An Evolutionary Approach for Robust Adaptation of Robot Behavior to Sensor Error

Joshua P. Hecker*
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131-0001
jhecker@cs.unm.edu

Melanie E. Moses
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131-0001
melaniem@cs.unm.edu

ABSTRACT

Evolutionary algorithms can adapt the behavior of individual agents to maximize the fitness of populations of agents. We use a genetic algorithm (GA) to optimize behavior in a team of simulated robots that mimic foraging ants. We introduce positional and resource detection error models into this simulation, emulating the sensor error characterized by our physical iAnt robot platform. Increased positional error and detection error both decrease resource collection rates. However, they have different effects on GA behavior. Positional error causes the GA to reduce time spent searching for local resources and to reduce the likelihood of returning to locations where resources were previously found. Detection error causes the GA to select for more thorough local searching and a higher likelihood of communicating the location of found resources to other agents via pheromones. Agents that live in a world with error and use parameters evolved specifically for those worlds perform significantly better than agents in the same error-prone world using parameters evolved for an error-free world. This work demonstrates the utility of employing evolutionary methods to adapt robot behaviors that are robust to sensor errors.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics—*autonomous vehicles, sensors*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems, intelligent agents*

General Terms

Experimentation, Algorithms

Keywords

distributed robotics; evolutionary algorithms; multi-agent systems; biologically-inspired systems

1. INTRODUCTION

Multi-agent simulations have been used to evolve behaviors which are then transferred into physical robots [21, 29]. 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 [3].

A particularly challenging class of problems for multi-robot systems is central-place foraging [15, 23]. 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, and can be instantiated into a number of real-world applications such as hazardous waste clean-up [24], land mine detection and removal [7, 12, 18], search and rescue [11], and extraplanetary exploration [4, 5, 30]. For such 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. We have developed a central-place foraging algorithm (CPFA) based on the foraging behavior of ants.

In our previous work, we conducted manipulative field studies on three species of *Pogonomyrmex* desert seed harvester ants [6]. Colonies were baited with dyed seeds distributed in a variety of pile sizes around each ant nest. We calculated foraging rates for each distribution by monitoring seeds of each color as they were brought into the nest. Results from these experiments showed that ants collected seeds faster when seeds were more clustered. Computer simulations used genetic algorithms to find individual ant behavioral parameters that maximized seed collection rate. 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 [25].

We also observed how individual parameters and overall fitness change with different distributions of resources and different numbers of simulated and physical agents in the simulated central-place foraging task. In [13], 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 40% decrease in fit-

---

*Corresponding author

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.

ness when using parameters on a distribution different from the one for which they were optimized.

In [8], we modified our multi-agent ant-foraging simulation to model the physical environment and hardware constraints of our iAnt robot platform. We adapted our existing genetic algorithm to evolve parameters for our iAnt robots, which were then transferred into the real robots. We showed that teams of three physical robots and simulated agents collected resources twice as fast as a single robot using the same strategy. Simulated teams also collected three to four times as many resources as the real robot teams, and 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.

In this paper, we present a multi-agent simulation which emulates physical robot experiments that mimic foraging ants. Our simulation evolves parameters in a parsimonious model of biological ant behavior. We investigate the effects of sensor error on simulation performance and on individual parameters of the evolved model. We demonstrate the utility of this approach by measuring the number of resources that robots collect in perfect (error-free) worlds and in imperfect worlds (with sensor error) using parameters evolved with and without sensor error. All experiments described here are done in simulation, however, we consider the types of error that characterize our physical iAnt robot platform, and will incorporate parameters from these simulations into physical robots in future work.

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 [21, 29], or evolved directly in real robots through embodied evolution [32]. Following principles outlined by Brooks in [2], 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 [10]. 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 [9, 19, 22].

State-of-the-art robotic simulators such as Stage [31] and ARGoS [26] can be used to model large robot teams with realistic physical kinematics, but 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 ARGoS was modified to incorporate an actuator noise model in [27] and generated 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 [16, 17], rather than lower-level motor controllers or basic directional responses. 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 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. 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.

Parameter	Description	Initialization Function
p_{st}	Probability of stopping travel	$\mathcal{U}(0,1)$
p_{sp}	Probability of stopping following pheromone	$\mathcal{U}(0,1)$
p_{ss}	Probability of stopping search	$\mathcal{U}(0,1)$
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 9 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 4 parameters are initially sampled from a uniform distribution, and the last 5 from exponential distributions within the stated bounds.

3. METHODS

We present our simulated model of ant behavior (Subsection 3.1), detailed pseudocode and diagrams explaining our CPFA (Subsection 3.2), probabilistic models for physical sensor error characterized by the iAnt robot platform and implemented in our multi-agent system (Subsection 3.3), and the design of our simulated experiments (Subsection 3.4).

3.1 Simulated Ant Behavior

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 [1, 6, 20]. 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 [13, 25].

We incorporate key behaviors observed in our previous field studies on desert seed-harvester ants [6] into our multi-agent simulation. We model probabilistic actions and state transitions using nine evolvable parameters, detailed in Table 1. These are simplifications of our earlier CPFA algorithm presented in [8]:

- **State transitions:** Agents switch among three behaviors:
  - **Traveling:** In the absence of information, an agent at the nest will select a random direction and begin traveling. At each step of traveling, agents have a probability p_{st} of transitioning into search behavior.
  - **Trail following:** At each step of trail following, agents have a probability p_{sp} of abandoning the trail before reaching the end and transitioning into (uninformed) search behavior.
  - **Searching:** At each step of searching, agents who have not found a resource have a probability p_{ss} of returning to the nest.
- **Correlated random walk:** Simulated agents explore a two-dimensional grid 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 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 [14], travel distance [33], and encounter rates with other ants [28]. 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 trail, or follow a pheromone trail:
  - **Site fidelity:** An agent 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:** An agent lays a pheromone trail to 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, an agent follows a pheromone trail to a previously found resource location if $F_{fp}(c) < \mathcal{U}(0,1)$, where $F_{fp}(x) = 1 - e^{-\lambda_{fp} * (x+1)}$. Trails are selected with probability proportional to their value.
  - **Pheromone decay:** Pheromone trails decay exponentially over time t as $e^{-\lambda_{pd} * t}$. Trails 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 induce the agents 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.

As with our physical robots, we simulate pheromone 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.

3.2 Search Algorithm

Pseudocode for our biologically-inspired CPFA 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 Subsection 3.1. Figure 1(a) shows a state diagram of the algorithm, and Figure 1(b) illustrates an example of one possible cycle through the search behavior loop.

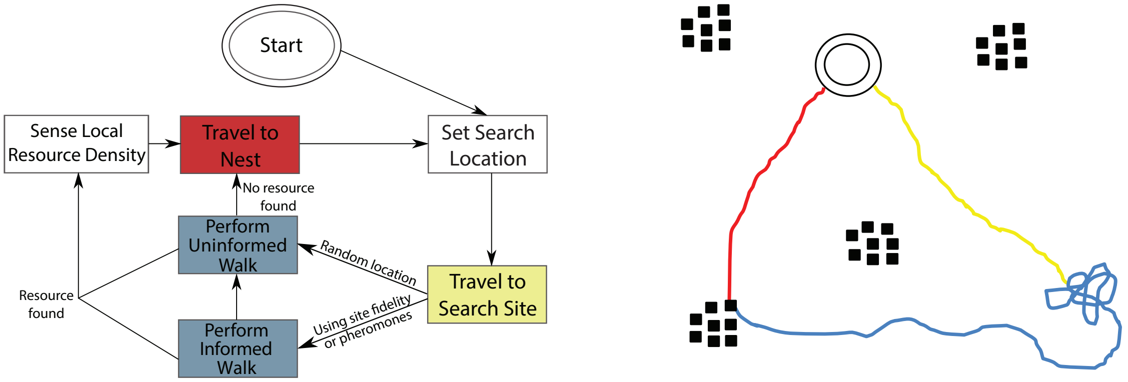
Algorithm 1 Biologically-Inspired CPFA

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

3.3 Physical Sensor Error

Comparisons between simulated and physical robot performance reveal two sensing components which 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. Accuracy measurements show that robots can be as much as 100 cm off from their estimated position, and this error is generally additive over time.



(a)

(b)

Figure 1: (a) State diagram describing the flow of behavior for simulated agents during an experiment, and (b) an example of a single cycle through this search behavior loop. The agent begins its search at a central nest site (double circle) and **sets a search location**. The agent then **travels to the search site** (yellow line). Upon reaching the search location, the agent **searches for resources** (blue line) until a resource (black squares) is found. After searching, the agent **travels to the nest** (red line).

Resource detection is accomplished using a down-facing camera to read barcode-style QR tags. Tag detection error is empirically estimated at approximately 50% based on measurements made during physical testing.

For this paper, we select straightforward probabilistic error models for positional measurement and resource detection in our simulated agents, emulating the error empirically observed in iAnt robots. Positional error is modeled by perturbing the physical position of an agent from (x, y) to (x', y'), such that $x' \sim \mathcal{N}(x, \sigma)$ and $y' \sim \mathcal{N}(y, \sigma)$, where $\sigma \in [0, 100]$. That is, (x', y') is sampled from a normal distribution with mean equal to the true position (x, y) and σ varied from 0 to 100 centimeters. We impose this positional perturbation twice: once when an agent finds a resource, and again when an agent leaves the nest using site fidelity or following a pheromone trail to a known location. This emulates the error a real robot experiences during localization (after finding a tag) and when traveling to a known position.

Resource detection error is modeled as a fixed detection error rate $d_e \in [0, 1)$. A value of 0.0 implies an agent has a 0% chance of missing a resource, whereas a value of 0.4 means an agent has a 40% chance of missing a resource.

3.4 Experimental Design

Simulated teams of six agents search for resources on a 90 x 90 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.

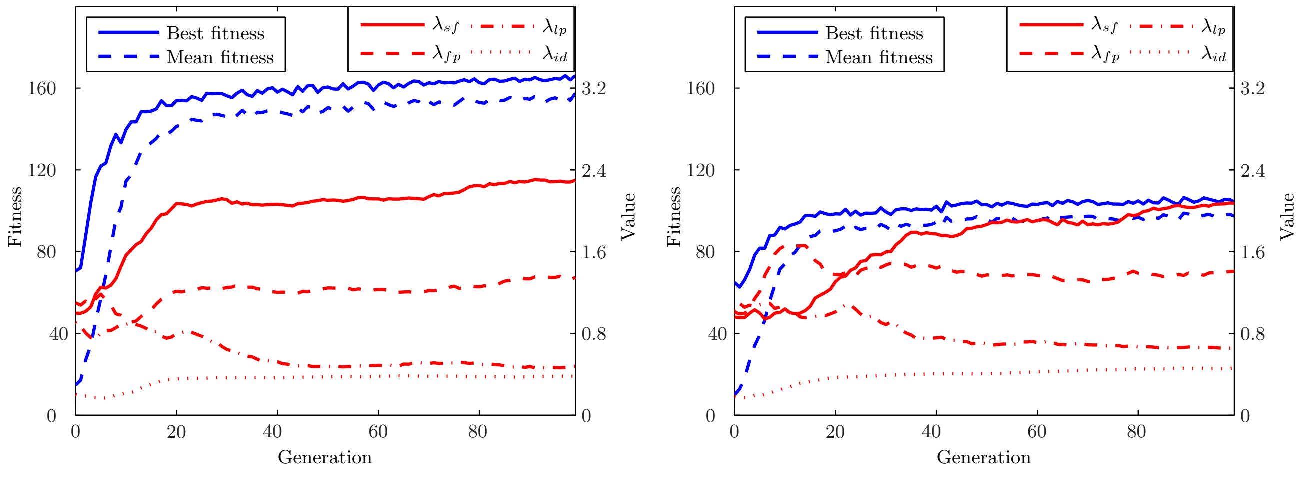
256 identical resources are placed on the grid (each resource occupies a single grid cell) in a power law distribution using clusters of varying size and number: one large pile of 64, four medium piles of 16, 16 small piles of 4, and 64 randomly scattered. Each individual pile (or single resource) is 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 100 teams is evolved for 100 generations using recombination and mutation (details provided in Table 2). Each team's parameter set is randomly initialized using 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 it's own grid, but the grids are identical. During each generation, all 100 teams undergo eight evaluations on randomly placed resource distributions; fitness is evaluated as the sum total of resources collected by each team over an entire generation. Two individuals 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 an error-free world to a world with error. We compare the performance for the swapped case to results using the originally evolved parameters (i.e. the parameters optimized for the worlds with error). For these experiments, we measure the resources collected across 100 randomly placed distributions. 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 simulated robot teams.

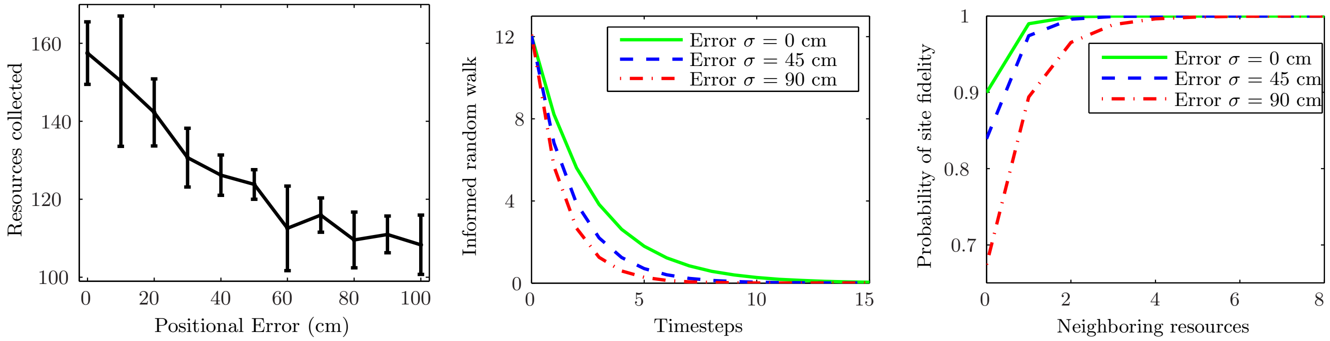
Name	Value
Population size	100 teams of 6 agents
Grid size	90 x 90
Crossover rate	10%
Crossover method	Independent assortment
Generations	100
Evaluations per generation	8
Mutation rate	10%
Selection	Tournament (k = 2)

Table 2: Settings used for the genetic algorithm in our multi-agent simulation. Specific values and methods were selected based on their performance during preliminary experiments.



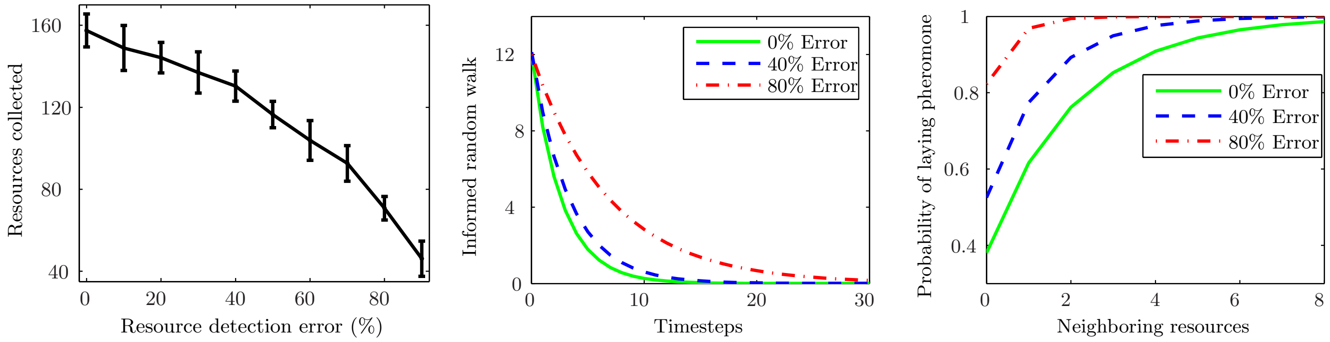
(a) (b)

Figure 2: Best and mean fitness curves and parameter evolution for rates of site fidelity (λ_{sf}), following pheromone (λ_{fp}), laying pheromone (λ_{lp}), and informed random walk decay (λ_{id}). (a) Simulated foraging with no error. (b) Simulated foraging with positional error of 45 cm and resource detection error of 40%.



(a) (b) (c)

Figure 3: Results from simulations incorporating varying amounts of positional error and no resource detection error. (a) Increasing positional error decreases the number of resources collected. (b) The decay rate of an informed walk (λ_{id}) is faster with larger positional error. (c) The rate at which agents use site fidelity (λ_{sf}) is lower with larger positional error.



(a) (b) (c)

Figure 4: Results from simulations incorporating varying amounts of resource detection error and no positional error. (a) Increasing resource detection error decreases the number of resources collected. (b) The decay rate of an informed walk (λ_{id}) is slower with larger resource detection error. (c) The rate at which agents lay pheromone (λ_{lp}) is higher with larger resource detection error.

4. RESULTS

We present results for teams of six simulated robots searching for resources under varying conditions of positional and detection error. Unless otherwise noted, results for each experimental treatment are averaged over ten replicates.

Figure 2 shows fitness curves and parameter evolution for simulated foraging over distributions of resources. Figure 2(a) plots results for an error-free world, including best and mean fitness over 100 generations and parameters influencing agents' use of information ($\lambda_{sf}, \lambda_{fp}$, and λ_{lp}), as well as the informed walk decay rate (λ_{id}). Figure 2(b) shows the same metrics for simulations with positional error of 45 cm and resource detection error of 40%. In both cases, we observe fitness and individual parameters stabilizing after 50 generations. Simulations with error converge to a fitness level approximately 60% of the fitness achieved in simulations without error. The rate of site fidelity (λ_{sf}) is approximately 10% lower, the rate of laying pheromone (λ_{lp}) is 40% higher, the rate of following pheromone (λ_{fp}) is 5% higher, and the informed random walk decay rate (λ_{id}) is 20% higher.

Figure 3 shows how positional error affects resource collection (fitness) and the evolution of parameters governing informed random walks and the probability of using site fidelity. In Figure 3(a), we observe that increasing positional error decreases the amount of resources collected by the simulated teams. Note that this inverse relationship begins to plateau (i.e. the slope of the line nears zero) as positional error approaches 100 cm.

Figure 3(b) shows the decay rate of an informed random walk (λ_{id}) for standard deviations of 0, 45, and 90 cm. Here we see that the informed walk decays faster with larger positional error. Figure 3(c) shows the rate of site fidelity use (λ_{sf}) for the same deviations of positional error. In this case, the probability of using site fidelity given a fixed number of neighboring resources is lower for larger positional error.

Figure 4 shows how resource detection affects resource collection (fitness) and the evolution of parameters governing informed random walks and the probability of laying pheromone. In Figure 4(a), we observe decreasing resource collection with increasing resource detection error. Note that the slope is generally increasing as the detection error rate goes up.

The rate of decay of an informed walk (λ_{id}) for 0%, 40%, and 80% resource detection error is presented in Figure 4(b). Here we observe that an informed walk decays slower with larger error in resource detection. Figure 4(c) shows the rate of laying pheromones (λ_{lp}) for the same detection error rates. The probability of laying a pheromone trail for a fixed number of neighboring resources is higher for larger detection error rates.

We analyze the effects of swapping parameters from a perfect world with no error to worlds with 45 or 90 cm positional error, 40% or 80% detection error rates, and a combination of 45 cm positional error together with 40% detection error. Results, shown in Figure 5, are averaged over 100 replicates for each specific error case. We observe a significant effect for parameter swapping with positional error of 45 cm ($t(198) = 4.5, p < 0.001$) and 90 cm ($t(198) = 5.2, p < 0.001$). We also see a significant effect with resource detection error of 40% ($t(198) = 21, p < 0.001$) and 80% ($t(198) = 32, p < 0.001$). Parameter swapping was significant in the combined error case as well ($t(198) = 2.2, p = 0.030$). In all cases, the parameter sets originally evolved for worlds with error outperformed the parameters evolved for an error-free world: agents collected 2.0% more resources using imperfect parameters evolved for 45 cm positional error, 2.5% more for 90 cm positional error, 11% more for 40% detection error, 29% more for 80% detection error, and 1.1% more for combined 45 cm positional error and 40% detection error.

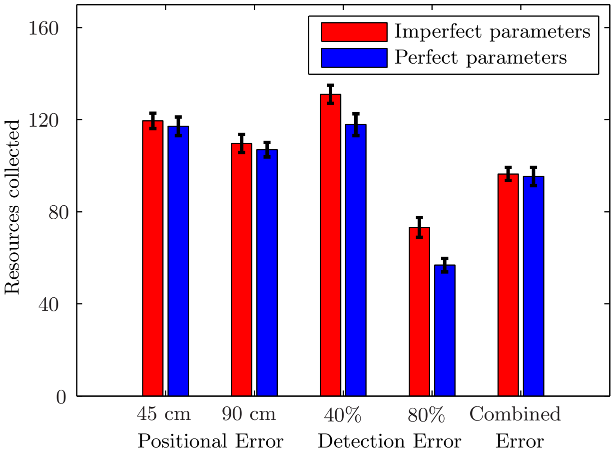


Figure 5: Results for swapping parameters evolved for a perfect world (i.e. error-free) into an imperfect world (i.e. with error) compared to results using parameters evolved for an imperfect world in that (imperfect) world. 2.0% more resources are collected using imperfect parameters evolved for 45 cm positional error, and 2.5% more were collected using parameters evolved for 90 cm positional error. 11% more resources are collected using imperfect parameters evolved for 40% detection error, and 29% more were collected using parameters evolved for 80% detection error. 1.1% more resources are collected using imperfect parameters evolved for a combination of 45 cm positional error and 40% detection error. Agents collect significantly more resources in all cases.

5. DISCUSSION

In this paper, we presented results from experiments with teams of simulated robots using a central-place foraging algorithm (CPFA) to search for resources under varying conditions of sensor error. A genetic algorithm (GA) was used to evolve parameter sets which correspond to the behaviors of the robot teams, inspired by seed-harvester ants. We considered two types of error, positional error and resource detection error, and we explored the effects of both types of error on overall resource collection and on individual evolved parameters. Evolved parameters were swapped from error-free worlds into worlds with error and tested for ability to collect resources in comparison to results using the original parameters.

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 a resource's location can direct 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 laying of a pheromone trail. Robots with positional errors may perform better with faster decay in their 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.

Fitness curves and parameter evolution results (Fig. 2) demonstrate the ability of the GA to reliably converge. The speed of convergence verifies the decision to evolve populations for 100 generations, and the small variance in both fitness and parameter evolution curves (after convergence) supports the usage of 8 evaluations per generation.

Fig. 3(a) shows the effect of positional error on resource collection, and we observe that this effect plateaus as positional error becomes large. We hypothesize that this diminishing effect results from location becoming irrelevant, i.e., positional error is so great that agents are essentially searching at random. The decay rate of informed random walks (Fig. 3(b)) and the rate of site fidelity (Fig. 3(c)) reflect less reliance on positional information when there is more error in that information. We observe that the GA evolves a faster informed walk decay and a lower likelihood of using site fidelity, implying that agents perform better when they spend less time exploiting resource density in imperfect worlds.

Results for resource collection over varying detection error rates (Fig. 4(a)) show that the influence of detection error on resource collection grows with increasing error. This growth occurs because of the additive effect of detection error on counting neighboring resources: agents with a high error rate have both a small chance of finding a resource *and* a small chance of properly estimating the density of resources in the area. The informed walk decay rate (Fig. 4(b)) and the rate of laying pheromone (Fig. 4(c)) compensate for this additive error by compelling agents to be more sensitive to resource density information. The GA evolves a slower informed walk decay rate, allowing more time to find resources when resources are harder to detect. The GA also evolve a higher probability of laying pheromones when a small number of resources are detected, reflecting the likelihood that a large number of resources in that location remain undetected.

Results for parameters swapped from error-free worlds into worlds with error (Fig. 5) show that parameters evolved specifically for imperfect worlds outperform parameters evolved in perfect worlds in all cases. This observation supports the utility of our method, namely, employing evolutionary methods to 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.

However, we observe only 1.1% improvement in resource collection for the combined error case using imperfect parameters (Fig. 5, far right), a smaller performance increase than either positional or detection error alone. In fact, the parameters evolved for worlds with combined error (Fig. 2(b)) are similar to those evolved for worlds with no error (Fig. 2(a)), which explains this small increase in performance. We hypothesize that the opposing influence of positional error versus detection error on the evolved parameters confounds the ability of the GA to improve performance in worlds with both types of error. For example, the informed random walk decay rate decays faster with increasing positional error (Fig. 3(b)) and slower with increasing detection error (Fig. 4(b)). We plan to simplify the agents' behaviors in future work; we are confident that simplifying the GA's fitness landscape will resolve these confounding effects.

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 perform parameter swapping experiments in real robots identical to the simulation tests presented here. iAnt data will be collected on physical sensor error observed during localization, traveling, and resource detection. The error will be incorporated in simulations, and parameters will be evolved to maximize fitness of iAnts with error. Physical experiments will be conducted by transferring parameters from simulated worlds with and without error into the real robots, then observing their ability to collect resources using each parameter set.

6. ACKNOWLEDGMENTS:

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

7. REFERENCES

[1] B. D. 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.

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

[3] 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, 1992.

[4] S. A. Curtis, J. Mica, J. Nuth, G. Marr, M. Rilee, and M. Bhat. ANTS (Autonomous Nano-Technology Swarm): An artificial intelligence approach to asteroid belt resource exploration. Technical report, 2000.

[5] S. A. Curtis, W. Truszkowski, M. L. Rilee, and P. E. Clark. ANTS for Human Exploration and Development of Space. In *IEEE Aerospace Conference*, volume 1, pages 1–161, 2003.

[6] 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.

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

[8] 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. *Swarm Intelligence*, 7461:252–259, 2012.

[9] N. Jakobi. Half-baked , Ad-hoc and Noisy : Minimal Simulations for Evolutionary Robotics. In *Fourth*

European Conference on Artificial Life, pages 348–357. MIT Press, 1997.

[10] N. Jakobi, P. Husbands, and I. Harvey. Noise and The Reality Gap : The Use of Simulation in Evolutionary Robotics. *Advances in Artificial Intelligence*, 704-720, 1995.

[11] 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 *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 739–743. IEEE, 1999.

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

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

[14] 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.

[15] M. J. Matarić. Learning to Behave Socially. *Third International Conference on Simulation of Adaptive Behavior*, (617):453—-462, 1994.

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

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

[18] J. McLurkin. Using Cooperative Robots for Explosive Ordnance Disposal. *Massachusetts Institute of Technology Artificial Intelligence Laboratory*, pages 1–10, 1996.

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

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

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

[22] 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.

[23] L. Panait and S. Luke. Learning ant foraging behaviors. *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, 2004.

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

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

[26] C. Pinciroli, V. Trianni, R. O. Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, T. Stirling, A. Guti, L. M. Gambardella, and M. Dorigo. ARGoS : a Pluggable , Multi-Physics Engine Simulator for Heterogeneous Swarm Robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, number December 2010, pages 5027–5034, 2011.

[27] 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.

[28] 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.

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

[30] 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.

[31] R. Vaughan. Massively multi-robot simulation in stage. *Swarm Intelligence*, 2008.

[32] 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.

[33] M. Wittlinger, R. Wehner, and H. Wolf. The Ant Odometer: Stepping on Stilts and Stumps. *Science*, 312(5782):1965–1967, 2006.