DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch153

Evolving Error Tolerance in Biologically-Inspired iAnt Robots

Joshua P. Hecker[1], Karl Stolleis[1], Bjorn Swenson[1], Kenneth Letendre[1,2] and Melanie E. Moses[1,2,3]

[1]Department of Computer Science, University of New Mexico, Albuquerque, NM 87131
[2]Department of Biology, University of New Mexico, Albuquerque, NM 87131
[3]External Faculty, Santa Fe Institute, Santa Fe, NM 87501
{jhecker,melaniem}@cs.unm.edu {kstolleis,kletendr}@unm.edu

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.

Introduction

Multi-agent simulations have been used to evolve behaviors which are then transferred into physical robots (Nelson et al., 2004; Singh and Parhi, 2011). 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 (Brooks, 1992).

A particularly challenging class of problems for multi-robot systems is central-place foraging (Matarić, 1994; Panait and Luke, 2004). 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 (Parker, 1998), land mine detection and removal (Gage, 1995; Kong et al., 2006), search and rescue (Kitano et al., 1999), and extraplanetary exploration (Curtis et al., 2003; Tunstel et al., 2008). 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 (Hecker et al., 2012; Hecker and Moses, 2013). 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.

Previous Work

We conducted manipulative field studies on three species of *Pogonomyrmex* desert seed-harvester ants (Flanagan et al., 2012). 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 (Paz Flanagan et al., 2011).

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 (Letendre and Moses, 2013). 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 (Hecker et al., 2012). 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 (Hecker and Moses, 2013). 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.

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 (Nelson et al., 2004; Singh and Parhi, 2011), or evolved directly in real robots through embodied evolution (Watson et al., 2002). Following principles outlined by Brooks (1991), 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 (Jakobi et al., 1995). 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 (Nolfi et al., 1993; Miglino et al., 1995; Jakobi, 1997).

State-of-the-art robotic simulators such as Stage (Vaughan, 2008) and ARGoS (Pinciroli et al., 2011) 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 Pinciroli et al. (2012) recently modified ARGoS to incorporate an actuator noise model, generating performance matching results from positional error observed in real robots.

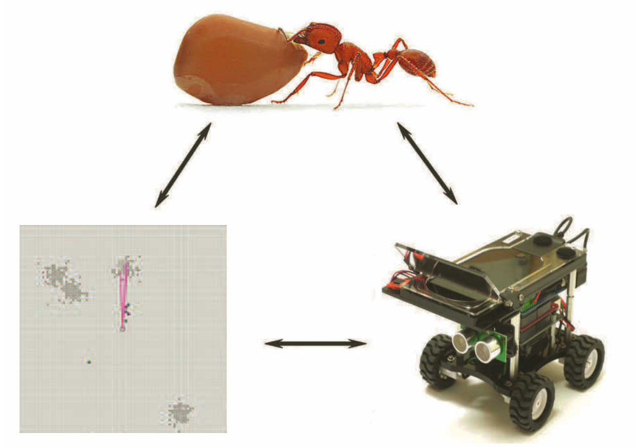


Figure 1: Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform.

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 (Matarić, 1997a,b). 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 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.

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.

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 (Moses, 2005; Beverly et al., 2009; Flanagan et al., 2012). 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 (Paz Flanagan et al., 2011; Letendre and Moses, 2013).

We incorporate key behaviors observed in our previous field studies on desert seed-harvester ants (Flanagan et al., 2012) into our multi-agent simulation and physical iAnt robots. We model probabilistic actions and state transitions using eight evolvable parameters, detailed in Table 1. These are simplifications of our earlier CPFA algorithm (Hecker et al., 2012). Modifications have been made since our most recent work in an effort to increase parsimony (Hecker and Moses, 2013), 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 (Mallon and Franks, 2000), travel distance (Wittlinger et al., 2006), and encounter rates with other ants (Prabhakar et al., 2012). 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.

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 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.

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 maintaining 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.

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 2(a) shows a state diagram of the algorithm, and Figure 2(b) illustrates an example of one possible cycle through the search behavior loop. An iAnt robot is shown in Figure 2(c).

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 $F_{lp}(c) > \mathcal{U}(0,1)$ then
      Create pheromone waypoint for $l_f$
      Pheromones followed by robots at nest
      Pheromones decay over time
    else
      if $F_{sf}(c) > \mathcal{U}(0,1)$ and $F_{fp}(c) < \mathcal{U}(0,1)$ then
        Return to $l_f$
        Conduct informed correlated random walk
      else
        Check for pheromone
        if pheromone found and
        $F_{fp}(c) > \mathcal{U}(0,1)$ and $F_{sf}(c) < \mathcal{U}(0,1)$ 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
```

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.

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

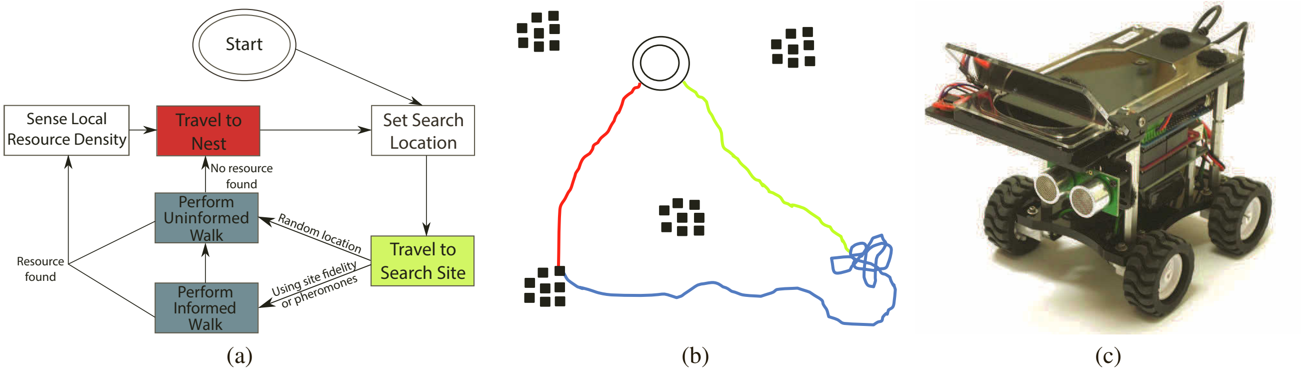


Figure 2: (a) State diagram describing the flow of behavior for individual robots during an experiment, (b) an example of a single cycle through this search behavior loop, and (c) an iAnt robot with Velcro for attaching reflective markers (motion capture was used for a previous experiment (Bezzo et al., 2013), 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).

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 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 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 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.

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 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

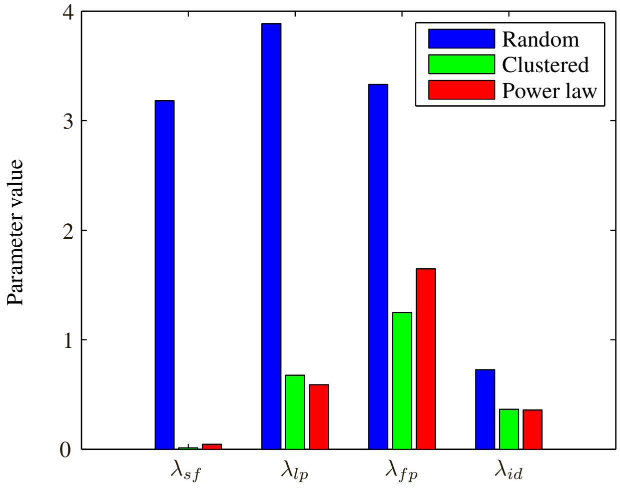
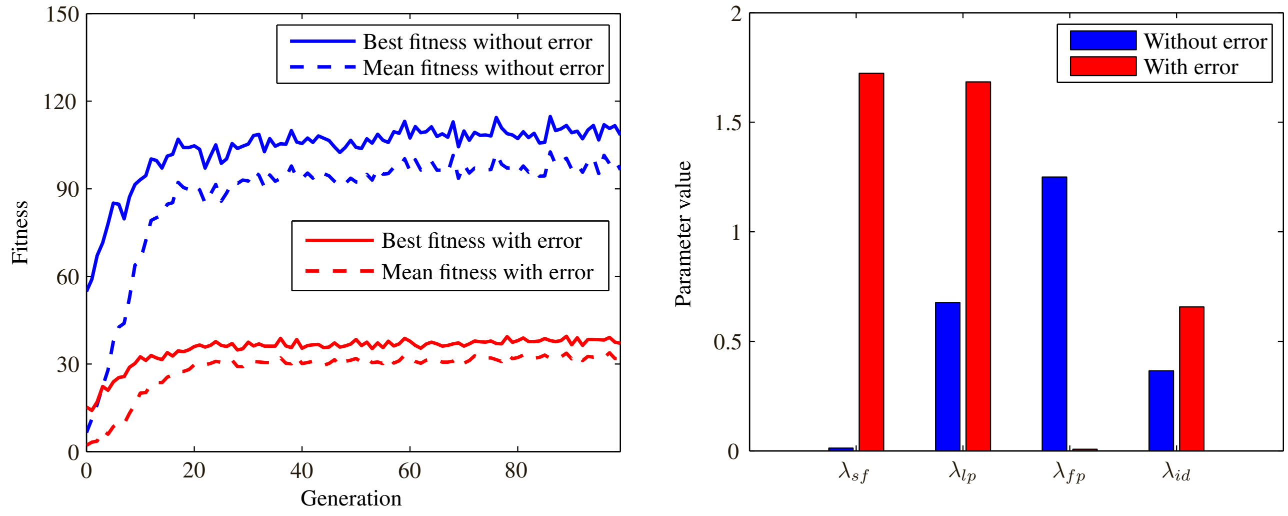


Figure 3: Parameter values for rates of site fidelity (λ_{sf}), laying pheromone (λ_{lp}), following pheromone (λ_{fp}), and informed random walk decay (λ_{id}) for random, clustered, and power law distributed resources.

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 4 shows fitness curves and parameter values adapted for simulated foraging for resources on a clustered distribution. Figure 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 4(b) 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}). 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 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 ($t(103) = 0.16, p = 0.87$).



(a)

(b)

Figure 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}).

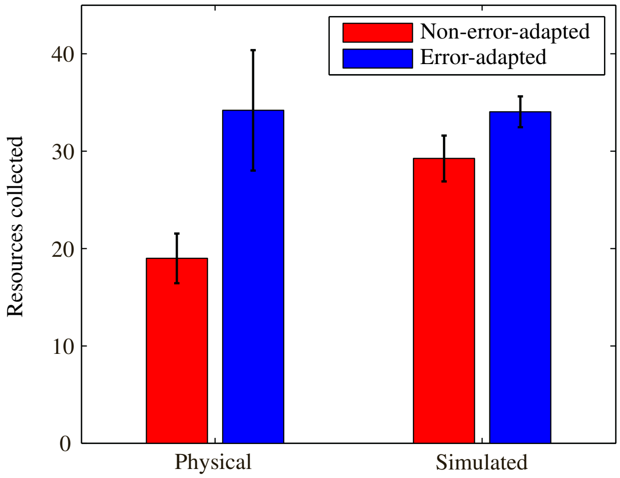


Figure 5: Results for physical and simulated robots foraging in a world with error using parameters adapted for a world with error, and parameters adapted for an error-free world. 80% more resources are collected using error-adapted parameters in physical robot teams, and 16% more are collected in simulated teams. Robots collected significantly more resources in both cases. Physical and simulated robots using error-adapted parameters are not significantly different.

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) 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. 4(a)) demonstrate the ability of the GA to reliably converge. Parameter values (Fig. 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. 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.

Acknowledgments

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

References

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

Bezzo, N., Hecker, J., Stolleis, K., Moses, M., and Fierro,

R. (2013). Exploiting Heterogeneous Robotic Systems in Cooperative Missions. *Robotica (in review)*.

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

Brooks, R. A. (1992). Artifical Life and Real Robots. In *Proceedings of ECAL*, pages 3–10.

Curtis, S. A., Truszkowski, W., et al. (2003). ANTS for Human Exploration and Development of Space. In *IEEE Aerospace Conference*, volume 1, pages 1–161.

Flanagan, T. P., Letendre, K., Burnside, W. R., Fricke, G. M., and Moses, M. E. (2012). Quantifying the Effect of Colony Size and Food Distribution on Harvester Ant Foraging. *PLoS ONE*, 7(7):e39427.

Gage, D. W. (1995). Many-Robot MCM Search Systems. In *Autonomous Vehicles in Mine Countermeasures Symposium*, number April, pages 4–7.

Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex Machina: Ant Swarm Foraging From Physical to Virtual and Back Again. *Swarm Intelligence*, 7461:252–259.

Hecker, J. P. and Moses, M. E. (2013). An Evolutionary Approach for Robust Adaptation of Robot Behavior to Sensor Error. In *GECCO 2013 (in press)*.

Jakobi, N. (1997). Half-baked , Ad-hoc and Noisy : Minimal Simulations for Evolutionary Robotics. In *Proceedings of ECAL*, pages 348–357. MIT Press.

Jakobi, N., Husbands, P., et al. (1995). Noise and The Reality Gap : The Use of Simulation in Evolutionary Robotics. *Advances in Artificial Intelligence*, 704-720.

Kitano, H., et al. (1999). Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In IEEE SMC'99 Conference Proceedings, volume 6, pages 739–743. IEEE.

Kong, C. S., Peng, N. A., and Rekleitis, I. (2006). Distributed coverage with multi-robot system. In *Proceedings of ICRA*, number May, pages 2423–2429. Ieee.

Letendre, K. and Moses, M. E. (2013). Synergy in Ant Foraging Strategies : Memory and Communication Alone and In Combination. In *GECCO 2013 (in press)*.

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

Matarić, M. J. (1994). Learning to Behave Socially. *Proceedings of SAB*, (617):453—-462.

Matarić, M. J. (1997a). Behaviour-based control : examples from navigation , learning , group behavior. *Journal of Experimental & Theoretical AI*, 9(2-3):323–336.

Matarić, M. J. (1997b). Reinforcement learning in the multirobot domain. *Autonomous Robots*, 4:73–83.

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

Moses, M. E. (2005). *Metabolic scaling from individuals to societies*. PhD thesis, University of New Mexico.

Nelson, A., et al. (2004). Evolution of neural controllers for competitive game playing with teams of mobile robots. *Robotics and Autonomous Systems*, 46(135-150).

Nolfi, S., Floreano, D., et al. (1993). How to evolve autonomous robots: Different approaches in evolutionary robotics. *Artificial Life*, 4:190–197.

Panait, L. and Luke, S. (2004). Learning ant foraging behaviors. *Proceedings of ALIFE9*.

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

Paz Flanagan, T., Letendre, K., Burnside, W., Fricke, G.M., and Moses, M.. How Ants Turn Information into Food. *IEEE Conference on ALife*, pages 178–185, 2011.

Pinciroli, C., Trianni, et al. (2011). ARGoS : a Pluggable, Multi-Physics Engine Simulator for Heterogeneous Swarm Robotics. In *IEEE/RSJ Proceedings of IROS*, number December 2010, pages 5027–5034.

Pinciroli, C., Trianni, V., et al. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6:271–295.

Prabhakar, B., Dektar, K. N., et al. (2012). The Regulation of Ant Colony Foraging Activity without Spatial Information. *PLoS Computational Biology*, 8(8):e1002670.

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

Tunstel, E., Dolan, J., et al. (2008). Mobile Robotic Surveying Performance for Planetary Surface Site Characterization. In *Proceedings of PerMIS'08*, pages 200–205.

Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence*.

Watson, R. A., et al. (2002). Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.

Wittlinger, M., et al. (2006). The Ant Odometer: Stepping on Stilts and Stumps. *Science*, 312(5782):1965–1967.