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

Joshua P. Hecker, Kenneth Letendre, Karl Stolleis, Daniel Washington, and
Melanie E. Moses

University of New Mexico,
Albuquerque, NM 87111
`{jhecker,mmoses}@cs.unm.edu`
`{kletendr}@unm.edu`
`http://cs.unm.edu/`

Abstract. Ants use individual memory and pheromone communication to achieve effective collective foraging. 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 at random and in clustered distributions. Teams of three robots that forage based on individual memory without communication collect RFID tags from all three distributions approximately twice as fast as a single robot using the same strategy. Adding pheromone-like communication in the teams of three robots improves foraging success. Our simulation system mimics the foraging behaviors of the robots and replicates our results, with slight improvements in the three robot teams. 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 robotic swarms 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.

Keywords: swarm intelligence, robots, agent-based models, social insect foraging, genetic algorithms

1 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 [4, 5, 8]. 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 [4, 8, 21]. Biologically-inspired decentralized approaches in particular have enhanced scalability and robustness by removing single points of failure from communication bottlenecks and rigid control structures. Thus far such approaches have not yet reached the level of emergent coordination observed in natural systems [28].

Our contribution is inspired by colonies of seed harvester ants who forage for seeds in a desert environment using a combination of individual memory and information sharing through pheromone trails. Our robots are equipped with a sensor suite which mimics the real ants: time-based odometry approximates physical location analogous to the ants' stride integration [33], and ultrasound ranging measures distance to objects and corrects for drift similar to an ant's landmark-based navigation [16]. Like ants, the robots use individual memory and communication of previously successful search locations to improve search performance. Our robots search for radio-frequency identification (RFID) tags, and upon finding them, return to a central nest.

The search algorithm utilized by individual members of the swarm is 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 [11, 18]. We duplicate parameters from the ant model in the robots. For example the robots movement during uninformed search replicates the correlated random walk of virtual ants that was evolved by the GA to produce colonies that find seeds quickly. 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 virtual implementation allows us to compare results from identical experiments *in machina* as implemented in physical robots and *in silico* in the ABM (as in [7, 19]. We conduct additional experiments with the ABM in which we scale up the size of the swarm, the number of tags, and the size of the area in which the virtual robots search. Because we see similar foraging success in simulated and robotic swarms with 1 and 3 individuals, these trials suggest future capabilities of swarms of 30 and 100 robots.

2 Background

2.1 Swarm Robotics

Swarm robotics is necessitated by problems that are inherently too complex or difficult for a single robot, and by the need to develop systems that are cheaper, more adaptive, and robust to failures, errors and dynamic environments [5, 8]. 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 [19, 20, 7]: sensors are imperfect and may fail, collisions with obstacles (including other robots) are common, and real environments are dynamic, changing in response to external factors and the activities of the robots themselves. Further, approaches must balance the benefit of centralized information exchange with the scalability of decentralized approaches [24, 2, 27]. Even highly decentralized robot interactions show diminishing returns in which interference between robots can make swarm efficiency decrease as the swarm size grows [17].

Recent work has demonstrated the feasibility of swarms in which collectively intelligent behaviors emerge from distributed interactions among robots. Simultaneous localization and mapping (SLAM) enables robots to infer knowledge about unknown environments [10, 1, 22]. Localization space-trails (LOST) facilitate a shared world view between robots without a global coordinate system through the use of local landmarks and waypoints [32].

Simple low-cost platforms have been designed specifically to form robot swarms, e.g., [21, 7, 6, 30], but a great challenge exists in transforming a set of simple mobile components into a functional swarm. Robotic swarms have not yet approached the emergent intelligence of biological swarms [28], but a promising approach is to use evolutionary algorithms to determine the parameters of individual behavior that result in effective collective action [26, 9, 31, 12].

2.2 Biological Ants

Our algorithms are largely inspired by foraging in desert seed-harvester ants of the genus *Pogonomyrmex* [13]. 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 searching behavior. The forager searches by moving in a correlated random walk, where the probability of turning is dependent on whether the forager expects to find seeds in the area (informed by pheromone trails or previous foraging success) or not. An informed ant has an initially high tendency to turn, keeping the ant in a small area. Over time, if a seed is not found, degree of turning decreases, which straightens out the search path, and the ant tends to wander farther from its initial search location [12].

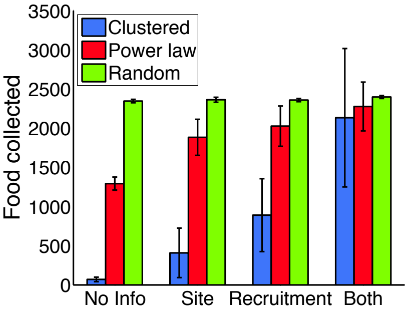
An ant with no prior expectations of finding seeds will use a smaller degree of turning and therefore explore a relatively larger area. When a forager finds a seed, it brings it directly back to the nest. Foragers often return to the location where they previously found a seed, in a process called site fidelity[23, 3, 13]. Seeds are hard to find, so the duration of a foraging trip, which includes travel time and search time, is dominated by the time spent searching for a seed [23, 3]. Effective search strategies for foragers will minimize the time spent searching for seeds (which minimizes the risk of foraging in the hot, dry desert) while maximizing the number of seeds collected. It is unclear exactly how often these ants lay and follow pheromone trails [16, 15, 25], but our recent work indicates laying and following pheromone trails to dense piles of food may be an effective component of these ants' foraging strategies [18, 12].

2.3 Agent-based Model

We have used Genetic Algorithms (GAs) to find the optimal balance of site fidelity and pheromone communication in simulated ant colonies [18]. We simulated ant foraging using a set of agent-based models (ABMs) of foragers on a grid, with parameters optimized by a GA to specify how ants travel from the nest, search, and use site fidelity and pheromone communication. GAs are an optimization technique that simulates the process of evolution by natural selection [14], just as biological ants undergo evolutionary pressure to maximize foraging success (among other goals and constraints). Therefore GAs were an appealing method for selecting parameters for our ant foraging model [29, 26]. The foraging success of virtual ants evolved by the GA is shown in Figure 1.

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 real world parameters. All *in machina* experiments have been duplicated *in silico*, and results are presented side by side to allow comparison.

Fig. 1: Bars represent number of seeds collected during simulated foragin trials by colonies of 100 foragers. Colonies forage on clustered, random, and power law distributed, after optimization by GA to mazimize food collection rate on those distributions. Simulations using site fidelity, pheromone recruitment, both methods together, or neither (no information use) are used as the fitness function in a GA that selects parameters governing travel from the nest, turning during the foragers' search behavior, and use of site fidelity and/or recruitment.



3 Methods

3.1 Hardware

While our algorithms and architecture are intended to be used with a variety of platforms, swarm performance will depend on the particular specifications of the hardware on which those algorithms are implemented. Our robots are built using easily obtained off-the-shelf components (Table 1) at a total cost of $450 per robot. System architecture is based on the Arduino open-source hardware platform, allowing for straightforward programming in a C++-style language based on Wiring. Sensor error is described in Table 2.

3.2 Search Algorithm

The search behavior used by the robots to locate RFID tags is shown in Fig. 2.

1. Set Search Site Location: The robot begins at the nest in the center and selects an initial search site location, encoded as a direction, d, and heading, h. This location is initially chosen at random, but may be influenced by memory or communication in subsequent foraging trips.
2. Travel to Search Site(yellow path) Travelling robots iterate through behaviors to avoid collisions with other robots, correct for motor drift, and communicate events with the coordination server.
3. Search for Tag (blue path): The robot moves in a correlated random walk with direction at time t drawn from a normal distribution centered around direction θ_{t-1} and standard deviation $SD = \omega + \gamma / t_s^{\delta}$, where ω determines the degree of turning during an uninformed search (i.e. at a random location), and γ / t_s^{δ} determines an additional degree of turning at the beginning of an informed search, and which decreases over time spent searching. Equation 1 results tight turns in an initially small area that expand to explore a larger area over time.
4. Travel to Nest (pink path): The robot leaves the location of the found tag, stepping toward the known nest location. The robot lays a pheromone on its return trip if count C of other tags detected in the 8-cell neighborhood of the collected tag is > 1. Pheromone evaporates exponentially with time.

Table 1: Robots components

Component	**Description**
Chassis	The Open Source Robotics OSbase chassis is a four-motor, treaded differential drive platform powered by a 7.4V LiPo rechargeable battery
Microprocessor	The Arduino Uno is an open-source, low-cost development board for the Atmel ATmega328 microprocessor. The ATmega328 is an 8-bit, 16 Mhz processor with 32K of onboard memory for program storage, and +5V logic and onboard power regulation
Motor Shield	The SparkFun Electronics L298 H-Bridge motor driver board controls the four onboard motors. The shield attaches via stacking headers onto the Arduino Uno
Wireless Shield	The SparkFun Electronics WiFly Shield provides wireless communication via standard 802.11b/g TCP protocol through the Roving Networks RN-131C module. Control is via the Uno SPI bus
Compass	The SparkFun Electronics HMC6352 digital compass board uses the Honeywell 6352 compass chip to report magnetic headings with a published accuracy of 2.5°. Module communication is via Arduino Uno TWI bus
GPS	The US GlobalSat EM-406a GPS receiver has a ceramic chip antenna and reports data via NEMA-formatted strings over the Arduino Uno hardware serial port with a 1Hz update rate. The GPS is not used in these experiments
Ultrasound	The Devantech SRF-05 ultrasonic rangefinder provides distance measurements up to 4 meters and communicates via two standard Arduino Uno digital pins
RFID Reader/Writer	The Parallax RFID module reads and writes data using standard 125kHz RFID tags and communicates over Arduino Uno pins using serial port emulation via the Arduino library

Table 2: Sensor error: Compass data show errors in degrees at a fixed heading. GPS data measures error from true location across 5000 data points collected in each location. Odometry data are from 20 trials for each location and measures ability to navigate to a point 5 meters away using programmed motor turning rate, time and compass heading. Ultrasound errors are measured from a concrete barrier 0.5 and 1.5 meters away.

	Location 1	**Location 2**
Compass (°)	2.5 ($\sigma = 1.7$)	3.0 ($\sigma = 0.78$)
GPS (m)	6.6 ($\sigma = 3.6$)	13 ($\sigma = 5.7$)
Odometry (cm)	20 ($\sigma = 7.7$)	22 ($\sigma = 5.5$)
Ultrasound (cm)	1.3 ($\sigma = 0.38$)	4.1 ($\sigma = 4.1$)

5. Set Next Search Location: On subsequent trips, d and h are determined by either returning to the previously found tag location if $C > 0$, or following a pheromone to a location identified by another robot.

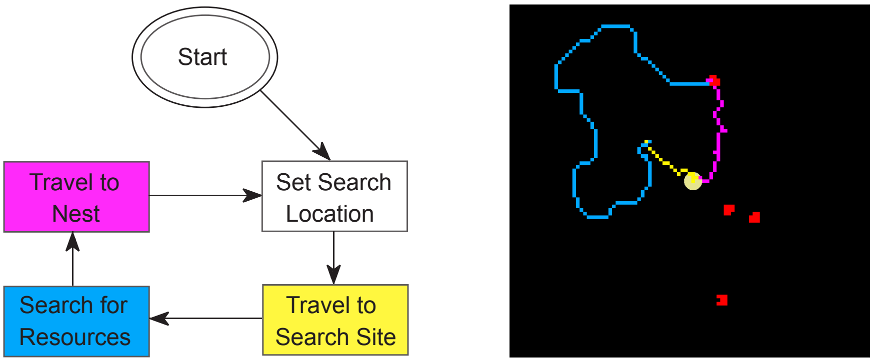


Fig. 2: A robot begins its search at a globally shared central nest site (center circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for tags** (blue line) until tags (red squares) are found or a probabilistic timeout occurs. After searching, the robot **travels to the nest** (purple line).

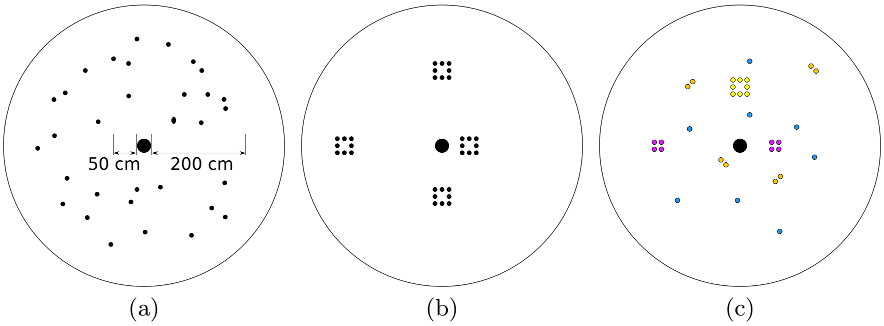
3.3 Experimental Design

We conducted experiments on outdoor concrete surfaces. Each trial runs for a maximum of one hour. A cardboard cylinder marks the center point and represents a home or 'nest' to which the robots return once they have located a tag. This center point is used for localizing and error correction by the robots' ultrasonic sensors. All robots involved in a trial are initially placed near the cylinder to minimize dead reckoning error. We program each robot to stay within a 3m radius 'virtual fence' to deter drift outside of the experimental area.

In every experiment, 32 RFID tags are arranged in one of three different patterns: random, clustered, or power law. The random layout has tags scattered throughout a ring between 50 cm and 200 cm in a uniform distribution (Figure 3(a)). The clustered layout has four piles of eight tags placed at 90° intervals at 50, 100, 150, and 200 cm in relation to the central nest (Figure 3(b)). The power law layout 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 (Figure 3(c)). Experiments are replicated under identical conditions for individual robots and for groups of three bots.

Robot locations are continually transmitted over WiFi 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. Note that tags can only be read once, simulating seed retrieval. The central server also acts as a coordinator for virtual pheromone trails. 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. Each location's associated pheromone value is decayed over time by the server; 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 randomly selects a location from the list (if available) and transmits it to the robot.

Our simulations are design to replicate the behavior of the robots and their experimental area. We measured the physical dimensions of the robots, their speed while traveling and searching, and the range over which their RFID reader can detect an RFID tag. We built the simulation with spatial dimensions that reproduce the properties of the robot, their 3-m radius experimental area, and the distribution of tags in this area. Like the real robots, simulated robots avoid collisions by turning to the right to move past other robots. We allow the simulated robots to search for tags for an amount of time equivalent to an hour, which we calibrated by the speed of the robots as they search and travel around the experimental area. In addition to simulating the 3-m radius area to which the physical robots were restricted, we also simulated the behavior of the robots in a much larger area in which movement is not restricted to 3 m of the nest, and 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- and 3-robot swarms, and also scaled up to 30 and 100 robot swarms to observe the scaling properties of the system.



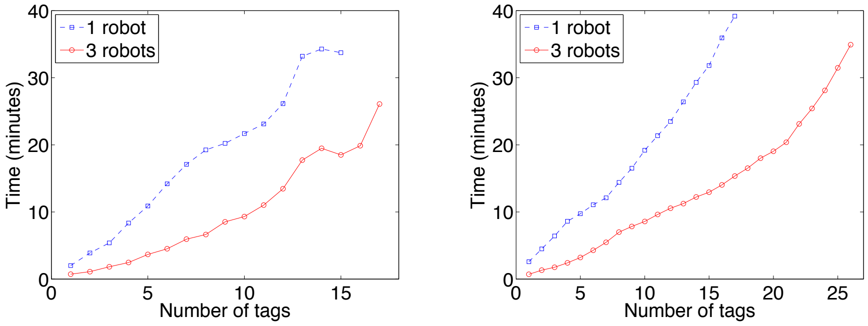
(a) (b) (c)

Fig. 3: 32 RFID tags layed out in (a) random, (b) clustered, and (c) power law distributions.

4 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, result for each experimental treatment are averaged over five robot experiments and twenty experiments in simulation. Error bars indicate one standard deviation of the mean.

Time to collect 32 randomly distributed tags in 5 physical and 20 virtual experiments is shown in Figure 4. 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. The simulated experiments show slightly better scaling than the real robots. It is not surprising that simulated teams of 3 robots are faster than real teams of 3 robots because real robots have more difficulty with avoiding each other, physical hardware limitations, imperfect localization and the possibility that real robots confuse each other with the nest.



(a) Physical (b) Virtual

Fig. 4: Time to collect tags in a random distribution for one and three robots in physical (5 replicates) and virtual (20 replicates) experiments.

As more tags are found over the course of each experiment, it becomes increasingly difficult to find new, remaining tags, and the foraging rate sometimes decelerates. This ceiling effect limits our ability to observe differences between 1 and 3 robot teams. Figure 5 shows data from both physical and virtual experiments for one and three robots. We also analyzed time to collect 25% of the tags from the random, clustered, and power law distributions. We observe improved performance with three robots which collect 25% of the tags 2.8 times faster than one robot in the physical experiment and 2.3 times faster in the model.

Figure 6 illustrates the the rate of tag collection per minute of experiment time for physical and virtual swarms. Each bar denotes the collection rate for a swarm size over a particular tag distribution. This provides a normalized comparison between swarm sizes as well as distributions, regardless of overall experiment runtime which may vary between trials. We were not able to distinguish a significant effect of tag distribution 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 afforded by simulation (GLM: $p < 0.001$; $n = 120$). In the simulations, the greatest rate of tag collection was in the clustered distribution, followed by the power law distribution, followed by random. Note that this the reverse of the pattern with respect to distribution in Figure 1, a result of greater overall density of tags in the robots' experimental area, and therefore greater ease of discovering piles, relative to the food densities used in our previous modeling work.

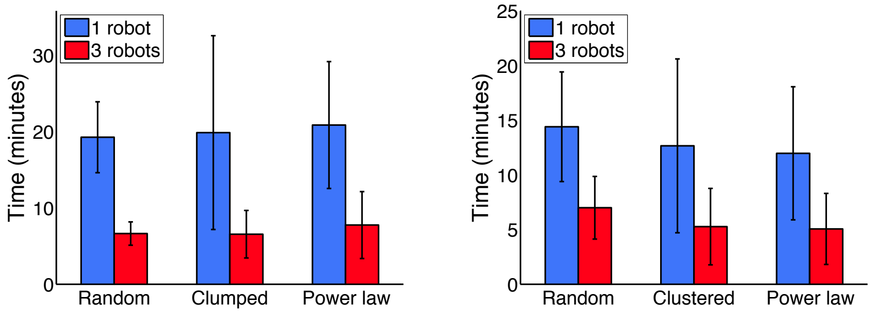
We look at the effects of pheromone trails on tag collection rate in Figure 7. Figure 7(a) compares physical and virtual results for three robots using pheromones while searching for tags in a power law distribution. Results from the physical experiment are averaged over three trials. Figures 7(b) and 7(c) show simulated results for 1, 3, 30, and 100 robots collecting power law distributed tags in an unbounded world.

5 Discussion

We have used Agent Based Models (ABM) and Genetic Algorithms (GA) to translate foraging behaviors of seed harvesting ants into algorithms for robotic swarms searching for RFID tags. We tested two sets of algorithms: one in which robots rely on individual memory of locations of previously found tags (mimicking site fidelity), and one in which robots communicate locations of previously found tags (mimicking pheromones) as waypoints to a central server that acts as the robots' nest'. We tested each approach in single robots and teams of 3 robots, and observed that 3 robots find tags approximately twice as fast as 1 robot when using site fidelity. Pheromone-like communication improves foraging success robots in simulation. We did not observe that pheromones improved foraging in real robots, but in addition to small sample size, we attribute the lack of success primarily to errors that were propagated by miscommunication. Pheromones decrease performance when robots get lost and communicate incorrect locations to other robots. In simulation we found that pheromones improved foraging in 3-robot teams by 10% to 50% (depending on distribution) over site fidelity alone. Additionally, in simulation, the combination of pheromones and site fidelity provided an approach that is scalable to swarms of 100 robots. We suspect that improving the robots' ability to navigate will reduce this problem. The close correspondence between simulation and real robots in smaller swarms make us optimistic that these results could be replicated in large robot swarms.

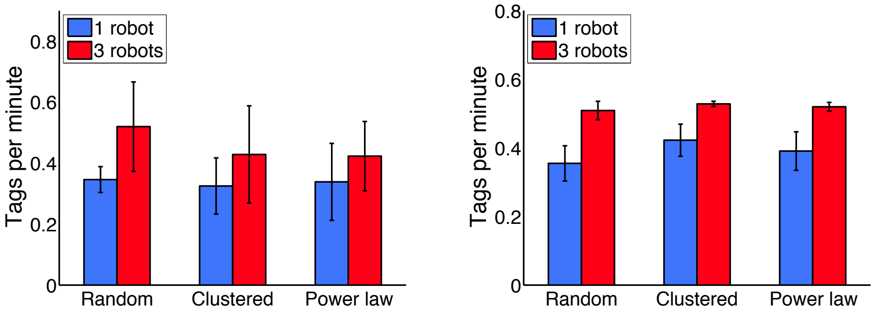
As in the ants, we found that site fidelity is an effective strategy for foraging. This behavior has several benefits. First, it is extremely simple and easily encoded into very simple devices, including devices much simpler than the robots we used here. Second, the approach is highly parallelizable because it requires no communication among robots. Third, it leads to effective and small teams.

Our simulations of ants and our simulations of robots show that adding pheromone communication increases foraging success, particularly on clustered distributions (Figs. 1, 6). We demonstrated that it is possible to implement pheromone communication in robots by having robots report the location where they found a tag to a central server if the robot saw at least 2 additional tags in the vicinity. The server then 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. Simulations show more success with pheromones because simulated ants don't get lost or miscommunicate. Simulations suggest that this approach is highly scalable. When we scale up to 100 robots, each robot is about half as efficient as a single robot, meaning that teams of 100 robots collect resources 50 times faster than a single robot (Fig. 7). This per-robot decline is largely due to the



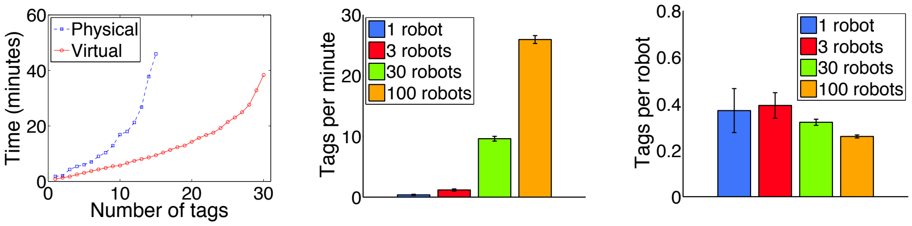
(a) Physical (b) Virtual

Fig. 5: Time to collect 25% of the tags from three different distributions for one and three robots.



(a) Physical (b) Virtual

Fig. 6: Rate of tag discovery calculated as total tags found normalized by experiment length in minutes.



(a) (b) (c)

Fig. 7: Effects of using pheromone trails on tag collection

increased distance the simulated robots travel–an unavoidable consequence of central place foraging.

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 approach, similar to the approach by [7] helps to lay a foundation to further explore the interplay between simulation and experiments with real robots. Our next steps are to use GAs to optimize parameters that lead to maximum efficiency and/or robustness in the ABM, and then import those parameters into the robots. For example, currently the robots' turning angles during their random walk are based on a rough approximation of how our simulated ants evolved to forage from GAs. In future work, we will take the same approach and evolve optimal parameters given the physical attributes of the robots already encoded in the ABM. We will also evolve parameters to determine the optimal balance between reliance on individual memory versus pheromone communication. We will extend our analysis to different kinds of resource distributions, including ones that may be dynamic by encoding tags with resources that appear and disappear over time. Finally, this work will be extended by simulating and replicating in our robots, features of some large ant colonies–the use of mobile nests (as exemplified by army ants) and the use of multiple nests (as exemplified by invasive argentine ants).

References

1. T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, 2006.
2. 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.
3. B. Beverly, H. McLendon, S. Nacu, S. Holmes, and D. Gordon. How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behavioral Ecology*, 20(3):633–638, 2009.
4. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems.* Oxford University Press, USA, 1999.
5. Y. Cao, A. Fukunaga, and A. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.
6. M. Dorigo, M. Birattari, et al. Swarmanoid, the movie. In *AAAI-11 Video Proceedings.* AAAI Press, 2011. Winner of the "AAAI-2011 Best AI Video Award".
7. M. Dorigo, D. Floreano, et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. Technical report, Technical Report TR/IRIDIA/2011-014, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2011.
8. M. Dorigo and E. Sahin. Swarm robotics–special issue editorial. *Autonomous Robots*, 17(2-3):111–113, 2004.
9. M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. Labella, G. Baldassarre, S. Nolfi, J. Deneubourg, F. Mondada, D. Floreano, et al. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2):223–245, 2004.
10. H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.
11. T. Flanagan, K. Letendre, W. Burnside, G. Fricke, and M. Moses. How Ants Turn Information into Food. *Proceedings of the 2011 IEEE Conference on Artificial Life*, pages 178–185, 2011.

12. T. Flanagan, K. Letendre, W. Burnside, G. Fricke, and M. Moses. How Ants Turn Information into Food. *Proc. of the 2011 IEEE Conf. on ALife*, 2011.
13. T. Flanagan, K. Letendre, and M. E. Moses. Quantifying the Effect of Colony Size and Food Distribution on Harvester Ant Foraging. *PLoS ONE*, in review.
14. D. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley Professional, 1989.
15. D. Gordon. The spatial scale of seed collection by harvester ants. *Oecologia*, 95(4):479–487, 1993.
16. B. Hölldobler. Recruitment behavior, home range orientation and territoriality in harvester ants, Pogonomyrmex. *Behav. Ecol. and Sociobio.*, 1(1):3–44, 1976.
17. M. Krieger, J. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, 2000.
18. K. Letendre and M. E. Moses. Ant foraging strategies: Site fidelity and recruitment alone and in combination. in review.
19. R. Mayet, J. Roberz, T. Schmickl, and K. Crailsheim. Antbots: A feasible visual emulation of pheromone trails for swarm robots. *Swarm Intell.*, pages 84–94, 2011.
20. C. Moeslinger, T. Schmickl, and K. Crailsheim. Emergent flocking with low-end swarm robots. *Swarm Intelligence*, pages 424–431, 2011.
21. F. Mondada, G. Pettinaro, I. Kwee, A. Guignard, L. Gambardella, D. Floreano, S. Nolfi, J. Deneubourg, and M. Dorigo. SWARM-BOT: A swarm of autonomous mobile robots with self-assembling capabilities. In *Proc. of the Intl. Workshop on Self-organisation and Evolution of Social Behaviour*, pages 307–312, 2002.
22. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.
23. M. Moses. *Metabolic scaling from individuals to societies.* PhD thesis, University of New Mexico, 2005.
24. M. Moses and S. Banerjee. Biologically Inspired Design Principles for Scalable, Robust, Adaptive, Decentralized Search and Automated Response (RADAR). *Proceedings of the 2011 IEEE Conference on Artificial Life*, pages 30–37, 2011.
25. J. Mull and J. MacMahon. Spatial variation in rates of seed removal by harvester ants (Pogonomyrmex occidentalis) in a shrub-steppe ecosystem. *American Midland Naturalist*, pages 1–13, 1997.
26. S. Nolfi and D. Florin. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines.* MIT Press, 2000.
27. L. Parker. Designing control laws for cooperative agent teams. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 582–587. IEEE, 1993.
28. A. Sharkey. Robots, insects and swarm intelligence. *Artificial Intelligence Review*, 26(4):255–268, 2006.
29. R. Solé, E. Bonabeau, J. Delgado, P. Fernández, and J. Marín. Pattern formation and optimization in army ant raids. *Artificial Life*, 6(3):219–226, 2000.
30. A. Stranieri, E. Ferrante, A. E. Turgut, V. Trianni, C. Pinciroli, M. Birattari, and M. Dorigo. Self-organized flocking with a heterogeneous mobile robot swarm. In T. Lenaerts, M. Giacobini, H. Bersini, P. Bourgine, M. Dorigo, and R. Doursat, editors, *Advances in Artificial Life, ECAL 2011: 11th European Conference on the Synthesis and Simulation of Living Systems*, pages 789–796. MIT Press, Cambridge, MA, 2011.
31. 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.
32. R. Vaughan, K. Stoy, G. Sukhatme, and M. Mataric. LOST: Localization-space trails for robot teams. *IEEE Robotics and Automation*, 18(5):796–812, 2002.
33. M. Wittlinger, R. Wehner, and H. Wolf. The ant odometer: stepping on stilts and stumps. *Science*, 312(5782):1965, 2006.