm produces more efficient total resource collection compared to each algorithm acting alone.

While our system is designed to be a demonstration platform for swarm robotics research, this work provides a foundation for designing and implementing autonomous robot swarms that can function outside of the academic research laboratory. The ability of robot swarms to tolerate sensor noise, adapt to variable environments, distribute work across large teams, and identify and exploit heterogeneously-distributed resources are all critical factors for successful remote exploration missions on distant worlds.

Contents

List of Figures **xii**

List of Tables **xxi**

1 Introduction to Biologically-Inspired Foraging Robot Swarms **1**

1.1 Related Work in Evolving Swarm Behaviors 5

1.2 Simulation Design and History 6

1.3 Contributions and Organization 7

2 *Formica ex Machina*: Ant Swarm Foraging From Physical to Virtual and Back Again **11**

2.1 Abstract 11

2.2 Introduction 12

2.3 Background 13

2.4 Methods 15

2.4.1 Hardware 15

2.4.2 Search Algorithm 15

2.4.3 Experimental Design 16

2.5 Results 18

2.6 Discussion 19

2.7 Acknowledgments: 22

3 Evolving Error Tolerance in Biologically-Inspired iAnt Robots 23

3.1 Abstract 23

3.2 Introduction 24

3.2.1 Previous Work 25

3.2.2 Background 26

3.3 Methods 28

3.3.1 Ant Behavior Model 28

3.3.2 Search Algorithm 31

3.3.3 Physical Sensor Error 32

3.3.4 Experimental Design 33

3.4 Results 36

3.5 Discussion 38

3.6 Acknowledgments 40

4 Beyond Pheromones: Evolving Error-Tolerant, Flexible, and Scalable Ant-Inspired Robot Swarms 42

4.1 Abstract 42

4.2 Introduction 43

4.3 Related Work 47

4.3.1 Automatic Design of Swarm Foraging Behaviors 47

4.3.2 Foraging in Desert Harvester Ants 49

4.3.3 Foundations of the CPFA 50

4.4 Methods 52

4.4.1 Central-Place Foraging Algorithm 52

4.4.2 Genetic Algorithm 57

4.4.3 iAnt Robot Platform 59

4.4.4 Physical Sensor Error Model 60

4.4.5 Experimental Setup 61

4.4.6 Performance Evaluation 62

4.5 Results 65

4.5.1 Error Tolerance 66

4.5.2 Flexibility 69

4.5.3 Scalability 73

4.6 Discussion 77

4.7 Conclusions 80

4.8 Acknowledgements 81

5 Exploiting Clusters for Complete Resource Collection in Biologically-Inspired Robot Swarms 87

5.1 Abstract . . . 87

5.2 Introduction . . . 88

5.3 CPFA Background . . . 89

5.4 Methods . . . 91

5.4.1 Central-Place Foraging and Evolutionary Algorithms . . . 91

5.4.2 Predicting Optimal Clusters . . . 93

5.4.3 Exploiting Clusters . . . 94

5.4.4 Experimental Setup and Evaluation . . . 96

5.5 Results . . . 99

5.6 Discussion . . . 104

5.7 Acknowledgements . . . 106

6 Concluding Remarks 107

6.1 Major Findings . . . 110

6.2 Foraging Robot Swarms Within the Context of Evolutionary Robotics 112

6.3 The Exploration-Exploitation Tradeoff . . . 113

6.4 Extensions to Foraging Robot Swarms . . . 114

6.4.1 Alternative Search Strategies Inspired by the Immune System 114

6.4.2 Alternative Recruitment Strategies Inspired by Foraging Ants 115

6.4.3 Furthering Swarm Robotics Research and Technology Through Student Competition 117

6.5 Future Work 117

Appendices **121**

A Beyond Pheromones: Electronic Supplementary Material **122**

References **125**

List of Figures

1.1 Three major revisions of our iAnt robot platform. (a) Version 1 had a Surveyor SRV-1 chassis, Arduino Uno microcontroller, Ardumoto motor shield, and WiFly communication shield. Sensors included a magnetometer, ultrasonic rangefinder, GPS antenna, and RFID reader/writer. (b) Version 2 added an iPod Touch to provide iAnts with a forward-facing camera to detect a central nest beacon and downward-facing camera to detect QR tags. (c) Version 3 replaced the SRV-1 with a custom-designed laser-cut chassis, lower gear motors to increase torque, and larger capacity batteries to minimize robot down time. 8

2.1 A robot begins its search at a globally shared central nest site (center circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for tags** (blue line) until tags (red squares) are found. After searching, the robot **travels to the nest** (purple line). 16

2.2 32 RFID tags scattered in a ring between 50 cm and 200 cm in (a) the uniform random distribution. The clustered distribution (b) has four piles of eight tags placed at 90° intervals at 50, 100, 150, and 200 cm in relation to the central nest. The power law distribution (c) uses piles of varying size and number: one large pile of eight tags at 125 cm, two medium piles of four tags at 75 and 175 cm, four small piles of two tags at 50, 100, 150, and 200 cm, and eight randomly placed tags. 18

2.3 Time for 1 and 3 robots, real and simulated, to collect tags arranged in (a) random and (b) power law distributions using only site fidelity, and (c) for 3 robots on a power law distribution using pheromones and site fidelity. 19

2.4 Rate of tag discovery per minute of experiment time for 1 and 3 (a) physical and (b) simulated robots in the 3 m area using only site fidelity, as well as (c) 1, 3, 30, and 100 simulated robots collecting tags in a large world with site fidelity and pheromones. 20

3.1 Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform. 28

3.2 (a) State diagram describing the flow of behavior for individual robots during an experiment, (b) an example of a single cycle through this search behavior loop, and (c) an iAnt robot with Velcro for attaching reflective markers (motion capture was used for a previous experiment, but not for any of the observations in this paper). The robot begins its search at a central nest site (double circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for resources** (blue line) until a resource (black squares) is found. After sensing the local resource density, the robot **travels to the nest** (red line). 32

3.3 Parameter values for rates of site fidelity (λ_{sf}), laying pheromone (λ_{lp}), following pheromone (λ_{fp}), and informed random walk decay (λ_{id}) for random, clustered, and power law distributed resources. . . 35

3.4 Results for simulated foraging on a clustered resource distribution with and without error. (a) Best and mean fitness curves. (b) Parameter values for rates of site fidelity (λ_{sf}), laying pheromone (λ_{lp}), following pheromone (λ_{fp}), and informed random walk decay (λ_{id}). . 36

3.5 Results for physical and simulated robots foraging in a world with error using parameters adapted for a world with error, and parameters adapted for an error-free world. 80% more resources are collected using error-adapted parameters in physical robot teams, and 16% more are collected in simulated teams. Robots collected significantly more resources in both cases. Physical and simulated robots using error-adapted parameters are not significantly different. 38

4.1 (a) An iAnt robot. (b) A swarm of iAnt robots foraging for resources around a central illuminated beacon. 44

4.2 We use a GA to **evolve** a foraging strategy (CPFA parameter set) that maximizes resource collection for specified classes of error model, environment, and swarm size. We then **evaluate** the foraging strategy in multiple experiments with simulated and physical robots and record how many resources were collected. We repeat this for different error models, environments, and swarm sizes. We **analyze** flexibility by evolving parameters for one condition and evaluating them in another. 53

4.3 (a) State diagram describing the flow of behavior for individual robots during an experiment. (b) An example of a single cycle through this search behavior. The robot begins its search at a central nest site (double circle) and **sets a search location**. The robot then **travels to the search site** (solid line). Upon reaching the search location, the robot **searches for resources** (dotted line) until a resource (square) is found and collected. After sensing the local resource density, the robot **returns to the nest** (dashed line). 54

4.4 256 resources are placed in one of three distributions: (a) the clustered distribution has four piles of 64 resources. (b) The power law distribution uses piles of varying size and number: one large pile of 64 resources, 4 medium piles of 16 resources, 16 small piles of 4 resources, and 64 randomly placed resources. (c) The random distribution has each resource placed at a uniform random location. . . . 61

4.5 Best and mean fitness, measured as foraging efficiency (resources collected per hour, per swarm) for simulated swarms foraging on (a) clustered, (b) power law, and (c) random resource distributions with and without real-world sensor error. Results are for 100 replicates. . 66

4.6 Foraging efficiency (resources collected per hour, per swarm) using error-adapted and non-error-adapted parameters for (a) 6 robots foraging in a simulation that includes sensor error and (b) 6 physical robots. Asterisks indicate a statistically significant difference ($p < .001$). 67

4.7 For error-adapted and non-error-adapted swarms foraging on clustered resources, (a) the probability of laying pheromone as a function of the count c of resources in the neighborhood of the most recently found resource (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{lp}$), and (b) the pheromone waypoint decay rate (λ_{pd}). Asterisks indicate a statistically significant difference ($p < .001$). 68

4.8 Foraging efficiency (resources collected per hour, per swarm) using parameters adapted to different resource distributions for (a) 6 robots foraging in a simulation that includes sensor error and (b) 6 physical robots. Asterisks indicate a statistically significant difference ($p < .001$). 70

4.9 For error-adapted swarms (top) and non-error-adapted swarms (bottom), (a,c) the probability of returning to a site (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{sf}$) and (b,d) the probability of laying pheromone (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{lp}$) given the number of resources c in the neighborhood of a found resource. 83

4.10 Foraging efficiency (resources collected per hour, per robot) of 1, 3, and 6 robots foraging on a power law distribution for (a) swarms in a simulation that includes sensor error and (b) physical swarms. All results are statistically different ($p < 0.001$). 84

4.11 Foraging efficiency (resources collected per hour, per robot) in simulated swarms of 1 to 768 robots foraging without sensor error. Data are shown on a log scale, and linear regression lines are shown for log-transformed data. Per-robot efficiency is shown for four cases: using the full CPFA parameter set adapted to swarm size (slope = -0.17, $R^2 = 0.96$), using the full CPFA with parameters fixed to values evolved for a swarm size of 6 (slope = -0.19, $R^2 = 0.83$), using parameters adapted to swarm size without information (i.e. the CPFA without memory and communication; slope = -0.14, $R^2 = 0.95$), and using parameters fixed to values evolved for a swarm size of 6 without information (slope = -0.21, $R^2 = 0.91$). All linear fits are statistically significant ($p < 0.001$). 85

4.12 (a) Swarm size versus best evolved uninformed search variation (ω) (slope = -0.035, $R^2 = 0.94, p < 0.001$) (see Fig. A.2 for statistical distribution). (b) Swarm size versus best evolved probability of laying pheromone when two resources are found in the resource neighborhood (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$) (slope = -0.040, $R^2 = 0.84, p < 0.001$) (see Fig. A.2). 86

5.1 Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform. 90

5.2 The central-place foraging algorithm, or CPFA, describes the flow of behavior for individual robots in the swarm during the foraging task (reprinted from our prior work [58]). 92

5.3 The central-place cluster exploitation algorithm, or CPCEA, describes the flow of behavior for robots after switching from foraging with the CPFA to exploiting the clustered regions predicted by OptimalEM (Algorithm 3). 95

5.4 Top: 256 resources are placed one of five different clustered distributions: (a) 1×256 has 1 randomly placed pile of 256 resources, (b) 2×128 has 2 piles of 128 resources, (c) 4×64 has 4 piles of 64 resources, (d) 8×32 has 8 piles of 32 resources, and (e) 16×16 has 16 piles of 16 resources. Bottom: (f)–(j) The residual resource distribution after 224 resources have been collected. 96

5.5 Median cumulative collection time (in timesteps) for 1 to 256 resources in each of five resource distributions. 100

5.6 Mean absolute error ε_c (Eq. 5.3) when predicting the number of clusters in each of five resource distributions, with increasing numbers of resources collected before switching to cluster exploitation. 101

5.7 The effect r (Eq. 5.4) of clustering for robot swarms that switch from the CPFA to the CPCEA after some fraction of resources have been collected, compared to swarms that do not switch and forage only with the CPFA. 102

5.8 Total collection time for robot swarms foraging without information using a correlated random walk, for swarms foraging only with the CPFA, for swarms that switch from the CPFA to the CPCEA after 224 resources have been collected, and for idealized swarms with perfect information (all resource locations are known *a priori*). . . . 103

6.1 The Swarmie hardware platform developed through a UNM partnership with NASA Kennedy Space Center's (KSC) Swamp Works laboratory. The Swarmie platform is an integrated system with adaptive software and a simulator that allows rapid testing of new algorithms in software and rapid porting to hardware for verification in physical experiments. 118

A.1 The best evolved parameters for simulated swarms of 1 to 768 robots foraging without sensor error on a power law distribution. (a) Swarm size versus probability of switching to searching behavior (p_s) (slope = -0.058, $R^2 = 0.30, p = 0.10$), (b) swarm size versus probability of switching to traveling behavior (p_r) (slope = -0.00075, $R^2 = 0.79, p < 0.001$), (c) swarm size versus uninformed search correlation (ω) (slope = -0.035, $R^2 = 0.94, p < 0.001$), (d) swarm size versus rate of pheromone decay (λ_{pd}, Eq. 4.5) (slope = 0.011, $R^2 = 0.73, p = 0.0016$), (e) swarm size versus probability of laying pheromone (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$) (slope = -0.040, $R^2 = 0.84, p < 0.001$), (f) swarm size versus probability of returning to a site (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{sf}$) (NS, $p = 0.27$), and (g) swarm size versus decay rate of informed search (λ_{id}, Eq. 4.3) (NS, $p = 0.38$). 123

A.2 Statistical distributions of parameters evolved for simulated swarms of 1 to 768 robots foraging without sensor error on a power law distribution. (a) Swarm size versus probability of switching to searching behavior (p_s), (b) swarm size versus probability of switching to traveling behavior (p_r), (c) swarm size versus uninformed search correlation (ω), (d) swarm size versus rate of pheromone decay (λ_{pd}, Eq. 4.5), (e) swarm size versus probability of laying pheromone (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$), (f) swarm size versus probability of returning to a site (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{sf}$), and (g) swarm size versus decay rate of informed search (λ_{id}, Eq. 4.3). Gray dots indicate outliers beyond interquartile range; an asterisk represents an outlier at $\omega = 9.3$. . . . 124

List of Tables

3.1 Set of 8 parameters evolved in simulation guided by genetic algorithms. At the start of a simulated run, parameters in each colony are initialized using randomly sampled values from their associated initialization function. The first 3 parameters are initially sampled from a uniform distribution, and the last 5 from exponential distributions within the stated bounds. 31

4.1 Set of 7 CPFA parameters evolved by the GA 54

Chapter 1

Introduction to Biologically-Inspired Foraging Robot Swarms

An innate sense of curiosity drives humans to seek out the unknown, pushing back the boundaries of our understanding as we risk our lives to cross vast expanses of land, sea, air, and most recently, space. The inherent human desire for exploration is perhaps best illustrated in a quote from Michael Collins, American astronaut and Command Module Pilot for Apollo 11: *"It's human nature to stretch, to go, to see, to understand. Exploration is not a choice, really; it's an imperative."*

Human space exploration may be an imperative. But, as Michael Collins and his spacefaring colleagues fundamentally understood, it is also extremely challenging, highly dangerous, and increasingly expensive. To mitigate these difficulties, government space agencies have designed and built robotic rovers that act as surrogate explorers on the Moon and Mars, receiving instructions from human controllers on Earth to conduct a variety of scientific experiments by analyzing air and soil samples

with an array of instruments, including high-resolution cameras. Results of these measurements are transmitted back to Earth for further study. These robots are superb achievements of science and engineering, wholly eliminating the risk to human life while significantly expanding our capacity for extraplanetary observation. However, these technological advancements and risk mitigations have done little to curtail expenses: the total cost of the recent NASA Mars Science Laboratory mission, which includes the Mars rover *Curiosity*, is approximately US$2.5 billion [6].

In addition to the substantial expense required to build, launch, land, and control an extraplanetary rover, a lack of full autonomy in these remote exploration systems means that a large team of scientists and engineers is required to direct a single robot's actions from Earth, approximately 140 million miles away [127]. Because of the long communication time delays, as well as the relatively small Earth-Mars communication window [138], the Mars Exploration Rover *Opportunity* required approximately 11 years to travel 26 miles (the length of a marathon) at an average speed of 0.43 m/h. Furthermore, all of the previous robotic exploration missions have used a monolithic robot design, meaning that all exploration functionality is contained within a single rover. Although each rover is built with a limited amount of computational and telecommunication redundancy, the majority of the onboard components act as single points of failure: if a rover's mobility system or power generator were to fail catastrophically, for example, there would be no "backup" robot to take its place.

Here we consider foraging robot swarms as an alternative to the typical monolithic rover design utilized for past extraplanetary exploration missions [27]. Robot swarms emulate the collective behaviors of social animals such as ants, exhibiting robustness to sensor noise, flexibility through a wide variety of tasks, and scalability for different swarm sizes [17]. Such swarms, typically comprised of large numbers of relatively cheap robots built from low-cost components with short-range sensors, avoid single

points of failure and reduce overhead expenses. Most importantly, robots in the swarm can function autonomously and without a centralized controller, although each robot may also share information with its neighbors through local, peer-to-peer communication [17].

In spite of the advantages of robot swarms over the traditional monolithic rover, there are nonetheless several notable caveats that should be taken into consideration. First, as a result of the economical construction costs, individual robots in these swarms experience increased sensor error and a higher likelihood of hardware failure compared to state-of-the-art monolithic robot systems. To make matters worse, the environments envisioned in the literature for swarm robotics applications (e.g. disaster zones [97] and ocean currents [64], as well as extraplanetary surfaces [121]) are generally variable, unknown, and stochastic in nature. To address this problem, we pose the following question: How should researchers design, build, and program robot swarms to perform large-scale real-world tasks efficiently and robustly?

In this work, we propose one possible solution to this problem: a central-place foraging algorithm (CPFA) inspired by the seed collection behavior of harvester ants. Our CPFA governs the behaviors of individual robots as members of a swarm. We use a genetic algorithm (GA) to optimize the foraging performance of simulated robots by evolving behavior transitions in a tailor-made agent-based model, then we transfer the optimized transition parameters to physical robots. The chapters of this manuscript contribute to solving this problem in five aspects:

Robustness. We mitigate hardware fragility by building a swarm of identical, redundant, and interchangeable physical iAnt robots.

Error-tolerance. We reduce the effects of sensor noise by adapting robot foraging strategies to the real-world error experienced by the physical robots.

Flexibility. We produce generality by evolving and evaluating our robot swarms in

complex and variable environments with heterogeneously distributed resources.

Scalability. We observe an increase in the foraging performance of the swarm, but a decrease in the performance of each individual robot, as the size of the swarm grows.

Thoroughness. We augment our swarm foraging system with a cluster prediction and exploitation algorithm that produces efficient, complete resource collection.

Our approach has several advantages over other state-of-the-art swarm robotics systems. We restrict ourselves to using only low-cost hardware components, providing a low barrier to entry for other researchers, developers, and users. We compensate for imperfect sensing and hardware fragility by using an adaptive, stochastic, biologically-inspired algorithm to control high-level robot behaviors. These simple, parsimonious, modular behaviors provide the GA with a manageable fitness landscape, which ensures evolutionary convergence within a reasonably short time. In this way, we design our robot swarm to take on the entire, integrated task of central-place foraging in a concurrent fashion without requiring a modularized, component-level analysis. The algorithm's flexibility permits our simulated robots to adapt their behaviors in an effort to mitigate sensor error and maximize foraging performance in unknown environments. Our agent-based simulation models the physical environment and hardware constraints of our iAnt robot platform. This parallel physical and simulated approach facilitates the seamless transfer of evolved parameters from simulation into real robots, as well as the ability to tune the simulation to improve its representation of the robots over time. As a result of this iterative process, our simulated robot agents and physical iAnt robots are less sensitive to the ubiquitous *reality gap* that exists in the correspondence between simulated and physical robots.

1.1 Related Work in Evolving Swarm Behaviors

Our GA evolves parameters to control the high-level behaviors we have observed and modeled in ants. These parameters control the sensitivity threshold for triggering behaviors, the likelihood of transitioning from one behavior to another, and the length of time each behavior should last. Several previous projects have taken an approach similar to our own, using learning and optimization techniques to tune a fixed repertoire of higher-level swarm foraging behaviors, rather than lower-level motor controllers or basic directional responses. Matarić [84, 85] used reinforcement learning to train robots to switch between behaviors through positive and negative reinforcement related to foraging success. Similar to Matarić [84], Balch [7] trained robot teams to perform multiple foraging tasks simultaneously using Q-learning. Labella et al [74] implemented adaptive swarm foraging, observing emergent division of labor using only local information and asynchronous communication. Liu and Winfield [79] used a genetic algorithm to tune a macroscopic probabilistic model of adaptive collective foraging, optimizing division of labor and minimizing energy use. Francesca et al [38] used a parameter optimization algorithm to automatically construct probabilistic behavioral controllers for swarm aggregation and foraging tasks. These previous studies have tested swarms on simple foraging tasks that required no communication. Instead, we focus on more difficult foraging tasks in which communication among robots increases collective foraging efficiency. Efficient foraging in environments with more complex resource distributions necessitates more complex foraging strategies. In our study, robots alter the environment by collecting food and by laying pheromones, and those alterations affect future robot behavior. Therefore, these foraging strategies cannot be practically represented by the finite state machines often used in prior work [79, 38].

1.2 Simulation Design and History

We use a two-dimensional agent-based model to simulate our foraging robot swarms. Each agent in the simulation represents an idealized, non-physics-based robot that explores a discrete, gridded environment by moving from cell to cell. We do not explicitly model actuators or sensors; instead, movement and resource sensing are locally restricted to the eight-cell Moore neighborhood surrounding each robot to mimic the capabilities of the real iAnts. Each robot plans a path between its current position (x, y) and its target $(\hat{x}, \hat{y})$ (e.g. from the nest to a pheromone waypoint) by selecting the Moore neighborhood cell that minimizes the distance between (x, y) and $(\hat{x}, \hat{y})$. All operations are performed synchronously via a global simulation clock, a simple solution that is feasible because we do not model robot collisions. This parsimonious simulation facilitates the rapid testing of many different foraging strategies, which is required to evolve appropriate solutions for varied and complex environments.

In Chapter 2, we used an established ant-foraging model written in C++ [77, 109] as a proof-of-concept simulation of our ant-inspired robot swarms. In Chapter 3, this predecessor simulator was entirely rewritten in Objective-C, the same programming language used in the iPods onboard our iAnts. To improve the correspondence between simulated and physical robots, we implemented a rotation delay (one second per one-quarter of a radian), and adjusted the size of the grid (125×125 cells) and the runtime of the simulation (7200 ticks) to match real iAnt velocity (16 cm/s) and the resource detection window (8×8 cm) during a 1 hour experiment in a 100m^2 arena. This tuning ensured that the simulated robots would explore their environment at approximately the same speed as the real robots. We also measured positional error and resource detection error in our real robots, then incorporated models of these errors into our simulation. Positional error is modeled as a symmetric Gaussian, and is applied when a simulated robot finds a resource, as well as when a robot leaves

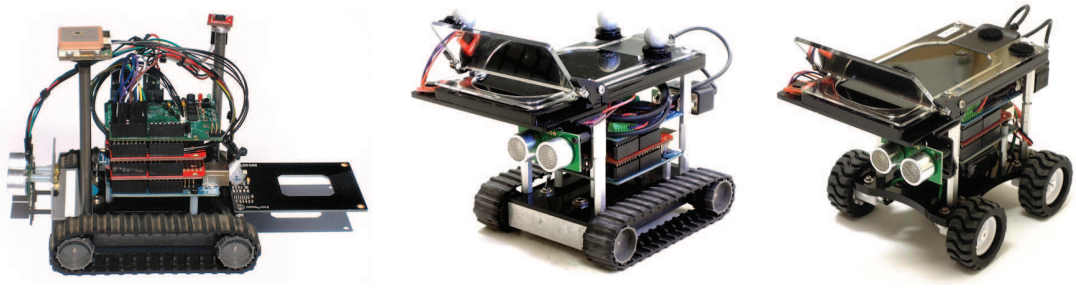
the nest using site fidelity or following a pheromone waypoint. Detection error is modeled as a fixed probability, and is applied every time a simulated robot attempts to detect a resource. Chapter 4 used a functionally similar version of the simulation from Chapter 3, though many supplementary features and code refactorizations were added. In particular, we incorporated additional tuning of the error models and movement behavior based on comprehensive observations of the physical robots. Finally, in Chapter 5, we added new methods to apply the expectation-maximization algorithm onto sets of resource locations, then select the optimal number of clusters for a given set based on Bayesian information criterion. The functionality of the simulation in this chapter was otherwise identical to the previous chapters.

In general, this repeated, back-and-forth tuning process between the simulated and real robots was especially valuable in producing a simulation that better represented the actions of our iAnt swarms. Because our simulated and real swarms follow stochastic actions while they forage, it is difficult to conclude precisely how accurate our simulation is at a given moment (i.e. at a specific code revision). Nevertheless, this work demonstrates several points of correspondence between the simulated and real robots, particularly when foraging strategies are evolved in simulation and evaluated in iAnt swarms, which support the efficacy of our iterative procedure for generating efficient swarm foraging behavior.

1.3 Contributions and Organization

Our main contributions are divided into four chapters and summarized below. All chapters are published (or in press) in peer-reviewed conference proceedings or journals.

Chapter 2. This chapter presents the predecessor to our ant-inspired foraging al-



(a) (b) (c)

Figure 1.1: Three major revisions of our iAnt robot platform. (a) Version 1 had a Surveyor SRV-1 chassis, Arduino Uno microcontroller, Ardumoto motor shield, and WiFly communication shield. Sensors included a magnetometer, ultrasonic rangefinder, GPS antenna, and RFID reader/writer. (b) Version 2 added an iPod Touch to provide iAnts with a forward-facing camera to detect a central nest beacon and downward-facing camera to detect QR tags. (c) Version 3 replaced the SRV-1 with a custom-designed laser-cut chassis, lower gear motors to increase torque, and larger capacity batteries to minimize robot down time.

gorithm and simulated multi-agent swarm simulator[1], as well as an alpha version of our physical iAnt robot platform (Figure 1.1(a)). This section of the manuscript chiefly presents a proof of concept for our swarm robotics system, meaning that the results are improved upon and further analyzed in later chapters. We test the ability of individual robots and teams of three robots to collect tags distributed in random and clustered distributions in simulated and real environments. This work demonstrates the feasibility of programming our robot teams for collective tasks such as retrieval of dispersed resources, mapping, and environmental monitoring. This chapter is published in *Lecture Notes in Computer Science* as a conference proceedings of the 8th International Conference on Swarm Intelligence (ANTS 2012), Copyright ©2012, Springer-Verlag Berlin Heidelberg [55].

Chapter 3. This chapter presents a codified version of the central-place foraging

[1] A proof-of-concept simulation based on Letendre and Moses' agent-based ant foraging model [77]

algorithm (CPFA[2]) that is easier to interpret, as well as the beta version of our iAnt robot platform (Figure 1.1(b)), and an updated multi-agent simulation[3] that more accurately reflects physical reality. We use a genetic algorithm (GA) to optimize behavior in a team of simulated robots, then transfer the evolved behaviors into physical iAnt robots. We introduce positional and resource detection error models into our simulation to characterize the empirically-measured sensor error in our physical robots. This work extends state-of-the-art biologically-inspired robotics, evolving high-level behaviors that are robust to sensor error and meaningful for phenotypic analysis. This chapter is published in *Advances in Artificial Life* as a conference proceedings of the 12th European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013) [59].

Chapter 4. This chapter presents the final, definitive version of the CPFA[4] and the simulated[5] and physical iAnt robot swarms (Figure 1.1(c)). We describe a swarm robotics system that emulates ant behaviors which govern memory, communication, and movement, as well as an evolutionary process that tailors those behaviors into foraging strategies that maximize performance under varied and complex conditions[6]. The system evolves appropriate solutions to different environmental challenges. Analysis of the evolved behaviors reveals the importance of interactions among behaviors, and of the interdependencies between behaviors and environments. The effectiveness of interacting behaviors depends on the uncertainty of sensed information, the resource distribution, and the swarm size. Such interactions could not be manually specified, but are effectively evolved in simulation and transferred to physical robots. This work is the first to demonstrate high-level robot swarm behaviors

[2]See Table 3.1 for an itemized description of the genome used in this chapter

[3]The source code for the simulation used to produce results for this chapter is freely available online at https://github.com/BCLab-UNM/iAnt-Sim/tree/722016d73e4968d07a305c2fcce8342e9dff08a9

[4]See Table 4.1 for an itemized description of the genome used in this chapter

[5]The source code for the simulation used to produce results for this chapter is freely available online at https://github.com/BCLab-UNM/iAnt-Sim/tree/859d76b60c902ca1dcb8d3d18894440c5b99d226

[6]See Figure 4.2 for a visual description (flowchart) of the methodology

that can be automatically tuned to produce efficient collective foraging strategies in varied and complex environments. This chapter is published in the journal *Swarm Intelligence*, Copyright ©2015, Springer Science+Business Media New York [58].

Chapter 5. This chapter presents a novel extension to the CPFA that mitigates the diminishing returns encountered when simulated[7] and physical iAnt robot swarms (Figure 1.1(c)) are tasked with the complete collection of all resources from a predefined search area. We describe our existing ant-inspired robot swarm foraging system that searches for and collects resources from a variety of distributions, and a new cluster prediction and exploitation algorithm that augments swarm foraging by directing robots to residual resources. By characterizing the cumulative resource collection time for a robot swarm foraging in a variety of clustered resource distributions, we can identify the relationship between the "clusteredness" of the distribution and the change in the resource collection rate over time. This work demonstrates the feasibility of efficient, complete resource collection using simple, range-limited robot swarms programmed with ant-inspired foraging behaviors. This chapter is in press as a conference proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) [54].

[7]The source code for the simulation used to produce results for this chapter is freely available online at https://github.com/BCLab-UNM/iAnt-Sim/tree/89fbd9e7e6527dd0af2a1801367a34df50ec4c1f

Chapter 2

Formica ex Machina: Ant Swarm Foraging From Physical to Virtual and Back Again

2.1 Abstract

Ants use individual memory and pheromone communication to forage efficiently. We implement these strategies as distributed search algorithms in robotic swarms. Swarms of simple robots are robust, scalable and capable of exploring for resources in unmapped environments. We test the ability of individual robots and teams of three robots to collect tags distributed in random and clustered distributions in simulated and real environments. Teams of three real robots that forage based on individual memory without communication collect RFID tags approximately twice as fast as a single robot using the same strategy. Our simulation system mimics the foraging behaviors of the robots and replicates our results. Simulated swarms of 30 and 100 robots collect tags 8 and 22 times faster than teams of three robots. This work

demonstrates the feasibility of programming large robot teams for collective tasks such as retrieval of dispersed resources, mapping, and environmental monitoring. It also lays a foundation for evolving collective search algorithms *in silico* and then implementing those algorithms *in machina* in robust and scalable robotic swarms.

2.2 Introduction

One goal of swarm robotics is to engineer groups of simple, low-cost robots that can cooperate as a cohesive unit to accomplish collection and exploration tasks such as mapping, monitoring, search and rescue, and foraging for resources in unmapped environments [15, 21, 32]. Ideally, robotic swarms are capable of exploring unknown environments without the benefit of prior knowledge to guide them. Individuals must adapt to sensor error and motor drift, and the swarm must function given variation, errors, and failures in individual robots.

Biology often provides inspiration for approaches to achieve these design goals [15, 29, 32, 90]. Biologically-inspired decentralized approaches have enhanced scalability and robustness by removing single points of failure from communication bottlenecks and rigid control structures. Such approaches have not yet reached the level of emergent coordination observed in natural systems [119].

Our robots are designed to mimic colonies of seed harvester ants who forage using a combination of individual memory and pheromone trails. Robots are equipped with a sensor suite which mimics the real ants: time-based odometry approximates physical location analogous to the ants' stride integration [141], and ultrasound ranging measures distance to objects and corrects for drift similar to an ant's landmark-based navigation [61]. Pheromone-like communication of previously successful search locations is used to improve search performance. Robots search for radio-frequency identification (RFID) tags, and upon finding them, return to a central 'nest.' Robot loca-

tions are transmitted over one-way wireless communication to a server for data logging; occasional two-way communication allows virtual pheromones to direct robots to previously found tag locations.

We program our robots with search algorithms derived from our previous work that used an agent-based model (ABM) guided by genetic algorithms (GA) to replicate foraging behaviors of seed harvester ants [109, 77]. We duplicate parameters from the ant model in the robots. We modified the ABM to replicate the constraints of the robot hardware, and to model the behavior and environment of the robots in their search for RFID tags. This parallel physical and simulated implementation allows us to compare results from analogous experiments *in machina* as implemented in physical robots and *in silico* in the ABM (as in [31, 86]). In additional ABM experiments we scale up the size of the swarm, the number of tags, and the size of the area in which the simulated robots search.

2.3 Background

Swarm robotics: Like ant colonies and other complex biological systems, robotic swarms have potential to utilize efficient, robust, distributed approaches to physical tasks. Effective algorithms for swarm robotics must extend beyond simulation to intelligently deal with the complexities of navigating in real environments [31, 86, 89]. Our approach balances the benefit of centralized information exchange with the scalability of decentralized autonomous search [9, 91, 105]. We use evolutionary algorithms to determine the parameters of individual behavior that result in effective collective action, as in [33, 109, 100, 126].

Biological ants: Our algorithms are largely inspired by foraging in *Pogonomyrmex* desert seed-harvester ants [35]. These foragers typically leave their colony's single nest, travel in a relatively straight line to some location on their territory, and

then switch to a correlated random walk to search for seeds.

When a forager finds a seed, it brings it directly back to the nest. Foragers often return to the location where they previously found a seed in a process called site fidelity [12, 35, 93], which reduces future search times. It is unclear exactly how often these ants lay and follow pheromone trails [47, 61, 95], but our recent work indicates occasional laying of pheromone trails to dense piles of food may be an effective component of these ants' foraging strategies [109, 77].

Models: We used GAs to find the optimal balance of site fidelity and pheromone communication in simulated ant colonies [77]. We simulated ant foraging using a set of ABMs of foragers on a grid, with parameters that specify how ants travel from the nest, search, and use site fidelity and pheromone communication. These parameters are optimized by a GA to maximize seed intake rate. Previous simulations show that ants increase foraging rates with rare pheromone use ($< 10\%$ of foraging trips), particularly in the clustered distribution where the intake rate doubles with the addition of pheromone [77].

The ant foraging ABM was modified to model our swarm robots and our experimental setup. The simulation provides both a theoretical benchmark and a basic architecture for using GAs to optimize simulated robots within the constraints imposed by the physical hardware. All *in machina* experiments have been duplicated *in silico*.

2.4 Methods

2.4.1 Hardware

Our robots use an Arduino microcontroller with a compass, ultrasound, wireless card, and RFID reader. These allow the robots to localize at a central 'nest,' measure distance (object 100 cm away: mean error = 2.7 cm, σ = 2.24), and calculate odometry (round trip of 10 m: mean error = 21 cm, $\sigma = 6.6$). Robots avoid collisions by rotating clockwise until the object has been cleared.

2.4.2 Search Algorithm

The search behavior used by the robots to locate RFID tags is shown in Fig. 2.1.

1. Set Search Location: Robots begin at the nest in the center and randomly select an initial search site location, encoded as a distance d and heading h.

2. Travel to Search Site (yellow path) Traveling robots go straight to the search location while avoiding collisions with other robots, correcting for motor drift, and communicating events to the server for later analysis.

3. Search for Tag (blue path): The robot moves in a correlated random walk with direction θ at time t drawn from a normal distribution centered around direction θ_{t-1} and standard deviation $SD = \omega + \gamma/t_s^{\delta}$. ω determines the degree of turning during an uninformed search. In a search informed by memory or communication, γ/t_s^{δ} determines an initial additional degree of turning which decreases over time spent searching. This mimics ants' tight turns in an initially small area that expand to explore a larger area over time [109].

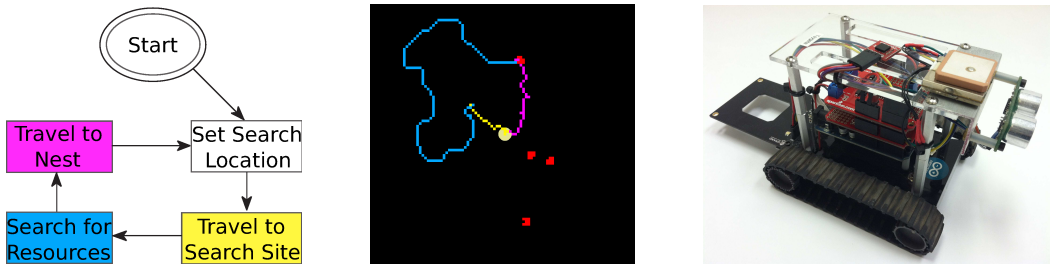


Figure 2.1: A robot begins its search at a globally shared central nest site (center circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for tags** (blue line) until tags (red squares) are found. After searching, the robot **travels to the nest** (purple line).

4. Travel to Nest (pink path): The robot returns to the known nest location. In pheromone experiments, the tag location (d, h) is reported to the server if C ¿ 1, where C is the count of other tags detected in the 8-cell neighborhood of the collected tag in the simulation or discovered in one 360° rotation of the real robot.

5. Set Next Search Location: On subsequent trips, d and h are determined by either returning to the previously found tag location if $C > 0$, otherwise d and h are communicated from the pheromone list on the server.

2.4.3 Experimental Design

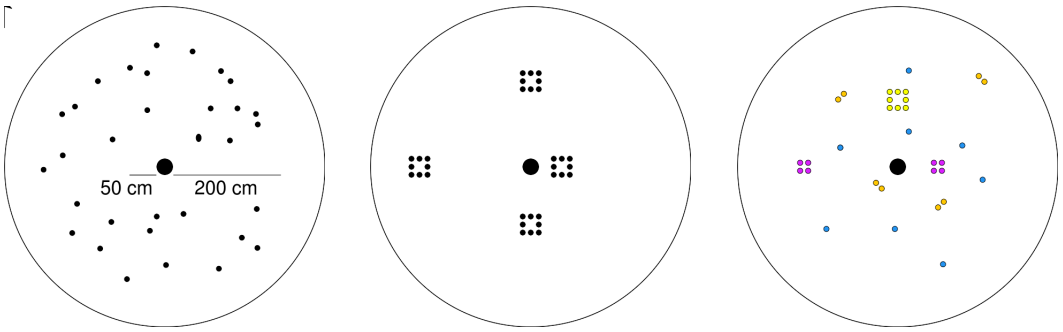
Each experimental trial on a concrete surface runs for a maximum of one hour. A cylinder marks the center 'nest' to which the robots return once they have located a tag. This center point is used for localization and error correction by the robots' ultrasonic sensors. All robots involved in a trial are initially placed near the cylinder. We program each robot to stay within a 3 m radius 'virtual fence'. In every experiment, 32 RFID tags are arranged in one of three different patterns: random,

clustered, or power law (Fig. 2.2). Experiments are replicated under identical conditions for individual robots and for groups of three bots.

Robot locations are continually transmitted over one-way WiFi communication to a central server and logged for analysis. When a tag is found, its unique identification number is transmitted back to the server, providing us with a detailed record of tag discovery. Tags can only be read once, simulating seed retrieval. The central server also acts as a coordinator for virtual pheromone trails using two-way communication. Locations deemed important enough to require a pheromone value (i.e. those with two or more tags discovered by the robot) are added to a list data structure with a pheromone value of 1. Each location's associated pheromone value p_i is decayed exponentially over time by the server: $p_{t+1} = p_t * .995^{\eta}$, where η is the number of seconds between time t and $t + 1$. When a location's pheromone value has dropped below a threshold of 0.001, it is removed from the list. As each robot returns to the nest, the server selects a location from the list (if available) and transmits it to the robot.

Our simulations replicate the physical dimensions of the robots, their speed while traveling and searching, and the area over which they can detect an RFID tag, with spatial dimensions that reflect the distribution of tags in the 3 m area. Like the real robots, simulated robots avoid collisions by turning to the right to move past other robots, and search for a simulated hour.

We also simulated the behavior of the robots in a much larger area in which tags are distributed in the same density but in such large numbers that even large swarms of robots collect only a fraction of the available tags. We simulated 1, 3, 30, and 100 robots to observe the scaling properties of the system.



(a) Random (b) Clustered (c) Power law

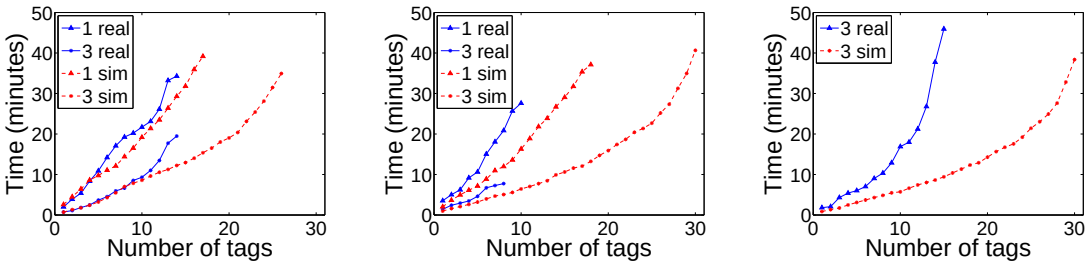
Figure 2.2: 32 RFID tags scattered in a ring between 50 cm and 200 cm in (a) the uniform random distribution. The clustered distribution (b) has four piles of eight tags placed at 90° intervals at 50, 100, 150, and 200 cm in relation to the central nest. The power law distribution (c) uses piles of varying size and number: one large pile of eight tags at 125 cm, two medium piles of four tags at 75 and 175 cm, four small piles of two tags at 50, 100, 150, and 200 cm, and eight randomly placed tags.

2.5 Results

We analyze the rates at which robots retrieve tags from each distribution, individually or in teams of three, in real robots and in simulation. Unless otherwise noted, results for each experimental treatment are averaged over five robot experiments and twenty simulations. Error bars indicate one standard deviation.

Time to collect 32 tags is shown in Fig. 2.3. In robots and in simulation, three robots collect tags faster than one robot, however, the speedup varies over the course of the experiments (i.e., the red and blue lines are not parallel). When we average time to collect n tags, where n varies between 1 and the maximum number of tags collected, we find that 3 robots collect tags approximately twice as fast as 1 robot.

Figure 2.4 shows the the rate of tag collection per minute of experiment time for physical and simulated robots. Each bar denotes the collection rate over a particular tag distribution. We were not able to distinguish a significant effect of tag distri-



(a) Random (b) Power law (c) Power law w/ pheromones

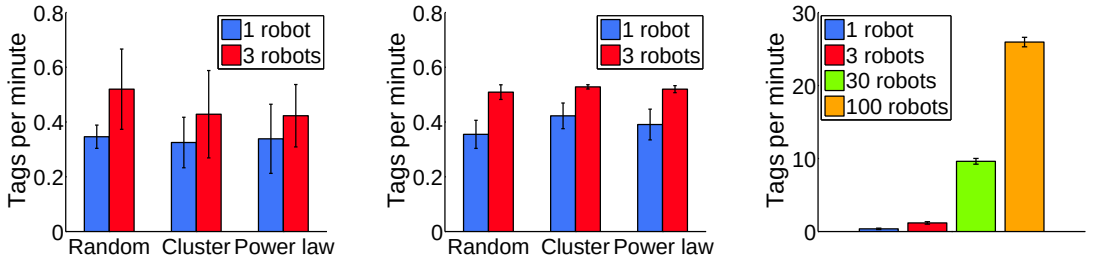
Figure 2.3: Time for 1 and 3 robots, real and simulated, to collect tags arranged in (a) random and (b) power law distributions using only site fidelity, and (c) for 3 robots on a power law distribution using pheromones and site fidelity.

bution on tag collection rate by the robots (General Linear Model [GLM]: $p > 0.1$; $n = 18$); but we did find a significant effect of distribution on tag collection rate using the larger sample size in simulation (GLM: $p < 0.001$; $n = 120$). In the simulations, the fastest tag collection was in the clustered distribution, followed by power law and then random distributions.

2.6 Discussion

We used ABMs and GAs to translate foraging behaviors of seed harvesting ants into algorithms for teams of RFID tag–seeking robots. We tested two algorithms: one in which robots rely on individual memory of locations of previously found tags (mimicking site fidelity), and one in which robots share tag locations as waypoints (mimicking pheromones) via a server that acts as the robots' nest.

Three robots find tags approximately twice as fast as 1 robot when using site fidelity. Site fidelity is an effective foraging strategy in ants and robots. It is extremely simple and easily encoded into very simple devices, including devices much simpler than the robots we used here. The approach is also highly parallelizable because it



(a) Physical (b) Simulated (c) Simulated, large world

Figure 2.4: Rate of tag discovery per minute of experiment time for 1 and 3 (a) physical and (b) simulated robots in the 3 m area using only site fidelity, as well as (c) 1, 3, 30, and 100 simulated robots collecting tags in a large world with site fidelity and pheromones.

requires no communication among robots or the server.

Our approach, similar to [31], lays a foundation to explore the interplay between simulation and experiments with real robots. Simulated and real experiments with 1 and 3 robots using site fidelity show similar foraging rates (Fig. 2.3(a),(b) and Fig. 2.4(a),(b)), although simulated robots are slightly faster. This results from real robots having more difficulty with avoiding each other, physical hardware limitations, imperfect localization, and the possibility that real robots confuse each other with the nest.

Simulated foraging is highly scalable whether using site fidelity alone, or site fidelity augmented with pheromones when multiple tags are found in the same location. When we scale up to 100 robots in unbounded environments with many tags, teams of 100 robots collect resources 66 times faster than a single robot (Fig. 2.4(c)). This 34% decline in per-robot efficiency results from increased travel distance–an unavoidable consequence of central place foraging [72].

We implemented pheromone communication in real robots by having robots report found tag locations to a central server. Mimicking a strategy that was effective

in our ant simulations, robots communicated a location as a waypoint to the server if the robot saw at least 2 additional tags in the vicinity. The server implements a simple pheromone algorithm and reports those locations to other robots. When we add this pheromone-like behavior to our robots, we observe robots clearing large clusters of tags faster; however, pheromones decreased the average tag collection rate in real robots relative to tag collection using only site fidelity. We attribute the lack of success primarily to error propagation: pheromones decrease performance when robots get lost and communicate incorrect locations to other robots, similar to [5].

Our results suggest that the approach of combining individual memory with communication at a central nest can transform simple robots into effective swarms that are scalable and robust to the loss or malfunction of a few individuals. Results of our 3 robot experiments include several instances in which one robot became lost or malfunctioned, but the other two robots continued their task. Such systems could be used for search and rescue, searching for resources or obstacles, and even biomedical applications using nano-robots.

Our next steps are to use a GA to optimize parameters that maximize efficiency and/or robustness in the robot ABM, and then import those parameters into the robots. For example, currently the robots report a pheromone to the server if there are 2 or more additional tags in the local neighborhood of the last tag found. We will use the GA to optimize the decision to lay pheromone and follow pheromone trails vs. returning to the last site food was found, optimizing the balance between shared and private information. Preliminary analysis suggests that the GA can evolve a pheromone-laying rule that significantly improves foraging over our current implementation. We will also extend analysis to different distributions, and will increase scalability by mimicking features of large ant colonies such as the use of mobile nests and of multiple nests.

2.7 Acknowledgments:

This work was funded by NSF EF #1038682 and DARPA CRASH #P-1070-113237.

Chapter 3

Evolving Error Tolerance in Biologically-Inspired iAnt Robots

3.1 Abstract

Evolutionary algorithms can adapt the behavior of individuals to maximize the fitness of cooperative multi-agent teams. We use a genetic algorithm (GA) to optimize behavior in a team of simulated robots that mimic foraging ants, then transfer the evolved behaviors into physical iAnt robots. We introduce positional and resource detection error models into our simulation to characterize the empirically-measured sensor error in our physical robots. Physical and simulated robots that live in a world with error and use parameters adapted specifically for an error-prone world perform better than robots in the same error-prone world using parameters adapted for an error-free world. Additionally, teams of robots in error-adapted simulations collect resources at the same rate as the physical robots. Our approach extends state-of-the-art biologically-inspired robotics, evolving high-level behaviors that are robust to sensor error and meaningful for phenotypic analysis. This work demonstrates the

utility of employing evolutionary methods to optimize the performance of distributed robot teams in unknown environments.

3.2 Introduction

Multi-agent simulations have been used to evolve behaviors which are then transferred into physical robots [99, 120]. Simulations rapidly generate multiple viable solutions, allowing researchers to test many possible scenarios and make informed decisions about which physical experiments to run. Such simulations should focus on physical fidelity by replicating the environment, hardware constraints, and sensor error of the real robots [19].

A particularly challenging class of problems for multi-robot systems is central-place foraging [83, 104]. For this task, robots are programmed to search an area for resources and aggregate these resources at a central location. Foraging is considered a canonical task for distributed robotics: foraging can be instantiated into a number of real-world applications such as hazardous waste clean-up [106], land mine detection and removal [42, 71], search and rescue [70], and extraplanetary exploration [27, 129]. For applications where the physical environment may vary over time and the distribution of resources is most likely unknown, evolutionary approaches allow robot teams to adapt their behavior to each particular scenario.

Our robots use a central-place foraging algorithm (CPFA) based on the foraging behavior of ants [55, 56]. The CPFA is parameterized by a GA in a multi-agent simulation which emulates the physical robot experiments. Our simulation evolves parameters in a parsimonious model of biological ant behavior, and our iAnt robots use these parameters to forage for resources in an experimental area. We investigate the effects of sensor error on physical and simulated robot performance. We demonstrate the utility of this approach by measuring the number of resources that robots

collect using parameters adapted and not adapted to error.

3.2.1 Previous Work

We conducted manipulative field studies on three species of *Pogonomyrmex* desert seed-harvester ants [35]. Colonies were baited with dyed seeds distributed in a variety of pile sizes around each ant nest. We calculated foraging rates for each distribution and found that ants collected seeds faster when seeds were more clustered. Computer simulations used genetic algorithms to find individual ant behavioral parameters that maximized the seed collection rate of the colony. Simulated ants foraging with those parameters mimicked the increase of seed collection rate with the amount of clustering in the seed distribution when ant agents were able to remember and communicate seed locations [109].

We also observed how individual parameters and overall fitness change with different distributions of resources and different numbers of simulated agents performing a central-place foraging task [77]. Parameters evolved for specific types of resource distributions were swapped and then fitness was measured for the new distribution; for example, parameters optimized for a clustered distribution were tested on random distributions of resources. Simulated agents incurred as much as a 50% decrease in fitness when using parameters on a distribution different from the one for which they were optimized.

We then modified our multi-agent central-place foraging simulation to model the physical environment and hardware constraints of our iAnt robot platform [55]. We adapted our existing GA to evolve parameters for our iAnt robots. The evolved parameters were then transferred into the physical robots. Simulated teams collected three to four times as many resources as the real robot teams. We hypothesized that this discrepancy resulted from a *reality gap* between the error-free simulated world

and the sensor error experienced by the physical robots.

Most recently, we incorporated a probabilistic error model into our multi-agent iAnt simulator in a workshop paper [56]. In this preliminary study, we added varying amounts of noise to agents' physical positions and their ability to detect resources, and analyzed the response of the genetic algorithm by observing individual behavioral parameters. We saw that increased positional error reduced resource collection, and induced the GA to select for a lower likelihood of returning to locations where resources were previously found. Increased detection error also reduced resource collection, as well as influencing the GA to select behaviors that searched local areas more thoroughly when only a few resources were detected. These behaviors indicated that the GA was able to evolve parameters appropriate to the sensor error used in the simulation.

We build on this prior work by *a*) simplifying the CPFA which improves performance and makes it easier to interpret why parameters are evolved to different values in different experiments; *b*) updating the iAnt simulator to more accurately reflect physical reality; *c*) testing the CPFA on new resource distributions; and *d*) implementing error-adapted parameters in experiments in physical robots.

3.2.2 Background

Research in evolutionary robotics (ER) primarily focuses on using evolutionary methods to develop controllers for autonomous robots. Controllers can be evolved in simulation and subsequently transferred into physical robots [99, 120], or evolved directly in real robots through embodied evolution [133]. Following principles outlined by [18], work in ER has focused on bridging the reality gap between simulated and real robots to improve the performance of evolved controllers in the physical world [67]. Neural networks have been used in combination with evolutionary methods to evolve

controllers for simulated robot agents with random sensor noise; controllers were subsequently transferred to real robots with varying degrees of success [101, 88, 66].

State-of-the-art robotic simulators such as Stage [131] and ARGoS [112] can be used to model large robot teams with realistic, complex physical kinematics, but they do not incorporate any learning or evolutionary methods that allow simulated agents to adapt to unknown environments. Neither simulator includes sensor noise in its standard implementation, however [111] recently modified ARGoS to incorporate an actuator noise model, generating performance matching results from positional error observed in real robots.

Previous work on multi-robot group foraging tasks used reinforcement learning to train robots on higher-level behaviors, rather than lower-level motor controllers or basic directional responses [84, 85]. Robots learned when to switch between behaviors in a fixed repertoire set through positive and negative reinforcement related to their foraging success. We follow this high-level learning approach in the design of our CPFA.

Our approach (see Figure 3.1) differs from previous approaches in that we do not attempt to evolve basic primitive behaviors from the ground up. Instead, we model existing biological ant behaviors that have evolved naturally over millions of years. We use a genetic algorithm to parameterize these behaviors in our simulated agents, then we transfer those behaviors into physical robots. Evolved parameters control the sensitivity threshold for triggering behaviors, the likelihood of transitioning from one behavior to another, and the length of time each behavior should last.

We extend the state of the art in evolutionary and biologically-inspired robotics by *i*) evolving high-level behaviors that are *ii*) robust to real-world sensor error and *iii*) meaningful for phenotype-level analysis.

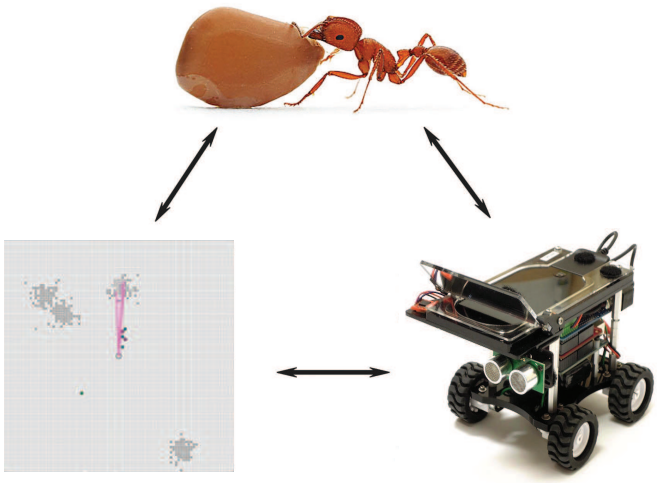


Figure 3.1: Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform.

3.3 Methods

We present our simulated model of ant behavior, detailed pseudocode and diagrams explaining our simplified CPFA, probabilistic models of physical sensor error in the iAnt robot platform and implemented in our multi-agent system, and the design of our simulated and physical experiments.

3.3.1 Ant Behavior Model

Pogonomyrmex seed-harvester ants follow a central-place foraging strategy to aggregate food at their colony's single nest. These foragers typically leave their nest, travel in a relatively straight line to some location on their territory, and then switch to a correlated random walk to search for seeds. A foraging ant who has located a seed brings it directly back to the nest. Foragers often return to a location where they have previously found a seed in a process called site fidelity [93, 12, 35]. Our recent work indicates that combining site fidelity with occasional laying of pheromone trails

to dense piles of food may be an effective component of these ants' foraging strategies [109, 77].

We incorporate key behaviors observed in our previous field studies on desert seed-harvester ants [35] into our multi-agent simulation and physical iAnt robots. We model probabilistic actions and state transitions using eight evolvable parameters, detailed in Table 3.1. These are simplifications of our earlier CPFA algorithm [55]. Modifications have been made since our most recent work in an effort to increase parsimony [56], such as removing the parameter for probabilistically abandoning a pheromone waypoint:

- **State transitions:** Robots switch between two behaviors:
  - **Traveling:** In the absence of information, a robot at the nest will select a random direction and begin traveling. At each step of traveling, robots have a probability p_s of transitioning to search behavior.
  - **Searching:** At each step of searching, robots who have not found a resource have a probability p_t of returning to the nest.
- **Correlated random walk:** Robots explore regions using a random walk with a fixed step size and a direction $\theta_t \sim \mathcal{N}(\theta_{t-1}, \sigma)$ at time t. The standard deviation σ determines how correlated the direction of the next step is with the direction of the previous step. σ depends on whether an agent has prior information through the use of site fidelity or pheromones:
  - **Uninformed search:** If an agent has not used site fidelity or pheromones, then $\sigma = \omega$.
  - **Informed search:** If an agent has arrived at a site by using site fidelity or pheromones, then $\sigma = \omega + (4\pi - \omega) * e^{-\lambda_{id} * t}$, where σ decays to ω as time t increases.

- **Information:** Previous ant studies have demonstrated the ability of ants to count event frequencies in estimating nest size [81], travel distance [141], and encounter rates with other ants [114]. In our simulation, when an agent finds a resource, it stores a count c of additional resources in the 8-cell neighborhood of the found resource. This count c represents an estimate of the density of resources in the local region, and the agent uses c to decide when to use site fidelity, lay a pheromone waypoint, or follow a pheromone waypoint:
  - **Site fidelity:** A robot returns to a previously found resource location if $F_{sf}(c) > \mathcal{U}(0,1)$, where $F_{sf}(x) = 1 - e^{-\lambda_{sf}*(x+1)}$.
  - **Laying pheromone:** A robot creates a pheromone waypoint for a previously found resource location if $F_{lp}(c) > \mathcal{U}(0,1)$, where $F_{lp}(x) = 1 - e^{-\lambda_{lp}*(x+1)}$. New pheromone trails are initialized with a value of 1.
  - **Following pheromone:** Upon returning to the nest, a robot follows a pheromone waypoint to a previously found resource location if $F_{fp}(c) > \mathcal{U}(0,1)$, where $F_{fp}(x) = 1 - e^{-\lambda_{fp}*(9-x)}$. Waypoints are selected with probability proportional to their pheromone value.
  - **Pheromone decay:** Pheromone waypoints decay exponentially over time t as $e^{-\lambda_{pd}*t}$. Waypoints are removed from the simulation once their value drops below a threshold of 0.001.

Four parameters that are of interest in our analysis are the informed search decay rate (λ_{id}), the rate of using site fidelity (λ_{sf}), the rate of laying pheromone (λ_{lp}), and the rate of following pheromone (λ_{fp}). Lower values of informed search decay (λ_{id}) cause the robots to use a less correlated random walk, and thus a more random and thorough local search, for a longer period of time when they have information pertaining to a high density of resources at a particular location.

In both simulated and physical robots, we simulate pheromone trail use by main-

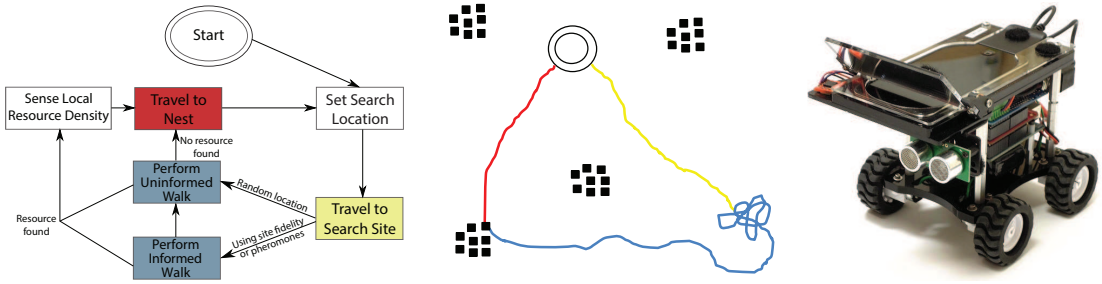
Parameter	Description	Initialization Function
p_t	Probability of traveling	$\mathcal{U}(0,1)$
p_s	Probability of searching	$\mathcal{U}(0,1)$
ω	Uninformed search correlation	$\mathcal{U}(0,4\pi)$
λ_{id}	Informed search decay	$exp(5)$
λ_{lp}	Rate of laying pheromone	$exp(1)$
λ_{fp}	Rate of following pheromone	$exp(1)$
λ_{sf}	Rate of site fidelity	$exp(1)$
λ_{pd}	Rate of pheromone decay	$exp(10)$

Table 3.1: Set of 8 parameters evolved in simulation guided by genetic algorithms. At the start of a simulated run, parameters in each colony are initialized using randomly sampled values from their associated initialization function. The first 3 parameters are initially sampled from a uniform distribution, and the last 5 from exponential distributions within the stated bounds.

taining a list of waypoints. Pheromone strength of each waypoint evaporates over time (λ_{pd}). Physical marking is not possible with real robots, and therefore our simulated agents follow the same protocol.

3.3.2 Search Algorithm

CPFA pseudocode is shown in Algorithm 1. Note that probabilities of using site fidelity ($F_{sf}(c)$), laying pheromone ($F_{lp}(c)$), and following pheromone ($F_{fp}(c)$) are generated using the equations discussed in the previous subsection. Figure 3.2(a) shows a state diagram of the algorithm, and Figure 3.2(b) illustrates an example of one possible cycle through the search behavior loop. An iAnt robot is shown in Figure 3.2(c).



(a) (b) (c)

Figure 3.2: (a) State diagram describing the flow of behavior for individual robots during an experiment, (b) an example of a single cycle through this search behavior loop, and (c) an iAnt robot with Velcro for attaching reflective markers (motion capture was used for a previous experiment, but not for any of the observations in this paper). The robot begins its search at a central nest site (double circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for resources** (blue line) until a resource (black squares) is found. After sensing the local resource density, the robot **travels to the nest** (red line).

3.3.3 Physical Sensor Error

Two sensing components are precise in simulation but error-prone in our physical iAnt robot platform: positional measurement and resource detection. Our physical robots use a combination of ultrasonic distance, magnetic compass headings, time-based odometry, and an on-board forward-facing camera to estimate their position within the experimental area. Resource detection is accomplished using a downward-facing camera to read barcode-style QR tags.

We measured positional error in five physical robots while localizing to measure the absolute position of a found resource, and while traveling to a location informed by site fidelity or pheromones. We replicated each test 20 times per robot; means and standard deviations for both types of positional error were calculated using 120 samples each. For robots localizing at a true position of $(0\text{ cm}, 0\text{ cm})$, we observed

a measured position of $(-18 \pm 79 \text{ cm}, -15 \pm 47 \text{ cm})$, whereas robots traveling to a true position of $(0 \text{ cm}, 0 \text{ cm})$ had a measured position of $(1.6 \pm 45 \text{ cm}, 64 \pm 110 \text{ cm})$.

Positional error is modeled by perturbing the physical position of an agent from (x, y) to (x', y'), such that $x' \sim \mathcal{N}(x + \hat{x}, \sigma_x)$ and $y' \sim \mathcal{N}(y + \hat{y}, \sigma_y)$. That is, (x', y') is sampled from a normal distribution with mean equal to the true position (x, y) offset by $(\hat{x}, \hat{y})$, and standard deviation (σ_x, σ_y). We impose this positional perturbation twice: once when a robot finds a resource, and again when a robot leaves the nest using site fidelity or following a pheromone waypoint to a known location.

We observed resource detection error for physical robots searching for resources, and for robots searching for neighboring resources. Resource-searching robots attempt to physically align with a QR tag, using small left and right rotations and forward and backward movements to center the tag in their down-facing camera. Robots searching for neighboring resources do not use this alignment strategy, but instead simply rotate 360°, scanning for a tag every 10° with their down-facing camera. We replicated each test 20 times for three different robots; means for both types of resource detection error were calculated using 60 samples each. We observed that resource-searching robots detected 55% of tags and neighbor-searching robots detected 43% of tags. Resource detection is modeled as a fixed probability $d_r = 0.55$ for resource-searching robots, and $d_n = 0.43$ for neighbor-searching robots.

3.3.4 Experimental Design

Each experimental physical trial on a 100 m^2 concrete surface runs for 30 minutes. An illuminated beacon marks the center 'nest' to which the robots return once they have located a resource. This center point is used for localization and error correction by the robots' ultrasonic sensors, magnetic compass, and front-facing camera. All robots involved in a trial are initially placed near the beacon. Robots are programmed to

stay within a 5 m 'virtual fence' of the beacon. In every experiment, 256 QR tags are arranged in 4 randomly placed clusters of 64 tags each.

Robot locations are continually transmitted over one-way WiFi communication to a central server and logged for later analysis. When a tag is found, its unique identification number is transmitted back to the server, providing us with a detailed record of tag discovery. Tags can only be read once, simulating seed retrieval. The central server also acts as a coordinator for pheromone waypoints using two-way communication. As each robot returns to the nest, the server selects a location from the list (if available) and transmits it to the robot.

Simulated teams of five robots search for resources on a 125 x 125 cellular grid. The system architecture replicates the physical dimensions of our real robots, their speed while traveling and searching, and the area over which they can detect resources. The spatial dimensions of the grid reflect the distribution of resources over a 100 m^2 physical area, and agents search for a simulated half hour. 256 identical resources are placed on the grid (each resource occupies a single grid cell) in one of three distributions: random (each resource placed at a random location), clustered (4 randomly placed clusters of 64 resources each), or power law (1 large cluster of 64, 4 medium clusters of 16, 16 small clusters of 4, and 64 randomly scattered). Each individual pile is placed at a new random, non-overlapping location for each fitness evaluation in an effort to avoid bias or convergence to a specific resource layout.

A population of 200 teams is evolved for 100 generations using recombination and mutation. Each team's parameter set is randomly initialized using uniform independent samples from each parameter's initialization function (see Table 3.1, column 3); agents within a team use identical parameters throughout the simulation. Each team forages for resources on its own grid, but the grids are identical. During each generation, all 200 teams undergo eight evaluations with different random placements of tag clusters; fitness is evaluated as the sum total of resources collected by each

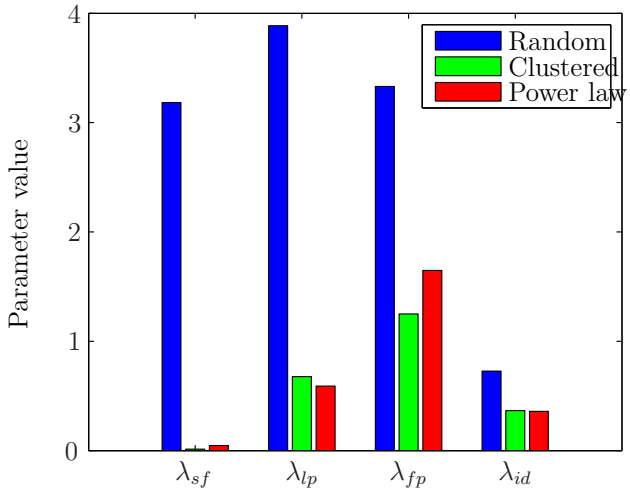


Figure 3.3: Parameter values for rates of site fidelity (λ_{sf}), laying pheromone (λ_{lp}), following pheromone (λ_{fp}), and informed random walk decay (λ_{id}) for random, clustered, and power law distributed resources.

team in the eight runs of a generation. Two individual teams are chosen through tournament selection and recombined through independent assortment: each parameter has a 10% chance of being selected from the second individual, otherwise it is selected from the first individual. Once selected, each parameter has a 10% chance of mutation.

We additionally conduct a series of *parameter swapping* experiments, in which we transfer a parameter set evolved in a simulated error-free world to a simulated world with error. We compare the performance for parameters adapted to error to results using the original parameters not adapted to error. For these experiments, we average the resources collected across multiple replicates. In this way, we can determine the importance of including error in our model by testing whether it has a significant effect on the evolved behavior of the physical and simulated robot teams.

3.4 Results

We present results for teams of five physical and simulated robots searching for resources in worlds with and without sensor error. Unless otherwise noted, results for each experimental treatment are averaged over five physical replicates and ten simulated replicates. Error bars indicate one standard deviation of the mean.

Figure 3.3 shows parameter values influencing robots' use of information (λ_{sf}, λ_{lp}, and λ_{fp}), as well as the informed walk decay rate (λ_{id}) for random, clustered, and power law distributed resources. We observe similar values for all four parameters for clustered and power law distributions: robots evolve a high rate of following pheromones and a low rate of using site fidelity. Robots foraging on random distributions evolve both a high rate of following pheromones and a high rate of using site fidelity, but the effective probability of using either behavior are actually low because of the dependencies between them (see Algorithm 1).

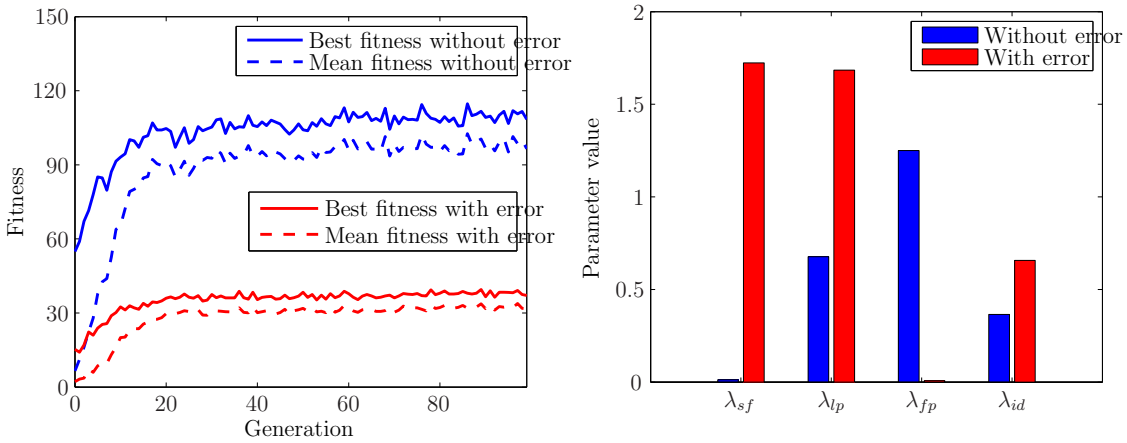


Figure 3.4: Results for simulated foraging on a clustered resource distribution with and without error. (a) Best and mean fitness curves. (b) Parameter values for rates of site fidelity (λ_{sf}), laying pheromone (λ_{lp}), following pheromone (λ_{fp}), and informed random walk decay (λ_{id}).

Figure 3.4 shows fitness curves and parameter values adapted for simulated foraging for resources on a clustered distribution. Figure 3.4(a) plots best and mean fitness over 100 generations for worlds with and without positional and resource detection error modeled on our physical iAnt robots. We observe fitness stabilizing after approximately 20 generations. Simulations with error converge to a fitness level approximately 33% of the fitness achieved in simulations without error. Figure 3.4(b) shows parameter values influencing robots' use of information ($\lambda_{sf}, \lambda_{lp}$, and λ_{fp}), as well as the informed walk decay rate (λ_{id}). Robots foraging in an error-free world evolve a high rate of following pheromones (1.2) and a low rate of using site fidelity (0.013), whereas robots in a world with error evolve a high rate of site fidelity (1.7) and a low rate of following pheromones (0.0071). Additionally, in worlds with error, robots are 2.4 times more likely to lay pheromones, and their informed random walk decays 1.8 times faster than in an error-free world.

We analyze the performance of physical and simulated robots foraging in a world with error using parameters adapted specifically for the error-prone world. We compare the results to robots in a world with error using parameters adapted for an error-free world. Figure 3.5 shows the effects of parameter swapping on resource collection for physical and simulated robots (simulated results are averaged over 100 replicates). We observe an 80% improvement using the error-adapted parameters in physical robot teams, and a 16% improvement in simulated robot teams. We were able to distinguish a significant effect of parameter swapping in physical robots ($t(8) = 5.1, p < 0.001$) and in simulation ($t(198) = 17, p < 0.001$). Although simulated robots collect more resources than physical robots when using non-error-adapted parameters, we find that physical and simulated robots using error-adapted parameters are not significantly different ($t(103) = 0.16, p = 0.87$).

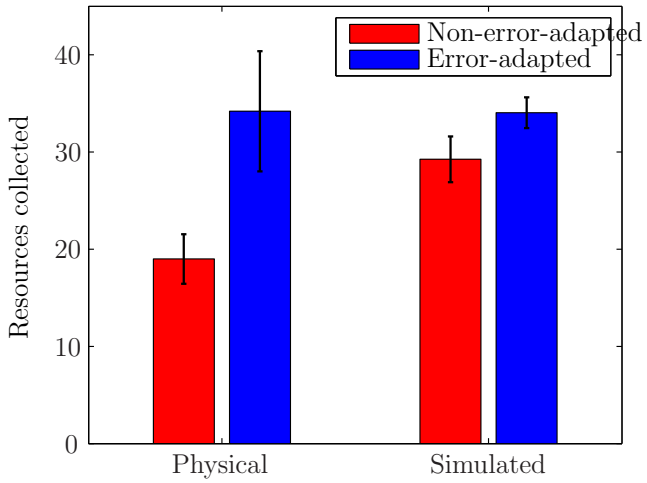


Figure 3.5: Results for physical and simulated robots foraging in a world with error using parameters adapted for a world with error, and parameters adapted for an error-free world. 80% more resources are collected using error-adapted parameters in physical robot teams, and 16% more are collected in simulated teams. Robots collected significantly more resources in both cases. Physical and simulated robots using error-adapted parameters are not significantly different.

3.5 Discussion

Teams of physical and simulated robots used a central-place foraging algorithm (CPFA) to search for resources with and without sensor error. A genetic algorithm (GA) was used to evolve parameter sets which corresponded to robot team behaviors inspired by seed-harvester ants. We considered two types of error, positional error and resource detection error, and we explored the effects of error on overall resource collection and on individual evolved parameters. Error-adapted parameters improved performance of physical and simulated robots in worlds with error. We observed that teams of robots in error-adapted simulations collected resources at the same rate as physical robots.

Both positional and detection errors have the potential to confound a robot's ability to properly use information to exploit resources clustered via site fidelity

or pheromones. Large positional errors in the estimation of resource locations can cause robots to perform informed random walks in regions without resources, thereby wasting time in detailed searches of the wrong areas. Errors in detecting resources can cause robots to underestimate the numbers of resources in a local area, so that robots fail to take advantage of memory or communication to return or recruit other agents to resource-rich locations.

Evolutionary algorithms have the potential to mitigate sensing errors by selecting for parameters which perform optimally given imperfect conditions. For example, robots experiencing errors in resource detection benefit from a lower threshold of resource density detection for triggering creation of a pheromone waypoint. Robots with positional errors perform better with a faster decaying informed random walk, so that they quickly abandon detailed searches when there is a high probability that resources are not in remembered or communicated locations.

Parameter values for simulated robots foraging on random, clustered, and power law distributed resources (Fig. 3.3) illustrate the GA's ability to evolve sets of behaviors for each distribution. Parameters for clustered and power law distributions are similar, demonstrating the ability of the GA to focus on exploiting clumped resources when available. The lack of clustering in the random distribution induces the GA to effectively disable site fidelity and pheromone following behaviors, thus causing the adapted robot teams to concentrate on random exploration.

Fitness curves for simulations with and without error (Fig. 3.4(a)) demonstrate the ability of the GA to reliably converge. Parameter values (Fig. 3.4(b)) demonstrate the ability of the GA to evolve distinct sets of behaviors for an error-free world compared to a world with error.

Results for parameters swapped from error-free worlds into worlds with error (Fig. 3.5) show that parameters adapted for imperfect worlds outperformed parameters

adapted for perfect worlds. Teams of physical and simulated robots collected similar numbers of resources, particularly when using parameters adapted for error. Thus, evolutionary methods effectively adapt robot behavior to sensor error. These results also mirror observations from our previous work in which genetic algorithms were used to evolve optimal parameter sets for specific types of resource distributions.

The work presented here motivates estimation of real robot error, evolution of parameters to fit with that error, and programming of those evolved parameters into real robots. In future work, we will conduct additional physical and simulated robot experiments using different numbers and distributions of resources, arena sizes, numbers of robots, and modes of communication to test whether simulations and physical experiments continue to correspond as closely as we have observed here.

3.6 Acknowledgments

This work was funded by NSF EF #1038682 and DARPA CRASH #P-1070-113237.

Algorithm 1 Biologically-Inspired CPFA

```
Disperse from nest to random location
while experiment running do
    Conduct uninformed correlated random walk
    if resource found then
        Count number of resources c near current location l_f
        Return to nest with resource
        if F_lp(c) > U(0,1) then
            Create pheromone waypoint for l_f
            Pheromones followed by robots at nest
            Pheromones decay over time
        else
            if F_sf(c) > U(0,1) and F_fp(c) < U(0,1) then
                Return to l_f
                Conduct informed correlated random walk
            else
                Check for pheromone
                if pheromone found and F_fp(c) > U(0,1) and F_sf(c) < U(0,1) then
                    Travel to pheromone location l_p
                    Conduct informed correlated random walk
                else
                    Choose new random location
                end if
            end if
        end if
    end if
end while
```

Chapter 4

Beyond Pheromones: Evolving Error-Tolerant, Flexible, and Scalable Ant-Inspired Robot Swarms

4.1 Abstract

For robot swarms to operate outside of the laboratory in complex real-world environments, they require the kind of error tolerance, flexibility, and scalability seen in living systems. While robot swarms are often designed to mimic some aspect of the behavior of social insects or other organisms, no systems have yet addressed all of these capabilities in a single framework. We describe a swarm robotics system that emulates ant behaviors which govern memory, communication, and movement, as well as an evolutionary process that tailors those behaviors into foraging strategies that maximize performance under varied and complex conditions. The system

evolves appropriate solutions to different environmental challenges. Solutions include: *i*) increased communication when sensed information is reliable and resources to be collected are highly clustered, *ii*) less communication and more individual memory when cluster sizes are variable, and *iii*) greater dispersal with increasing swarm size. Analysis of the evolved behaviors reveals the importance of interactions among behaviors, and of the interdependencies between behaviors and environments. The effectiveness of interacting behaviors depends on the uncertainty of sensed information, the resource distribution, and the swarm size. Such interactions could not be manually specified, but are effectively evolved in simulation and transferred to physical robots. This work is the first to demonstrate high-level robot swarm behaviors that can be automatically tuned to produce efficient collective foraging strategies in varied and complex environments.

4.2 Introduction

Robot swarms are appealing because they can be made from inexpensive components, their decentralized design is well-suited to tasks that are distributed in space, and they are potentially robust to communication errors that could render centralized approaches useless. A key challenge in swarm engineering is specifying individual behaviors that result in desired collective swarm performance without centralized control [69, 140]; however, there is no consensus on design principles for producing desired swarm performance from individual agent behaviors [17]. Moreover, the vast majority of swarms currently exist either as virtual agents in simulations or as physical robots in controlled laboratory conditions [139, 17] due to the difficulty of designing robot swarms that can operate in natural environments. For example, even mundane tasks such as garbage collection require operating in environments far less predictable than swarms can currently navigate. Furthermore, inexpensive

components in swarm robotics lead to increased sensor error and a higher likelihood of hardware failure compared to state-of-the-art monolithic robot systems.

This calls for an integrated approach that addresses the challenge of designing collective strategies for complex and variable environments [98, 53]. Pfeifer et al [110] argue that biologically-inspired behaviors and physical embodiment of robots in an ecological niche can lead to adaptive and robust robots. Here we describe such an approach for robot swarm foraging, demonstrate its effectiveness, and analyze how individual behaviors and environmental conditions interact in successful strategies.

This paper describes a robot swarm that forages for resources and transports them to a central place. Foraging is an important problem in swarm robotics because it generalizes to many real-world applications, such as collecting hazardous materials and natural resources, search and rescue, and environmental monitoring [80, 107, 139, 17]. We test to what extent evolutionary methods can be used to generate error-tolerant, flexible, and scalable foraging behaviors in simulation and in physical experiments conducted with up to 6 iAnt robots. The iAnt is an inexpensive platform (shown in Figure 4.1) capable of movement, memory, and communication, but with substantial sensing and navigation error [59].

(a) (b)

Figure 4.1: (a) An iAnt robot. (b) A swarm of iAnt robots foraging for resources around a central illuminated beacon.

Our approach to developing foraging strategies emulates biological processes in two ways. First, robot behaviors are specified by a central-place foraging algorithm (CPFA) which mimics the foraging behaviors of seed-harvester ants. Second, we use a genetic algorithm (GA) to tune CPFA parameters to optimize performance in different conditions. The GA-tuned CPFA is an integrated strategy in which movement, sensing, and communication are evolved and evaluated in an environment with a particular amount of sensing and navigation error, a particular type of resource distribution, and a particular swarm size. Our iAnt robots provide a platform to test how well the GA can evolve behaviors that tolerate realistic sensing and navigation error, and how much those errors affect foraging performance given different resource distributions and swarm sizes.

This study builds on important previous work in which robot swarms mimic a specific component of ant foraging behavior. For example, substantial attention has been given to pheromone communication [108, 117, 24], and others have imitated ant navigation mechanisms, cooperative carrying, clustering, and other isolated behaviors [21, 15, 116, 125, 11]. Rather than imitating a specific behavior for a specific subtask, we evolve strategies that use different combinations of navigation, sensing, and communication to accomplish a complete foraging task. This approach mimics the way that ant foraging strategies evolve in nature. Ants do not decompose the foraging problem into subtasks; rather, from a small set of behaviors, each species of ant has evolved an integrated strategy tuned to its own particular environment. We emulate not just the behaviors, but also the evolutionary process that combines those behaviors into integrated strategies that are repeatedly tested in the real environments in which each species forages.

Our study is the first to evolve foraging behaviors that are effective in varied and complex environments. Previous studies have developed or evolved foraging behaviors for randomly distributed resources [7, 28, 80], while others have studied

foraging from one or two infinite sources [60, 38]. However, previous studies have not attempted to evolve strategies that are sufficiently flexible to perform well in both of those environments, nor have they developed strategies that are effective at collecting from more complex distributions. We show that foraging for resources in heterogeneous clusters requires more complex communication, memory, and environmental sensing than strategies evolved in previous work. This is important for robot swarms operating outside of controlled laboratory environments because the features of natural landscapes are heterogeneous, and the complex topology of natural landscapes has a profound impact on how animals search for resources [130, 68, 135]. In particular, the patchiness of environments and resources affects which foraging behaviors are effective for seed-harvesting ants [25].

This work provides an automated process to adapt the high-level behaviors of individual foragers to optimize collective foraging performance in complex environments with varied resource distributions. Experiments show the evolution of complex strategies that are effective when resources are clustered heterogeneously, the automatic adaptation of these strategies to different distributions, and the evolution of a generalist strategy that is effective for a variety of resource distributions (even when the distributions are not known *a priori*). We additionally evolve foraging behaviors that are tolerant of real-world sensing and navigation error, and scalable (in simulation) to large swarm sizes. The novelty of the approach is that it takes into account interactions between the various behaviors that compose a foraging task (e.g., exploration, exploitation by individuals, and recruitment), and interdependencies between behaviors and the environmental context in which the behaviors evolve. The utility of this approach is evident in two examples of how behaviors adapt and interact: *i*) greater amounts of communication evolve in experiments with clustered resource distributions, reliable sensors, and small swarms; and *ii*) given a variety of pile sizes, robots evolve to exploit small piles using individual memory and to exploit large piles using pheromone recruitment. More generally, we show that efficient and

flexible strategies can emerge when simple behaviors evolve in response to complex and variable environments.

In summary, this work makes three main contributions: *i*) we evolve a complete foraging strategy composed of behaviors that interact with each other and that adapt to the navigation and sensing errors of the robots, the environment, and the size of the swarm; *ii*) we automatically tune foraging behaviors to be effective in varied and complex environments; and *iii*) we analyze the evolved foraging strategies to understand how effective strategies emerge from interactions between behaviors and experimental conditions.

4.3 Related Work

This paper builds on a large body of related research in robot swarm foraging behaviors, ant foraging behaviors, and our own prior work developing the CPFA and iAnt robot platform.

4.3.1 Automatic Design of Swarm Foraging Behaviors

The most common automatic design approach in swarm foraging is evolutionary robotics (ER). Research in ER primarily focuses on using evolutionary methods to develop controllers for autonomous robots [87, 100]. Previous work in ER has evolved neural networks to control lower-level motor functions in simulated robot agents; controllers were subsequently transferred to real robots with success on several different tasks [8, 2, 113]. One drawback of this approach is that the evolved neural controllers are a black box – it is often not clear why a particular controller is good for a particular task. Additionally, task generalization is difficult because evolved solutions are often overfit to specific design conditions [38]. Our approach

mitigates these problems by tuning a simple set of behaviors inspired by foraging ants. Because the behaviors are simple, the evolved parameters are relatively easy to interpret. Additionally, because the GA fine-tunes predefined, high-level behaviors, it avoids overfitting solutions to idiosyncratic features of either simulated or physical conditions.

Our GA evolves parameters to control the high-level behaviors we have observed and modeled in ants. These parameters control the sensitivity threshold for triggering behaviors, the likelihood of transitioning from one behavior to another, and the length of time each behavior should last. Several previous projects have taken an approach similar to our own, using learning and optimization techniques to tune a fixed repertoire of higher-level swarm foraging behaviors, rather than lower-level motor controllers or basic directional responses. Matarić [84, 85] used reinforcement learning to train robots to switch between behaviors through positive and negative reinforcement related to foraging success. Similar to Matarić [84], Balch [7] trained robot teams to perform multiple foraging tasks simultaneously using Q-learning with a shaped reinforcement reward strategy. Labella et al [74] implemented adaptive swarm foraging, observing emergent division of labor using only local information and asynchronous communication. Liu and Winfield [79] used a genetic algorithm to tune a macroscopic probabilistic model of adaptive collective foraging, optimizing division of labor and minimizing energy use. Francesca et al [38] used a parameter optimization algorithm to automatically construct probabilistic behavioral controllers for swarm aggregation and foraging tasks. These previous studies have tested swarms on simple foraging tasks that required no communication. Instead, we focus on more difficult foraging tasks in which communication among robots increases collective foraging efficiency. Efficient foraging in environments with more complex resource distributions necessitates more complex foraging strategies. In our study, robots alter the environment by collecting food and by laying pheromones, and those alterations affect future robot behavior. Therefore, these foraging strategies cannot be

practically represented by the finite state machines often used in prior work [79, 38].

4.3.2 Foraging in Desert Harvester Ants

The CPFA mimics foraging behaviors used by desert seed-harvester ants. Desert harvester ants collect seeds that are scattered in space and remain available for long time periods, but foraging under hot, dry conditions limits seed collection to short time windows during which not all available resources can be collected [50]. We emulate harvester ant foraging strategies which have evolved to collect many seeds quickly, but not exhaustively collect all available seeds. Colonies must adapt their foraging strategies to seasonal variations in environmental conditions and competition with neighbors [1].

Foragers initially disperse from their central nest in a travel phase, followed by a search phase [34] in which a correlated random walk is used to locate seeds [26]. Foragers then navigate home to a remembered nest location [61]. Seed-harvester ants typically transport one seed at a time, often searching the surrounding area and sometimes sampling other seeds in the neighborhood of the discovered seed [61]. Letendre and Moses [77] hypothesized that this behavior is used to estimate local seed density.

Ants can sense direction using light polarization, remember landmarks [61], and, even in the absence of visual cues, measure distance using odometry [142, 124]. These mechanisms enable ants to navigate back to previously visited sites and return to their nest [61], sometimes integrating visual cues to rapidly remember and straighten their homebound paths [96].

It is frequently observed that an individual ant will remember the location of a previously found seed and repeatedly return to that location [61, 26, 12]. This behavior is called *site fidelity*. When foragers return to a site using site fidelity, they

appear to alter their search behavior such that they initially search the local area thoroughly, but eventually disperse to search more distant locations [35]. We model this process using a biased random walk that is initially undirected and localized with uncorrelated, tight turns [109, 77]. Over time, successive turning angles become more correlated, causing the path to straighten.

Many ants also lay pheromone trails from their nest to food patches [51, 16, 20, 122, 65]. Foragers at the nest then follow these pheromone trails, which direct the ants to high-quality food patches via the process of recruitment. Trails are reinforced through positive feedback by other ants that follow trails with a probability that increases as a function of the chemical strength of the trail. Recruitment by pheromone trails is rare in seed harvesters except in response to very large and concentrated seed piles [46, 49].

4.3.3 Foundations of the CPFA

In prior work, we observed and modeled ants foraging in natural environments [35], parameterized those models using a GA that maximized seed collection rates for different resource distributions [109, 77], and instantiated those foraging parameters in robot swarms [55, 56, 59]. This process has led to the robot foraging algorithms we describe here.

Flanagan et al [35] conducted manipulative field studies on three species of *Pogonomyrmex* desert seed-harvesters. In order to test behavioral responses to different food distributions, colonies were baited with seeds clustered in a variety of pile sizes around each ant nest. Ants collected seeds faster when seeds were more clustered. An agent-based model (ABM) simulated observed foraging behaviors, and a GA was used to find individual ant behavioral parameters that maximized the seed collection rate of the colony. Simulated ants foraging with those parameters mimicked the in-

crease of seed collection rate with the amount of clustering in the seed distribution when ant agents were able to remember and communicate seed locations using site fidelity and pheromones [109].

Letendre and Moses [77] tested the ABM and observed how model parameters and foraging efficiency changed with different distributions of resources. Simulations showed that both site fidelity and pheromone recruitment were effective ways to collect clustered resources, with each behavior increasing foraging success on clustered seed distributions by more than 10-fold, compared to a strategy which used no memory or communication. Both site fidelity and pheromones were beneficial, but less so, with less clustered seed distributions. Further, simulations demonstrated an important synergy between site fidelity and pheromone recruitment: each behavior became more effective in the presence of the other behavior [92].

Letendre and Moses [77] also showed that a GA could effectively fine-tune the repertoire of ant foraging behaviors to different resource distributions. Parameters evolved for specific types of resource distributions were swapped and fitness was measured for the new distribution; for example, parameters evolved for a clustered distribution were tested on random distributions of resources. Simulated agents incurred as much as a 50% decrease in fitness when using parameters on a distribution different from the one for which they were evolved.

The robot algorithms and experiments described in this paper are informed by insights from these studies and simulations of ant foraging: *i*) the success of a foraging strategy depends strongly on the spatial distribution of resources which are being collected, and *ii*) memory (site fidelity) and communication (pheromones) are critical components of foraging strategies when resources are clustered.

We simplified and formalized the behaviors from Letendre and Moses [77] into a robot swarm foraging algorithm, the CPFA, in Hecker and Moses [56]. In this work,

we showed that a GA, using a fitness function that included a model of iAnt sensing and navigation errors, could evolve CPFA parameters to generate behaviors that improved performance in physical iAnt robots. The CPFA is designed to provide a straightforward way to interpret parameters evolved by the GA in order to assess how movement patterns, memory, and communication change in response to different sensor errors, resource distributions, and swarm sizes. The CPFA also reflects the fact that our physical robots lack the ability to lay chemical pheromone trails. Instead, pheromones are simulated in a list of pheromone-like waypoints (described below).

The work presented here is a comprehensive study of the GA, CPFA, and iAnt platform. We extend our previous results by performing a systematic analysis of *i*) error tolerance to adapt CPFA parameters to improve performance given errors inherent to the iAnt robots, *ii*) flexibility to forage effectively for a variety of resource distributions in the environment, and *iii*) scalability to increasing swarm size with up to 6 physical robots and up to 768 simulated robots.

4.4 Methods

The design components of our system include the central-place foraging algorithm (CPFA), the genetic algorithm (GA), the physical iAnt robots, the sensor error model, and the experimental setup. The error tolerance, flexibility, and scalability of our robot swarms are tested under different experimental conditions. The framework for our approach is shown in Figure 4.2.

4.4.1 Central-Place Foraging Algorithm

The CPFA implements a subset of desert seed-harvester ant foraging behaviors (see Subsection 4.3.2) as a series of states connected by directed edges with transition

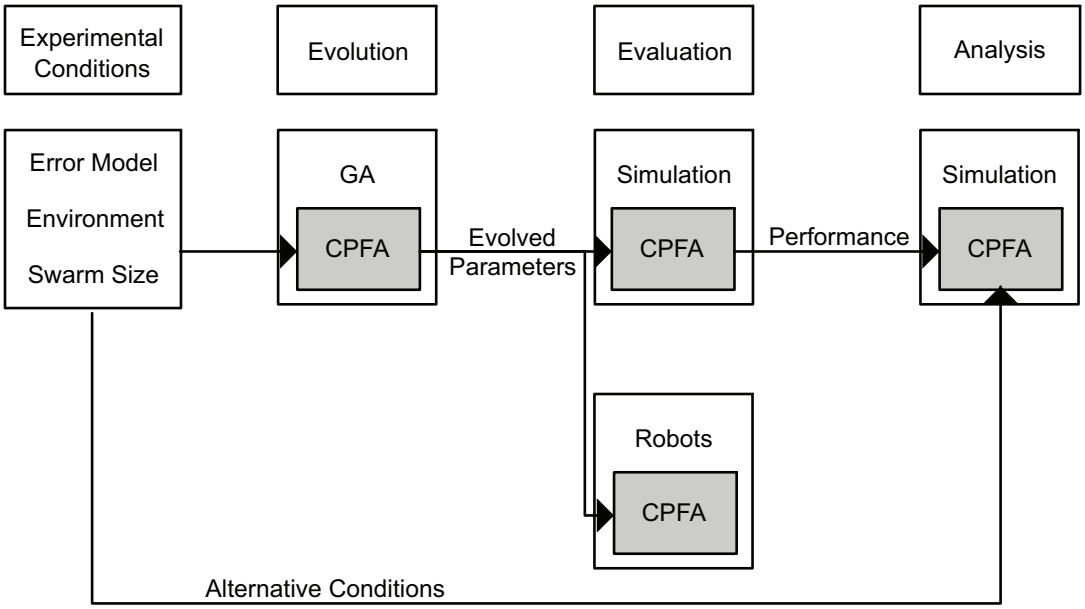


Figure 4.2: We use a GA to **evolve** a foraging strategy (CPFA parameter set) that maximizes resource collection for specified classes of error model, environment, and swarm size. We then **evaluate** the foraging strategy in multiple experiments with simulated and physical robots and record how many resources were collected. We repeat this for different error models, environments, and swarm sizes. We **analyze** flexibility by evolving parameters for one condition and evaluating them in another.

probabilities (Figure 4.3). The CPFA acts as the high-level controller for our simulated and physical iAnt robots. Parameters governing the CPFA transitions are listed in Table 4.1, and CPFA pseudocode is shown in Algorithm 2.

Each robot transitions through a series of states as it forages for resources:

- **Set search location:** The robot starts at a central nest and selects a dispersal direction, θ, initially from a uniform random distribution, $\mathcal{U}(0, 2\pi)$. In subsequent trips, the robot may set its search location using site fidelity or pheromone waypoints, as described below.

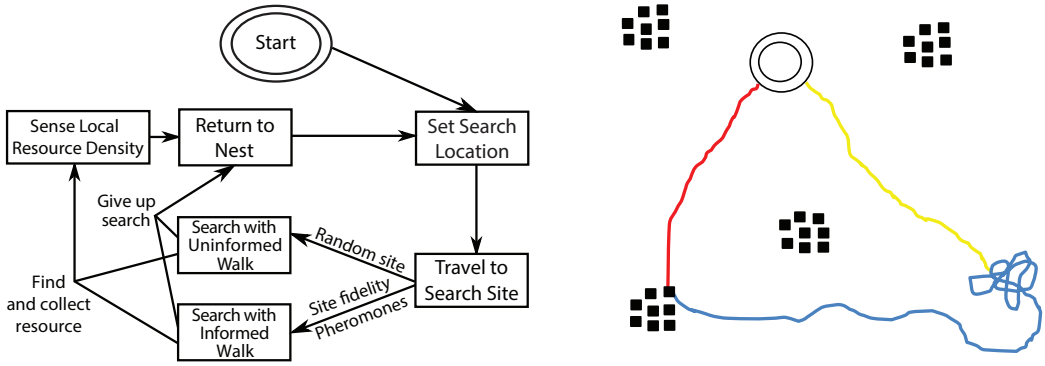


Figure 4.3: (a) State diagram describing the flow of behavior for individual robots during an experiment. (b) An example of a single cycle through this search behavior. The robot begins its search at a central nest site (double circle) and **sets a search location**. The robot then **travels to the search site** (solid line). Upon reaching the search location, the robot **searches for resources** (dotted line) until a resource (square) is found and collected. After sensing the local resource density, the robot **returns to the nest** (dashed line).

- **Travel to search site:** The robot travels along the heading θ, continuing on this path until it transitions to searching with probability p_s.

- **Search with uninformed walk:** If the robot is not returning to a previously found resource location via site fidelity or pheromones, it begins searching using a correlated random walk with fixed step size and direction θ_t at time t, defined

Table 4.1: Set of 7 CPFA parameters evolved by the GA

Parameter	Description	Initialization Function
p_s	Probability of switching to searching	$\mathcal{U}(0,1)$
p_r	Probability of returning to nest	$\mathcal{U}(0,1)$
ω	Uninformed search variation	$\mathcal{U}(0,4\pi)$
λ_{id}	Rate of informed search decay	$exp(5)$
λ_{sf}	Rate of site fidelity	$\mathcal{U}(0,20)$
λ_{lp}	Rate of laying pheromone	$\mathcal{U}(0,20)$
λ_{pd}	Rate of pheromone decay	$exp(10)$

by Equation 4.1:

$$\theta_t = \mathcal{N}(\theta_{t-1}, \sigma) \tag{4.1}$$

The standard deviation σ determines how correlated the direction of the next step is with the direction of the previous step. Robots initially search for resources using an uninformed correlated random walk, where σ is assigned a fixed value in Equation 4.2:

$$\sigma \leftarrow \omega \tag{4.2}$$

If the robot discovers a resource, it will collect the resource by adding it to a list of collected items, and transition to sensing the local resource density. Robots that have not found a resource will give up searching and return to the nest with probability p_r.

- **Search with informed walk:** If the robot is informed about the location of resources (via site fidelity or pheromones), it searches using an informed correlated random walk, where the standard deviation σ is defined by Equation 4.3:

$$\sigma = \omega + (4\pi - \omega)e^{-\lambda_{id}t} \tag{4.3}$$

The standard deviation of the successive turning angles of the informed random walk decays as a function of time t, producing an initially undirected and localized search that becomes more correlated over time. This time decay allows the robot to search locally where it expects to find a resource, but to

straighten its path and disperse to another location if the resource is not found. If the robot discovers a resource, it will collect the resource by adding it to a list of collected items, and transition to sensing the local resource density. Robots that have not found a resource will give up searching and return to the nest with probability p_r.

- **Sense local resource density:** When the robot locates and collects a resource, it records a count c of resources in the immediate neighborhood of the found resource. This count c is an estimate of the density of resources in the local region.
- **Return to nest:** After sensing the local resource density, the robot returns to the nest. At the nest, the robot uses c to decide whether to use information by *i*) returning to the resource neighborhood using site fidelity, or *ii*) following a pheromone waypoint. The robot may also decide to communicate the resource location as a pheromone waypoint.

Information decisions are governed by parameterization of a Poisson cumulative distribution function (CDF) as defined by Equation 4.4:

$$\textsc{Pois}(c, \lambda) = e^{-\lambda} \sum_{i=0}^{\lfloor c \rfloor} \frac{\lambda^i}{i!} \tag{4.4}$$

The Poisson distribution represents the probability of a given number of events occurring within a fixed interval of time. We chose this formulation because of its prevalence in previous ant studies, e.g., researchers have observed Poisson distributions in the dispersal of foragers [62], the density of queens [128], and the rate at which foragers return to the nest [114].

In the CPFA, an event corresponds to finding an additional resource in the immediate neighborhood of a found resource. Therefore, the distribution $\textsc{Pois}(c, \lambda)$

describes the likelihood of finding at least c additional resources, as parameterized by λ. The robot returns to a previously found resource location using site fidelity if the Poisson CDF, given the count c of resources, exceeds a uniform random value: $\textsc{Pois}(c, \lambda_{sf}) > \mathcal{U}(0, 1)$. Thus, if c is large, the robot is likely to return to the same location using site fidelity on its next foraging trip. If c is small, it is likely not to return, and instead follows a pheromone to another location if pheromone is available. If no pheromone is available, the robot will choose its next search location at random. The robot makes a second independent decision based on the count c of resources: it creates a pheromone waypoint for a previously found resource location if $\textsc{Pois}(c, \lambda_{lp}) > \mathcal{U}(0, 1)$.

Upon creating a pheromone waypoint, a robot transmits the waypoint to a list maintained by a central server. As each robot returns to the nest, the server selects a waypoint from the list (if available) and transmits it to the robot. New waypoints are initialized with a value of 1. The strength of the pheromone, γ, decays exponentially over time t as defined by Equation 4.5:

$$\gamma = e^{-\lambda_{pd} t} \tag{4.5}$$

Waypoints are removed once their value drops below a threshold of 0.001. We use the same pheromone-like waypoints in simulation to replicate the behavior of the physical iAnts.

4.4.2 Genetic Algorithm

There are an uncountable number of foraging strategies that can be defined by the real-valued CPFA parameter sets in Table 4.1 (even if the 7 parameters were limited to single decimal point precision, there would be 7^{10} possible strategies). We address

this intractable problem by using a GA to generate foraging strategies that maximize foraging efficiency for a particular error model, resource distribution, and swarm size.

The GA evaluates the fitness of each strategy by simulating robots that forage using the CPFA parameter set associated with each strategy. Fitness is defined as the foraging efficiency of the robot swarm: the total number of resources collected by all robots in a fixed time period. Because the fitness function must be evaluated many times, the simulation must run quickly. Thus, we use a parsimonious simulation that uses a gridded, discrete world without explicitly modeling sensors or collision detection. This simple fitness function also helps to mitigate condition-specific idiosyncrasies and avoid overfitted solutions, a problem noted by Francesca et al [38].

We evolve a population of 100 simulated robot swarms for 100 generations using recombination and mutation. Each swarm's foraging strategy is randomly initialized using uniform independent samples from the initialization function for each parameter (Table 4.1). Five parameters are initially sampled from a uniform distribution, $\mathcal{U}(a, b)$, and two from exponential distributions, $exp(x)$, within the stated bounds. Robots within a swarm use identical parameters throughout the hour-long simulated foraging experiment. During each generation, all 100 swarms undergo 8 fitness evaluations, each with different random placements drawn from the specified resource distribution.

At the end of each generation, the fitness of each swarm is evaluated as the sum total of resources collected in the 8 runs of a generation. Deterministic tournament selection with replacement (tournament size = 2) is used to select 99 candidate swarm pairs. Each pair is recombined using uniform crossover and 10% Gaussian mutation with fixed standard deviation (0.05) to produce a new swarm population. We use elitism to copy the swarm with the highest fitness, unaltered, to the new population – the resulting 100 swarms make up the next generation. After 100 generations, the

evolutionary process typically converges on a set of similar foraging strategies; the strategy with highest fitness at generation 100 is kept as the best foraging strategy.

We repeat the evolutionary process 10 times to generate 10 independently evolved foraging strategies for each error model, resource distribution, and swarm size. We then evaluate the foraging efficiency of each of those 10 strategies using 100 new simulations, each of which uses the CPFA with specified parameters and a new random placement of resources.

4.4.3 iAnt Robot Platform

iAnt robots are constructed from low-cost hardware and range-limited sensors. Our iAnt robot design has been updated and enhanced over three major revisions to improve experimental repeatability and to decrease the reality gap between simulated and physical robot performance.

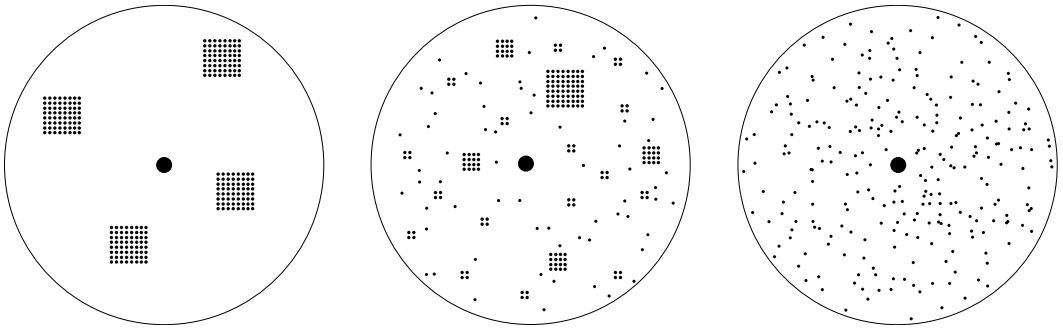
The current iAnt platform (see Figure 4.1) is supported by a custom-designed laser-cut chassis, low-geared motors to provide high torque, and a 7.4V battery that provides consistent power for 60 minutes. The iAnt uses an Arduino Uno microcontroller, combined with an Ardumoto motor shield, to coordinate low-level movement and process on-board sensor input. Sensors include a magnetometer and ultrasonic rangefinder, as well as an iPod Touch to provide iAnts with forward-facing and downward-facing cameras, in addition to computational power. Robots use the OpenCV computer vision library to process camera images. The forward-facing camera is used to detect a central nest beacon, and the downward-facing camera is used to detect QR matrix barcode tags. iAnt cost is approximately $500, with an assembly time of approximately 2 hours. Detailed platform specifications and assembly instructions are available online [94].

4.4.4 Physical Sensor Error Model

Two sensing components are particularly error-prone in our iAnt robot platform: positional measurement and resource detection. In prior work, we reduced the reality gap between simulated and physical robots by measuring sensing and navigation error, then integrating models of this error into our agent-based simulation [59]. In this work, the goal is to understand ways in which behaviors evolve to mitigate the effects of error on foraging performance.

We measured positional error in 6 physical robots while localizing to estimate the location of a found resource, and while traveling to a location informed by site fidelity or pheromones. We replicated each test 20 times for each of 6 robots, resulting in 120 measurements from which we calculated means and standard deviations for both types of positional error. We performed a linear regression of the standard deviation of positional error on the distance from the central beacon and observed that standard deviation ς increased linearly with localization distance d_l, $\varsigma = 0.12d_l - 16$ cm ($R^2 = 0.58, p < 0.001$), and travel distance d_t, $\varsigma = 0.37d_t + 0.02$ cm ($R^2 = 0.54, p < 0.001$).

We also observed resource detection error for physical robots searching for resources, and for robots searching for neighboring resources. Resource-searching robots attempt to physically align with a QR tag, using small left and right rotations and forward and backward movements to center the tag in their downward-facing camera. Robots searching for neighboring resources do not use this alignment strategy, but instead simply rotate 360°, scanning for a tag every 10° with their downward-facing camera. We replicated each test 20 times for each of 3 robots; means for both types of resource detection error were calculated using 60 samples each. We observed that resource-searching robots detected 55% of tags and neighbor-searching robots detected 43% of tags.



(a) Clustered (b) Power law (c) Random

Figure 4.4: 256 resources are placed in one of three distributions: (a) the clustered distribution has four piles of 64 resources. (b) The power law distribution uses piles of varying size and number: one large pile of 64 resources, 4 medium piles of 16 resources, 16 small piles of 4 resources, and 64 randomly placed resources. (c) The random distribution has each resource placed at a uniform random location.

4.4.5 Experimental Setup

- **Physical:** Each physical experiment runs for one hour on a 100 m^2 indoor concrete surface. Robots forage for 256 resources represented by 4 cm^2 QR matrix barcode tags. A cylindrical illuminated beacon with radius 8.9 cm and height 33 cm marks the center nest to which the robots return once they have located a resource. This center point is used for localization and error correction by the robots' ultrasonic sensors, magnetic compass, and forward-facing camera. All robots involved in an experiment are initially placed near the beacon. Robots are programmed to stay within a 'virtual fence' that is a radius of 5 m from the beacon. In every experiment, QR tags representing resources are arranged in one of three distributions (see Figure 4.4): clustered (4 randomly placed clusters of 64 resources each), power law (1 large cluster of 64, 4 medium clusters of 16, 16 small clusters of 4, and 64 randomly scattered), or random (each resource placed at a random location).

  Robot locations are continually transmitted over one-way WiFi communication

to a central server and logged for later analysis. Robots do not pick up physical tags, but instead simulate this process by reading the tag's QR code, reporting the tag's unique identification number to a server, and returning within a 50 cm radius of the beacon, providing a detailed record of tag discovery. Tags can only be read once, simulating tag retrieval.

- **Simulated:** Swarms of simulated robot agents search for resources on a 125 x 125 cellular grid; each cell simulates an 8 x 8 cm square. The simulation architecture replicates the physical dimensions of our real robots, their speed while traveling and searching, and the area over which they can detect resources. The spatial dimensions of the grid reflect the distribution of resources over a 100 m^2 physical area, and agents search for a simulated hour. Resources are placed on the grid (each resource occupies a single grid cell) in one of three distributions: clustered, power law, or random. We use the same resource distribution as in the physical experiments, although physical and simulated resources are not in the same locations. Instead, each individual pile is placed at a new random, non-overlapping location for each fitness evaluation to avoid bias or convergence to a specific resource layout. We use an error model to emulate physical sensing and navigation errors in some simulations (see Subsection 4.4.4).

4.4.6 Performance Evaluation

Here we describe the methods and metrics used to empirically evaluate the error tolerance, flexibility, and scalability of our iAnt robot swarms. We use these metrics to measure the ability of the GA to tune CPFA parameters to maximize the foraging efficiency of swarms under varying experimental conditions. We define *efficiency* as the total number of resources collected within a fixed one hour experimental window. In some cases we measure efficiency per swarm, and in others we measure efficiency per robot. Efficiency per swarm serves as the GA fitness function when evolving

populations of robot swarms in our agent-based simulation. We characterize error tolerance, flexibility, and scalability by comparing E_1 and E_2, where E_1 and E_2 are efficiency measurements under two different experimental conditions. In addition to using performance metrics to measure efficiency changes, our analysis also reveals evolutionary changes in parameters that lead to these changes in efficiency.

Error Tolerance

We measure how well simulated and physical robots mitigate the effects of the error inherent to iAnts. In simulation, error tolerance is measured only in experiments in which simulated robots forage using the model of iAnt sensor error described in Subsection 4.4.4. For robots foraging with such error, error tolerance is defined as:

$$\frac{E_2 - E_1}{E_1} \times 100\% \tag{4.6}$$

where E_1 is the efficiency of a strategy evolved assuming no error, and E_2 is the efficiency of a strategy evolved in the presence of error. This set of experiments demonstrates the ability of our system to increase foraging success given realistic sensor error. Note that simulated robots foraging in the presence of error can never outperform robots foraging without error, and that physical robots always forage in the presence of the inherent iAnt robot error.

Flexibility

Flexibility is defined as:

$$\frac{E_2}{E_1} \times 100\% \tag{4.7}$$

where E_1 is the efficiency of the best strategy evolved for a given resource distribution, and E_2 is the efficiency of an alternative strategy evolved for a different resource distribution but evaluated on the given resource distribution. A strategy that is 100% flexible is one that has been evolved for a different distribution but is equally efficient on the target distribution. We measure flexibility the same way in physical and simulated robots.

We measure flexibility by evolving swarms of 6 simulated robots foraging independently on each of the three resource distributions (see Figure 4.4). When the evolution is complete, we then evaluate each of the three evolved strategies on all three distributions: the one for which they were evolved, as well as the other two (see Figure 4.2). For example, a robot swarm is evolved to forage on power-law-distributed resources, then the swarm is evaluated for efficiency on the power law distribution, as well as the clustered and random distributions.

Scalability

Scalability is defined using Equation 4.7, where E_1 is the efficiency of 1 robot, and E_2 is the efficiency *per robot* of a larger swarm. Note that E_1 and E_2 are defined per robot for scalability, while E_1 and E_2 are defined per swarm for error tolerance and flexibility. We measure scalability from 1 to 6 physical robots, and from 1 to 768 simulated robots.

We measure scalability by evolving swarms of 1, 3, and 6 simulated robots foraging on a power law distribution in a world with error, using the experimental setup described in Subsection 4.4.5. When the evolution is complete, we then evaluate physical and simulated swarms of 1, 3, and 6 robots using the parameters evolved specifically for each swarm size.

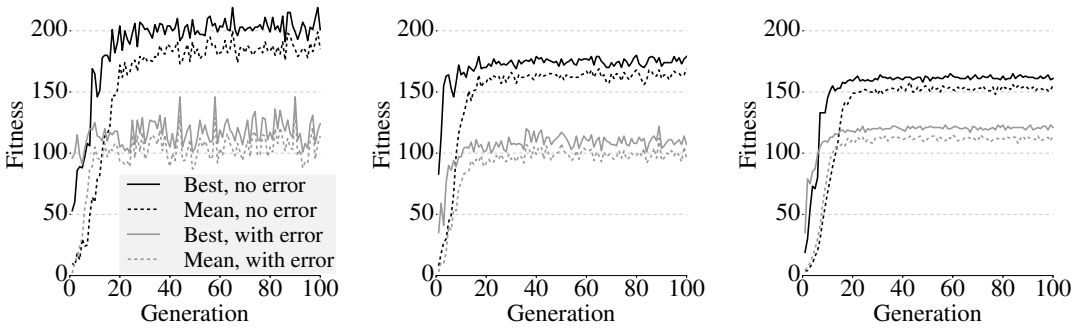
We can measure scalability more thoroughly in simulation, where we analyze 1

to 768 simulated robots in a large simulation space: a 1323 x 1323 cellular grid, replicating an approximate 11,000 m^2 physical area. We evolve simulated swarms foraging for 28,672 resources divided into groups: 1 cluster of 4096 resources, 4 clusters of 1024, 16 clusters of 256, 64 clusters of 64, 256 clusters of 16, 1024 clusters of 4, and 4096 resources randomly scattered. We then evaluate each evolved foraging strategy on the swarm size for which they were evolved. We additionally evaluate a fixed set of parameters evolved for a swarm size of 6 (i.e. parameters are evolved for a swarm size of 6, but evaluated in swarm sizes of 1 to 768) to test the flexibility of a fixed strategy for different numbers of robots.

Finally, we test the effect on site fidelity and pheromones by evolving simulated swarms using the large experimental setup described above, except with information use disabled for all robots in the swarm. Because robots are not able to remember or communicate resource locations, the CPFA parameters λ_{id}, λ_{sf}, λ_{lp}, and λ_{pd} no longer affect robot behavior. This restricts the GA to evolving strategies that govern only the movement patterns specified by the search and travel behaviors (p_r, p_s, and ω). We compare the efficiency of such strategies to the efficiency of swarms using the full CPFA to evaluate how much memory and communication improve foraging performance for different swarm sizes.

4.5 Results

Results below compare parameters and foraging efficiency of the best evolved foraging strategies, where *efficiency* is the total number of resources collected by a robot swarm during an hour-long experiment. Results that compare parameters show means and standard deviations of the 10 foraging strategies evolved in simulation; error bars (when shown) indicate one standard deviation of the mean. Results that compare foraging efficiency show the single best of those 10 strategies evaluated 100



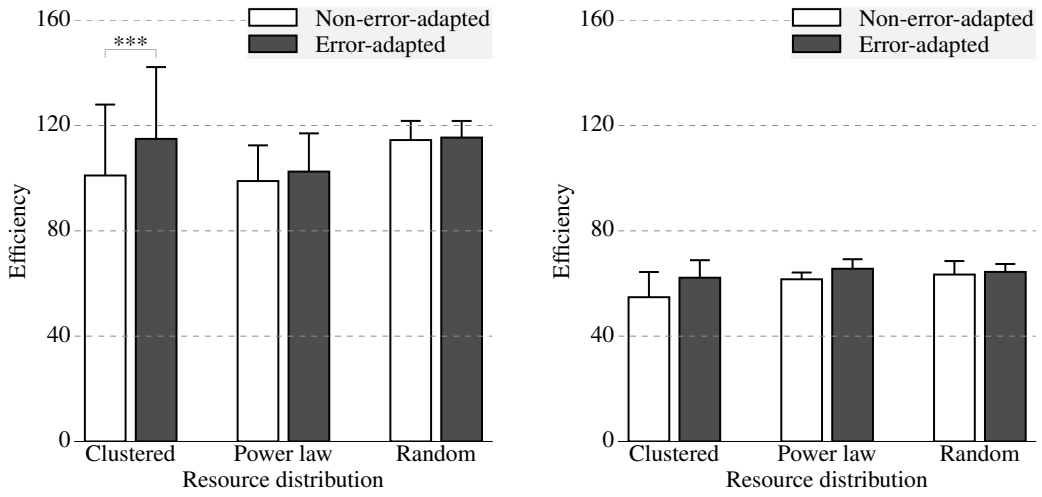
(a) Clustered (b) Power law (c) Random

Figure 4.5: Best and mean fitness, measured as foraging efficiency (resources collected per hour, per swarm) for simulated swarms foraging on (a) clustered, (b) power law, and (c) random resource distributions with and without real-world sensor error. Results are for 100 replicates.

times in simulation and 5 times in physical iAnt robots, for each error model, resource distribution, and swarm size.

4.5.1 Error Tolerance

Figure 4.5 shows best and mean fitness curves for simulated robot swarms foraging with and without sensor error on clustered, power law, and randomly distributed resources. Robot swarms adapted for randomly distributed resources have the most stable fitness function, followed by power-law-adapted and cluster-adapted swarms. Fitness stabilizes for all three distributions after approximately 20 generations. Real-world sensor error has the largest effect on power-law-adapted swarms, reducing mean fitness by 44% by generation 100 (mean fitness without error = 170, mean fitness with error = 96). Sensor error reduces mean fitness by 42% for cluster-adapted swarms (without error = 190, with error = 110), and by 25% for random-adapted swarms (without error = 160, with error = 120). Thus, not surprisingly, robots with error are always less efficient than robots without error. In idealized simulations without



(a) Simulation (b) Physical

Figure 4.6: Foraging efficiency (resources collected per hour, per swarm) using error-adapted and non-error-adapted parameters for (a) 6 robots foraging in a simulation that includes sensor error and (b) 6 physical robots. Asterisks indicate a statistically significant difference ($p < .001$).

robot error, efficiency is higher for the more clustered distributions; but when the model of iAnt error is included, efficiency is highest for randomly dispersed resources.

Figure 4.6 shows the efficiency of simulated and physical robot swarms foraging on clustered, power law, and random resource distributions using error-adapted and non-error-adapted parameters. The GA evolves error-adapted swarms that outperform non-error-adapted swarms in worlds with error. The error-adapted strategies improve efficiency on the clustered and power law distributions: error tolerance (Eq. 4.6) is 14% and 3.6% for simulated robots, and 14% and 6.5% for physical robots (Fig. 4.6). The effect of error-adapted parameters in simulated robots foraging on the clustered distribution was significant ($t(198) = 3.6, p < 0.001$), and the effect for simulated robots on the power law distribution was marginally significant ($t(198) = 1.8, p = 0.07$). Efficiency was not significantly different for simulated or physical robots foraging on randomly distributed resources.

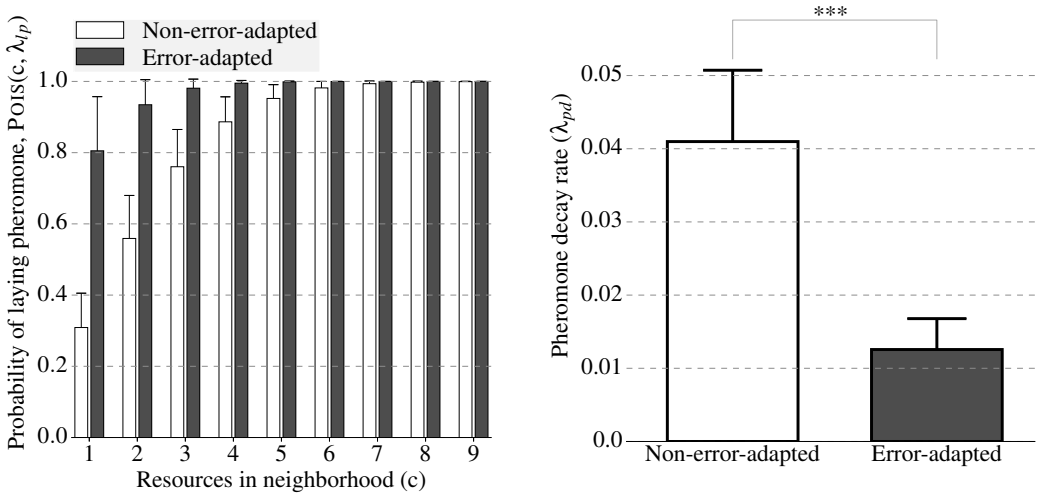


Figure 4.7: For error-adapted and non-error-adapted swarms foraging on clustered resources, (a) the probability of laying pheromone as a function of the count c of resources in the neighborhood of the most recently found resource (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{lp}$), and (b) the pheromone waypoint decay rate (λ_{pd}). Asterisks indicate a statistically significant difference ($p < .001$).

Figure 4.7 compares the probability of laying pheromone (Fig. 4.7(a)) and the rate of pheromone decay (Fig. 4.7(b)) in error-adapted and non-error-adapted swarms foraging for clustered resources. Error-adapted strategies are significantly more likely to use pheromones than non-error-adapted strategies when 4 or fewer resources are detected in the local neighborhood of a found resource (i.e. when $c \leq 4$, see Fig. 4.7(a)). We interpret the increase in pheromone use for small c as a result of sensor error (only 43% of neighboring resources are actually detected by iAnts). The evolved strategy compensates for the decreased detection rate by increasing the probability of laying pheromone when c is small. In other words, given sensor error, a small number of *detected* tags indicates a larger number of *actual* tags in the neighborhood, and the probability of laying pheromone reflects the probable number of tags actually present.

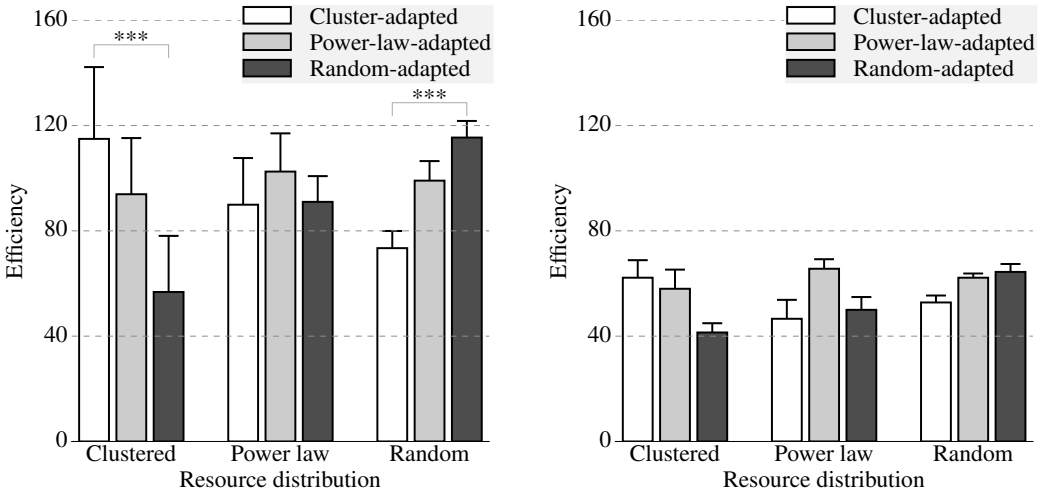
In error-adapted swarms, pheromone waypoints are evolved to decay 3.3 times slower than in swarms evolved without sensor error (Fig. 4.7(b)). Slower pheromone decay compensates for both positional and resource detection error. Robots foraging in worlds with error are less likely to be able to return to a found resource location, as well as being less likely to detect resources once they reach the location, therefore they require additional time to effectively make use of pheromone waypoints.

Sensor error affects the quality of information available to the swarm. These experiments show that including sensor error in the clustered simulations causes the GA to select for pheromones that are laid under more conditions and that last longer. This increased use of pheromones is unlikely to lead to overexploitation of piles because robots will have error in following the pheromones and in detecting resources. Thus, while pheromones can lead to overexploitation of found piles (and too little exploration for new piles) in idealized simulations [77], overexploitation is less of a problem for robots with error.

Figures 4.5–4.7 show that error has a strong detrimental effect on the efficiency of swarms foraging for clustered resources. Swarms foraging on random distributions are only affected by resource detection error; however, the efficiency of cluster-adapted swarms is reduced by both positional and detection error. Generally speaking, different types of error affect different strategies in different ways [56]. In situations where resources are clustered, as is often the case in the real world [135, 25, 136], it is beneficial to adapt to the sensor error experienced by real robots.

4.5.2 Flexibility

Figure 4.8 shows the efficiency of simulated and physical robot swarms evolved on one resource distribution (clustered, power law, or random), then evaluated on all three distributions. All results are for 6 simulated or physical robots foraging with



(a) Simulated (b) Physical

Figure 4.8: Foraging efficiency (resources collected per hour, per swarm) using parameters adapted to different resource distributions for (a) 6 robots foraging in a simulation that includes sensor error and (b) 6 physical robots. Asterisks indicate a statistically significant difference ($p < .001$).

error. As expected, robot swarms evolved for each of the three distributions perform best when evaluated on that distribution. That is, cluster-adapted swarms perform best on the clustered distribution, power-law-adapted swarms perform best on the power law distribution, and random-adapted swarms perform best on the random distribution. Strategy specialization is best illustrated in foraging experiments on the clustered distribution: the cluster-adapted strategies are twice as efficient as the random-adapted strategies.

Figure 4.8 demonstrates that the GA is able to evolve both specialist and generalist strategies. If the resource distribution is known *a priori*, then the robot swarm will be most efficient when using a specialist strategy adapted for that distribution. However, power-law- adapted strategies are sufficiently flexible (Eq. 4.7) to function well on all three distributions. Simulated robot swarms using power-law-adapted parameters are 82% as efficient as cluster-adapted swarms when evaluated

on a clustered distribution, and 86% as efficient as random-adapted swarms when evaluated on a random distribution. The power-law-adapted strategy is also the most flexible strategy for physical robot swarms: power-law-adapted swarms are 93% as efficient as cluster-adapted swarms on a clustered distribution, and 96% as efficient as random-adapted swarms on a random distribution.

While Figure 4.8 demonstrates the expected result that specialist strategies are most efficient, Figure 4.9 illustrates several ways in which strategies are specialized. The four-panel figure shows the probability of exploiting information about resource density in the local neighborhood of a found resource in worlds with error (top) and worlds without error (bottom) by returning to the site via site fidelity (Fig. 4.9(a,c)) or laying pheromone (Fig. 4.9(b,d)). Error-adapted swarms evolved to forage for clustered distributions show large and consistent differences from swarms evolved for power law distributions: they are 3.5 times less likely to return to a site via site fidelity with a single resource in the local neighborhood (Fig. 4.9(a)), and 7.8 times more likely to lay pheromone (Fig. 4.9(b)). Non-error-adapted swarms evolved to forage for clustered distributions are equally likely to return to a site via site fidelity with a single resource in the local neighborhood (Fig. 4.9(c)), but twice as likely to lay pheromone (Fig. 4.9(d)), compared to swarms evolved for power law distributions. In all cases, swarms evolved for random distributions have a significantly lower probability of returning to a site via site fidelity or pheromones.

These results show differences in how each strategy is evolved to use information for different resource distributions, and how these strategies adapt to error by changing how swarms communicate information. Cluster-adapted strategies make frequent use of both memory (site fidelity) and communication (pheromones). Power-law-adapted strategies are nearly equally likely to use memory as cluster-adapted strategies (Fig. 4.9(a,c)), but they are less likely to use pheromones (Fig. 4.9(b,d)). In contrast, swarms foraging on random distributions neither benefit from information,

nor evolve to use it. This result also helps to explain why random-adapted swarms with error experience a relatively small change in fitness (Fig. 4.5(c)): information is irrelevant for random-adapted strategies, therefore error in information has no effect on swarms using these strategies.

The differences among the strategies are most evident when the local resource density estimate c is small: site fidelity and laying pheromones are both effectively absent in random strategies, but they are important components of strategies for clustered distributions. Additionally, it is particularly likely that c will be small in the environment during evaluation when resources are distributed at random. Thus, for clustered distributions, robots are both more likely to lay pheromones for any given c, and more likely to detect large c in the environment, further increasing the probability that pheromones will be laid. This illustrates that the likelihood of a particular behavior being used depends both on the rules that have evolved and on the environment in which it is evaluated.

This point is further illustrated by considering the response to encountering large c: the random strategy evolves a non-zero probability of using site fidelity and laying pheromones when nine resources are discovered. However, the probability of encountering a cluster with nine adjacent resources is vanishingly small in a random resource distribution. Since that condition is never encountered, there is no selective pressure on behaviors under that condition. Thus, the probability of laying pheromone in a random-adapted strategy is effectively zero because the GA evolves zero probability for the cases that are actually encountered.

When interpreting Figure 4.9, it is important to note tradeoffs and interactions among behaviors. If a robot decides to return to a site via site fidelity, it necessarily cannot follow pheromone (Alg. 2, lines 11–16). Thus, the decision to return to a site via site fidelity preempts the decision to follow pheromones, such that the probability of following pheromone is at most $1 - \textsc{Pois}(c, \lambda_{sf})$. However, a robot can both lay a

pheromone to a site (Alg. 2, lines 8–9) and return to that site via site fidelity (Alg. 2, lines 11–13). Furthermore, a robot can return to its own previously discovered site by following its own pheromone. This alternative method of returning to a previously found resource by a robot following its own pheromone may in part explain the lower values of $\textsc{Pois}(c, \lambda_{sf})$ for the error-adapted clustered strategy: $\textsc{Pois}(c, \lambda_{sf})$ may be low because $\textsc{Pois}(c, \lambda_{lp})$ is high (Fig. 4.9(a,b)).

These strategies produced by the GA logically correspond with the resource distribution for which they were evolved. All of the resources in the clustered distribution are grouped into large piles, so finding a single resource is predictive of additional resources nearby. Power-law-adapted swarms are more selective when deciding to share a resource location because robots encounter both large piles and small piles, as well as randomly scattered resources; thus, power-law-adapted swarms have evolved to be more cautious when laying pheromones to avoid recruiting to low-quality resource sites. The power-law-adapted strategies are also the most variable in their use of site fidelity and pheromones, suggesting that many combinations of the two are effective given a distribution with a variety of pile sizes.

4.5.3 Scalability

Figure 4.10 shows the efficiency per robot of simulated and physical swarms with 1, 3, and 6 robots foraging on a power law resource distribution in a world with error. Not surprisingly, we observe that both simulated and physical swarms collect more resources as swarm size increases, however larger swarms are less scalable (Eq. 4.7, where E_1 and E_2 are defined per robot). In simulation, scalability to 3 robots is 89%, while scalability to 6 robots is 79% (Fig. 4.10(a)); in physical experiments, scalability to 3 robots is 68%, while scalability to 6 robots is 56% (Fig. 4.10(b)).

The simulation accurately represents the efficiency of a single robot, but increas-

ingly overestimates swarm efficiency as swarm size increases: 1 simulated robot is 1.1 times more efficient than 1 physical robot, while a simulated swarm of 3 robots is 1.4 times more efficient than a physical swarm of 3, and a simulated swarm of 6 is 1.6 times more efficient than a physical swarm of 6. We hypothesize that this increasing discrepancy is a result of inter-robot interference in the real world that is not captured in the simulation.

Figure 4.11 shows how efficiency per robot changes as swarm size increases from 1 to 768 robots. As in Figure 4.10, there is an increase in overall swarm efficiency, but a decrease in per-robot efficiency, as swarm size scales up. The solid line in Figure 4.11 shows how per-robot foraging efficiency scales when robots forage on a power law distribution (without sensor error) and robots are able to adapt behaviors to swarm size (slope on logged axes $= -0.17$, $R^2 = 0.96$, $p < 0.001$). The scalability (Eq. 4.7) for 768 robots using the full CPFA is 27%. We compare the efficiency of subsets of the full CPFA at different swarm sizes to assess which behaviors contribute most to scalability.

The other three lines in Figure 4.11 show how efficiency scales when swarms are prevented from adapting the full CPFA to the environment in which they are evaluated. The dashed line shows the efficiency of swarms that use a fixed set of parameters evolved for a swarm size of 6 (i.e. parameters are evolved for a swarm size of 6, but evaluated in swarm sizes of 1 to 768). Comparing the solid line to the dashed line shows how adapting to swarm size improves efficiency. The difference in efficiency (Fig. 4.11, solid vs. dashed) increases as swarm size increases. For example, adapting to a swarm size of 24 improves overall swarm efficiency by 4.0%, and adapting to a swarm size of 768 improves swarm efficiency by 51%.

The dash-dotted line shows the efficiency of swarms that adapt to swarm size but are unable to use information (site fidelity and pheromones are disabled so that CPFA parameters λ_{id}, λ_{sf}, λ_{lp}, and λ_{pd} have no effect on robot behavior). By comparing

the efficiency of swarms with and without information (Fig. 4.11, solid vs. dash-dotted), we observe that adapting to use information improves swarm efficiency by an average of 46% across all swarm sizes.

Finally, the dotted line shows swarms that are restricted in both of the ways described above: information use is disabled, and parameters are fixed to those evolved for swarms of size 6. By comparing the dash-dotted line to the dotted line, we can observe how the GA evolves the remaining parameters that govern robot movement (p_r, p_s, and ω) in order to adapt to swarm size. The GA is able to adapt movement to scale up more efficiently: adapting movement parameters to a swarm size of 24 improves swarm efficiency by 6.8%, and adapting movement parameters to a swarm size of 768 improves swarm efficiency by 59%. Thus, parameters governing movement improve efficiency more than parameters governing information use (59% vs. 46%, respectively, for swarms of 768).

The scaling exponents are remarkably similar for swarms under the 4 conditions shown in Figure 4.11 (slopes ranging from -0.14 to -0.21): those that adapt to swarm size, those with behaviors adapted only to a swarm of 6 robots, those that do not use individual memory or pheromone communication, and those with behaviors adapted to a swarm of 6 robots that do not use memory or communication. The cause of these similar exponents is unclear. Central-place foraging produces diminishing returns as swarm size increases because the central nest imposes a constraint on swarm efficiency – robots in larger swarms have to travel farther to collect more resources. However, it is not obvious why that should lead to similar scaling exponents for all four cases. Other researchers have focused on inter-robot interference as the main cause of sub-linear scaling [82, 76], but we observe sub-linear scaling even without including collisions in the simulation.

Figures 4.12(a) and 4.12(b) show two ways in which the GA evolves different strategies for different swarm sizes. Both parameters are drawn from the single best

strategy evolved for each swarm size. Figure 4.12(a) shows that the variation in the uninformed random walk (ω) declines with swarm size. Other movement parameters are also correlated with swarm size: robots in larger swarms use the straight motion of the travel behavior for a longer period of time (i.e. p_s decreases; see Fig. A.1(a)), and they are less likely to give up searching and return to the nest (i.e. p_r decreases; see Fig. A.1(b)). These three trends result in robots in large swarms using more directed motion to disperse farther to cover a larger area and reduce crowding.

Figure 4.12(b) shows how the GA evolves the probability of laying pheromone for different swarm sizes. The probability of laying pheromone decreases with swarm size when two resources are found in the local neighborhood of a found resource (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$). This decreasing trend is observed for all numbers of neighboring resources (this follows from Eq. 4.4). Additionally, pheromone waypoints decay faster as swarm size increases (λ_{pd}) (Fig. A.1(d)). Small swarms may evolve to lay pheromones more often because they deplete piles more slowly than larger swarms. The preference for less pheromone laying and faster pheromone decay in larger swarms may be advantageous to avoid the problem of overshoot in real ant foraging [137], where pheromones can adversely affect foraging rates by recruiting ants to previously depleted food sources.

The two remaining parameters evolved by the GA, the rate of site fidelity (λ_{sf}) and the decay rate of the informed random walk (λ_{id}), show no significant correlation with swarm size (Fig. A.1(f,g)). Figure A.2 shows the full distributions for the parameters of all 10 strategies evolved by the GA in simulation. We see the same trends in the median parameter values as we see in the best parameter values in Figure A.1, but we also observe some outlier strategies that are substantially different from the best performing strategies. For example, an asterisk in Figure A.2(c) corresponds with an outlier strategy which performs at 37% of the efficiency of the best strategy. This particular outlier evolved by converging on an unusually high

rate of pheromone use coupled with ineffective spatial dispersal. Such premature convergence on suboptimal strategies is common in evolutionary computation, but because we repeat the evolutionary process multiple times (see Section 4.4.2), we can evolve a rich variety of interactions among parameters and transfer only the most effective parameter sets into physical robots.

4.6 Discussion

We have described a central-place foraging algorithm (CPFA) whose parameters are evolved by a genetic algorithm (GA) to maximize foraging performance under different experimental conditions. Experiments show that the system successfully evolves parameters appropriate to a wide variety of conditions in simulation, and these lead to successful foraging in iAnt robots. We show that foraging for heterogeneously distributed resources requires more complex strategies than foraging for the randomly distributed resources that have been the focus of previous work. Strategies that automatically tune memory and communication substantially increase performance: Figure 4.8(a) shows that the more complex strategy doubles foraging efficiency for clustered resources compared to a simpler strategy evolved for randomly distributed resources. The same behaviors that allow flexible foraging for different resource distributions can also adapt to tolerate real-world sensing and navigation error (Fig. 4.6) and scale up to large swarm sizes (Fig. 4.11). This system contributes to solving a key challenge in swarm robotics: it automatically selects individual behaviors that result in desired collective swarm foraging performance under a variety of conditions.

The error tolerance, flexibility, and scalability of this system arise from interactions among the set of behaviors specified in the CPFA, and dependencies between those behaviors and features of the environment. These interactions allow a small set of 7 parameters (Table 4.1) to generate a rich diversity of foraging strategies, each

tuned to a particular amount of sensing and navigation error, a particular type of resource distribution, and a particular swarm size. Post-hoc analysis of evolved parameters reveals that pheromone-like communication is one among many important components of the evolved strategies, and interactions among multiple behaviors (i.e., memory, environmental sensing, and movement patterns) are important for generating flexible strategies. Further, the relative importance of pheromone communication varies with sensing and navigation error, resource distribution, and swarm size.

Several examples illustrate how the parameters are automatically adapted to features of specific foraging problems. The power-law-distributed resources are placed in a range of pile sizes, so effective strategies balance the use of random exploration to find scattered resources, individual memory to collect resources from small piles, and recruitment to collect resources from large piles. This balance is altered when the simulations include real-world sensing and navigation error. When error is included, the power law strategy uses less pheromone laying and less site fidelity (Fig. 4.9(a,b). vs. Fig. 4.9(c,d), light gray bars); thus, search automatically becomes more random when information is less reliable due to error. In contrast, the cluster-adapted strategy uses *more* pheromone communication when robots have error: pheromones are laid more often and evaporate more slowly (Fig. 4.7), and robots reduce rates of site fidelity in order to follow pheromones more (Fig. 4.9(a) vs. Fig. 4.9(c), white bars). Sensing and navigation errors have the least effect on foraging performance when resources are distributed at random (Fig. 4.5), and random-adapted strategies are unaffected by error (Fig. 4.9, dark gray bars) because those strategies do not evolve to use information.

Thus, introducing more complex resource distributions reveals effects of sensing and navigation error that are not apparent in simpler foraging problems. Understanding how error affects foraging for heterogeneously distributed resources, and having an automated way to adapt to those effects, are both important given that

landscapes in the real world have complex resource distributions [130, 135], and that robots in the real world have error. Additionally, real-world scenarios will have variable numbers of robots to achieve different tasks. We demonstrate that systematic changes in behaviors are adaptive in larger swarms. We find that power-law-adapted robots in larger swarms evolve to disperse more (Fig. 4.12(a)) and communicate less (Fig. 4.12(b)), and that parameters governing movement have a greater effect on scaling performance than parameters governing communication (59% vs. 46% improvement). Thus, the same parameters that adapt to improve performance for different distributions and error cases can also be automatically tuned to improve performance for variable swarm sizes.

Our approach differs from prior work in that we focus on finding combinations of individual behaviors that result in collective foraging success. We make no attempt to evolve low-level controllers, nor do we attempt to evolve new ways to remember, communicate, or move. We focus the GA on identifying combinations of parameters governing individual behaviors that maximize collective performance. This mirrors the natural evolutionary process that has shaped the successful foraging strategies of different ant species by tuning and combining a common set of existing behaviors. The results show significant performance improvements when parameters are evaluated in the same context in which they are evolved. The success of the evolved foraging strategies demonstrates that this approach is a practical method to generate effective foraging strategies from interactions among foraging behaviors and the specified foraging environment.

Experiments with this swarm robotics system can also test existing biological hypotheses and generate new ones, a potentially important role for robotics as suggested by Webb [134] and Garnier [43]. For example, the balance between communication and memory may shift in ants in response to resource distribution. This could be tested by comparing the typical distribution of resources foraged for by species that

rarely use pheromones [46, 49] to the distribution foraged for by species that use pheromones ubiquitously [3]. Our finding that individual robots in small swarms are more likely to lay pheromones than those in large swarms (Fig. 4.12(b)) conflicts with the hypothesis by Beckers et al [10] that communication increases with colony size. One potential explanation is that large colonies tend to forage for more clustered distributions, a factor not accounted for in our simulations in which all swarms foraged on a power law distribution. Thus, the relationship between colony size and pheromones may be driven by environmental differences in the niches of large and small colonies. How communication among individuals depends on colony size and resource distribution is worthy of further study in real ants, as well as in swarm robotics. More generally, our system provides a way to test how memory, communication, and movement interact in different foraging conditions with experimental control that is not possible with ants in natural environments.

4.7 Conclusions

This paper presents an ant-inspired swarm robotics system whose parameters are specified by a GA. The GA automatically selects individual behaviors that result in desired collective swarm foraging performance under a variety of conditions. This work emphasizes the importance of incorporating environmental conditions into the design process at the outset, rather than assuming idealized conditions and adapting them to environmental realities afterwards. It is the interactions with features of the specified foraging problem during the evolutionary process that generate complex and flexible behaviors. Foraging strategies emerge from the interactions among rules and dependencies in the foraging environment, including the amount of error in robot sensing and navigation, the complexity of the resource distribution, and the size of the swarm.

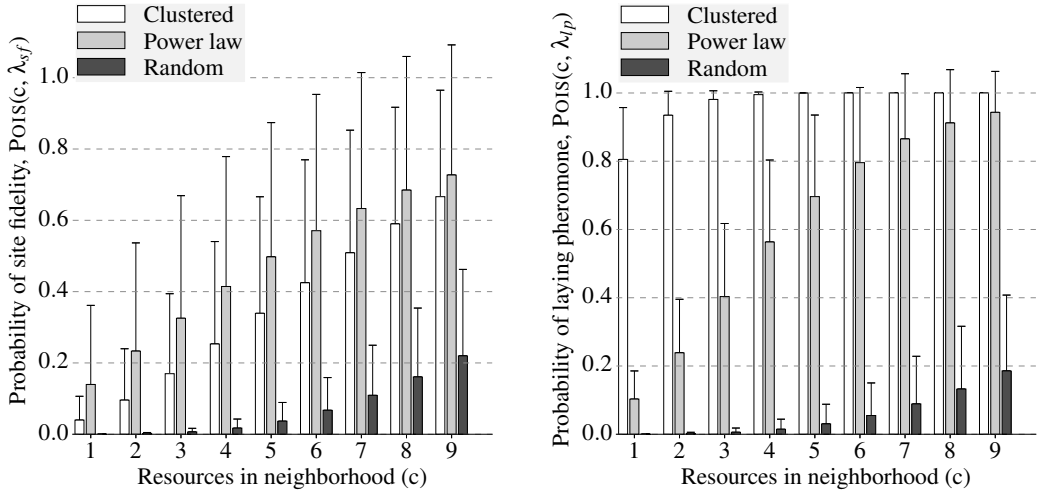
Our work demonstrates one approach toward the common goal of developing robot swarms that can function in the varied and complex conditions of the real world. Of course, real environments are vastly more complex than the conditions we have considered here. Future work should test whether and how a GA can adapt the CPFA to more complex environments, additional sources of robot error, and larger physical robot swarms. This work also provides a foundation for automatically evolving behaviors that interact with environmental conditions to accomplish other collective tasks, for incorporating other ant behaviors, and for adapting behavioral rules in response to sensed environmental conditions in real time. By demonstrating how a rich set of strategies can evolve from simple behaviors interacting with complex environments, we suggest that biologically-inspired swarm robotics can benefit from leveraging a larger set of biological behaviors to accomplish complex real-world tasks.

4.8 Acknowledgements

We gratefully acknowledge members of the Biological Computation Lab: Karl Stolleis, Bjorn Swenson, Justin Carmichael, Kenneth Letendre, and Neal Holtschulte, for their assistance with the iAnt swarm robotics project. We also thank Deborah Gordon for illuminating discussions about seed-harvesting ant behaviors, Alan Winfield for discussions about swarm foraging problems, and Lydia Tapia and Stephanie Forrest for constructive feedback on this manuscript. This work is supported by NSF grant EF-1038682, DARPA CRASH grant P-1070-113237, and a James S. McDonnell Foundation Complex Systems Scholar Award.

Algorithm 2 Central-Place Foraging Algorithm

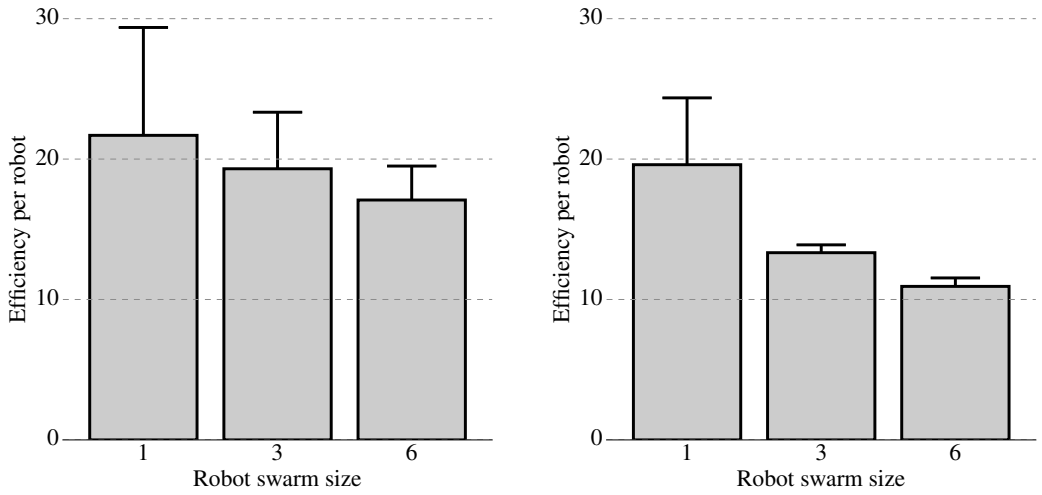
```
1: Disperse from nest to random location
2: while experiment running do
3:     Conduct uninformed correlated random walk
4:     if resource found then
5:         Collect resource
6:         Count number of resources c near current location l_f
7:         Return to nest with resource
8:         if POIS(c, λ_lp) > U(0, 1) then
9:             Lay pheromone to l_f
10:        end if
11:        if POIS(c, λ_sf) > U(0, 1) then
12:            Return to l_f
13:            Conduct informed correlated random walk
14:        else if pheromone found then
15:            Travel to pheromone location l_p
16:            Conduct informed correlated random walk
17:        else
18:            Choose new random location
19:        end if
20:    end if
21:    end while
```



(a) Error-adapted (b) Error-adapted

(c) Non-error-adapted (d) Non-error-adapted

Figure 4.9: For error-adapted swarms (top) and non-error-adapted swarms (bottom), (a,c) the probability of returning to a site (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{sf}$) and (b,d) the probability of laying pheromone (Eq. 4.4: $k \leftarrow c$, $\lambda \leftarrow \lambda_{lp}$) given the number of resources c in the neighborhood of a found resource.



(a) Simulated (b) Physical

Figure 4.10: Foraging efficiency (resources collected per hour, per robot) of 1, 3, and 6 robots foraging on a power law distribution for (a) swarms in a simulation that includes sensor error and (b) physical swarms. All results are statistically different ($p < 0.001$).

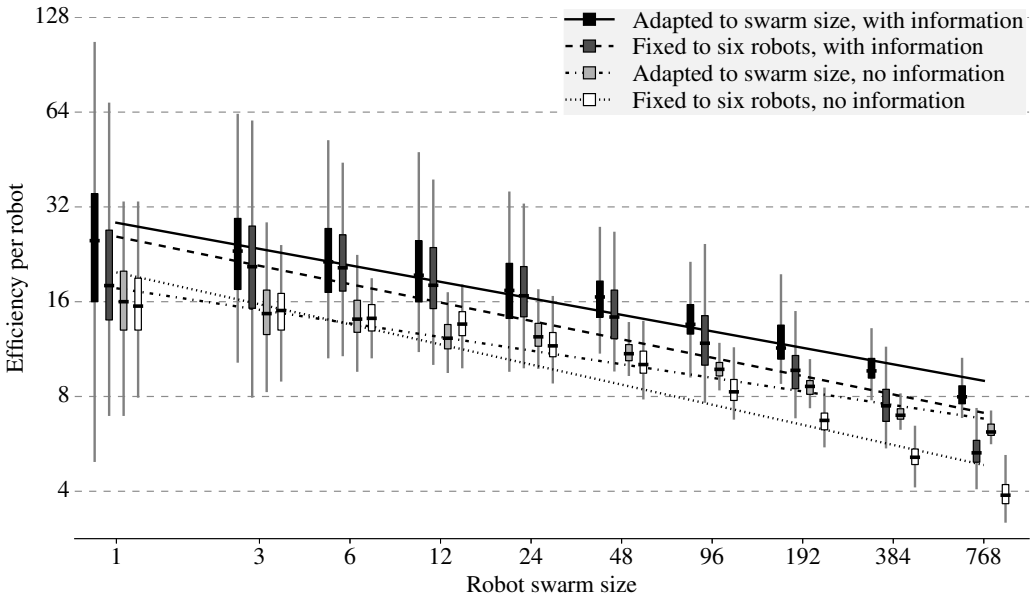
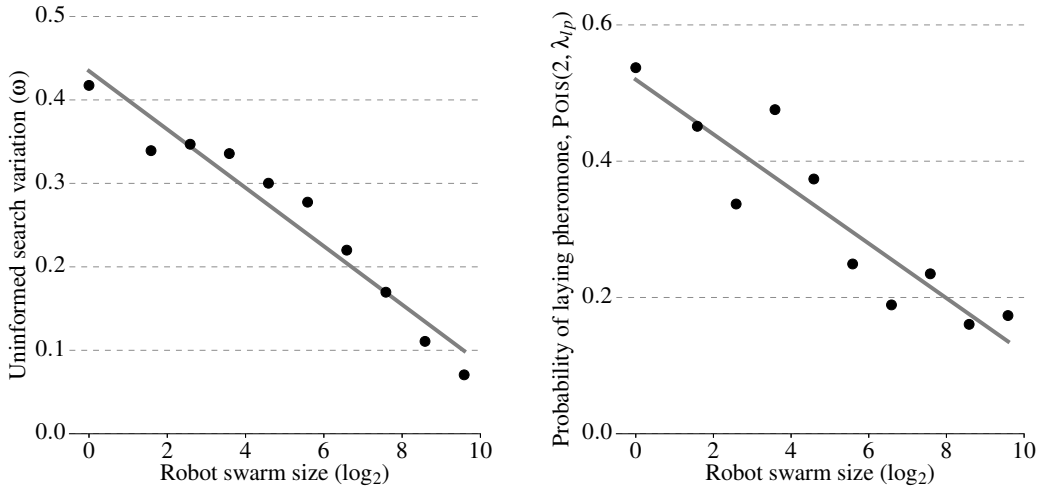


Figure 4.11: Foraging efficiency (resources collected per hour, per robot) in simulated swarms of 1 to 768 robots foraging without sensor error. Data are shown on a log scale, and linear regression lines are shown for log-transformed data. Per-robot efficiency is shown for four cases: using the full CPFA parameter set adapted to swarm size (slope $= -0.17$, $R^2 = 0.96$), using the full CPFA with parameters fixed to values evolved for a swarm size of 6 (slope $= -0.19$, $R^2 = 0.83$), using parameters adapted to swarm size without information (i.e. the CPFA without memory and communication; slope $= -0.14$, $R^2 = 0.95$), and using parameters fixed to values evolved for a swarm size of 6 without information (slope $= -0.21$, $R^2 = 0.91$). All linear fits are statistically significant ($p < 0.001$).



(a) (b)

Figure 4.12: (a) Swarm size versus best evolved uninformed search variation (ω) (slope = -0.035, $R^2 = 0.94, p < 0.001$) (see Fig. A.2 for statistical distribution). (b) Swarm size versus best evolved probability of laying pheromone when two resources are found in the resource neighborhood (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$) (slope = -0.040, $R^2 = 0.84, p < 0.001$) (see Fig. A.2).

Chapter 5

Exploiting Clusters for Complete Resource Collection in Biologically-Inspired Robot Swarms

5.1 Abstract

The complete collection of resources from a predefined search area is a challenging task for autonomous robot swarms. Because naturally-occurring resources are likely to be distributed in clusters, foraging robot swarms can identify and exploit these resource clusters to improve collection efficiency. We describe an ant-inspired robot swarm foraging system that searches for and collects resources from a variety of distributions, and a cluster prediction and exploitation algorithm that augments swarm foraging by directing robots to residual resources. By characterizing the cumulative resource collection time for a robot swarm foraging in a variety of clustered resource

distributions, we can identify the relationship between the "clusteredness" of the distribution and the change in the resource collection rate over time. Experiments show that collection efficiency is most significantly increased when robots switch from ant-inspired foraging to focused exploitation of clusters after approximately 90% of resources have been collected. Not surprisingly, clustering algorithms are most effective when resources are highly clustered in the environment. This work demonstrates the feasibility of efficient, complete resource collection using simple, range-limited robot swarms programmed with ant-inspired foraging behaviors.

5.2 Introduction

Robot swarms are appealing because they can be made from inexpensive components, their decentralized design is well-suited to tasks that are distributed in space, and they are potentially robust to communication errors that could render centralized approaches useless. Central-place foraging is a canonical task for robot swarms, and can be instantiated into a number of real-world resource collection tasks, including hazardous waste clean-up [107], humanitarian demining [42, 71], and in-situ resource utilization [27, 129]. Total, or complete, collection of all resources is a challenging task, and one that may prove effectively intractable when robots are placed in complex environments with unknown and variable resource distributions. Additionally, naturally-occurring resource distributions are likely to be spatially heterogeneous [135, 25, 136], with some large clusters of resources that are relatively rare and thus difficult to find. However, foraging robot swarms that are able to identify and remember the locations of clusters can exploit them to improve complete collection efficiency.

The central-place foraging algorithm (CPFA) emulates ant behaviors which govern memory, communication, and movement, as well as an evolutionary process that

tailors those behaviors into foraging strategies that maximize error-tolerance, flexibility, and scalability under varied and complex environmental conditions [58]. These robot swarms use information to direct their search to clustered resources – individual robots are more likely to remember or communicate the locations of resources that are found in dense clusters. Swarms that are evolved for a particular environmental condition employ foraging strategies that integrate the movement patterns and communication strategy appropriate for that condition. In prior work, the evolved swarms foraged efficiently by maximizing resource collection during a 1 hour experimental window, but resource intake rates tend to drop sharply when only a small, sparsely distributed fraction of residual resources remain.

This work presents a novel extension to robot swarm foraging that mitigates the diminishing returns encountered during the complete collection task. Simulated robot swarms combine machine learning with statistical models to predict the location, size, and number of clusters remaining in the residual distribution after some fraction of resources have been collected. Swarms exploit clusters by switching from foraging with the CPFA to searching the cluster locations predicted by the statistical model. Robot swarms that predict and exploit resource cluster locations, particularly those with few remaining resources, significantly reduce diminishing returns when foraging in highly clustered distributions.

5.3 CPFA Background

The CPFA and its biological roots are described in detail in our previous work [58]. Here we summarize the essential features that the current work builds upon. The CPFA mimics foraging behaviors used by desert seed-harvester ants that have adapted to forage under hot, dry conditions. We emulate harvester ant foraging strategies, which have evolved to forage in short time windows during which not all

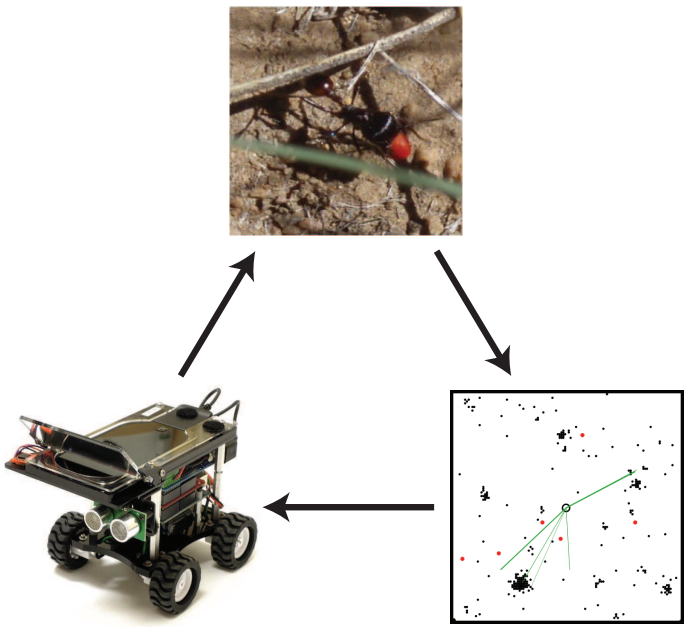


Figure 5.1: Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform.

available resources can be collected [50]. Foragers initially disperse from their central nest in a travel phase, followed by a search phase [34] in which a correlated random walk is used to locate seeds [26]. Foragers then navigate home to a remembered nest location [61]. Seed-harvester ants typically transport one seed at a time, sometimes sampling other seeds in the neighborhood of the discovered seed [61] to estimate local seed density [77]. Ants that detect high seed density are more likely to return to previously found food patches using individual memory or pheromone recruitment. When foragers return to a patch, they appear to alter their search behavior such that they initially search the local area thoroughly, but eventually disperse to search more distant locations [35].

We instantiated ant foraging behaviors in an algorithm (the CPFA) that governs simulated and physical iAnt robot swarms [55, 56, 59, 57] (Fig. 5.1). We demonstrated a close correspondence between the behaviors of simulated and physi-

cal robots; consequently, in this paper, we conduct experiments only with simulated robot swarms. Most recently, we demonstrated that the GA is able to evolve foraging strategies that are tolerant of real-world sensing and navigation error, flexible for a variety of resource distributions, and scalable to large swarm sizes [58]. The swarm foraging system evolved appropriate solutions to different environmental challenges. Solutions included: *i*) increased communication when sensed information was reliable and resources to be collected were highly clustered, *ii*) less communication and more individual memory when cluster sizes were variable, and *iii*) greater dispersal with increasing swarm size. Analysis of the evolved behaviors reveals the importance of interactions among behaviors, and of the interdependencies between behaviors and environments. The effectiveness of interacting behaviors depends on the uncertainty of sensed information, the resource distribution, and the swarm size.

Here we extend our previous results by *i*) characterizing the time for robot swarms to completely collect all available resources from a range of resource distributions, *ii*) predicting the location, size, and number of resource clusters in each distribution from a partial list of resource locations, and *iii*) identifying the most effective point in time to switch from ant-inspired foraging to exploitation of cluster locations.

5.4 Methods

5.4.1 Central-Place Foraging and Evolutionary Algorithms

The CPFA implements robot foraging behaviors as a series of states connected by directed edges with transition probabilities (Fig. 5.2). Each robot begins its search at a central nest site and sets a search location. Robots traveling to a random location with no prior information search using an uniformed correlated random walk. Robots traveling to a previously found resource location search using an informed

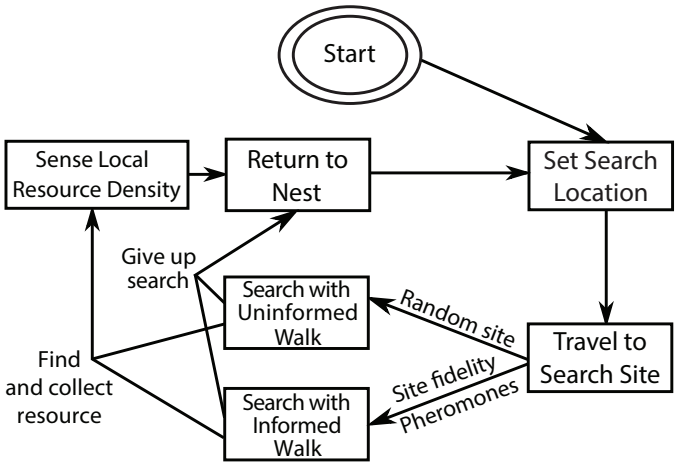


Figure 5.2: The central-place foraging algorithm, or CPFA, describes the flow of behavior for individual robots in the swarm during the foraging task (reprinted from our prior work [58]).

random walk that is initially undirected and localized, then becomes more directed and straighter over time. When a robot locates a resource, it first collects the resource, and then records a count of resources in the neighborhood of the found resource. Robots use this count to decide whether to exploit information through memory or communication. Robots who have not found a resource will probabilistically return to the nest.

We use a genetic algorithm (GA) to evolve a population of CPFA parameters that maximizes the foraging efficiency of simulated robot swarms evaluated in an agent-based model. These parameters control the sensitivity threshold for triggering CPFA behaviors, the likelihood of transitioning from one behavior to another, and the length of time each behavior should last.

5.4.2 Predicting Optimal Clusters

Expectation-maximization (EM) is a procedure for finding the means $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$ for the components of a Gaussian mixture model $\boldsymbol{\theta}$ using maximum likelihood estimation (MLE). We predict the most likely k-component mixture model $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by training EM on a given set of observed data $\boldsymbol{x}$. The predicted values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ identify the locations and sizes of clusters in the observed data $\boldsymbol{x}$

EM requires the number of mixture model components k to be determined *a priori*, which presents a challenge when training on partially observed data with an unknown number of clusters. We predict the optimal mixture model OptimalEM (Algorithm 3) for a set of discovered two-dimensional resource locations $\boldsymbol{x} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where N is the number of discovered resources. For each model EM$(\boldsymbol{x}, k)$, where $k > 0$, we calculate the maximum likelihood L and the number of free parameters f. We then calculate the Bayesian Information Criteria (BIC) for the model as:

$$\text{BIC} = 2\log(L) - f\log(N) \tag{5.1}$$

The number of free parameters for a two-dimensional Gaussian mixture model is defined as $f = 4k - 1$ [52]. The BIC evaluates the relative quality of the model by rewarding for goodness of fit, which is predicted by the maximum likelihood L, but penalizing for the complexity of the predicted model defined by the number of model components k. Comparing BIC values for competing mixture models trained on a shared set of input data is a common method for selecting the most representative model and estimating the number of clusters [37].

The OptimalEM function calculates the finite difference $\nabla BIC(k) = BIC(k) - BIC(k-1)$ between the BIC values of trained EM models for an increasing number

of clusters k. OptimalEM initializes k to 0, then increases k until $\nabla BIC(k) \geq 0$, at which point the function returns the predicted mixture model $\text{EM}(\boldsymbol{x}, k-1)$.

5.4.3 Exploiting Clusters

Robot swarms begin each experiment by foraging for resources using the CPFA (Fig. 5.2). After some fraction of resources N have been collected by the swarm, all robots return to the nest and compile the locations of the N resources into a single list $\boldsymbol{x}$. Note that this compilation process requires a one-time communication cost that scales linearly with the number of robots in the swarm. The list $\boldsymbol{x}$ containing the (x, y) positions of the discovered resources is divided, according to the function $\text{OptimalEM}(\boldsymbol{x}, N)$, into a k-component Gaussian mixture model $\boldsymbol{\theta}$ with means $\boldsymbol{\mu}$ and diagonal covariance matrices $\boldsymbol{\Sigma}$. Sampling each Gaussian component θ_i at one standard deviation of the mean μ_i produces a two-dimensional ellipse. The rectangular bounding box containing the ellipse is defined as a *cluster region* c_i

Algorithm 3 Select Optimal Clustering

```
1: function OptimalEM(x, N)
2:     k ← 0
3:     BIC[0] ← MAXFLT                    ▷ Initialize to maximum float
4:     do
5:         k ← k + 1
6:         EM[k] ← EM(x, k)                                  ▷ Train EM
7:         L ← EM[k].L                          ▷ Retrieve likelihood value
8:         BIC[k] ← BIC(L, k, N)                             ▷ Calculate BIC
9:     while BIC[k] − BIC[k − 1] < 0
10:    return EM[k − 1]
11: end function
```

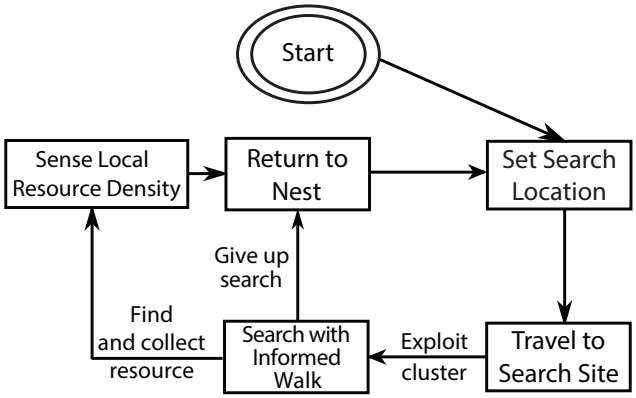
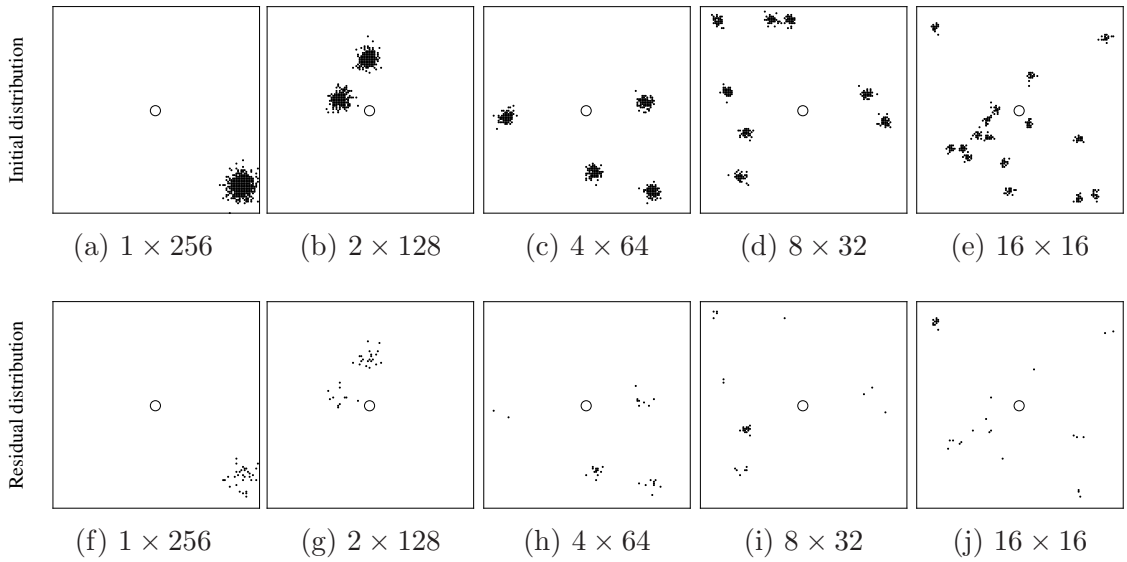


Figure 5.3: The central-place cluster exploitation algorithm, or CPCEA, describes the flow of behavior for robots after switching from foraging with the CPFA to exploiting the clustered regions predicted by OptimalEM (Algorithm 3).

with width $\Sigma_{i,0,0}$, height $\Sigma_{i,1,1}$, and center μ_i. Individual robots are then randomly assigned to search each of the clusters regions predicted by OptimalEM$(\boldsymbol{x}, N)$.

Each robot makes a behavioral switch from foraging using the CPFA to exploiting the clustered regions defined by the mixture model θ. After this behavioral switch, all robots follow the central-place cluster exploitation algorithm (CPCEA) shown in Figure 5.3. A robot at the nest starts its search by first choosing a cluster region c with random probability, then selecting an (x, y) point at random from within region c as its search location. The robot travels to the search location and begins searching, using an informed random walk to search locally where it expects to find a resource (i.e. within the predicted cluster region), but to straighten its path and disperse to another location if the resource is not found. Robots that have not found a resource will probabilistically give up searching and return to the nest. If a robot locates and collects a resource, it records a count of resources in the immediate neighborhood of the found resource, then returns to the nest.



(a) 1×256 (b) 2×128 (c) 4×64 (d) 8×32 (e) 16×16

(f) 1×256 (g) 2×128 (h) 4×64 (i) 8×32 (j) 16×16

Figure 5.4: Top: 256 resources are placed one of five different clustered distributions: (a) 1×256 has 1 randomly placed pile of 256 resources, (b) 2×128 has 2 piles of 128 resources, (c) 4×64 has 4 piles of 64 resources, (d) 8×32 has 8 piles of 32 resources, and (e) 16×16 has 16 piles of 16 resources. Bottom: (f)–(j) The residual resource distribution after 224 resources have been collected.

5.4.4 Experimental Setup and Evaluation

Swarms of 6 simulated robot agents search for resources on a 125×125 cellular grid; each cell simulates an 8×8 cm square. The simulation architecture replicates the physical dimensions of our real robots, their speed while traveling and searching, and the area over which they can detect resources. The spatial dimensions of the grid reflect the distribution of resources over a 100 m^2 physical area. 256 resources are placed on the grid (each resource occupies a single grid cell) in one of five clustered distributions (see Fig. 5.4(a)–(e)): 1×256 (1 randomly placed cluster of 256 resources, 2×128 (2 clusters of 128), 4×64 (4 clusters of 64), 8×32 (8 clusters of 32), and 16×16 (16 clusters of 16).

Following the evolutionary methods introduced in our previous work [58], we use

a GA to generate robot swarm foraging strategies, represented by CPFA parameter sets, that maximize resource collection efficiency for each of the five distributions. The GA evaluates the fitness of each strategy, where fitness is defined as the total number of resources collected by a swarm foraging with the CPFA during a simulated 1 hour time period. The evolution process requires 600,000 simulated hours to generate a foraging strategy for each distribution.

A population of 100 simulated robot swarms evolve for 50 generations using recombination and mutation. Each swarm's strategy is randomly initialized, and robots within a swarm use identical parameters throughout each hour-long experiment. During each generation, all 100 swarms undergo 12 fitness evaluations, each with a different random placement of clusters; Figure 5.4(a)–(e) shows representative sample placements. Deterministic tournament selection with replacement (tournament size = 2) is used to select 99 candidate swarm pairs, then each pair is recombined using uniform crossover and 10% Gaussian mutation with fixed standard deviation (0.05). We use elitism to copy the swarm with the highest fitness, unaltered, to the new population – the resulting 100 swarms make up the next generation. After 50 generations, the strategy with highest fitness is kept as the best foraging strategy. We repeat the evolutionary process 10 times to generate 10 independently evolved foraging strategies, then we evaluate the best of these 10 foraging strategies for each distribution on the complete resource collection task, for robot swarms foraging with and without the use of the OptimalEM function (Alg. 3) and the CPCEA (Fig. 5.3).

We define *collection time* as the amount of time required to collect some fraction of resources from one of the five experimental distributions, and *total collection time* as the total time required to collect all 256 resources. We define *relative change in efficiency* as:

$$\Delta E_r = \left| \frac{T_1 - T_2}{T_2} \right| \times 100\% \tag{5.2}$$

where T_1 is the total collection time for a robot swarm foraging only with the CPFA, and T_2 is the total collection time for a swarm that switches from the CPFA to the CPCEA after some fraction of resources have been collected.

We also evaluate the accuracy of our cluster prediction algorithm by comparing the number of clusters predicted by OptimalEM to the actual number of clusters discovered by robot swarms. We define *mean absolute error* in cluster prediction over n experimental replicates as:

$$\varepsilon_c = \frac{1}{n} \sum_{i=1}^{n} |k_i - c_i| \tag{5.3}$$

where k_i is the predicted number of clusters, and c_i is the actual number of clusters, for replicate i.

Finally, we measure the effect of switching from foraging with the CPFA to exploiting clusters after different numbers of resources have been collected: we test how switching to cluster exploitation after collecting N resources affects ΔE_r, where $N = 32i$ and $i \in \{1, 2, \ldots, 7\}$. That is, we measure the total collection time for robot swarms that switch to exploiting clusters after 32, 64, ..., 224 resources have been collected. In order to thoroughly explore the performance of ΔE_r when only a small fraction of resources remain, we also exhaustively test all integer values for N between 225 and 256. We define the *effect size* of switching to exploiting clustering as:

$$r = \frac{z}{\sqrt{M}} \tag{5.4}$$

according to Richler et al [41], where z is the z-score from the Mann-Whitney U test comparing the total collection time for swarms foraging only with the CPFA to the total collection time for swarms that switch from foraging with the CPFA to exploiting clusters, and M is the total number of samples. We also define the *degree of dispersion* in the total collection time for each swarm as:

$$d = median_i(|y_i - median_j(y_j)|) \tag{5.5}$$

where d is the median absolute deviation for a data set $\boldsymbol{y} = \{y_1, y_2, \ldots, y_M\}$ containing M samples.

5.5 Results

Results below compare resource collection times and cluster prediction errors across five resource distributions that vary in the number and size of clusters (Fig. 5.4(a)–(e)). All experiments are replicated 1000 times; medians and quartiles represent the distribution of resource collection times, means and standard deviations summarize cluster prediction error (Eq. 5.3), and the Mann-Whitney U test measures the effect of switching from the CPFA to the CPCEA.

Figure 5.5 shows the median cumulative collection time for robot swarms using only the CPFA to forage for each of the five clustered resource distributions. Swarms foraging on the 1×256 distribution collect 88% of the total resources in the shortest amount of time (224 resources collected in approximately 6,600 timesteps). Increasing the number of clusters (while simultaneously decreasing the resources per cluster) produces a corresponding increase in collection time for the first 224 resources: swarms foraging on the 16×16 distribution collect 88% of resources in

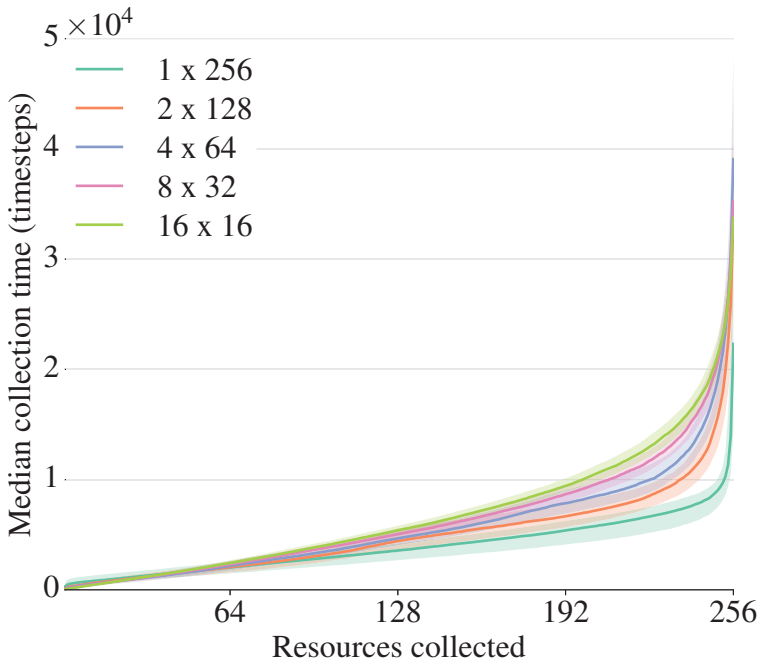


Figure 5.5: Median cumulative collection time (in timesteps) for 1 to 256 resources in each of five resource distributions.

approximately 13,000 timesteps, nearly twice the collection time required for the 1×256 distribution. In all five distributions, the time required to collect the first 224 resources (between 6,600 and 11,000 timesteps) is approximately half of the time required to collect the last 32 resources (between 16,000 and 29,000 timesteps). In other words, swarms foraging only with the CPFA spend between 63% and 75% of their time collecting the last 12% of the total resources in each distribution.

Figure 5.6 shows the mean absolute error of the OptimalEM cluster prediction algorithm (ε_c, Eq. 5.3) for each of the five resource distributions with an increasing number of resources collected before switching to exploiting clusters. Prediction error decreases monotonically with time for four of the five distributions; i.e., as expected, predication error decreases when predictions are made after robot swarms have sampled more of their environment. Error for the 16×16 distribution initially

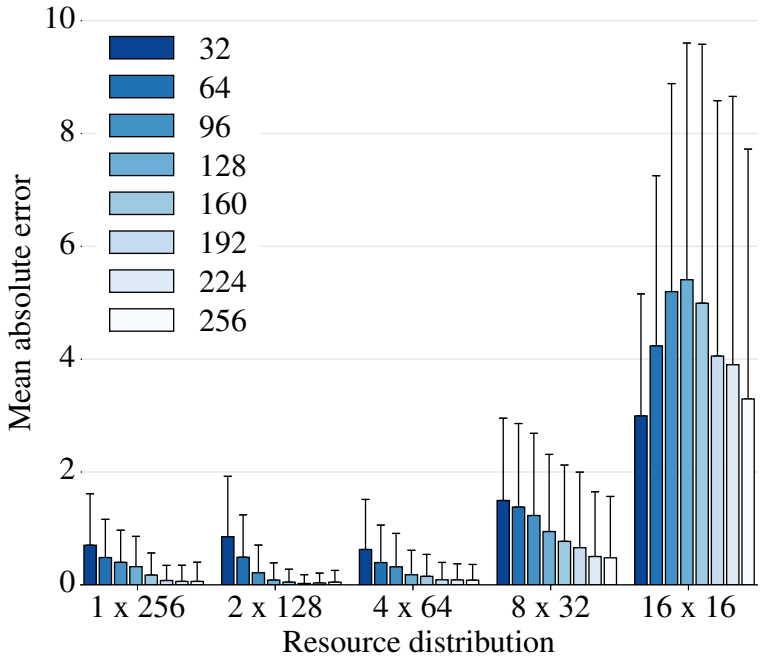


Figure 5.6: Mean absolute error ε_c (Eq. 5.3) when predicting the number of clusters in each of five resource distributions, with increasing numbers of resources collected before switching to cluster exploitation.

increases, then decreases after 160 resources have been collected. We also observe that prediction error is larger for large numbers of clusters.

Figure 5.7 shows the effect r (Eq. 5.4) for robot swarms that switch from foraging with the CPFA to exploiting clusters after collecting the indicated number of resources (on the x-axis), compared to swarms that do not switch and forage using only the CPFA (Eq. 5.4). A positive effect indicates that the swarms that switch to exploiting clusters outperform the swarm that do not switch, while a negative effect indicates the reverse. We observe that r generally increases as more resources are collected before switching to cluster exploitation. That is, swarms that collect more resources before clustering those resources, then exploiting the resulting clusters, perform better than swarms that collect fewer resources before clustering and exploiting. However, in four of the five distributions, change in effect size tends to

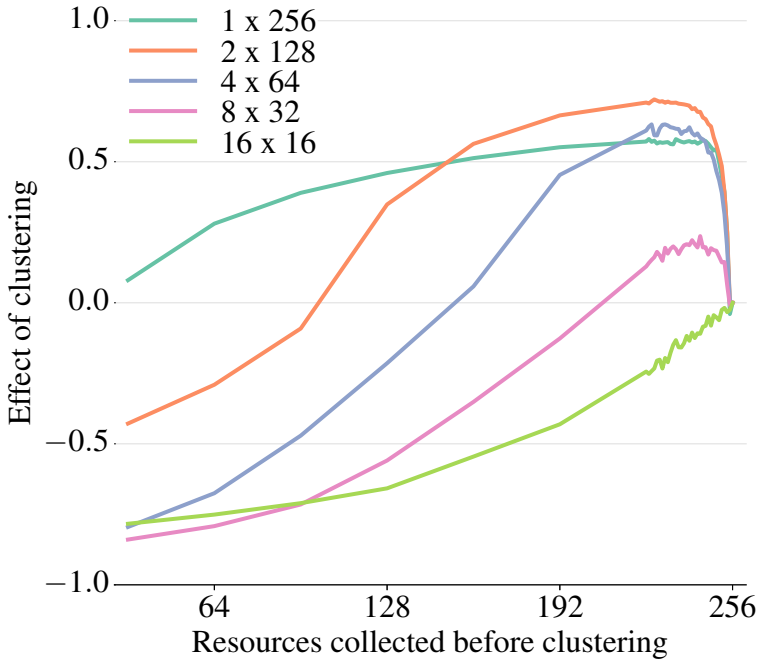


Figure 5.7: The effect r (Eq. 5.4) of clustering for robot swarms that switch from the CPFA to the CPCEA after some fraction of resources have been collected, compared to swarms that do not switch and forage only with the CPFA.

plateau after approximately 90% of resources have been collected. Swarms foraging on the 16×16 distribution that switch to clustering never outperform CPFA-only swarms, regardless of the number of resources collected before switching. Additionally, the effect for a given number of resources collected before switching generally decreases as the number of clusters increases. For example, swarms foraging on the 1×256 distribution that switch to exploiting clusters consistently outperform swarms foraging using only the CPFA, whereas swarms that switch to exploiting clusters on the 16×16 distribution perform consistently worse than CPFA-only swarms. In all five resources distributions, we observe the largest effects of exploiting clusters for swarms that switch after at least 224 resources have been collected, indicating that the majority of resources should be collected before switching to clustering.

Figure 5.8 shows the distribution of total collection times for robot swarms forag-

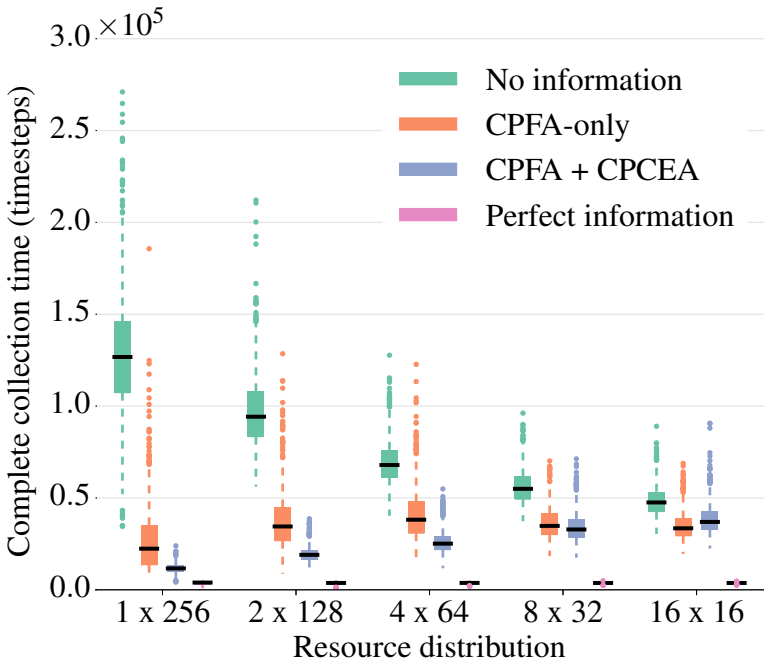


Figure 5.8: Total collection time for robot swarms foraging without information using a correlated random walk, for swarms foraging only with the CPFA, for swarms that switch from the CPFA to the CPCEA after 224 resources have been collected, and for idealized swarms with perfect information (all resource locations are known *a priori*).

ing only with the CPFA, and for swarms that switch from foraging with the CPFA to exploiting clusters after 224 resources have been collected. As a comparative benchmark, we also include the distribution of total collection times for swarms that forage without information using a correlated random walk (robots do not use memory or communication), and for idealized swarms that forage with perfect information (all resource locations are known *a priori*, therefore search is not required). Swarms foraging with the CPFA on the 1×256 distribution are 5.7 times more efficient than swarms that forage without information, and 82% less efficient than idealized swarms with perfect information. Swarms that switch to exploiting clusters with the CPCEA are 11 times more efficient than swarms that forage without information, and only 66% less efficient than swarms with perfect information. The advantage of foraging

using either the CPFA or the CPCEA decreases as the number of clusters increases and the distribution of resources become more random.

We observe that clustering and exploiting after 224 resources are collected reduces the total collection time required for the CPFA on four of the five distributions, increasing efficiency (ΔE_r, Eq. 5.2) by 92% for the 1×256 distribution ($z = 26$), 81% for 2×128 ($z = 32$), 52% for 4×64 ($z = 27$), and 6% for 8×32 ($z = 5.8$). The increase in efficiency was statistically significant in all four cases ($p < 0.001$). Additionally, the dispersion d (Eq. 5.5) of the total collection time decreases when swarms switch to exploiting clusters in four of the five distributions. We observe 6.1 times less dispersion for the 1×256 distribution, 3.5 times less for 2×128, 2.4 times less for 4×64, and 13% less for 8×32. This result demonstrates how cluster exploitation significantly reduces the variation in total collection time; for example, CPFA-only swarms foraging on the 1×256 distribution have a worse-case collection time of 26 simulated hours, compared 3.3 hours for swarms that switch to cluster exploitation.

5.6 Discussion

We have described a novel extension to our robot swarm central-place foraging algorithm (CPFA) that mitigates the diminishing returns encountered during the complete resource collection task. Experiments show collection efficiency is most significantly increased when robot swarms switch from ant-inspired foraging to directly exploiting clusters after approximately 90% of resources have been collected. As expected, cluster exploitation is most effective when resources are highly clustered in the environment.

The effect of clustering tends to level off after robot swarms collect more than 90% of resources, and eventually decreases to zero as all 256 resources are collected

(Fig. 5.7). We attribute this result to the diminishing returns of cluster exploitation relative to the performance of ant-inspired foraging: as expected, swarms that delay the use of clustering until very few resources remain experience a less significant gain from cluster exploitation. However, swarms actually perform better when they delay cluster exploitation until approximately 90% of resources are collected. Until that point, the CPFA is a more efficient foraging algorithm than cluster exploitation, which demonstrates the value of ant-inspired foraging. In summary, the resource location information gathered by foragers while collecting the first 90% of resources is vital to cluster exploiters collecting the last 10% of resources. This calls for a more thorough investigation of the interactions and interdependencies between exploration and exploitation in foraging robot swarms.

Interestingly, accurate estimation of the number of clusters is not necessarily predictive of efficient resource collection. For example, increasing and subsequently decreasing prediction error with increasing numbers of resources for the 16×16 distribution (Fig. 5.6) does not match the monotonically increasing effect of switching for the same distribution (Fig. 5.7). This result is likely indicative of latent cluster exploitation mechanisms that are not accounted for in the error measure ε_c (Eq. 5.3). Additionally, we hypothesize that decreasing neighborhood resource density may help to explain the corresponding decrease in resource collection rate for CPFA-only swarms (Fig. 5.5): sparsely distributed residual resources (Fig. 5.4(f)-(j)) tend not to trigger memory or communication behaviors crucial to ant-inspired foraging. Future work should implement mathematical models to formally specify the relationship between the number, size, and density of resource clusters, and the optimal point for switching to cluster exploitation.

Our work demonstrates a novel approach for augmenting ant-inspired robot swarm foraging with machine learning and statistical models. We show that combining our existing, biologically-inspired CPFA with a cluster exploitation algorithm

produces more efficient total resource collection compared to each algorithm acting alone. More generally, the results of this study support the efficacy of augmenting biologically-inspired methods with machine learning algorithms to generate new, robust techniques supported by solid mathematical foundations.

5.7 Acknowledgements

We gratefully acknowledge Lydia Tapia and Trilce Estrada for constructive suggestions on clustering techniques. This work is supported by NSF grant EF-1038682, DARPA CRASH grant P-1070-113237, and a James S. McDonnell Foundation Complex Systems Scholar Award.

Chapter 6

Concluding Remarks

As human beings, we are driven to explore distant new worlds as we seek to understand the fundamental nature of the Universe, and to unravel the mysteries of our own existence. Because of the inherent dangers of human spaceflight, modern extraplanetary exploration missions often use telerobotic, partially autonomous rovers in place of humans to investigate these remote worlds [138]. As the technology behind these machines continues to advance, the conceptual ambitions of researchers and scientists to design and build low-cost, durable, fully autonomous rovers are being realized [27, 121, 127]. Emerging research in biologically-inspired robotics suggests that swarms of inexpensive, robust, quintessentially autonomous robots are destined to surpass today's Mars rovers as the extraplanetary explorers of the future [55, 59, 58, 54].

With this contribution, we aim to advance the ambitious goal of designing and programming robots that can successfully navigate unknown and variable environments, such as extraplanetary surfaces. Accordingly, Chapters 2 through 5 describe in detail our swarm robotics system controlled by a central-place foraging algorithm (CPFA) whose parameters are evolved by a genetic algorithm (GA) to maximize for-

aging performance under different experimental conditions. This work demonstrates how robot swarms built from inexpensive components can successfully operate outside of the typical robot laboratory setting, employing evolutionary computation and machine learning to mitigate the adverse effects of unreliable information, variable environments, congestion bottlenecks, and sparse resources.

In Chapter 2, we translated a GA-evolved agent-based model of seed-harvester ants into a foraging algorithm for tag-seeking robot swarms. We investigated the benefits of private and shared information by conducting one set of experiments with swarms using individual memory (i.e. site fidelity), and a second set of experiments with swarms using shared communication (i.e. pheromone-like waypoints). We found that both memory and communication are advantageous behaviors for swarms collecting resources from large clusters: sensing the local resource density in the immediate neighborhood of a found resource is an effective mechanism for guiding robots to densely-packed, resource-rich locations. However, shared communication was less beneficial than individual memory as a result of positional errors that were propagated through the swarm, causing other robots to become lost and thus waste valuable exploration time.

In Chapter 3, we introduced models of positional and resource detection error into our robot swarm simulation to describe the empirically-measured sensor error in our physical iAnt robots. Using this simulation, we evolved foraging behaviors for simulated swarms in the presence these error models, then transferred these behaviors into iAnt robots and evaluated their performance. When compared to iAnts foraging using parameters evolved in a simulation without error, the error-adapted physical swarms collected more resources, and their foraging performance was not statistically different from error-adapted simulated swarms. Error-adapted swarms also employed distinctly different foraging strategies from non-error-adapted swarms, including a higher likelihood of using individual memory, and, for clustered distributions, a lower

likelihood of using shared communication for a given local density of resources.

In Chapter 4, we presented a systematic analysis of our swarm robotics system, specifically the system's ability to tolerate error, to flexibly forage in different environments, and to scale to large swarm sizes. We found that adapting to sensor error is most valuable when resources are clustered: cluster-adapted swarms are highly dependent on sensed information, and thus gain the most benefit from tuning their information-dependent behaviors to match the inherently noisy iAnt sensors. Evolving to efficiently forage for heterogeneously-distributed resources produced strategies that balanced the extent and thoroughness of search, using random exploration to find scattered resources, individual memory to collect resources from small piles, and recruitment to collect resources from large piles. We also found that power-law-adapted robots in larger swarms evolved to disperse more and communicate less, and that parameters governing movement had a greater effect on scaling performance than parameters governing communication. The success of these evolved foraging strategies demonstrated that our approach is a practical method to generate effective foraging strategies from interactions among foraging behaviors and the specified foraging environment.

Finally, in Chapter 5, we augmented our swarm robotics approach with machine learning and statistical models to improve collection efficiency for robots foraging on sparse clusters of resources. These robots combined a modified central-place foraging algorithm with a clustering procedure and a optimal model selection approach to predict and exploit the locations of residual resources. Experiments showed collection efficiency was most significantly increased when robot swarms switched from ant-inspired foraging to directly exploiting clusters after approximately 90% of resources had been collected. Not surprisingly, cluster exploitation was most effective when resources were highly clustered in the environment.

Taken as a whole, this dissertation presents a comprehensive swarm robotics

system, consisting of *i*) a set of high-level behaviors that constitute a **foraging strategy**, *ii*) an **agent-based model** that implements each foraging strategy in a simulated robot swarm, *iii*) a **genetic algorithm** that selects the most efficient foraging strategy according to the performance of the simulation, and *iv*) a swarm of physically-embodied **iAnt robots** that implement the most efficient strategy in the real world. Although our system was designed to be a demonstration platform for swarm robotics research, this work provides a foundation for designing and implementing autonomous robot swarms that can function outside of the academic research laboratory. The ability of robot swarms to tolerate sensor noise, adapt to variable environments, distribute work across large teams, and identify and exploit heterogeneously-distributed resources are all critical factors for successful remote exploration missions on distant worlds.

6.1 Major Findings

In the course of our research on foraging robot swarms, we have uncovered several key findings that are particularly novel, innovative, and/or unexpected. We highlight those findings here.

First, we used a relatively simple GA to quickly evolve a small set of integrated strategies that foraged efficiently in varied and complex environments. Previous studies have developed or evolved foraging behaviors for randomly distributed resources [7, 28, 80], while others have studied foraging from one or two infinite sources [60, 38]. However, previous studies have not attempted to evolve strategies that are sufficiently flexible to perform well in both of those environments, nor have they developed strategies that are effective at collecting from more complex distributions. We showed that foraging for resources in heterogeneous clusters requires more complex communication, memory, and environmental sensing than strategies evolved in

previous work.

As we tested these evolved foraging strategies for different robot swarm sizes, we observed a sub-linear scaling relationship between the size of the swarm and the resources collected per robot. Other researchers have focused on inter-robot interference as the main cause of sub-linear scaling [82, 76], but we observe sub-linear scaling even without including collisions in the simulation. We hypothesized that the central-place foraging paradigm produces diminishing returns as swarm size increases because the central nest imposes a "bottleneck" constraint on swarm efficiency – robots in larger swarms have to travel farther to collect more resources. This bottleneck could potentially be mitigated by increasing the number of nests in proportion to the size of the swarm, a scaling technique inspired by the flexible recruitment strategies of Argentine ants [36].

We also found that individual robots in smaller swarms were more likely to lay pheromones than those in larger swarms, a result that conflicts with a previous prediction that communication in ant colonies increases with colony size [10]. Contrary to the field studies conducted in previous work, however, our experiments were purposefully designed to fix all other known factors that could potentially influence foraging rate: territory size, quantity and distribution of resources, as well as all aspects of the individuals in the swarm. In reality, territory size is thought to scale with ant colony size [63] and colony growth rate [48]; colonies with larger territories naturally have access to more resources; and body size and ground speed vary widely across ant species [35]. Future studies should therefore incorporate some or all of these factors when considering the role of pheromones in swarm scalability.

Finally, we demonstrated the advantage of ant-inspired robot swarm foraging over a mathematical optimization technique. We also demonstrated a synergy between our ant-inspired foraging algorithm and a more traditional clustering algorithm [54]. Identifying and exploiting resources through machine learning and model selection

was disadvantageous until approximately 90% of the resources had been collected by the CPFA. However, the collection of these first 90% of resources was vital to foragers that switched to exploiting the remaining 10% of the identified resource clusters. In other words, our swarm foraging algorithm was essential to collecting the majority of resources, but also benefited from cluster exploitation in order to efficiently accomplish the complete collection task. These most recent findings support the efficacy of augmenting biologically-inspired methods with machine learning algorithms to generate new, robust search behaviors that function efficiently in a variety of environments.

6.2 Foraging Robot Swarms Within the Context of Evolutionary Robotics

Evolutionary robotics (ER) is an automated design approach for generating sensing, morphology, and control architectures for robots. ER is a integrated method of robot design, meaning that the robot (or robot swarm) is evolved as a single, cohesive unit, as oppose to traditional engineering approaches which generally follow the reductionist methodology of designing each of the robot's functions (e.g. mapping, path planning, kinematics, etc.) in isolation. Following a recent ER review article by Doncieux et al [30], we consider our work on foraging robot swarms as it relates to the state of the art in evolutionary robotics.

We generated a diverse set of foraging strategies by evolving our robot swarms in different environmental contexts. This approach mitigated premature convergence by providing an analogue of the archive system used in novelty search [75]. In this case, the diversity of the environments functioned as the novelty metric, driving the search process in order to produce a variety of different, efficient foraging strategies.

According to Doncieux et al, environment-driven evolution also provides selection pressure that can promote the development of novel foraging strategies, a hypothesis that is empirically supported by our findings in this manuscript. Although ER aims to design robots with "as little prior knowledge as possible," Doncieux et al also mention that "drawing inspiration from nature can be helpful" [30]. In fact, our work demonstrates that efficient and flexible foraging strategies can emerge from simple, biologically-inspired behaviors evolved in response to varied and complex environments.

6.3 The Exploration-Exploitation Tradeoff

The fundamental challenge for designing robot swarms that forage efficiently is deciding how to balance the tradeoff between exploration and exploitation. Foraging ant colonies, as well as any other species of foraging animal, face similar challenges. More specifically, *i*) how much time should be spent searching for new, previously undiscovered resources (exploration), vs. *ii*) how much time should be spent identifying, localizing, and returning to areas known to have high resource density (exploitation). The goal of determining the optimal tradeoff, commonly presented as the multi-armed bandit problem [45], is computationally intractable and has been investigated in numerous studies across a wide variety of disciplines, including machine learning [4], economics [73], and philosophy [23].

In our work, the genetic algorithm is able to appropriately balance exploration and exploitation for different environments by tuning a small set of behavioral parameters. We hypothesize that the GA's ability to select efficient foraging strategies for each resource distribution stems from the fact that the evolutionary process is able to indirectly sample the environment via the local resource density sensing procedure. That is, repeated environmental sampling by individual robots facilitates

an intensive testing of the behaviors governing information decisions (i.e. individual memory and shared communication), and the GA responds to the results of this testing (in the form of foraging efficiency) by promoting the strategies that provide the most effective explore-exploit tradeoff for a given environment. An alternative approach might be to attempt to balance exploration and exploitation in a more direct, straightforward manner, for example with a single real-valued parameter controlling the probability of exploring vs. exploiting. However, the lack of input from the environment would prevent such an approach from discerning the characteristics of different environments, and thus it would be difficult for this approach to achieve efficient foraging for a variety of resource distributions. Instead, our approach unifies the effectiveness of our foraging robot swarms with the environment in which they are evaluated, and therefore ensures that selection pressure from the environment drives the evolution of efficient foraging behavior (see Section 6.2 for a more detailed discussion of this phenomena).

6.4 Extensions to Foraging Robot Swarms

We have developed several innovative extensions that use our swarm robotics system to study new interdisciplinary problems, and to further our goal of reaching a wide range of researchers, developers, and students.

6.4.1 Alternative Search Strategies Inspired by the Immune System

Cells in the immune system protect biological organisms by quickly detecting and removing pathogens that may otherwise cause harm. As with colonies of seed-harvester ants, these immune cells have evolved strategies that enable them to efficiently search

for pathogens in a variety of environments. We previously characterized the movement patterns of T cells in the lymph node as following a heavy-tailed, Lévy-like distribution [39]. This Lévy walk-based search strategy is particularly efficient at finding target objects that are clustered, but sparsely distributed in space [132].

We modified our swarm robotics system to incorporate this T cell-inspired Lévy walk as an alternative search strategy, and compared the efficiency of heavy-tailed movement to the correlated random walk used in the CPFA [39]. As in previous experiments with the standard CPFA, we evolved the step length of the Lévy walk for efficient search on different target distributions. We observed a small, but statistically significant improvement in efficiency for simulated robot swarms using heavy-tailed search, and we saw repeated convergence of the evolved step length to a value consistent with the heavy-tailed, Lévy-like walk observed in T cells.

Another collaborative study focused on T cells and robots that visit some locations more frequently than others [40]. Results of this study showed similar distributions of step length for T cells and robots, which may be indicative of some underlying mechanism that is inherent to efficient spatial search in both biological and mechanical systems. These results also support the existence of a previously hypothesized adaptive T cell response to environmental cues, such as dendritic cells (DCs) carrying antigen indicative of a foreign pathogen.

6.4.2 Alternative Recruitment Strategies Inspired by Foraging Ants

Chemical pheromones provide foraging ants with a stigmergic, mass recruitment method that is highly scalable, fully decentralized, and generally tolerant of environments with little or no volatility. Robot swarms that mimic ant pheromones, on the other hand, are restricted to foraging in tightly controlled environments that

require complex, monolithic infrastructure. For example, swarm researchers have constructed elaborate stigmergic mechanisms using an always-on ink pen and white paper flooring [123]; a tightly-coupled video camera, video projector, and vision processing system [44]; and a phosphorescent-painted floor combined with ultraviolet light emitters [86].

Ants also use simpler, more primitive recruitment strategies such as tandem running and group raids, which include a local recruitment display to stimulate nest mates to return to high-quality food patches [22]. Robot swarms mimic these short-range recruitment strategies using robot-to-robot physical connections [72], nearest-neighbor local communication [118], and robot-chain path formation [102]. These swarms employ relatively simple communication schemes that do not require global coordination or preexisting infrastructure in order to collectively forage for resources or aggregate in target areas.

We recently used our swarm robotics system to demonstrate that nest recruitment strategies are at least as efficient as pheromone recruitment strategies for many environments [78]. Nest recruitment is relatively simple to implement in robot swarms, while pheromone recruitment requires robot- and environment-specific infrastructure. Further, the foraging success of nest recruiters depends only on local, agent-to-agent communication, while pheromone recruiters often depend on global coordination with a single point of failure. The results of this study suggest that research in swarm robotics should focus less on mimicking ant stigmergy, and more on designing and evaluating new decentralized information-sharing protocols that are more scalable and easier to implement in natural environments as foraging strategies for real robots.

6.4.3 Furthering Swarm Robotics Research and Technology Through Student Competition

Successful exploration of the Moon, Mars, and asteroids requires the location and retrieval of local resources on extraplanetary surfaces. Technologies are needed to find and collect materials such as ice (convertible into liquid water, hydrogen fuel and oxygen to support human life) and rocks, minerals and construction materials to build human shelters. In-situ resource utilization (ISRU) will be dramatically improved by robotic swarms able to efficiently locate, identify and collect resources over large and previously explored territory.

To that end, we have recently received funding from NASA's Minority University Research and Education Project (MUREP) to present an innovative swarm robotics challenge that pushes the state of the art in swarm robotics. Our cooperative NASA Swarmathon competition, which includes a three-year $1.8 million grant, will challenge 1,000 students at 50 minority-serving institutions across the country to further advance swarm robotics by programming teams of robots (Fig. 6.1) to autonomously search for and retrieve resources in unmapped environments. This technology has the potential to revolutionize space exploration programs that collect valuable materials. These samples have the potential to unravel mysteries about the origins of life.

6.5 Future Work

This work proposes and tests several fundamental hypotheses about foraging robot swarms. Even so, swarm robotics is a nascent field with numerous, equally significant research questions to pursue. Here we consider additional research directions to extend our existing swarm analysis as future work.

Figure 6.1: The Swarmie hardware platform developed through a UNM partnership with NASA Kennedy Space Center's (KSC) Swamp Works laboratory. The Swarmie platform is an integrated system with adaptive software and a simulator that allows rapid testing of new algorithms in software and rapid porting to hardware for verification in physical experiments.

We considered error tolerance in our system as the swarm's ability to mitigate noisy sensor data (i.e. errors in positional and resource detection information), but robots that function outside of the laboratory must also tolerate the loss of sensor and/or actuator functionality, as well as the complete loss of individual robots in the swarm. Following Bjerkenes and Winfield [14], future studies should consider a systematic study of fault tolerance in our system to assess the effect of hardware failure on foraging efficiency, as well as the ability of the GA to tune robot behaviors in order to mitigate these failures. Because some foraging behaviors are dependent on emergent properties (e.g. pheromone recruitment), we predict that experimentally measuring the mean time before failure (MTBF) in our foraging robot swarms would provide a quantitative benchmark to estimate our system's k-out-of-N reliability, the

approximate number of robots required to maintain emergent behavior [14]. This reliability metric should also be applied to future scalability studies (our previous scalability experiments did not implement hardware failure) in order to generate more accurate models of foraging efficiency for large swarms.

We previously demonstrated that our foraging strategies could be evolved in real time in swarms of 12 simulated robots using a distributed coevolutionary framework [57] based on work by O'Dowd et al [103]. Unpublished experiments with support vector machine (SVM) classifiers have also shown promise for accurate prediction of resource distributions given a small sample of the local resource density. Based on these empirical observations of the strong statistical correlation between resource distribution and local resource density, we have additionally proposed a lifelong learning architecture based on Ruvolo and Eaton's work [115] that could adapt robot swarm foraging strategies in real time using a cloud-based storage system. In future work, we plan to fully implement one or more of these real-time evolution system in our physical robot swarm, which would enable our swarm to adapt its behavior to previously unknown environments without the need for offline simulation and parameter transfer.

Finally, we plan to explore the feasibility of incorporating unmanned aerial vehicles (UAVs) with integrated cameras into our foraging robot swarm [13]. This type of heterogenous swarm would extend the visual sensing range of our existing system, in addition to significantly increasing the speed of local sensing. The top cover of our recently introduced Swarmie robot platform (Fig. 6.1) would provide the UAVs with an ideal "home base" from which to take off and land, as well as a charging station and long-distance transportation system. This heterogeneous swarm would be particularly advantageous in very large environments where ground-based search is inefficient, but local, short-range sensing is still required to complete the task assigned to the swarm. Ideal tasks for this type of swarm could include environmental

monitoring of farms or ecological research stations, locating and collecting water-ice on the Moon, or mining of raw materials from near-Earth asteroids.

Appendices

A Beyond Pheromones: Electronic Supplementary Material **100**

Appendix A

Beyond Pheromones: Electronic Supplementary Material

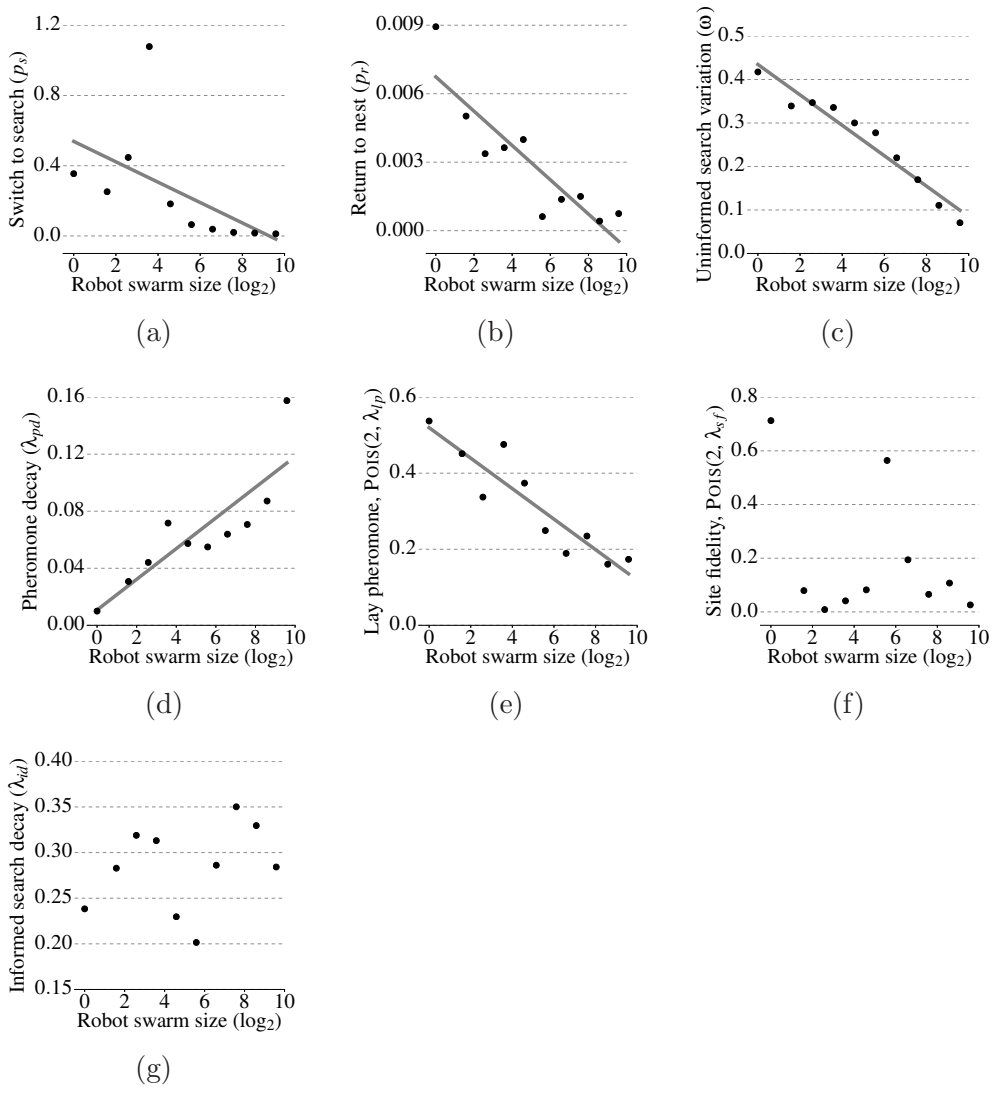


Figure A.1: The best evolved parameters for simulated swarms of 1 to 768 robots foraging without sensor error on a power law distribution. (a) Swarm size versus probability of switching to searching behavior (p_s) (slope = -0.058, $R^2 = 0.30, p = 0.10$), (b) swarm size versus probability of switching to traveling behavior (p_r) (slope = -0.00075, $R^2 = 0.79, p < 0.001$), (c) swarm size versus uninformed search correlation (ω) (slope = -0.035, $R^2 = 0.94, p < 0.001$), (d) swarm size versus rate of pheromone decay (λ_{pd}, Eq. 4.5) (slope = 0.011, $R^2 = 0.73, p = 0.0016$), (e) swarm size versus probability of laying pheromone (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$) (slope = -0.040, $R^2 = 0.84, p < 0.001$), (f) swarm size versus probability of returning to a site (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{sf}$) (NS, $p = 0.27$), and (g) swarm size versus decay rate of informed search (λ_{id}, Eq. 4.3) (NS, $p = 0.38$).

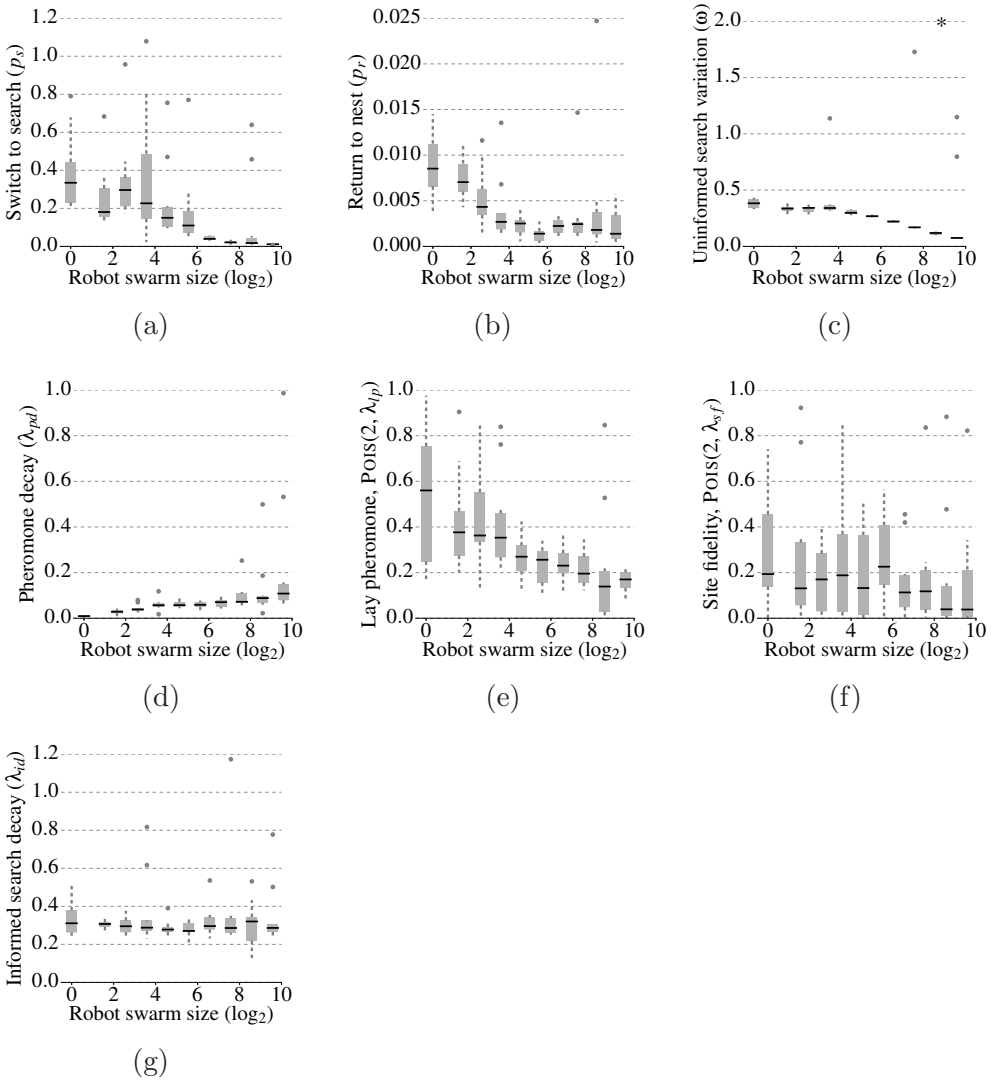


Figure A.2: Statistical distributions of parameters evolved for simulated swarms of 1 to 768 robots foraging without sensor error on a power law distribution. (a) Swarm size versus probability of switching to searching behavior (p_s), (b) swarm size versus probability of switching to traveling behavior (p_r), (c) swarm size versus uninformed search correlation (ω), (d) swarm size versus rate of pheromone decay (λ_{pd}, Eq. 4.5), (e) swarm size versus probability of laying pheromone (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{lp}$), (f) swarm size versus probability of returning to a site (Eq. 4.4: $k \leftarrow 2$, $\lambda \leftarrow \lambda_{sf}$), and (g) swarm size versus decay rate of informed search (λ_{id}, Eq. 4.3). Gray dots indicate outliers beyond interquartile range; an asterisk represents an outlier at $\omega = 9.3$.

References

[1] F. R. Adler and D. M. Gordon. Optimization, conflict, and nonoverlapping foraging ranges in ants. *The American Naturalist*, 162(5):529–543, 2003.

[2] C. Ampatzis. *On the evolution of autonomous time-based decision-making and communication in collective robotics*. PhD thesis, Université libre de Bruxelles, 2008.

[3] S. Aron, J. M. Pasteels, and J.-L. Deneubourg. Trail-laying behaviour during exploratory recruitment in the Argentine ant, Iridomyrmex humilis (Mayr). *Biology of Behavior*, 14:207–217, 1989.

[4] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[5] P. Bailis, R. Nagpal, and J. Werfel. Positional communication and private information in honeybee foraging models. In *Swarm Intelligence: 7th International Conference, ANTS 2010*, volume 6234, pages 263–274. Springer Berlin Heidelberg, Berlin, DE, 2010.

[6] D. N. Baker. NASA: Let academia lead space science. *Nature*, 488(7409):27–28, 2012.

[7] T. Balch. Reward and diversity in multirobot foraging. In *IJCAI-99 Workshop on Agents Learning About, From and With other Agents*, pages 92–99, San Francisco, CA, 1999. Morgan Kauffman Publishers.

[8] G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, and S. Nolfi. Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 37(1):224–239, Feb. 2007.

[9] S. Banerjee and M. Moses. 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, 2010.

[10] R. Beckers, S. Goss, J.-L. Deneubourg, and J. M. Pasteels. Colony size, communication, and ant foraging strategy. *Psyche*, 96(3-4):239–256, 1989.

[11] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. C. Pratt. Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proceedings of the IEEE*, 99(9):1470–1481, 2011.

[12] B. D. Beverly, H. McLendon, S. Nacu, S. Holmes, and D. M. Gordon. How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behavioral Ecology*, 20(3):633–638, 2009.

[13] N. Bezzo, J. P. Hecker, K. Stolleis, M. E. Moses, and R. Fierro. Exploiting heterogeneous robotic systems in cooperative missions. pages 1–23. arXiv:1509.00948.

[14] J. D. Bjerknes and A. F. T. Winfield. On fault tolerance and scalability of swarm robotic systems. In *Springer Tracts in Advanced Robotics: Distributed Autonomous Robotic Systems (DARS 2010)*, volume 83, pages 431–444. Springer Berlin Heidelberg, Berlin, DE, 2013.

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: From natural to artificial systems*. Oxford University Press, New York, NY, 1999.

[16] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine. Self-organization in social insects. *Trends in Ecology & Evolution*, 12(5):188–193, 1997.

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

[18] R. A. Brooks. New Approaches to Robotics. *Science*, 253(5025):1227–1232, 1991.

[19] R. A. Brooks. Artifical Life and Real Robots. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 3–10, Cambridge, MA, 1992. MIT Press.

[20] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-organization in biological systems*. Princeton University Press, Princeton, NJ, 2001.

[21] Y. U. Cao, A. S. Fukunaga, and A. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[22] D. Cassill. Rules of supply and demand regulate recruitment to food in an ant society. *Behavioral Ecology and Sociobiology*, 54(5):441–450, 2003.

[23] J. D. Cohen, S. M. McClure, and A. J. Yu. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942, 2007.

[24] B. D. Connelly, P. K. McKinley, and B. E. Beckmann. Evolving cooperative pheromone usage in digital organisms. In *IEEE Symposium on Artificial Life (ALife '09)*, pages 184–191, Piscataway, NJ, 2009. IEEE Press.

[25] T. O. Crist and J. W. Haefner. Spatial model of movement and foraging in harvester ants (Pogonomyrmex) (II): The roles of environment and seed dispersion. *Journal of Theoretical Biology*, 166(3):315–323, 1994.

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

[27] S. A. Curtis, W. Truszkowski, M. L. Rilee, and P. E. Clark. ANTS for human exploration and development of space. In *Proceedings of the 2003 IEEE Aerospace Conference*, volume 1, pages 1–161, Piscataway, NJ, 2003. IEEE Press.

[28] M. V. Dartel, E. Postma, J. van den Herik, and G. de Croon. Macroscopic analysis of robot foraging behaviour. *Connection Science*, 16(3):169–181, 2004.

[29] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 356–363, Cambridge, MA, 1991. MIT Press.

[30] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. E. G. Eiben. Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2(4):1–18, 2015.

[31] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Martinez Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2011.

[32] M. Dorigo and E. Sahin. Swarm robotics–special issue editorial. *Autonomous Robots*, 17(2-3):111–113, 2004.

[33] M. Dorigo, V. Trianni, E. ahin, R. Groß, T. H. LaBella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245, 2004.

[34] J. H. Fewell. Directional fidelity as a foraging constraint in the western harvester ant, Pogonomyrmex occidentalis. *Oecologia*, 82:45–51, 1990.

[35] T. P. Flanagan, K. Letendre, W. R. Burnside, G. M. Fricke, and M. E. Moses. Quantifying the effect of colony size and food distribution on harvester ant foraging. *PLoS ONE*, 7(7):e39427, 2012.

[36] T. P. Flanagan, N. M. Pinter-Wollman, M. E. Moses, and D. M. Gordon. Fast and flexible: Argentine ants recruit from nearby trails. *PLoS ONE*, 8(8):e70888, 2013.

[37] C. Fraley and A. E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[38] G. Francesca, M. Brambilla, A. Brutschy, V. Trianni, and M. Birattari. AutoMoDe: A novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112, 2014.

[39] G. M. Fricke, F. Asperti-Boursin, J. P. Hecker, J. L. Cannon, and M. E. Moses. From Microbiology to Microcontrollers: Robot Search Patterns Inspired by T Cell Movement. In *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (Advances in Artificial Life, ECAL 2013)*, pages 1009–1016, Cambridge, MA, 2013. MIT Press.

[40] G. M. Fricke, S. R. Black, J. P. Hecker, J. L. Cannon, and M. E. Moses. Distinguishing adaptive search from random search in robots and T cells. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference (GECCO '15)*, pages 105–112, New York, NY, 2015. ACM.

[41] C. O. Fritz, P. E. Morris, and J. J. Richler. Effect size estimates: Current use, calculations, and interpretation. *Journal of Experimental Psychology: General*, 141(1):2–18, 2012.

[42] D. W. Gage. Many-robot MCM search systems. In *Autonomous Vehicles in Mine Countermeasures Symposium*, pages 4–7, 1995.

[43] S. Garnier. From ants to robots and back: How robotics can contribute to the study of collective animal behavior. *Bio-Inspired Self-Organizing Robotic Systems*, 355:105–120, 2011.

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

[45] J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. Wiley, Hoboken, NJ, 2011.

[46] D. M. Gordon. The relation of recruitment rate to activity rhythms in the harvester ant, Pogonomyrmex barbatus (F. Smith) (Hymenoptera: Formicidae). *Journal of the Kansas Entomological Society*, 56(3):277–285, 1983.

[47] D. M. Gordon. The spatial scale of seed collection by harvester ants. *Oecologia*, 95(4):479–487, 1993.

[48] D. M. Gordon. The development of an ant colony's foraging range. *Animal Behaviour*, 49:649–659, 1995.

[49] D. M. Gordon. The regulation of foraging activity in red harvester ant colonies. *The American Naturalist*, 159(5):509–518, 2002.

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

[51] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[52] U. D. Gupta, V. Menon, and U. Babbar. Detecting the number of clusters during expectation-maximization clustering using information criterion. In *2010 Second International Conference on Machine Learning and Computing (ICMLC)*, pages 169–173, Piscataway, NJ, 2010. IEEE Press.

[53] E. Haasdijk, A. Eiben, and A. F. T. Winfield. Individual, social and evolutionary adaptation in collective systems. In *Handbook of Collective Robotics: Fundamentals and Challenges*, pages 295–336. Pan Stanford Publishing, Singapore, 2010.

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

[55] J. P. Hecker, K. Letendre, K. Stolleis, D. Washington, and M. E. Moses. Formica ex machina: Ant swarm foraging from physical to virtual and back again. In *Swarm Intelligence: 8th International Conference, ANTS 2012*, pages 252–259. Springer Berlin Heidelberg, Berlin, DE, 2012.

[56] J. P. Hecker and M. E. Moses. 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, 2013. ACM.

[57] J. P. Hecker and M. E. Moses. Real-time evolution of iAnt robot foraging strategies. In *ALIFE 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 835–836, Cambridge, MA, 2014. MIT Press.

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

[59] J. P. Hecker, K. Stolleis, B. Swenson, K. Letendre, and M. E. Moses. Evolving error tolerance in biologically-inspired iAnt robots. In *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (Advances in Artificial Life, ECAL 2013)*, pages 1025–1032, Cambridge, MA, 2013. MIT Press.

[60] N. Hoff, R. Wood, and R. Nagpal. Distributed colony-level algorithm switching for robot swarm foraging. In *Distributed Autonomous Robotic Systems: The 10th International Symposium*, pages 417–430. Springer, New York, NY, 2010.

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

[62] B. Hölldobler and E. O. Wilson. The multiple recruitment systems of the African weaver ant Oecophylla longinoda (Latreille) (Hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, 60(1):19–60, 1978.

[63] B. Hölldobler and E. O. Wilson. *The ants*. Harvard University Press, Cambridge, MA, 1990.

[64] M. A. Hsieh, E. Forgoston, T. W. Mather, and I. B. Schwartz. Robotic manifold tracking of coherent structures in flows. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4242–4247, Piscataway, NJ, 2012. IEEE Press.

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

[66] N. Jakobi. Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In *Fourth European Conference on Artificial Life*, pages 348–357, Cambridge, MA, 1997. MIT Press.

[67] N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Intelligence: 3rd European Conference on Artificial Life*, pages 704–720. MIT Press, Cambridge, MA, 1995.

[68] A. R. Johnson, J. A. Wiens, B. T. Milne, and T. O. Crist. Animal movements and population dynamics in heterogeneous landscapes. *Landscape Ecology*, 7(1):63–75, 1992.

[69] S. Kazadi. *Swarm engineering*. PhD thesis, California Institute of Technology, 2000.

[70] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, volume 6, pages 739–743, Piscataway, NJ, 1999. IEEE Press.

[71] C. S. Kong, N. A. Peng, and I. Rekleitis. Distributed coverage with multi-robot system. In *IEEE International Conference on Robotics and Automation*, pages 2423–2429, Piscataway, NJ, 2006. IEEE Press.

[72] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, 2000.

[73] K. Kyriakopoulos and C. Moorman. Tradeoffs in marketing exploitation and exploration strategies: The overlooked role of market orientation. *International Journal of Research in Marketing*, 21(3):219–240, 2004.

[74] T. H. Labella, M. Dorigo, and J.-L. Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):4–25, Sept. 2006.

[75] J. Lehman and K. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

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

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

[78] D. Levin, J. P. Hecker, M. E. Moses, and S. Forrest. Volatility and spatial distribution of resources determine ant foraging strategies. In *Proceedings of the European Conference on Artificial Life 2015 (ECAL 2015)*, pages 256–263, Cambridge, MA, 2015. MIT Press.

[79] W. Liu and A. F. T. Winfield. Modelling and optimisation of adaptive foraging in swarm robotic systems. *The International Journal of Robotics Research*, 29(14):1743–1760, 2010.

[80] W. Liu, A. F. T. Winfield, and J. Sa. Modelling swarm robotic systems: A case study in collective foraging. In *Towards Autonomous Robotic Systems (TAROS 07)*, volume 23, pages 25–32, Aberystwyth, UK, 2007. University of Wales.

[81] E. B. Mallon and N. R. Franks. Ants estimate area using Buffon's needle. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1445):765–770, 2000.

[82] M. J. Matarić. Minimizing complexity in controlling a mobile robot population. In *1992 IEEE International Conference on Robotics and Automation*, pages 830–835, Piscataway, NJ, 1992. IEEE Press.

[83] M. J. Matarić. Learning to Behave Socially. In *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*, pages 453–462, Cambridge, MA, 1994. MIT Press.

[84] M. J. Matarić. Behaviour-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.

[85] M. J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4:73–83, 1997.

[86] R. Mayet, J. Roberz, T. Schmickl, and K. Crailsheim. Antbots: A feasible visual emulation of pheromone trails for swarm robots. In *Swarm Intelligence: 7th International Conference, ANTS 2010*, volume 6234, pages 84–94, Berlin, DE, 2010. Springer Berlin Heidelberg.

[87] J.-A. Meyer, P. Husbands, and I. Harvey. Evolutionary robotics: A survey of applications and problems. In *Evolutionary Robotics: First European Workshop, EvoRobot98*, pages 1–22, Berlin, DE, 1998. Springer Berlin Heidelberg.

[88] O. Miglino, H. H. Lund, and S. Nolfi. Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life*, 2(4):417–434, Jan. 1995.

[89] C. Moeslinger, T. Schmickl, and K. Crailsheim. Emergent flocking with low-end swarm robots. *Swarm Intelligence*, pages 424–431, 2011.

[90] F. Mondada, G. C. Pettinaro, I. Kwee, A. Guignard, L. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. SWARM-BOT: A swarm of autonomous mobile robots with self-assembling capabilities. In *Proceedings of the Workshop on Self-Organisation and Evolution of Social Behaviour*, pages 307–312, Zürich, CH, 2002. ETH-Zürich.

[91] M. Moses and S. Banerjee. Biologically Inspired Design Principles for Scalable, Robust, Adaptive, Decentralized Search and Automated Response (RADAR). In *Proceedings of the 2011 IEEE Conference on Artificial Life*, pages 30–37, Piscataway, NJ, 2011. IEEE Press.

[92] M. Moses, T. Flanagan, K. Letendre, and M. Fricke. Ant colonies as a model of human computation. In *Handbook of Human Computation*, pages 25–37. Springer, New York, NY, 2013.

[93] M. E. Moses. *Metabolic scaling from individuals to societies.* PhD thesis, University of New Mexico, 2005.

[94] M. E. Moses, J. P. Hecker, and K. Stolleis. The iAnt project. http://iant.cs.unm.edu, 2014.

[95] J. F. Mull and J. A. MacMahon. Spatial variation in rates of seed removal by harvester ants (Pogonomyrmex occidentalis) in a shrub-steppe ecosystem. *American Midland Naturalist*, 138(1):1–13, 1997.

[96] M. Müller and R. Wehner. Path integration in desert ants, Cataglyphis fortis. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

[97] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen. Search and rescue robotics. In *Springer Handbook of Robotics*, pages 1151—-1173. Springer Berlin Heidelberg, Berlin, DE, 2008.

[98] A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, Apr. 2009.

[99] A. L. Nelson, E. Grant, and T. C. Henderson. Evolution of neural controllers for competitive game playing with teams of mobile robots. *Robotics and Autonomous Systems*, 46(135-150), 2004.

[100] S. Nolfi and D. Floreano. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press, Cambridge, MA, 2000.

[101] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. *Artificial Life*, 4:190–197, 1993.

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

[103] P. J. O'Dowd, A. F. T. Winfield, and M. Studley. The distributed co-evolution of an embodied simulator and controller for swarm robot behaviours. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4995–5000. IEEE, Sept. 2011.

[104] L. A. Panait and S. Luke. Learning ant foraging behaviors. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, pages 575–580, Cambridge, MA, 2004. MIT Press.

[105] L. E. Parker. Designing control laws for cooperative agent teams. In *IEEE International Conference on Robotics and Automation*, pages 582–587, Piscataway, NJ, 1993. IEEE Press.

[106] L. E. Parker. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[107] L. E. Parker. Path planning and motion coordination in multiple mobile robot teams. In *Encyclopedia of Complexity and System Science*, pages 5783–5800. Springer, New York, NY, 2009.

[108] D. Payton, M. Daily, B. Hoff, M. Howard, and C. Lee. Pheromone robotics. *Autonomous Robots*, 11(3):319–324, 2001.

[109] T. Paz Flanagan, K. Letendre, W. Burnside, G. M. Fricke, and M. Moses. How ants turn information into food. In *2011 IEEE Symposium on Artificial Life (ALIFE 2011)*, pages 178–185, Piscataway, NJ, 2011. IEEE Press.

[110] R. Pfeifer, M. Lungarella, and F. Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007.

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

[112] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, T. Stirling, A. Guti, L. M. Gambardella, and M. Dorigo. ARGoS: A pluggable, multi-physics engine simulator for heterogeneous swarm robotics. Technical Report December 2010, TR/IRIDIA/2010-026.002, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2011.

[113] G. Pini and E. Tuci. On the design of neuro-controllers for individual and social learning behaviour in autonomous robots: An evolutionary approach. *Connection Science*, 20(2-3):211–230, 2008.

[114] B. Prabhakar, K. N. Dektar, and D. M. Gordon. The regulation of ant colony foraging activity without spatial information. *PLoS Computational Biology*, 8(8):e1002670, 2012.

[115] P. Ruvolo and E. Eaton. ELLA: An efficient lifelong learning algorithm. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 507–515, 2013.

[116] E. Sahin. Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics*, 3342:10–20, 2005.

[117] J. A. Sauter, R. Matthews, H. Van Dyke Parunak, and S. Brueckner. Evolving adaptive pheromone path planning mechanisms. In *Proceedings of the First International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '02)*, pages 434–440, New York, NY, 2002. ACM.

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

[119] A. J. C. Sharkey. Robots, insects and swarm intelligence. *Artificial Intelligence Review*, 26(4):255–268, 2006.

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

[121] A. Stroupe, A. Okon, M. Robinson, T. Huntsberger, H. Aghazarian, and E. Baumgartner. Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. *Autonomous Robots*, 20(2):113–123, 2006.

[122] D. J. T. Sumpter and M. Beekman. From nonlinearity to optimality: Pheromone trail foraging by ants. *Animal Behaviour*, 66(2):273–280, 2003.

[123] J. Svennebring and S. Koenig. Building terrain-covering ant robots: A feasibility study. *Autonomous Robots*, 16(3):313–332, 2004.

[124] M. Thiélin-Bescond and G. Beugnon. Vision-independent odometry in the ant Cataglyphis cursor. *Naturwissenschaften*, 92(4):193–197, 2005.

[125] V. Trianni and M. Dorigo. Self-organisation and communication in groups of simulated and physical robots. *Biological Cybernetics*, 95(3):213–231, 2006.

[126] V. Trianni and S. Nolfi. Engineering the evolution of self-organizing behaviors in swarm robotics: A case study. *Artificial Life*, 17(3):183–202, 2011.

[127] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff. Autonomous and autonomic systems: A paradigm for future space exploration missions. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 36(3):279–291, 2006.

[128] W. R. Tschinkel and D. F. Howard. Colony founding by pleometrosis in the fire ant, Solenopsis invicta. *Behavioral Ecology and Sociobiology*, 12(2):103–113, May 1983.

[129] E. Tunstel, J. M. Dolan, T. Fong, and D. Schreckenghost. Mobile robotic surveying performance for planetary surface site characterization. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pages 200–205, New York, NY, 2008. ACM.

[130] M. G. Turner. Landscape ecology: The effect of pattern on process. *Annual Review of Ecology and Systematics*, 20:171–197, 1989.

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

[132] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley. Optimizing the success of random searches. *Nature*, 401(28):911–914, 1999.

[133] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, Apr. 2002.

[134] B. Webb. Using robots to understand animal behavior. *Advances in the Study of Behavior*, 38:1–58, 2008.

[135] J. A. Wiens, N. C. Stenseth, B. V. Horne, and R. A. Ims. Ecological mechanisms and landscape ecology. *Oikos*, 66(3):369–380, 1993.

[136] A. Wilby and M. Shachak. Harvester ant response to spatial and temporal heterogeneity in seed availability: Pattern in the process of granivory. *Oecologia*, 125(4):495–503, 2000.

[137] E. O. Wilson. Chemical communication among workers of the fire ant Solenopsis saevissima (Fr. Smith) 1. The organization of mass-foraging. *Animal Behaviour*, 10(1):134–147, 1962.

[138] A. Winfield. *Robotics: A very short introduction*. Oxford University Press, New York, NY, 2012.

[139] A. F. T. Winfield. Foraging robots. In *Encyclopedia of Complexity and Systems Science*, pages 3682–3700. Springer, New York, NY, 2009.

[140] A. F. T. Winfield, C. J. Harper, and J. Nembrini. Towards dependable swarms and a new discipline of swarm engineering. *Swarm Robotics*, 3342:126–142, 2005.

[141] M. Wittlinger, R. Wehner, and H. Wolf. The ant odometer: Stepping on stilts and stumps. *Science*, 312(5782):1965–1967, 2006.

[142] S. Wohlgemuth, B. Ronacher, and R. Wehner. Ant odometry in the third dimension. *Nature*, 411(6839):795–798, 2001.