

# Comparing Physical and Simulated Performance of a Deterministic and a Bio-inspired Stochastic Foraging Strategy for Robot Swarms

Qi Lu<sup>\*,1</sup>, Antonio D. Griego<sup>1</sup>, G. Matthew Fricke<sup>1,2</sup>, and Melanie E. Moses<sup>1,3,4</sup>

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

## I. INTRODUCTION

Robot swarms are particularly useful in spatially distributed tasks such as central place foraging, in which robots search for targets and transport them to a collection zone [1], [2]. Swarm foraging algorithms often mimic the stochastic behaviors of social animals, particularly social insects such as ant colonies [3], [4], [5].

In this work, we conduct physical and simulated experiments to compare two collective robot foraging algorithms: the central place foraging algorithm (CPFA) [4] and the distributed deterministic spiral search algorithm (DDSA) [6]. In previous work, these algorithms were compared using the Autonomous Robots Go Swarming (ARGoS) [7] simulator, and it was found that simple robot swarms operating in obstacle-free environments collected resources faster using the DDSA compared to the CPFA [6], at least for swarm sizes of up to 20 robots.

The foraging performance of robots can be measured by the number of targets retrieved in a fixed time. It is important to evaluate collective algorithms in physical robots [2] because it is not feasible to simulate all aspects

of a physical environment [8], and foraging performance can be altered by variable conditions and by sensor and actuation noise that affect localization, object retrieval, and collision avoidance. All of these components of the “reality gap” can alter the performance of algorithms real robotic experiments compared to simulations [9], [10].

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

This work implements two swarm foraging algorithms in the robot operating system (ROS) [11]. We systematically compare the foraging performance of the DDSA and the CPFA by measuring how quickly targets are collected in fixed time. We designed a set of experiments that we replicated in a Gazebo simulation [12] and in physical robots called “Swarmies” that search for, pick up, and collect physical objects (which we call targets) in outdoor arenas with various placements of targets and obstacles.

In previous work we compared the DDSA and the CPFA in intentionally simple simulations implemented in the footbot robot in ARGoS [6]. In contrast, in this paper, we describe simulations implemented in Gazebo that include more realistic physical processes that represent the localization, navigation, sensing, object pickup and drop off, and collision avoidance of the Swarmie robots that we implement in physical experiments. Still, our physical experiments include variability inherent to outdoor environments and sensor and actuator noise that is not fully simulated in our Gazebo simulations.

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

This work was supported by a James S. McDonnell Foundation Complex Systems Scholar Award and NASA MUREP #NNX15AM14A funding.

<sup>1</sup>Computer Science Department, <sup>2</sup>Center for Advanced Research Computing, <sup>3</sup>Biology Department, University of New Mexico, Albuquerque, <sup>4</sup>Santa Fe Institute, Santa Fe, NM, USA

\*lukey11@cs.unm.edu

DDSA is more degraded by conditions in the physical world, suggesting that the CPFA is more tolerant of real-world conditions.

The paper is organized as follows. The related work is presented in Section II. Sections III and IV summarize the CPFA and DDSA algorithms. Section V describes the physical robots and physical and simulated environments. Section VI describes the experiments, with results reported in Section VII. Section VIII discusses the strengths and limitations of stochastic and deterministic search strategies.

## II. RELATED WORK

Though swarm robot foraging has been studied for decades, replicated experimental analyses that compare different algorithms in simulation and in real robots are rare, particularly in outdoor environments [1], [2]. Many task partitioning and foraging algorithms have been simulated in the STAGE [13], [14], [15], the ARGoS [5], [16], [17] and Microsoft(R) Robotics Studio (MRDS) [18]. Physical foraging experiments have been conducted with *Foot-bots* equipped with grippers, IR sensors, and cameras for foraging tasks in [16], [17] and custom platforms like MinDART [19].

In practice, many complex physical experiments with swarm robots require human support [20] or simulation of some aspect of the foraging task. For example, the RoboTarium provides a testbed for remotely accessible physical robots [21], but localization is governed by an overhead camera. Other studies simulate physical pickup and drop-off of objects. For example, [22], [15] uses a group of e-puck robots and our prior work [4] used iAnt robots which detect targets but do not physically pick them up. Kilobots can operate autonomously to push items, but they have relatively limited mobility and only operate in controlled laboratory environments [23], [24]. Collaborative warehouse robots may require buried guide-wires or visual markers to navigate [25]. The Swarmanoid demonstrates an innovative heterogeneous physical swarm robotics system whose robots collaborate to solve a complex object retrieval task [26], but it was designed as a demonstration of swarm capabilities, not to replicate experiments to test algorithms in a physical environment.

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

### III. CENTRAL PLACE FORAGING ALGORITHM: CPFA

With the CPFA, robots mimic the foraging behaviors of *Pogonomyrmex* desert seed-harvester ants, social insects

which have evolved to cooperate without centralized control [27], [28]. Fig. 1a shows how individual robots transition between states in the CPFA based on various conditions (further detailed in [4]). Robots start from the central collection zone and travel towards a randomly selected location (State A) until they switch to searching using an uninformed correlated random walk [29] (State B). If a robot detects targets (Condition 3), it collects the closest one (State D) and measures the number of additional targets within its camera view by rotating 360° (State E). The robot uses the measured targets to decide whether to create a "pheromone waypoint" which adds the location and the strength to a list, mimicking ant pheromone trails [30], [31], [32]. The strength of waypoints decreases over time and waypoints can be added to the list by other robots. Robots communicate pheromone waypoints and may select waypoints probabilistically ranked by strength at the nest.

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

CPFA robots make real-time decisions based on a set of 7 real-valued parameters specifying the probabilities that govern the transitions in Fig. 1a. The CPFA parameters were selected by a genetic algorithm (GA) to maximize the number of collected targets in [4]. This is not feasible given the slow run-time of the Gazebo simulations or physical robots. Instead, we hand-tuned the parameters (included in our Github repository) based on the previously evolved results.

### IV. DISTRIBUTED DETERMINISTIC SEARCH ALGORITHM: DDSA

In contrast to the CPFA, the DDSA takes a geometric approach which exploits the optimality of spiral search demonstrated for single agents [33], [34] generalized to a swarm of robots. Robots using the DDSA start near the central collection zone and search for targets by following a pre-planned pattern of interlocking square spirals. When operating without error, noise, or collisions, the DDSA guarantees that robots will find the nearest targets first which minimizes transport cost. This provides complete coverage

of an area while minimizing repeated searches of the same location.

Each robot's path is calculated based on the number of robots  $r$ , the  $c^{th}$  circuit (where a circuit is one revolution of the spiral), and the distance  $g$  between the interlocking spirals. The distance  $g$  depends on the target detection range of the robot. For the Swarmie robots in these experiments,  $g = 0.41$  m. Equations 1 and 2 calculate the number of steps ( $F$ ) of each spiral path to the north (N) and south (S) directions by each of  $i$  robots on circuit  $c$ .  $c$  increases by one if robots complete their current circuit. Distances travelled east (E) and west (W) are similarly calculated. By solving the recurrence relation given in [6] we can simplify the DDSA formulation to the following two equations:

$$F_{c,i}^N = F_{c,i}^E = \begin{cases} i & \text{if } c = 1 \\ (2c - 3)r + 2i & \text{if } c > 1 \end{cases} \quad (1)$$

$$F_{c,i}^S = F_{c,i}^W = \begin{cases} 2i & \text{if } c = 1 \\ F_{c,i}^N + r & \text{if } c > 1 \end{cases} \quad (2)$$

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

In each subsequent foraging trip, the robot returns directly to the last location where it found a target (effectively implementing site fidelity for every foraging trip) where it resumes its spiral search. This relatively simple algorithm guarantees that the closest targets are found first, and due to site fidelity a robot will repeatedly return to a location so that it efficiently collects clusters of targets [35].

## V. DESCRIPTION OF SIMULATED & PHYSICAL ROBOTS

Our experiments run in a Gazebo [36] simulation and in outdoor arenas using the Swarmie robot platform, all of which were custom designed and built for the NASA Swarmathon swarm foraging competition [37].

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

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

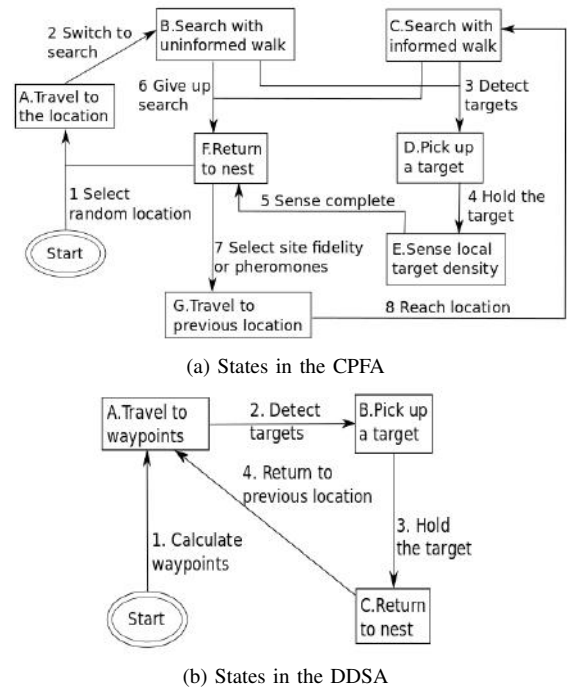


Fig. 1: Robot states in the CPFA and DDSA.

users. Complete build instructions for the Swarmie robot are publicly available<sup>1</sup>, and source code of the CPFA<sup>2</sup> and DDSA<sup>3</sup> are available on Github.

The targets collected by the robots are soft cubes with an AprilTag (2D barcode fiducials developed for robotics applications [38]) on each face. For the first two sets of experiments, targets were distributed in the arena in clusters of various sizes with locations determined at random. The collection zone in the center of the arena was a square area with AprilTags on its boundary. The camera detected the AprilTags which were translated into a location in space relative to the robot's position. This allows the robot to pick up targets in the arena and drop them off in the collection zone. Physical robots used a gripper with an actuated wrist for grabbing and dropping targets, which was also simulated in Gazebo. Target collection is an error-prone complex task. The average number of attempts to pick up a target is  $1.85 \pm 1.2$  in simulation and is  $1.96 \pm 1.2$  in physical experiments. In physical experiments, although robots attempted to visually confirm that a target was successfully picked up, robots sometimes drop targets or detect that a target was collected when it was not. On rare occasions targets were dropped after a collision or robots would steal targets from each other. More commonly, once targets were deposited in the collection zone, robots could accidentally push them out again. We manually counted and then removed collected targets from the collection zone to avoid these accidents. Recognition of targets and the collection zone was

<sup>1</sup><https://github.com/BCLab-UNM/Swarmathon-Robot>

<sup>2</sup><https://github.com/BCLab-UNM/CPFA-ROS>

<sup>3</sup><https://github.com/BCLab-UNM/DDSA-ROS>

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

Localization is a challenge in swarm robotics, particularly with low-cost robots using error-prone sensors and actuators. The robots in these experiments use an extended Kalman filter (EKF) [39] to fuse GPS information and odometry from an inertial measurement unit (IMU) and wheel encoders to determine position, orientation, and the locations of the collection zone and pheromone waypoints. We estimated the average accuracy of GPS localization to be 0.5 m, and we estimated how the IMU and encoders accumulate drift over time.

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

The architecture of the CPFA and DDSA implementation in ROS is shown in Fig. 2. Our *Rover GUI* either ran on a computer hosting Gazebo for a simulated swarm or connected to the physical robots in the swarm through a wireless network. The *GUI* acted as a communicator between users and robots. The results of the interaction between robots and objects in the simulation were sent to the ROS adapter.

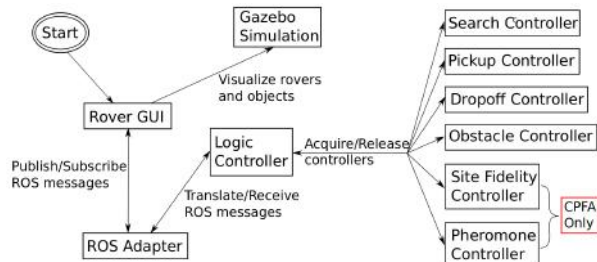


Fig. 2: The architecture of the CPFA and DDSA in ROS.

We deconstructed the robot control system into a series of functional controllers. Each controller was assigned a specific behavior. To fairly compare the CPFA and DDSA, we used the same implementation of the pickup, dropoff, and obstacle avoidance controllers. The deterministic spiral search and the ant-like stochastic search were implemented in the search controllers of the CPFA and DDSA, respectively. Site fidelity and pheromone controllers were only defined in the CPFA. Site fidelity in the DDSA was incorporated into the search controller as a feature. Different priorities were

assigned to controllers in different states. Higher priority controllers subsumed the roles of lower ones by suppressing their outputs. Robots switched controllers when they change their states. The logic controller handled transitions among all the controllers.

The ROS implementations of the CPFA and DDSA were directly loaded onto the Swarmie onboard Linux computer for physical robot experiments (see Fig. 3b). A demonstration video showing the CPFA and DDSA in the simulation and physical experiments are available on our YouTube playlist<sup>4</sup>.

## VI. EXPERIMENTAL SETUP

We evaluated four experimental configurations to measure the performance of the foraging algorithms. In the first two configurations, 128 targets were placed uniformly in a power-law distribution of cluster sizes which emulated the distribution of many resources in natural environments [40]. The targets were placed in 1 cluster of  $4 \times 8$  cubes, 2 clusters of  $4 \times 4$  cubes, 8 clusters of  $2 \times 2$  cubes, and 32 single cubes with each cluster location chosen at random. The first configuration had no obstacles, while the second configuration had the same distribution of targets and 4 obstacles ( $1 \text{ m} \times 0.5 \text{ m} \times 0.5 \text{ m}$  synthetic rocks). In the third configuration, 128 targets were placed in lines along the edges of the four arena walls (2 m). In the fourth configuration, four  $4 \times 8$  clusters of targets were placed in the four corners far (7.43 m) from the collection zone. The third and fourth configurations were designed to be more challenging with targets far from the center with no obstacles. In every experiment we placed 4 robots and a collection zone in the center of a  $14 \text{ m} \times 14 \text{ m}$  arena. Robots foraged for 20 min in each experiment.

Table. I summarizes the configurations and replicates of the experiments. Fig. 3a illustrates an example setup with obstacles in simulation while Fig. 3b shows the same setup in a physical experiment. We replicated all experiments 30 times in simulation. Physical experiments were repeated 15 times for the first two configurations (in which the locations of targets were chosen at random) and 5 times for the third and fourth configurations (in which targets were placed in fixed locations).

TABLE I: Experimental Setup and Replicates

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

## VII. RESULTS

We used interquartile-range notched box plots to visualize the statistical relationships between experiments [41]. Non-overlapping notches indicate the measurements were drawn from different distributions at the 95% confidence level.

<sup>4</sup><https://tinyurl.com/yceu6p9b>

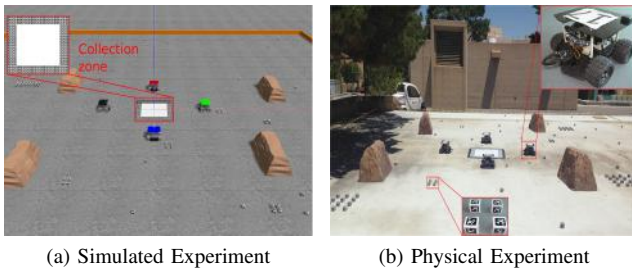


Fig. 3: Simulated and Physical experiments with 4 robots, 128 cubes, 4 obstacles and one central collection zone. Configuration 2 is shown, Target cluster sizes are described in Table 1, obstacles are placed 3 to 5 m from the center, and the exact location of each obstacle, target and target cluster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

Results were compared using the Mann-Whitney U test for simulated experiments and the student's t-test for physical experimental results (see [42], the t-test for small sample sizes). The statistical significance is indicated by asterisks in figures (\*\* indicates  $p < 0.01$ , \* indicates  $p < 0.05$ , and 'NS' indicates no statistical difference). The notches in the boxes indicate the 95% confidence interval of the medians.

The foraging performance of the DDSA and CPFA in simulated and physical experiments, with and without obstacles, is shown in Fig. 4. In simulations, the median number collected by the DDSA was 18% higher than the CPFA without obstacles (the U test two-tailed p-value was  $p = 0.0002$ ), and it was 26% higher than the CPFA with obstacles ( $p = 0.01$ ). In physical experiments, the CPFA was 25% higher than the DDSA ( $p = 0.04$ ) without obstacles and there was no significant difference with obstacles.

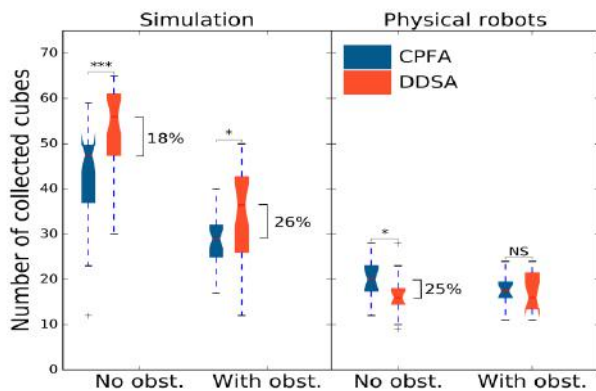


Fig. 4: Foraging performance of the DDSA and CPFA with and without obstacles, for 30 trials in simulation, and 15 trials in physical experiments using configurations 1 and 2 (shown in Fig. 3).

In the simulation, when resources were placed in the corners (configuration 4) the DDSA collected 26% more than the CPFA, with no difference between the two algorithms

with resources placed along the edges of the arena. There was no significant difference between algorithms in either configuration 3 or 4 in physical experiments (shown in Fig. 5).

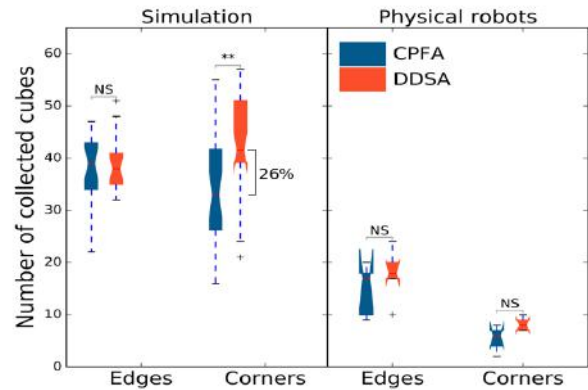


Fig. 5: Foraging performance with cubes lined to the edges and clustered in corners.

Fig. 6 summarizes the foraging performance across all experiments (240 simulations and 80 physical experiments). The DDSA collects 20% more targets than the CPFA in simulation, but there is no significant difference in physical environments. DDSA performance decreases more dramatically in physical experiments: the DDSA is 163% better in simulated vs. physical experiments, while the CPFA is only 95% better in simulated vs. physical experiments.

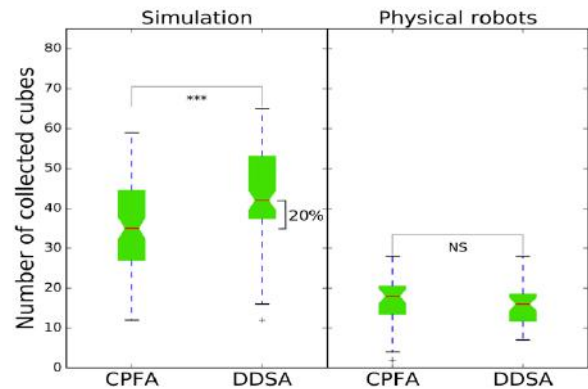
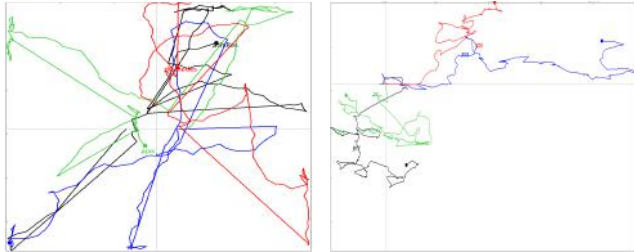


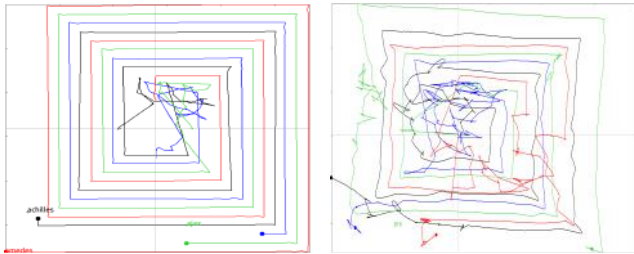
Fig. 6: Overall foraging performance with all experiments.

Fig. 7 shows traces of robots executing the CPFA and DDSA in different configurations. Panels (a) to (d) demonstrate baselines where each algorithm ran for 5 min without targets and obstacles. The traces in panels (a) and (c) illustrate the stochastic search pattern of the CPFA and the interlocking spiral pattern of the DDSA in simulation. The search patterns are still clear even with some drift in physical experiments without targets or obstacles (see (b) and (d)). Panels (e) to (h) demonstrate traces given targets and obstacles. In the simulations (see (e) and (g)), the characteristic search patterns of the CPFA and DDSA are

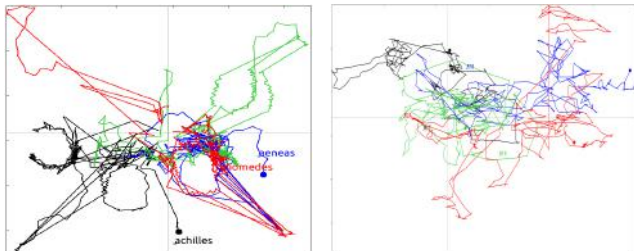
still evident, even though they are disrupted by direct paths to and from targets and empty regions outlining where targets are placed. However, in panel (h), the deterministic spiral is no longer visible because it is disrupted by robots interacting with each other, targets, and obstacles. In real environments with obstacles, the traces from the DDSA appear as random as those from the CPFA.



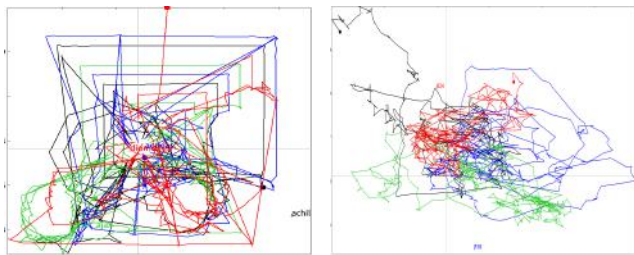
(a) No targets or obstacles, CPFA sim. (b) No targets or obstacles, CPFA phy.



(c) No targets or obstacles, DDSA sim. (d) Targets or obstacles, DDSA phy.



(e) Targets and obstacles, CPFA sim. (f) Targets and obstacles, CPFA phy.



(g) Targets and obstacles, DDSA sim. (h) Targets and obstacles, DDSA phy.

Fig. 7: Odometry traces of 4 robots in simulation (left column) and physical experiments (right column). Each robot path is a different colored line. Obstacles are not shown, but the empty areas in (e) and (g) imply their location.

## VIII. DISCUSSION

In a perfect environment without error or noise the DDSA outperforms the CPFA by collecting more targets in a fixed

time period. In the CPFA, the movement is random and some locations are visited multiple times while others are never visited at all. The DDSA guarantees complete coverage of the entire arena given sufficient time, and each location is visited only once. Our experiments in simulation confirm this expectation: the DDSA outperforms the CPFA (Fig. 4), even when the simulation includes physical interactions, collision avoidance, and some sensing and localization error. Even with resource placements far from the arena center, specifically designed to be difficult for the DDSA, it performs as well as or better than the CPFA in simulations (Fig. 5)

Not surprisingly, foraging performance was higher in both algorithms in simulations compared to physical experiments (Fig. 4 and Fig. 5). In prior ARGoS simulations [6], the DDSA collected targets 34% faster than the CPFA. Here our simulations include more realistic object pickup and dropoff and the complex physics of the Swarmie platform. In these simulations, the DDSA performed only 20% better than the CPFA (Fig. 6). The DDSA was no better than the CPFA in physical experiments, and in fact, the CPFA outperformed the DDSA in physical experiments without obstacles (Fig. 4). As more realism is included, the CPFA becomes as good as, or better than, the DDSA.

To understand why stochasticity affects performance, we recorded odometry traces of robots in simulation and physical robots. Without targets or obstacles in the arena, the essence of the stochastic CPFA and the DDSA spirals were evident in the odometry traces of physical robots (Fig. 7(b) and (d)). However, in contrast to the simulated traces, once targets and obstacles were placed in physical arenas, the DDSA spirals were disrupted so much that they were no longer distinguishable in robot paths (Fig. 7(f) and (h)).

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

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

## ACKNOWLEDGMENTS

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

## REFERENCES

- [1] Alan F. T. Winfield. Foraging Robots. Encyclopedia of Complexity and Systems Science, New York, NY, 3682-3700, 2009.
- [2] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1-41, 2013.
- [3] Erol Şahin. Swarm Robotics: From Sources of Inspiration to Domains of Application. *Swarm Robotics: SAB 2004 International Workshop*, Springer Berlin Heidelberg, 3342:10-20, 2005.
- [4] J. P. Hecker and M. E. Moses. Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, Springer US, 9(1):43-70, 2015.
- [5] Eliseo Ferrante, Ali E. Turgut, Edgar Duez-Guzmán, Marco Dorigo and Tom Wenseleers. Evolution of Self-Organized Task Specialization in Robot Swarms. *PLOS Computational Biology*, 11(8), 2015.
- [6] G. Matthew Fricke, Joshua P. Hecker, Antonio D. Griego, Linh T. Tran and Melanie E. Moses. A distributed deterministic spiral search algorithm for swarms. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, South Korea, 2016.
- [7] Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, and et al. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6:271-295, Springer. Berlin, Germany, 2012.
- [8] Roman Frigg and Stephan Hartmann. Models in science. The Stanford encyclopedia of philosophy. Stanford University. Spring 2012.
- [9] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. *European Conference on Artificial Life*. Berlin, Heidelberg, 704-720, 1995.
- [10] Jean-Baptiste Mouret, Sylvain Koos, and Stéphane Doncieux. Crossing the Reality Gap: a Short Introduction to the Transferability Approach, *CoRR*, 2013.
- [11] Morgan Quigley, Ken Conley, Brian Gerkey, and et al. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, 2009.
- [12] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *Intl. conf. IROS*, 3:2149-2154, 2004.
- [13] Brian P. Gerkey, Richard T. Vaughan and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, 317-323, 2003.
- [14] Wenguo Liu, Alan F. T. Winfield, Jin Sa, Jie Chen and Lihua Dou. Towards Energy Optimization: Emergent Task Allocation in a Swarm of Foraging Robots. *Adaptive Behavior*, 15(3):289-305, 2007.
- [15] Eduardo Castello, Tomoyuki Yamamoto, Fabio D. Libera, Wenguo Liu, Alan Winfield, Yutaka Nakamura, and Hiroshi Ishiguro. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence*, 10(1):1-31, 2016.
- [16] G. Pini, A. Brutschy, A. Scheidler, M. Dorigo and M. Birattari, Task Partitioning in a Robot Swarm: Object Retrieval as a Sequence of Subtasks with Direct Object Transfer, in *Artificial Life*, 20(3):291-317, July 2014.
- [17] Edgar Buchanan, Andrew Pomfret, and Jon Timmis. Dynamic Task Partitioning for Foraging Robot Swarms. *International Conference on Swarm Intelligence*. 9882. 113-124, 2016.
- [18] N. R. Hoff, A. Sagoff, R. J. Wood and R. Nagpal, Two foraging algorithms for robot swarms using only local communication, 2010 *IEEE International Conference on Robotics and Biomimetics*, Tianjin, 2010, pp. 123-130.
- [19] Paul E. Rybski, Amy Larson, Harini Veeraraghavan, Monica Anderson, and Maria Gini. Performance Evaluation of a Multi-Robot Search & Retrieval System: Experiences with MinDART. *Journal of Intelligent and Robotic Systems*, 52(3):363-387, 2008.
- [20] A. Rosenfeld, N. Agmon, O. Maksimov, and S. Kraus, Intelligent agent supporting human-multi-robot team collaboration, *Artificial Intelligence*, 2017.
- [21] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, The Robotarium: A remotely accessible swarm robotics research testbed, in *Robotics and Automation (ICRA)*, 2017 *IEEE International Conference on*. IEEE, 2017, pp. 16991706.
- [22] A. Brutschy, L. Garattoni, M. Brambilla, G. Francesca, G. Pini, M. Dorigo, and M. Birattari, The TAM: abstracting complex tasks in swarm robotics research, *Swarm Intelligence*, 9(1):1-22, 2015.
- [23] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal. Kilobot: A low cost robot with scalable operations designed for collective behaviors, *Robotics and Autonomous Systems*, 62(7):966975, 2014.
- [24] Simon Jones, Matthew Studley, Sabine Hauert, and Alan F. T. Winfield. *Evolving Behaviour Trees for Swarm Robotics*, 487-501, 2018.
- [25] J. Enright and P. R. Wurman, Optimization and Coordinated Autonomy in Mobile Fulfillment Systems. in *Automated action planning for autonomous mobile robots*, 2011, pp. 3338.
- [26] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, and A. Brutschy, Swarmanoid: a novel concept for the study of heterogeneous robotic swarms, *IEEE Robotics & Automation Magazine*, 20(4):6071, 2013.
- [27] Deborah M. Gordon and A. W. Kulig. Founding, Foraging, and Fighting: Colony Size and the Spatial Distribution of Harvester Ant Nests. *Ecological Society of America*, 77(8):2393-2409, 1996.
- [28] 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):1-9, July 2012.
- [29] Jennifer H. Fewell. Directional fidelity as a foraging constraint in the western harvester ant, *Pogonomyrmex occidentalis*. *Oecologia*, 82(1):45-51, 1990.
- [30] David J. T. Sumpter and Madeleine Beekman. From nonlinearity to optimality: pheromone trail foraging by ants. *Animal behaviour*, 66(2):273-280, 2003.
- [31] Duncan E. Jackson, Steven J. Martin, Francis L. W. Ratnieks and Mike Holcombe. Spatial and temporal variation in pheromone composition of ant foraging trails. *Behavioral Ecology*, 18(2):444-450, 2007.
- [32] Qi Lu, Joshua P. Hecker and Melanie E. Moses. The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, Daejeon, South Korea, 2016.
- [33] J. L. Bentley, B. W. Weide, and A. C. Yao, Optimal expected-time algorithms for closest point problems, *ACM Transactions on Mathematical Software (TOMS)*, 6(4):563-580, 1980.
- [34] R. A. Baeza-yates, J. C. Culberson, and G. J. E. Rawlins. Searching in the plane. *Information and computation*, 106(2):234-252, 1993.
- [35] G. Matthew Fricke, Diksha Gupta and Melanie E. Moses. Biologically-Inspired Distributed Spatial Search for Ground-Based Foraging Swarms, 5th Annual Biological Distributed Algorithms (BDA) Workshop, Washington, DC, 2017.
- [36] N. Koenig and A Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3:2149-2154, 2004.
- [37] Sarah M. Ackerman, G. Matthew Fricke, Joshua P. Hecker, Castro M. Hamed, Samantha R. Fowler, Antonio D. Griego, Jarett C. Jones J. Jake Nichol Kurt W. Leucht and Melanie E. Moses. The Swarmathon: An Autonomous Swarm Robotics Competition. *Swarms: From Biology to Robotics and Back at the IEEE International Conference on Robotics and Automation (ICRA) workshop*, 2018.
- [38] E. Olson. AprilTag: A robust and flexible visual fiducial system. 2011 *IEEE Intl. Conf. on Robotics and Automation*, 3400-3407, 2011.
- [39] Thomas Moore and Daniel Stouch. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. *Intelligent Autonomous Systems 13*, 335-348, Cham, 2016.
- [40] Mark E. Ritchie. *Scale, Heterogeneity, and the Structure and Diversity of Ecological Communities*. Berlin, Boston, 2009.
- [41] Robert McGill, John W. Tukey, and Wayne A. Larsen. Variations of Box Plots. *The American Statistician*, 32(1):12-16, 1978.
- [42] J. De Winter. Using the Students t-test with extremely small sample sizes. *Practical Assessment, Research and Evaluation*, 1-12, 2013.