



Multiple-place swarm foraging with dynamic depots

Qi Lu¹ · Joshua P. Hecker¹ · Melanie E. Moses^{1,2}

Received: 21 December 2016 / Accepted: 16 December 2017 / Published online: 9 January 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Teams of robots can be organized to collectively complete complex real-world tasks, for example collective foraging in which robots search for, pick up, and drop off targets in a collection zone. In the previously proposed central-place foraging algorithm (CPFA), foraging performance decreases as swarm size and search areas scale up: more robots produce more inter-robot collisions and larger search areas produce longer travel distances. We propose the multiple-place foraging algorithm with dynamic depots (MPFA_{dynamic}) to address these problems. Depots are special robots which are initially distributed in the search area and can carry multiple targets. Depots move to the centroids of the positions of local targets recently detected by robots. The spatially distributed design reduces robot transport time and reduces collisions among robots. We simulate robot swarms that mimic foraging ants using the MPFA_{dynamic} strategy, employing a genetic algorithm to optimize their behavior in the robot simulator ARGoS. Robots using the MPFA_{dynamic} find and collect targets faster than both the CPFA and the static MPFA. MPFA_{dynamic} outperforms the static MPFA even when the static depots are optimally placed using global information, and it outperforms the CPFA even when the dynamic depots deliver targets to a central location. Further, the MPFA_{dynamic} scales up more efficiently, so that the improvement over the CPFA and the static MPFA is even greater in large (50 × 50 m) areas. Including simulated error reduces foraging performance across all algorithms, but the MPFA still outperforms the other approaches. Our work demonstrates that dispersed agents that dynamically adapt to local information in their environment provide more flexible and scalable swarms. In addition, we illustrate a path to implement the MPFA_{dynamic} in the physical robot swarm of the NASA Swarmathon competition.

Keywords Swarm robotics · Foraging · Scalable system

1 Introduction

One major goal of swarm robotics research is to design robust, scalable, and flexible collective behaviors for multiple autonomous robots (Şahin 2005; Moses and Banerjee 2011; Brambilla et al. 2013). Simple rules and local interactions among individual robots result in desired collective swarm behavior by self-organized coordination mechanisms. Biological studies have revealed self-organized coordination

mechanisms in social insects which can be effectively implemented in swarm robotics systems (Camazine et al. 2001; Şahin 2005).

In this research, we focus on the foraging behavior of robot swarms. The challenge is to develop an effective, decentralized search-and-collection foraging algorithm for ant-like robot swarms (Gordon and Kulig 1996; Winfield 2009; Liu and Winfield 2010). Robots must retrieve objects from an environment and bring them back to a depot (or nest). Effective collective foraging requires coordination, navigation, and communication and is therefore a useful abstraction of many complex, real-world applications such as humanitarian de-mining, search and rescue, intrusion tracking, collection of hazardous materials, and space exploration (Winfield 2009; Brambilla et al. 2013). In particular, foraging is commonly used as a testbed for collective exploration, collective transport, and collective decision-making (Gazi and Passino 2004; Brambilla et al. 2013).

✉ Qi Lu
lukey11@unm.edu

Joshua P. Hecker
jhecker@cs.unm.edu

Melanie E. Moses
melaniem@cs.unm.edu

¹ Moses Biological Computation Laboratory, Department of Computer Science, University of New Mexico, Albuquerque, NM 87106, USA

² Santa Fe Institute, Santa Fe, NM 87501, USA

We propose the multiple-place foraging algorithm with dynamic depots (MPFA_{dynamic}). Depots are special robots which are able to carry multiple targets. Targets are objects such as mineral resources, hazardous waste, or any item that needs to be retrieved from the environment and gathered at a location. Foraging robots depart from a depot to forage for targets and then return to the closest depot to deliver these targets (the closest depot may be different from the one the robot departed from). Depots move to new locations based on the mean positions of the remaining targets sensed by the robots. The positions of the sensed targets are stored at each depot when each foraging robot returns to that depot. The stored positions are relative to the depot's current location so that no central controller is needed to facilitate information sharing across the swarm.

The final delivery of targets that are collected by the depots depends on the application. Targets may be processed at the dispersed locations where they are collected; they may be collected by another larger robotic agent that empties depots and delivers their contents to a central location; or, as the depots become full, they may drive the targets to the desired location. We explore the latter scenario in a subset of our experiments.

We compare the performance of the MPFA_{dynamic} with our previous MPFA_{static} proposed by Lu et al. (2016a) with uniformly-distributed static depots, and to the central-place foraging algorithm developed by Hecker and Moses (2015).

In order to assess the effectiveness of our approach, we also compare our results to algorithms with access to global information. We compare the MPFA_{dynamic} which uses only local information, to versions of the MPFA with global information describing the initial locations of all targets. These algorithms use the *k*-means++ clustering algorithm to determine the initial positions of the depots to minimize transport distance. We evaluate the MPFA with depots that have global information about target locations using both static depots (MPFA_{global_static}) and dynamic depots (MPFA_{global_dynamic}).

We test how quickly targets are collected using the five algorithms (CPFA, MPFA_{static}, MPFA_{global_static}, MPFA_{dynamic}, MPFA_{global_dynamic}) across different distributions of targets. We observe how much the mobile depots improve swarm foraging performance, specifically: (i) the time required to collect a fixed fraction of the targets (*foraging time*), (ii) the time required to detect and avoid collisions with other robots (*collision time*), (iii) the time that a robot spends searching for targets (*search time*), and (iv) the time that a robot spends traveling to and from a depot when collecting targets (*travel time*). We show that our proposed algorithm, MPFA_{dynamic}, outperforms both the CPFA and the MPFA_{static} on all performance criteria. We also show that MPFA_{dynamic} performs approximately as well as MPFA_{global_static} and MPFA_{global_dynamic} without depend-

ing on global communication. This is a significant advantage of MPFA_{dynamic} because global information is costly to obtain, and reliance on centralized communication is a single point of failure and efficiency bottleneck.

We also compare the scalability of the five algorithms by increasing the number of robots in the swarm and the size of the experimental arena. Our results show that MPFA_{dynamic} has better scalability than the other four algorithms: increasing the arena size has a smaller negative effect on the foraging time of swarms using MPFA_{dynamic}, and increasing swarm size in a large arena has a larger positive effect on the foraging time of those swarms. In addition, we implement the MPFA_{dynamic} with depots that transport their contents to a central depot, thus completing the central-place foraging task. We compare this implementation to the CPFA.

Finally, we demonstrate how we can use our existing ROS/Gazebo simulation and Swarmie hardware for the NASA Swarmathon competition (Secor 2016; Ackerman et al. 2018) to implement the dynamic MPFA in a physical robot swarm.

2 Related work

2.1 Central-place foraging

Central-place foraging is a canonical collective task commonly studied in swarm robotics (Şahin 2005; Brambilla et al. 2013). Robots depart from a centrally-placed depot to search for targets and return to this central place to deliver targets. The central-place foraging task can be instantiated into a number of real-world target collection applications, including crop harvesting (Sebbane 2012; Bac et al. 2014) and extra-planetary resource collection (Brooks and Flynn 1989; Landis 2004; Fink et al. 2005).

In prior work, Hecker and Moses (2015) introduced the central-place foraging algorithm (CPFA), which was designed to emulate seed-harvester ant behaviors governing memory, communication, and movement. The error-tolerance, flexibility, and scalability of the CPFA were evaluated on both simulated and real robot swarms. Hecker and Moses use a genetic algorithm (GA) to evolve foraging strategies that were tolerant of real-world sensing and navigation error, flexible for a variety of target distributions, and scalable to large swarm sizes.

The behaviors of the CPFA emulate harvester ant foraging that maximizes the number of targets collected in short foraging time periods (Gordon and Kulig 1996; Flanagan et al. 2012), but is not designed for complete target collection. The foraging efficiency of the CPFA was recently compared to the distributed deterministic spiral algorithm (DDSA), a deterministic benchmark for central-place foraging (Fricke et al. 2016) that is designed to collect the nearest targets first.

Results showed that robot swarms using the DDSA were faster at complete collection tasks than swarms using the CPFA.

However, the CPFA outperformed the DDSA by collecting more targets in fixed time windows for large swarms with more than 20 robots. The DDSA suffered from more robot collisions in more crowded environments. Since our goal for the MPFA is to increase foraging rates in large swarms, we build upon and compare to the CPFA in this work. We also focus on collecting targets quickly rather than complete target collection.

Although the CPFA is more scalable than the DDSA, CPFA swarms also exhibited diminishing returns as swarm size increased (i.e. sublinear scaling of foraging rate per robot given larger numbers of robots in the swarm). Diminishing returns are expected for central-place foraging because robots in larger swarms on average travel farther to collect more targets, and there are more collisions given more robots. As shown in Lu et al. (2016a), the MPFA mitigates those effects. We show in this work that adding dynamic depots to the MPFA further mitigates scaling limitations.

2.2 Multiple-place foraging

Previous work has demonstrated that a single, central depot cannot serve a large number of robots efficiently due to long travel times (Hecker and Moses 2015) and heavy crowding (Fricke et al. 2016). To mitigate this issue, we proposed the multiple-place foraging algorithm (MPFA) with multiple static depots, where robots are programmed to always return to the depot closest to the position of the target that the robot has found (Lu et al. 2016a, b).

The MPFA was primarily inspired by behaviors observed in groups of insects and primates, as well as the immune system. For example, polydomous colonies of Argentine ants are comprised of multiple nests spanning hundreds of square meters (Schmolke 2009; Flanagan et al. 2013); additionally, a study (Tindo et al. 2008) showed that wasps living in multiple nests have greater survival rates and increased productivity. Chapman et al. (1989) showed that communities of spider monkeys can be also considered as multiple central-place foragers (MCPF), where monkeys select a sleeping site close to current feeding areas, and the MCPF strategy entails the lowest travel costs. In another biological system, Banerjee and Moses (2010b) showed that the decentralized, sub-modular nature of the immune system increases the foraging efficiency of immune cells that aggregate in lymph nodes distributed throughout the body. These dispersed aggregation points (analogous to multiple nests) speed up immune response rates, particularly in large animals that may have trillions of immune cells. Recently dynamic lymph nodes that appear near sites of infection have been discovered (Moyron-

Quiroz et al. 2004), motivating the use of depots as dynamic aggregation points for robotic foraging.

The use of dynamic docks is introduced in the related work (Couture-Beil and Vaughan 2009). That work demonstrates that mobile docks mitigate the spatial interference and improve overall task performance when mobile robots execute a transportation task and periodically recharge from a docking station.

Multiple-place foraging also resembles the task allocation of global courier and delivery services, which use many distributed stores to collect and deliver packages efficiently. Several studies on task allocation in robot swarms have used biologically-inspired approaches in the deployment of homogeneous swarms of robots to multiple sites (Halász et al. 2007; Berman et al. 2008; Hsieh et al. 2008). These robots autonomously redistribute themselves among the candidate sites to ensure task completion by optimized stochastic control policies. In general, each swarm is modeled as a hybrid system where agents switch between maximum transfer rates and constant transition rates.

2.3 Foundations of the MPFA

In our original implementation of the MPFA (Lu et al. 2016a, b), robots were initially assigned in equal numbers to static collection points called nests. Nests were evenly placed in the environment, i.e. given 4 nests, each was placed at the center of one quadrant of a foraging arena with 1/4 of the robots assigned to each nest. The robots could autonomously switch to other nests as they foraged. If the location of a found target was closer to another nest, the robot (which had traveled a long distance from its initial depot and discovered this target) delivered this target to the closer depot. The transition from one depot to another one is shown in Fig. 3.

The use of multiple collection depots is the fundamental difference between the CPFA and the MPFA; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of multiple depots on swarm foraging efficiency.

2.3.1 The CPFA

There are several essential features of the CPFA that make it possible to implement the MPFA_{dynamic}. The CPFA implements site fidelity in which a robot remembers and returns to the location where it last found targets. The CPFA implements pheromone waypoints as a list of target-rich locations that have been found by robots. Depots report the list of waypoints to robots when they drop off targets.

When a robot finds a target, it senses the local density of targets and then uses that information to determine whether to use site fidelity to return to the location and whether to

communicate that information to other robots by reporting a pheromone waypoint to its depot.

How site fidelity, pheromone waypoints, and other details of the CPFA are implemented is described below in Sect. 3 and Algorithm 1 (where in line 7, the closest depot is always the single central depot in the case of the CPFA).

A final important feature of the CPFA in its implementation in real robots is the ability to reliably return to a depot. The CPFA and MPFA rely on the use of beacons that are detectable by any nearby robots. Our experiments with physical iAnt robots running the CPFA demonstrate that a light is an effective beacon that allows robots to reliably return to their nest (Hecker and Moses 2015). There are alternative beacons that can ensure that robots can reliably locate depots and other important locations. For example, colored LEDs on robots (Nouyan et al. 2009), speaker-induced sound gradients (Nurzaman et al. 2009), and images such as fiducials or roundels (Bezzo et al. 2015) can be used to mark important locations in space to which physical robots can reliably return.

2.3.2 The MPFA_{static}

The behavior of an individual robot in an MPFA foraging round is shown in Fig. 1. Each robot transitions through a series of states as it forages for targets. The states and transitions emulate foraging behaviors of ants. The MPFA differs from the CPFA in that the robots return to the closest depot in steps 4 and 5.

Robots initially disperse from depots and follow randomly selected travel paths (step 1). Upon reaching the end of the travel path, robots switch to searching for targets using an uninformed correlated random walk (in which the robot has no knowledge of target locations) observed in ants (step 2) (Fewell 1990). Robots navigate home to the depot closest to them after they collect a target (step 4) or give up searching (step 5) (as described in ants in Crist and MacMahon 1991). The search cycle for an individual robot foraging using uninformed search is shown in Fig. 2.

Robots that discover a target will sense the local target density before returning to their local depot (step 3 and step

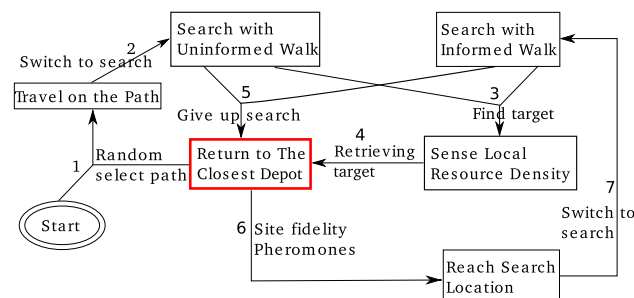


Fig. 1 The behavior of an individual robot implementing the MPFA

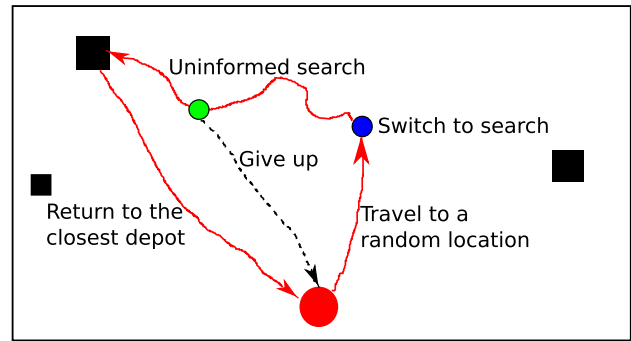


Fig. 2 A single cycle of uninformed search. Four states of a robot in the cycle are shown. A robot departs from a depot (large red circle), travels to a random location, and switches to searching using an uninformed random walk (dark blue circle). If the robot finds a target pile (largest black square), then it collects one target and delivers it to the closest depot. The robot also has a probability of giving up searching (bright green circle) and returning to the closest depot without finding a target (Color figure online)

4) (Hölldobler 1976). The density is the number of targets sensed in the local region by robots. The size of the region a robot can detect is described in Sect. 4.1. An individual robot may remember the location of a previously found target and repeatedly return to the same location, a process called *site fidelity* in ants (Beverly et al. 2009). Robots can also communicate using pheromones (Sumpter and Beekman 2003; Jackson et al. 2007) which are simulated as artificial waypoints (Campo et al. 2010) to recruit robots to known clusters of targets. This is also discussed in Sect. 3.1. Robots that return to a previously found target site using site fidelity or pheromone recruitment (step 6) will search the target site thoroughly using an informed correlated random walk (step 7). The search behaviors for an individual robot foraging using informed search is shown in Fig. 3.

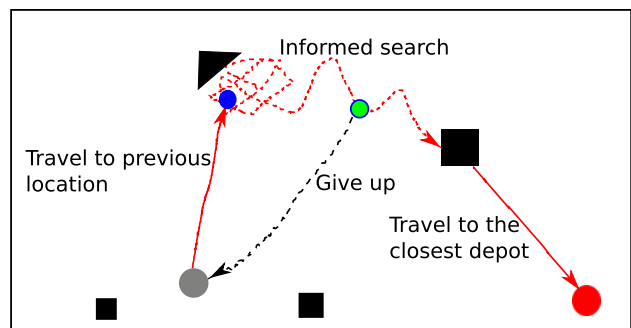


Fig. 3 A single cycle of informed search. Five states of a robot are shown. A robot departs from a depot (large gray circle) and travels to the previous location (dark blue circle), and switches to searching using an informed correlated walk. If it finds a target pile (largest black square), then it collects one target and delivers it to the closest depot (red circle in the lower right). The robot also has a probability of giving up searching (light green circle) and returning to the closest depot without finding a target (red circle) (Color figure online)

The search strategy is evolved by a genetic algorithm (GA); all robots use the same strategy, but make decisions probabilistically based on the interaction with the environment. Although robots are able to depart from and return to the nearest depot, robots still search globally, meaning that they are able to travel across the entire arena.

As in the CPFA, pheromone trails are simulated using pheromone waypoints. Different from the CPFA, pheromone waypoints are only reported to the closest depot to the robot when it arrives at the depot. Robots can only send and receive pheromone waypoints when they are returning to a depot.

We use an exponential decay function with a decay rate selected by the GA to simulate the pheromone decay process. After a certain amount of time, the pheromone waypoint will have decayed below a threshold and will be removed from the depot's list. When a robot arrives at the depot, it will probabilistically select a waypoint from that depot's list and travel to the location of the waypoint. The robot may also probabilistically choose to locally share information by sending pheromone waypoints to its current depot. Unlike the CPFA, the pheromone waypoints associated with a given depot are only locally available to robots returning to that depot.

Since robots always return to the closest depot with a found target, the sensed information relevant to a given target neighborhood is always associated with the depot closest to the position of the identified neighborhood. Thus, the robots only travel from the closest depot to any given pheromone waypoint.

In our recent work (Lu et al. 2016a), we conducted simulated experiments with the MPFA using multiple static depots (2, 4, 8, and 16). We ran the experiments with 256 targets and 24 robots in a 10×10 m (i.e. 10 m wide by 10 m long) arena. The results showed that the MPFA produces higher foraging rates and lower average travel time compared to the CPFA. Increasing the number of depots increases the foraging rate of the swarm and decreases the required travel time per target collected, while the search time per target collected is independent of the number of depots. In most experiments, 4 depots led to significantly faster foraging than the CPFA or 2 depots, but they were indistinguishable from 8 depots, and so we focus on experiments with 4 depots in this paper. We note that determining the optimal number of depots for a given number of robots, targets, and arena sizes is itself an interesting question that we leave to future work.

Because pheromone waypoints are distributed across multiple depots, MPFA swarms require less communication among robots, and individual robots spend less time traveling back to the closest depot to make use of the information. In contrast, CPFA swarms use pheromone waypoints that are globally available to the entire swarm; these robots, therefore, have access to more information, but individual robots take longer to travel back to the central depot and use the information. The GA balances these trade-offs automatically by

tuning the search strategies and optimizing the performance of each swarm, resulting in systematic changes in parameters governing pheromone laying and distance traveled from the depot as more depots are added.

In other recent work (Lu et al. 2016b), we compared the ability of the MPFA and the CPFA to maintain foraging efficiency as swarm size and target number increase. We increased the size of the swarm (4, 8, 16, 32, and 64 robots given 1024 targets) to test scalability and the number of targets (128, 256, 512, 1024, and 2048 targets given 32 robots) to test adaptability to different target densities.

The MPFA had higher foraging efficiency than the CPFA under increased swarm size and target number. Furthermore, robots using the MPFA spent less time avoiding collisions and required less travel time to collect each target.

3 Methods

Previous MPFA experiments (Lu et al. 2016a, b) were conducted using uniformly-spaced static depots, which outperformed central-place foraging swarms, but were not capable of dynamically adapting to different target distributions. In this work, we aim to further improve swarm foraging performance with depots that move to the centroid of known nearby targets in order to minimize the time and distance for foraging robots to transport those targets.

If all of the positions of the targets are known, then we can use this positional information to calculate the optimal location of depots to minimize travel distance to all targets. This problem is analogous to clustering targets based on their distances to the closest depot, where the sum of distances between targets to the center of the cluster is minimum.

Given the locations of all targets in the arena, the k -means++ clustering algorithm (Arthur and Vassilvitskii 2007) will calculate the locations of depots to minimize the travel distance required to collect all targets. Figure 4 shows an example of a dynamically allocated depot, in which six piles of targets are classified into four clusters and four depots are placed at the centroids of these clusters. This implementation would require global knowledge of all target locations, which violates one of the key features of swarm robotics systems: all sensing and communication must be local (Brambilla et al. 2013).

We use globally informed MPFA algorithms to provide points of comparison for our proposed multiple-place foraging algorithm with dynamic depots (MPFA_{dynamic}), an extension to our recent work in which depots move to new locations based on the locations of the targets sensed by robots. Depots always move to the centroid of recently sensed targets, which are maintained in a list and updated whenever site fidelity or pheromone waypoints are used. If site fidelity is not used, or if pheromone waypoints decay, then those

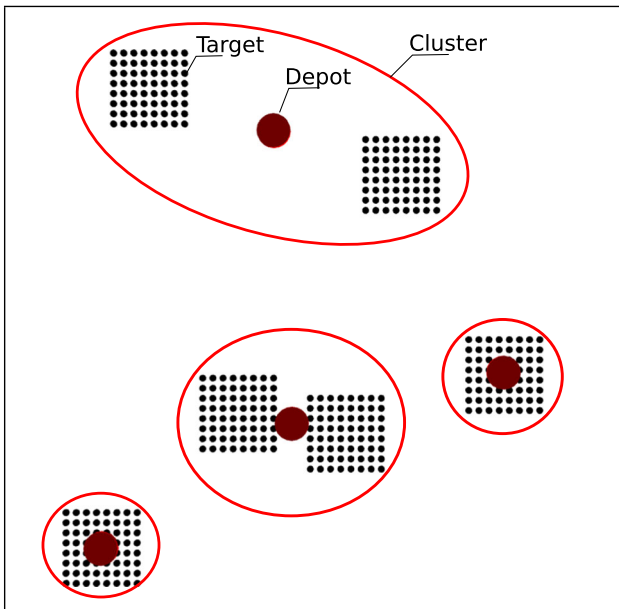


Fig. 4 An example of a dynamically allocated depot using the k -means++ clustering algorithm. The targets (black squares) are classified into four clusters (red ellipses). Depots (dark red solid circles) are placed at the centroids of these clusters (Color figure online)

sensed targets are removed from the list and no longer contribute to the dynamic calculation of the depot’s centroid.

The use of mobile depots is the fundamental difference between $MPFA_{static}$ and $MPFA_{dynamic}$; all other components of the two foraging algorithms are kept deliberately identical in order to test for the effect of mobile depots on foraging efficiency.

As in $MPFA_{static}$, depots are initially distributed uniformly in $MPFA_{dynamic}$, and robots are evenly distributed to each depot. Depots move to new locations based on the positional information of observed targets sensed by foraging robots. Figure 5 shows how a depot moves based on the sensed positional information of targets reported by foraging robots.

We assume robots can sense targets within camera range, but cannot precisely measure the positions of these targets. Therefore, a robot only reports its current position and the number of targets detected; the robot’s current position approximates the centroid of the targets that it has detected. Each depot is allocated to the centroid c_t of the sensed targets at time t , where c_t is defined by Eq. 1:

$$c_t = \frac{1}{N} \sum_{i=1}^N w_i p_i \tag{1}$$

where w_i is the number of sensed targets at location p_i , and N is the total number of different locations where robots have sensed targets.

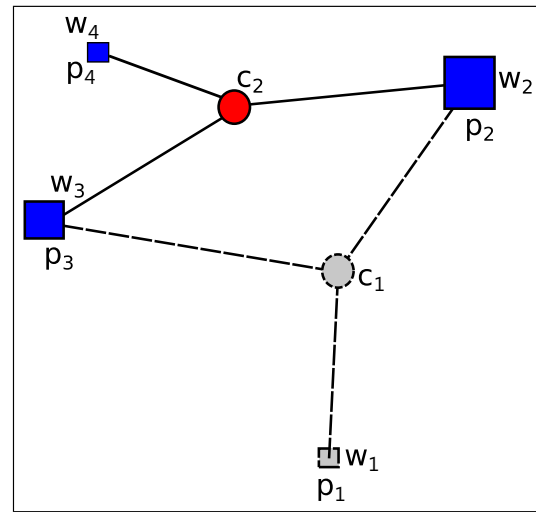


Fig. 5 Depot movement in $MPFA_{dynamic}$. A depot (gray circle) is at the centroid c_1 of the sensed targets (dark blue squares) at positions p_1 , p_2 , and p_3 , where w_1 , w_2 , and w_3 are the number of targets sensed by robots at each position, respectively. After some time, if targets at position p_1 are completely collected by robots, then the pheromone waypoints at p_1 will decay. If, at the same time, w_4 targets are sensed at a new location p_4 , then the depot will move to the centroid c_2 of the sensed targets (red circle) at positions p_2 , p_3 , and p_4 (Color figure online)

Table 1 Parameters for robot controllers

| Parameter | Description | Initialization |
|----------------|---------------------------------|----------------|
| σ | Uniformed search variation | $U(0, \pi)$ |
| p_s | Prob. of switching to search | $U(0, 1)$ |
| p_r | Prob. of giving up search | $U(0, 1)$ |
| λ_{id} | Rate of informed search decay | $Exp(0.2)$ |
| λ_{sf} | Rate of following site fidelity | $U(0, 20)$ |
| λ_{lp} | Rate of laying pheromone | $U(0, 20)$ |
| λ_{pd} | Rate of pheromone decay | $Exp(0.1)$ |

3.1 Implementation of robot controllers

Our robots mimic seed-harvester ant behaviors that have evolved over millions of years. We encode these behaviors into a robot controller (see Algorithm 1) using the same set of seven real-valued parameters that define the CPFA (see Table 1) specifying movement, sensing, and communication:

Uniformed search variation Uniformed robots forage using a correlated random walk with fixed step length and direction $\theta_t = \mathcal{N}(\theta_{t-1}, \sigma)$, where θ_{t-1} is the turning angle from the previous step, and σ is the uniformed search variation (or standard deviation), which determines the turning angle of the next step.

Algorithm 1 Multiple-Place Foraging Algorithm

```

1: Disperse from depot to random location
2: while experiment running do
3:   Conduct uninformed correlated random walk
4:   if target found then
5:     Collect it
6:     Sense targets  $c$  near current location  $l_f$ 
7:     Return to the closest depot to deliver target
8:     if  $\text{POIS}(c, \lambda_{lp}) > U(0, 1)$  then
9:       Lay pheromone to  $l_f$ 
10:    end if
11:    if  $\text{POIS}(c, \lambda_{sf}) > U(0, 1)$  then
12:      Return to  $l_f$ 
13:      Conduct informed correlated random walk
14:    else if pheromone found then
15:      Travel to pheromone location  $l_p$ 
16:      Conduct informed correlated random walk
17:    else
18:      Choose new random location
19:    end if
20:  end if
21: end while
    
```

Probability of switching to search Robots start at a depot and select a direction θ from a uniform random distribution $\mathcal{U}(0, 1)$, then travel in this direction away from the depot (see Fig. 2). Robots have a probability p_s of switching to an uninformed correlated random walk, where higher values of p_s indicate shorter travel distances from the depot.

Probability of giving up search At each step of the correlated random walk, robots that have not discovered a target may give up searching and return to the closest depot with probability p_r .

Rate of informed search decay If robots return to a previous location via site fidelity or pheromone waypoint, they search using an informed correlated random walk (see Fig. 3), with standard deviation $\hat{\sigma}$ defined by Eq. 2:

$$\hat{\sigma} = \sigma + (2\pi - \sigma)e^{-\lambda_{id}t} \tag{2}$$

As time t increases, $\hat{\sigma}$ decays to σ , producing an initially undirected and localized search that becomes more correlated over time. This time decay allows robots to search locally where they expect to find a target, but to straighten their path and move to another location if no target is found.

Rate of following site fidelity The probability of a robot returning to a previous target location via site fidelity is governed by the Poisson cumulative distribution function (CDF) defined by Eq. 3:

$$\text{POIS}(k, \lambda_{sf}) = e^{-\lambda_{sf}} \sum_{i=0}^{\lfloor k \rfloor} \frac{\lambda_{sf}^i}{i!} \tag{3}$$

where k is the number of additional targets detected in a previous location and the parameter λ_{sf} is the average number of detected targets. The Poisson CDF models the probability of following site fidelity given the number of detected targets k appropriately. The probability is highest when $k = \lambda_{sf}$. Robots return to previous locations via site fidelity if the parameterized Poisson CDF exceeds a uniform random value, $\text{POIS}(k, \lambda_{sf}) > \mathcal{U}(0, 1)$, simulating a random sampling process that is weighted by the probability of following site fidelity for a given k . Otherwise, robots follow pheromone waypoints to previous target locations if pheromones are available. If no pheromone exists, robots return to traveling and searching using the uninformed correlated random walk.

Rate of laying pheromone The probability of creating a pheromone waypoint is also governed by the Poisson CDF (Eq. 3). Robots create waypoints for previous target locations if $\text{POIS}(k, \lambda_{lp}) > \mathcal{U}(0, 1)$, where k is also the number of targets detected in a previous location.

Rate of pheromone decay Pheromone waypoint strength γ decays exponentially over time t as defined by Eq. 4:

$$\gamma = e^{-\lambda_{pd}t} \tag{4}$$

3.2 Evolving swarm behavior

Robot controllers are evolved using a genetic algorithm (GA) to optimize the collective behavior of the entire robot swarm, where every robot in the swarm uses the same controller. The controller is evolved in one set of simulations and evaluated in another set of simulations which are replicated 100 times. We run each foraging algorithm until the robot swarm collects the expected percentage of targets. Fitness is simply defined as the number of targets collected in a specified foraging time. In Hecker and Moses (2015) the foraging time was set to 1 h.

There are an uncountable number of foraging strategies that can be defined by the real-valued parameters of the CPFA and MPFA. Given 100 real values of each parameter, there would be 100^7 possible strategies. Additionally, the online decision making of each robot depends on interactions with environmental conditions. For example, following site fidelity is determined by the condition of $\text{POIS}(k, \lambda_{lp}) > \mathcal{U}(0, 1)$ as described in Sect. 3.1. The sampled value from $\mathcal{U}(0, 1)$ is random at each time, and the decision to use site fidelity depends on the value of k and the sampled random value. The GA provides a way to sample both parameter space and the effectiveness of the foraging algorithm evaluated in different environmental conditions.

The parameters in Table 1 are independently evolved 16 times in order to generate 16 independent foraging strate-

gies for each of the five foraging algorithms in each target distribution. Thus we have a total of 240 separate evolutionary runs (3 distributions \times 5 algorithms \times 16 replicates). Each of these evolutionary experiments follows the process described in Experiment 1 in Sect. 4.

In Hecker and Moses (2015) we demonstrated that the evolved CPFA controllers could be effectively transferred into physical robots, a process also described in related work (Nelson et al. 2004; Singh and Parhi 2011). Such controllers could be effectively tuned by the GA to mitigate the real-world error inherent in physical robots (Hecker et al. 2013). We describe steps toward similarly implementing MPFA_{dynamic} in real robots in Sect. 6.3.

We implement our GA using GAlib (Wall 1996). For each generation of the GA, we evaluate each candidate set of 7 parameters on 10 different random placements of targets (see Fig. 6) to determine fitness. We use a 50% uniform crossover rate and a 5% Gaussian mutation rate with a standard deviation of 0.02, and elitism to keep the fittest parameter set.

We set termination criteria of the GA in order to hasten parameter convergence, running for a maximum of 100 generations. The GA terminates based on three criteria: the convergence of fitness values, the diversity of parameter sets, and the number of generations. The GA will stop if the fitness has converged and the diversity is low; otherwise, it will terminate after 100 generations.

In our GA, 89% of the evolutionary runs terminate based on the convergence of fitness and low diversity. Across 16 independent evolutionary runs, all evolved parameter sets were nearly equally fit: the standard deviation in fitness was at most 5% of the mean fitness value of these 16 independently evolved parameter sets. We chose the fittest parameter set to evaluate foraging performance.

4 Experimental configuration

We conducted four sets of experiments using the swarm robot simulator Autonomous Robots Go Swarming (ARGoS) (Pinciroli et al. 2012) to evolve parameters and then test foraging performance. In the first set of experiments, we compared the foraging times of MPFA_{dynamic} to the CPFA and MPFA_{static}, as well as to the two idealized versions of the MPFA that rely upon global knowledge of target locations to determine depot locations, MPFA_{global_static} and MPFA_{global_dynamic}. These experiments were conducted with 24 robots in a 10 \times 10 m arena.

In the second set of experiments, we tested the scalability of these algorithms to larger arena sizes. We examined the rate of increase in foraging times with increasing arena size (24 robots in arenas that increase from 10 \times 10 m to 16 \times 16 m). In the third set of experiments, we tested the performance

Table 2 Experimental configuration

| Experiments | Arena width (m) | Number of robots |
|-------------|-----------------|------------------|
| 1 | 10 | 24 |
| 2 | 10, 12, 14, 16 | 24 |
| 3 | 50 | 96 |
| 4 | 10 | 28 |

of each algorithm in a very large arena (50 \times 50 m) with 96 robots.

In the fourth set of experiments, we account for transportation by the mobile depots to a single central collection point. In these experiments, each of the four mobile depots is a modified robot that carries targets to a central collection point; thus, we also add 4 robots to the CPFA experiments, so foraging performance is evaluated with each having 28 robots that ultimately deliver targets to a central place.

For the first set of experiments, the parameters for the CPFA and MPFAs were each evolved separately as described in Sect. 3. We select the set of evolved parameters which has the shortest foraging time from the 16 sets of evolved parameters for the experiment. These sets of evolved parameters are subsequently used for the corresponding CPFA and MPFAs in the second, third and fourth experiments.

The configuration of the four sets of experiments is summarized in Table 2. Each experiment has one central depot in the CPFA, and four depots for each of the four MPFAs. In the fourth experiment, we include a central depot and four dynamic depots in the MPFA_{dynamic} simulations.

Foraging time is measured as the time for the entire swarm to collect 88% of the 384 placed targets. This percentage was chosen because it is the inflection point in CPFA foraging performance (Hecker et al. 2015) after which there is an exponential increase in collection time and very high variance in performance due to the sparsity of remaining targets.

In the first set of experiments, we additionally measure the times for different components of the foraging time: travel time, search time, and collision time, each of which is described in Sect. 5.

Each of the five algorithms is tested on three different classes of target distribution: targets placed in a uniform random distribution, targets placed in a partially clustered distribution, and targets placed in a highly clustered distribution. Examples of targets placed in each distribution are shown in Fig. 6.

The partially clustered distribution uses a power law distribution of cluster sizes: 128 clusters that contain a single target, 32 clusters with 4 targets each, and 8 clusters with 16 targets each, for a total of 384 targets. This power law distribution of cluster sizes emulates that of many natural resource

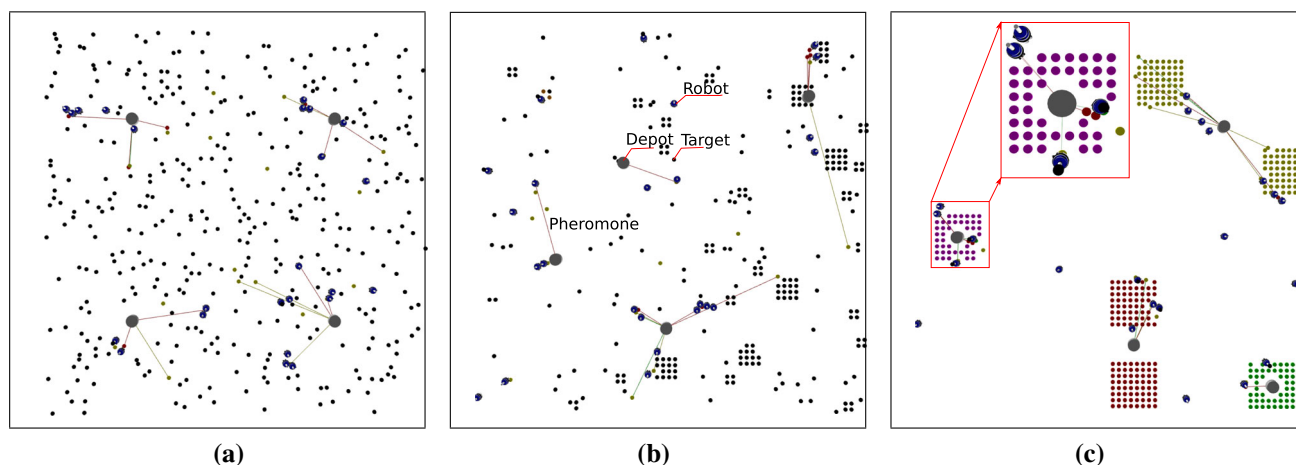


Fig. 6 The placement of depots and targets in ARGoS. 384 targets (small points) and 24 robots (middle-sized points) are placed in a 10×10 m arena, and 4 depots (large points) are distributed. The targets are unclustered and spread in a uniform random distribution in **a**, partially

clustered in **b**, and clustered into 6 equally-sized piles in **c**. The colored rays indicate pheromone waypoints with different strength that eventually evaporate and disappear. A small area is magnified in **c** to show targets, robots, and a depot in the center

distributions in real-world environments (Ritchie 2009). The fully clustered distribution has 6 clusters of 64 targets each.

Each experiment is replicated 100 times. For each replicate, the individual targets, or centers of target clusters, are chosen at random so that each replicate has a different target placement consistent with the distribution for that experiment. Thus, there are 1500 experimental runs (3 distributions \times 5 algorithms \times 100 replicates) for the first set of experiments, 6000 experimental runs (one for each of 4 arena sizes) for the second set of experiments, 1500 runs for the third set of experiments, and 600 runs for the fourth set of experiments, for a total of 9600 separate experimental runs.

4.1 ARGoS implementation

Our implementation includes a C++-based robot controller library, and an XML configuration file. The C++ controller specifies the robot's functionality and interaction with the ARGoS environment, while the XML file contains all of the information to set up the size of arena, the type of robots, the physics engines, the parameters of robot controllers, the simulation accuracy, and the distributions of targets, depots, and robots. Source code is available on GitHub,¹ and demonstration videos are available on our YouTube playlist.²

We use the ARGoS 8.5 cm radius foot-bots to represent our robots with a movement speed of 16 cm/s, while the movement speed of a depot is set to be the same. The step size of the simulation is 32 ticks per second, which was chosen to balance simulation accuracy and speed. Depots have a

15 cm radius and targets are cylinders with a 5 cm radius. The distance robots can sense targets is $2\sqrt{2}$ times the target radius.

Each pheromone trail is represented by a starting waypoint and an ending waypoint at a depot. Waypoints provide positional information maintained in lists in which pheromone strength of each waypoint decreases exponentially over time, as described by Eq. 4 above. Waypoints are removed once their values drop below a threshold of 0.001.

In the simulation, robots are able to identify and remember the exact locations of depots and the locations of sites visited in the last foraging round, but this is not realistic for physical robot hardware. To test potential pitfalls of transferring the behavior of simulated robots to physical robots, we simulate sensor errors that reflect those of iAnt robots.

Following the method used by Fricke et al. (2016), we simulate sensor error by applying Gaussian noise when robots attempt to return to a previous location via pheromones or site fidelity, mimicking that of the iAnt robots as described in Hecker et al. (2013). The standard deviation around the intended location increases with the distance the robot travels to its intended destination position, p . This reflects the greater accumulation of odometry errors over longer distances. The distance p is multiplied by a noise coefficient, e , in order to change noise severity. Noise is generated by: $noise \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = e \times p$. For example, given the maximum travel distance of CPFA swarms to the corner of an arena, $p = 7.0$ m, and a noise coefficient of $e = 0.4$, returning robots will arrive within approximately 3 m of their goal destination 68% of the time. MPFA swarms, which have shorter average travel distances (as shown below), and there-

¹ <https://github.com/BCLab-UNM/MPFA>.

² <https://www.youtube.com/playlist?list=PLkjRv85y76xl7mWU18YNJcbuBLz0t-1cC>.

fore lower modeled error, will return to previous locations with higher accuracy.

5 Results

We compare $MPFA_{dynamic}$ to the CPFA, $MPFA_{static}$, $MPFA_{global_static}$, and $MPFA_{global_dynamic}$. We replicate each experiment in 100 trials and report the median time for the swarm to collect targets in each experiment. We also examine several components of foraging time: travel time, search time, and collision time. We test the scalability of the algorithms by increasing the arena size and swarm size and examining the trends in foraging time. We demonstrate that $MPFA_{dynamic}$ is faster than the CPFA and $MPFA_{static}$, and similar in performance to $MPFA_{global_static}$ and $MPFA_{global_dynamic}$. We present our results in notched box plots to show which results are statistically different. We used the Mann-Whitney U test to compare the results of the $MPFA_{dynamic}$ to each of the four other algorithms. The statistical significance is explicitly indicated by asterisks in figures ($p < 0.001$). Additionally, the notch on each plot indicates the 95% confidence interval of the medians. If the notches of two boxes do not overlap, this indicates a statistically significant difference between the medians.

5.1 Foraging performance

5.1.1 Foraging time

In our simulation, the foraging time of each swarm is the time required to collect 88% (as described above in Sect. 4) of the targets. The configuration of each experiment is shown in Table 2. Figure 7 shows the time for each algorithm to collect

88% of the targets for three different classes of distributions of targets.

Our experiments show that $MPFA_{dynamic}$ outperforms the CPFA and the $MPFA_{static}$ in all three distributions. The $MPFA_{dynamic}$ is 47% faster than the CPFA in the partially clustered distribution and 18% faster than the $MPFA_{static}$ in the clustered distribution. Surprisingly, the $MPFA_{dynamic}$ is either faster than both globally informed algorithms in the clustered distribution or statistically indistinguishable from them in the partially clustered distribution. It is slightly slower than $MPFA_{global_dynamic}$ in the random distribution.

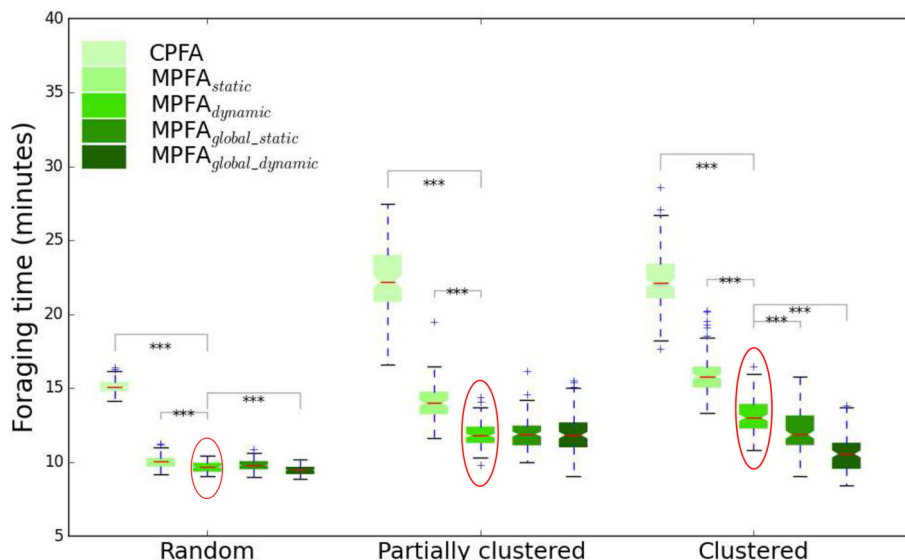
5.1.2 Robustness to error

We examine the effect of localization error on foraging performance. Figure 8 shows foraging time for swarms given simulated error with a noise coefficient $\epsilon = 0.4$. This error results in robots returning to pheromone or site fidelity waypoints at the far corner of a 10×10 m arena being normally distributed around the intended destination, with 68% of the robots within 3 m of the intended destination, a substantial amount of error when searching for targets that are 5 cm in radius. Our experiments show that the foraging times of all algorithms increase moderately (on average by 16%) with this level of error. However, $MPFA_{dynamic}$ still outperforms the CPFA and $MPFA_{static}$ in all three distributions with statistical significance levels similar to the error-free evaluations.

5.2 Search and travel time

Foraging time is composed of two distinct activities. When a robot departs from a depot, it travels to a location where it starts a localized search for targets. Once a target is dis-

Fig. 7 Foraging times for CPFA and MPFA swarms of 24 robots in a 10×10 m arena. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from $MPFA_{dynamic}$ which is emphasized by red ellipses. The performance of each algorithm is represented by a notched box plot in a different shade, ordered left to right, lightest to darkest in the same order indicated in the legend. The notches indicate the 95% confidence interval of the median so that overlapping ranges of the notches indicate statistically indistinguishable results at the $p = 0.05$ level (Color figure online)



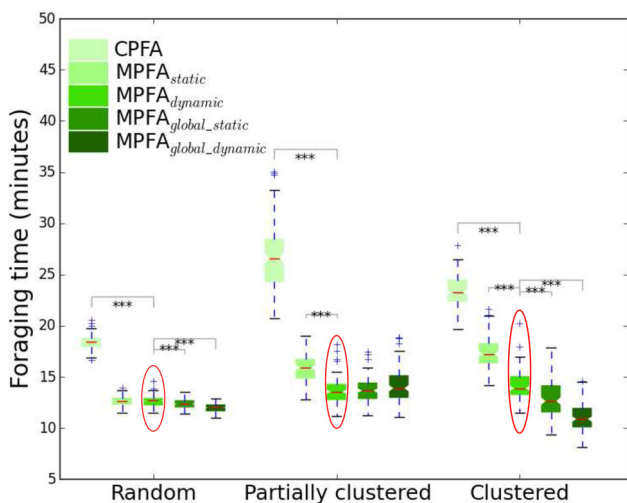


Fig. 8 Foraging times for CPFA and MPFA swarms of 24 robots with noise $\epsilon = 0.4$ in a 10×10 m arena (Color figure online)

covered, the robot takes approximately the same *travel time* back to the depot as it took to travel to the search location.

We measure the total travel time and search time spent by all robots in the swarm. The summed travel and search time of all robots in each swarm are shown in Fig. 9. In the $MPFA_{dynamic}$, travel time is reduced in all cases. Compared to the CPFA, the $MPFA_{dynamic}$ is up to 62% faster (in the clustered distribution); compared to the $MPFA_{static}$ it is up to 30% faster (in the clustered distribution). Robots using the $MPFA_{dynamic}$ also search faster in all cases. Compared to the CPFA it is up to 51% faster (in the partially clustered distribution), and compared to the $MPFA_{static}$ (up to 13.6% faster in the partially clustered distribution). It is also faster than the globally informed MPFAs in the partially clustered distribution. It is slightly slower than $MPFA_{global_dynamic}$ in the clustered distribution.

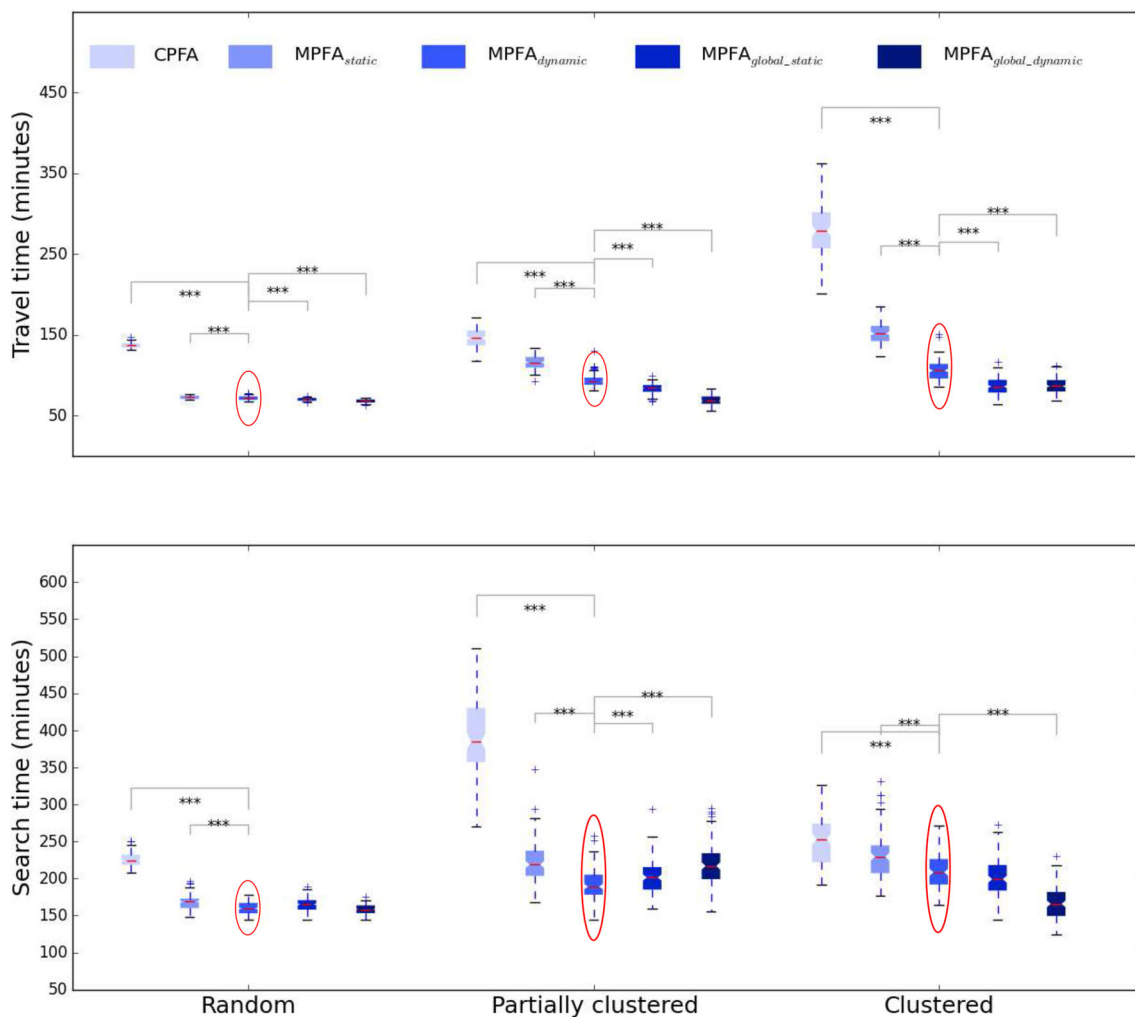


Fig. 9 The search and travel time (per swarm) for the CPFA and MPFAs (Color figure online)

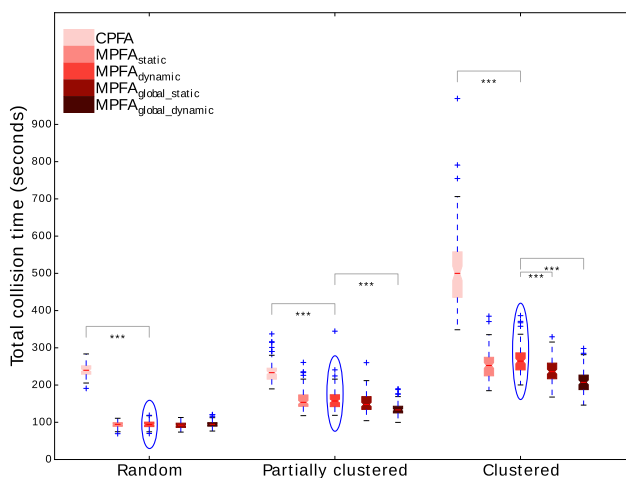


Fig. 10 Total time spent (per swarm) avoiding collisions for the CPFA and MPFAs. The boxplot of $MPFA_{dynamic}$ is emphasized by blue ellipses (Color figure online)

5.3 Collision time

In our simulation, if the distance between two robots is less than 25 cm, each robot will implement collision avoidance. Each robot senses the location of the other and turns left or right in order to avoid a collision, moving approximately 8 cm before resuming traveling. The collision avoidance takes time and will increase foraging times, particularly when the swarm size is large.

Collision time is the time spent to avoid a collision. The total collision time of each swarm is the sum of the total collision avoidance times for all robots in the swarm (shown in Fig. 10). The collision time for $MPFA_{dynamic}$ is less than the collision time for the CPFA in all cases, but it is more than the collision time for the globally informed algorithm with dynamic depots in the partially clustered distribution and for both globally informed algorithms in the clustered distribution. Not surprisingly, collision time is lowest in the random distribution where targets and robots are most dispersed, and highest in the clustered distribution where robots crowd around clustered target locations.

5.4 Scalability

We tested the foraging performance of $MPFA_{dynamic}$ with increased arena sizes and swarm sizes. Figure 11 shows the foraging performance in different arena sizes. Not surprisingly, foraging time increases as the arena size increases. $MPFA_{dynamic}$ outperforms the CPFA and $MPFA_{static}$ in all arena sizes and all three distributions. Its performance is similar to $MPFA_{global_static}$ and $MPFA_{global_dynamic}$.

The increase in foraging time appears to be linear with the length of the foraging arena. However, in the clustered target environment, $MPFA_{dynamic}$ (slope=2.55),

$MPFA_{global_static}$ (slope=2.56), and $MPFA_{global_dynamic}$ (slope=2.21) have improved scalability compared to the CPFA (slope=5.04) and $MPFA_{static}$ (slope=4.61) as evidenced by the more shallow increase in per-robot foraging time with arena size. The slope of the regression for $MPFA_{dynamic}$ is not significantly different from that of $MPFA_{global_static}$ and $MPFA_{global_dynamic}$.

To further test scalability, we create an arena 25 times larger (50×50 m) than the basic (10×10 m) arena and we measure foraging times for swarms of 96 robots.

Figure 12 shows foraging performance in this larger arena. $MPFA_{dynamic}$ still outperforms the CPFA (up to 30% in the clustered distribution) and $MPFA_{static}$ (up to 13% in the clustered distribution) in most cases. The $MPFA_{dynamic}$ is either better than or statistically indistinguishable from the $MPFA_{global_static}$ and $MPFA_{global_dynamic}$ in all cases. These results suggest that the $MPFA_{dynamic}$ is particularly effective for very large swarms and foraging areas.

5.5 Transport to a central depot

Two caveats should be considered in interpreting the above comparisons of the MPFA algorithms to the CPFA. First, because we consider the mobile depots to be robotic agents, this means that the MPFA swarms have four more robots than the CPFA swarms. Second, in cases where the MPFA is used, but targets must ultimately be collected at a central location, the mobile depots would need to transport targets to a single central depot (as is done in the CPFA).

To make a more fair experimental comparison, we added four robots to the CPFA swarm, and we modified $MPFA_{dynamic}$ so that when mobile depots are full (in this case containing 24 targets), they deliver those targets to a single central depot. Foraging robots carrying targets to that depot pause their motion while the depot is traveling to and from the central depot. A demonstration video is available on YouTube.³

Figure 13 compares the $MPFA_{dynamic}$ with central delivery to the CPFA. Central delivery increases the foraging time of the $MPFA_{dynamic}$ by 5.5% and adding 4 additional robots to the CPFA decreased foraging time by 11%.

However, the $MPFA_{dynamic}$ with central delivery is still significantly faster than the CPFA: 22% faster in the random distribution, 36% faster in the partially clustered, and 32% faster in the clustered distribution. Thus, even with central delivery, the MPFA is on average 30% faster than the CPFA.

³ <https://www.youtube.com/watch?v=nihPNSKm4xw&index=7&t=5s&list=PLkjRv85y76xl7mWU18YnJcbuBLz0t-1cC>.

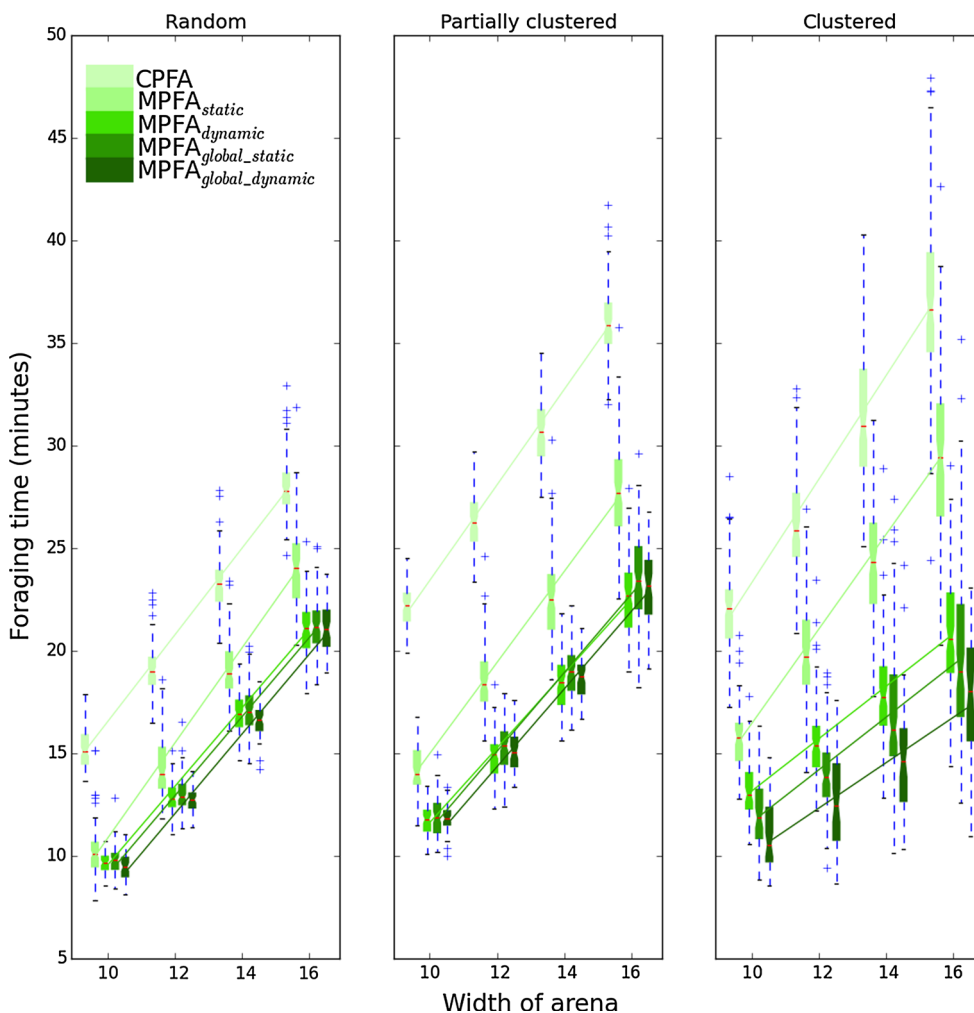


Fig. 11 The foraging time for each swarm for increasing arena sizes. Results are for 100 trials and data for each swarm is shown by the box plot. The lines show the best-fit linear regression (Color figure online)

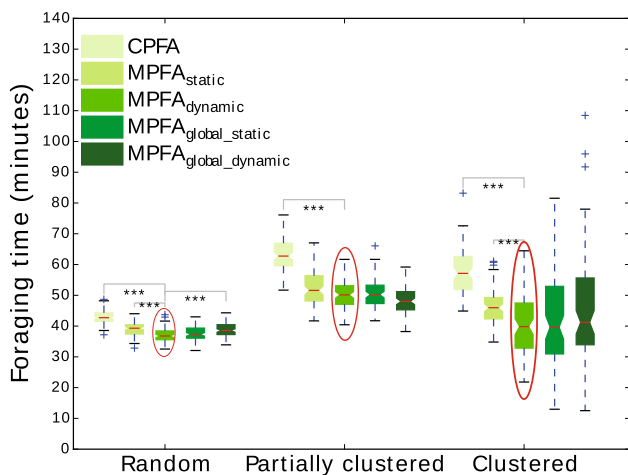


Fig. 12 The foraging time for each swarm of 96 robots in a 50×50 m arena. Results are for 100 replicates for each algorithm. Asterisks indicate a statistically significant difference ($p < 0.001$). The boxplot of $MPFA_{dynamic}$ is emphasized by red ellipses (Color figure online)

6 Discussion

This paper examines the foraging performance of swarms using the multiple-place foraging algorithm with dynamic depots ($MPFA_{dynamic}$). We test 4 variants of the multiple-place foraging algorithm and a central-place foraging algorithm (the CPFA). Because these ant-inspired algorithms are designed for collecting targets quickly rather than for the complete collection of all targets, we report the time required to collect 88% of the available targets in each experiment.

In the first set of experiments with 24 robots in a 10×10 m arena (Fig. 7), the average foraging time of $MPFA_{dynamic}$ across the three target distributions is 41% faster than the centralized CPFA, and 13% faster than $MPFA_{static}$. Its foraging times are similar to $MPFA_{global_static}$ and $MPFA_{global_dynamic}$, illustrating that dynamic depots that respond only to local information are as effective as global methods that require more information to be collected and communicated.

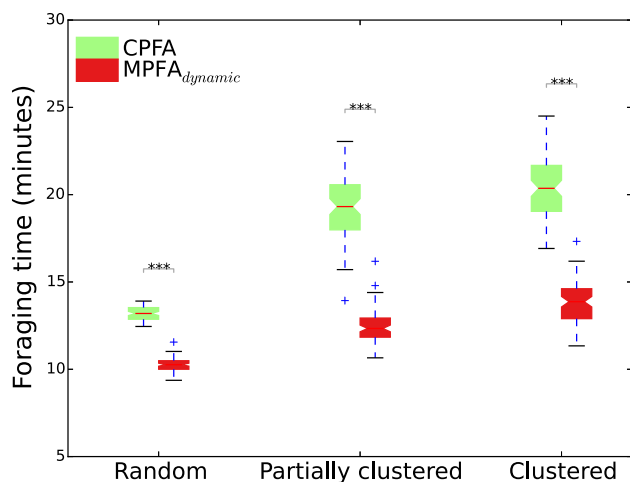


Fig. 13 Foraging times for CPFA swarm of 28 robots and MPFA_{dynamic} swarms of 24 robots in a 10×10 m arena. Depots deliver collected targets to the central placed depot when they have 24 targets. Results are for 100 trials with each swarm. Asterisks indicate a statistically significant difference of the medians ($p < 0.001$) from MPFA_{dynamic} (Color figure online)

Foraging times are reduced in all versions of the MPFA compared to the CPFA, primarily because travel times are dramatically reduced by an average of 49% over all three distributions. Travel times are reduced the most in partially clustered and clustered distributions, and in those distributions MPFA_{dynamic} also has reduced travel times relative to MPFA_{static} (see Fig. 9). The same comparisons are true for search time, but the differences are smaller: MPFA_{dynamic} is 33% faster than the CPFA and 9% faster than MPFA_{static} on average. Collision avoidance times are on average 47% lower for all versions of the MPFA compared to the CPFA (see Fig. 10). Since larger swarms produce more inter-robot collisions and reduce foraging performance, a more efficient collision avoidance strategy for reducing collision time will be included in future work, informed by the adaptive bucket-brigade foraging method introduced in Lein and Vaughan (2009).

In addition to having faster foraging times for all arena sizes and all target distributions, MPFA_{dynamic} is also more scalable than the CPFA and MPFA_{static}. Figure 11 shows that the increase in foraging times with arena size is smaller on the clustered distribution for MPFA_{dynamic} (slope=2.55) compared to the CPFA (slope=5.04) and MPFA_{static} (slope=4.61). MPFA_{dynamic} foraging times are particularly faster for large arenas and clustered targets (e.g., 21% faster than MPFA_{static} in a 16×16 m arena, and 30% faster in the 50×50 m arena in Fig. 12).

We also demonstrated how the MPFA_{dynamic} can be used to complete the central-place foraging task faster than the CPFA. For these experiments, the mobile depots deliver their contents to a central depot when they are full, and the CPFA

is given 4 additional robots for a more fair comparison. The transport time of a small number of trips to the central depot is minimal and has little effect on the total foraging time. Figure 13 shows that the MPFA_{dynamic} is still 30% faster than the CPFA.

Thus, by using mobile depots that adapt to local conditions, MPFA_{dynamic} is an efficient and scalable solution that minimizes the central-place bottleneck of the CPFA and improves foraging times compared to MPFA_{static} without requiring any global information.

6.1 Online decision-making in response to local information

Real-time adaptive response is a key component of MPFA_{dynamic}. Foraging robots adaptively respond to the targets they detect in the environment by making a real-time decision to communicate pheromones or to return to a previous search location using site fidelity. Depots make real-time adjustments each time a foraging robot drops off a target in order to move toward the centroid of the known target locations. The CPFA and MPFA_{static} are both effective algorithms (Hecker and Moses 2015; Lu et al. 2016a); however, the additional real-time decision-making of mobile depots decreases foraging times in all of our experiments, and the decrease is greatest in the largest arenas and for clustered target distributions (Fig. 12).

MPFA_{dynamic} is particularly effective compared to MPFA_{static} for highly clustered targets. Foraging robots adaptively respond to clusters by using pheromones and site fidelity; in turn, depots respond to the observations of the foraging robots by moving closer to clusters of targets. Thus, both foragers and depots respond to the environment to reduce the time to collect targets. The adaptive communication of foragers reduces search time, and the adaptive movement of depots reduces travel time. Real-time adaptation to communicated information about target locations is particularly valuable when targets are highly clustered because each target found in a cluster confers more information about the location of other targets in that cluster (Flanagan et al. 2011).

The benefits of dynamic depot movement are likely to be even greater when targets are ephemeral, i.e. appearing and disappearing over time, and when the targets themselves are mobile because depots can move to new locations where targets appear so that they can be collected quickly (Levin 2016).

In addition to real-time decision-making, robots also respond adaptively to their environments over evolutionary time. Our previous work showed that robots adjust dispersal parameters and the rate of communication to avoid overcrowding between depots and nearby piles when they are tested in environments with clustered targets (Lu et al.

2016a). This results in scalable algorithms, and scalability is improved further with MPFA_{dynamic}.

6.2 Broader implications for scalable design

A fundamental problem in computer science is the design of scalable solutions that perform well as the problem size increases. As computational systems interact more with the environment in which they are situated, particularly if they navigate physical space using stochastic movement, they become increasingly analogous to biological systems (Kleinberg 2007). In biology, scaling theory investigates how efficiently targets can be moved through spatial networks (West et al. 1997; Savage et al. 2008; Banavar et al. 2010). Scaling theory makes predictions beyond individual organisms, to explain the efficiency of ant colonies (Hou et al. 2010), societies (Moses and Brown 2003; Brown et al. 2011), and even computer chip design (Moses et al. 2016).

MPFA_{dynamic} offers a new perspective on the scaling problem. The use of multiple depots in the MPFA improves scaling compared to the CPFA, and having adaptive and dynamic mobile depots increases scalability even further. This advantage is particularly apparent when the targets to be transported are grouped into clusters, rather than randomly scattered, and when transport distances are very large (i.e., MPFA_{dynamic} is nearly twice as fast as the CPFA and MPFA_{static} for clustered targets in the largest 50 × 50 m arena as shown in Fig. 12). This suggests that adaptive mobile agents in robotic swarms can mitigate the inherent scaling inefficiencies of central-place transport. The experiments in Fig. 13 show that this holds even when the dispersed depots transport targets to a central nest.

The success of MPFA_{dynamic} also provides insight into biological mechanisms that improve scalability. While most biological scaling theory focuses on fixed, centralized transport networks, there are biological systems that have features similar to the depots of the MPFA. For example, the immune system, with multiple lymph nodes distributed throughout the search space of an organism, results in a highly scalable immune response with trillions of cells (Banerjee and Moses 2010a). Our prior works suggest that the partially distributed architecture of the immune system (one in which lymph nodes act as depots) is critical for overcoming the inherent scaling limitations of transporting targets (Moses and Banerjee 2011).

There is also evidence of mobile depots in the largest colonies of ants: invasive Argentine ant colonies are composed of a network of mobile nests connected by trails, and the dynamic patterns of recruitment and allocation of foragers to nests increases foraging efficiency (Flanagan et al. 2013; Lanan 2014). These examples suggest that in biological systems, as well as in robotic swarms, adaptive, decentralized, and mobile aggregation points increase search efficiency.



Fig. 14 The physical robot on which components of the CPFA have been implemented

Thus, biological systems have evolved architectures with the same advantages of MPFA_{dynamic}: faster search and foraging, fewer collisions, and reduced travel time.

6.3 The path to implementation

Our simulations suggest that the MPFA_{dynamic} is robust to the errors that we previously identified as important in our iAnt physical robots, namely error in returning to locations indicated by site fidelity or pheromone waypoints. When we included substantial error in our simulations (leading to robots being up to 3 m away from intended destinations), it reduced foraging performance by an average of 16% (see Fig. 8) across all of the MPFA and CPFA experiments, but the MPFA continued to be faster than the CPFA.

However, we do not expect that foraging performance in real robots will be as fast as it is in the simulation. In order to implement multiple-place foraging with dynamic depots in a physical robot swarm, we will use our existing robot platform, designed by our lab for the NASA Swarmathon competition (Secor 2016; Ackerman et al. 2018). The code for the competition is available on GitHub.⁴ Swarmathon robots are outfitted with a grasping apparatus that facilitates the pick up and drop off of target cubes (see Fig. 14). Structural modifications will be required to convert four Swarmathon robots into mobile depots capable of holding collected targets inside of a container.

Swarmathon robots are considerably larger and more powerful than the foot-bots modeled in ARGoS. Swarmathon robots run the Robot Operating System (ROS), a distributed message-passing framework with an extensive, user-supported package that helps streamline algorithm implementation (Quigley et al. 2009). Other swarm algorithms, including the DDSA and components of the CPFA, have been implemented in ROS and subsequently tested in the multi-robot simulator Gazebo (Koenig and Howard

⁴ <https://github.com/BCLab-UNM/SwarmBaseCode-ROS>.

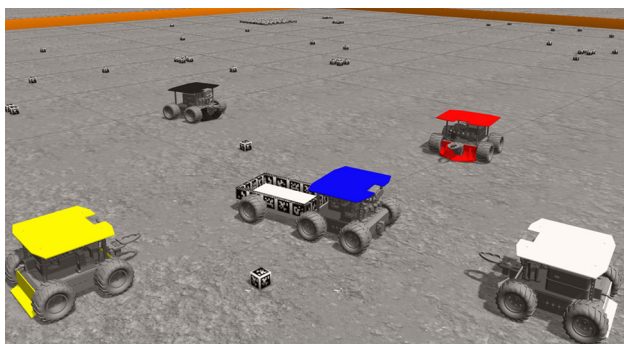


Fig. 15 A mobile depot with blue cover and four foraging robots simulated in Gazebo

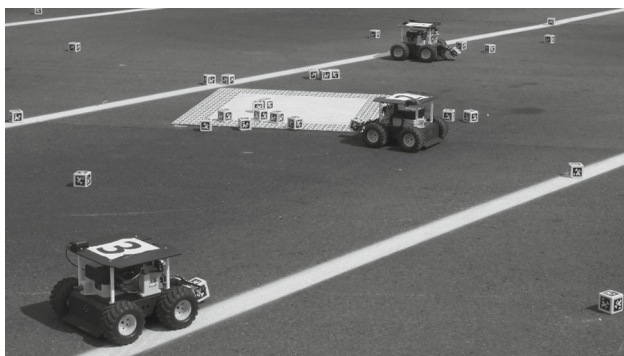


Fig. 16 A swarm of 6 robots (3 shown) implementing central place foraging in a 23×23 m arena

2004). Based on our experience with these existing foraging algorithms, we implemented a dynamic depot with Swarmathon robots in Gazebo (Fig. 15). A demonstration video showing central-place foraging in Gazebo and physical Swarmathon robots, as well as a mobile depot simulated in Gazebo is available.⁵ The next step is making a straightforward extension to the simulation to include multiple depots implementing pheromone waypoints associated with each depot and centroid estimation by each depot in order to fully implement MPFA_{dynamic}.

The biggest benefit of implementing the MPFA on the ROS and Gazebo system developed for the NASA Swarmathon is that code is very easily transferred from Gazebo onto the onboard Linux computer on the Swarmathon robots. The ease of this transfer is evidenced by the 19 college teams that successfully transferred their Gazebo code to up to 6 Swarmathon robots that operated in outdoor arenas up to 23×23 m for the Swarmathon competition (see Fig. 16). These teams showed that Swarmathon robots can reliably return to collection points, and implement site fidelity and recruitment to waypoints. Full implementation of the MPFA

in 24 physical robots in an outdoor environment is the next step to demonstrate truly scalable foraging swarms of robots.

Acknowledgements This work was supported by a James S. McDonnell Foundation Complex Systems Scholar Award and NASA MUREP #NNX15AM14A funding for the Swarmathon. We thank the UNM Center for Advanced Research Computing for computational resources used in this work. We gratefully acknowledge members of the Moses Biological Computation Lab for their assistance with the dynamic multiple-place foraging swarm robotics project: thanks to G. Mathew Fricke for discussing the statistical analysis of our results and implementing MPFA in Gazebo, and to Antonio Griego for developing the CPFA algorithm in ARGoS. Thanks also to Carlo Pinciroli for discussing the implementation in ARGoS.

References

- Ackerman, S. M., Fricke, G. M., Hecker, J. P., Hamed, K. M., Fowler, S. R., Griego, A. D., Jones, J. C., Nichol, J. J., Leucht, K. W., & Moses, M. E. (2018). The swarmathon: An autonomous swarm robotics competition. In *2018 IEEE international conference on robotics and automation (ICRA)* (in review).
- Arthur, D. & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07* (pp. 1027–1035). Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Bac, C. W., Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*(6), 888–911.
- Banavar, J. R., Moses, M. E., Brown, J. H., Damuth, J., Rinaldo, A., Sibly, R. M., et al. (2010). A general basis for quarter-power scaling in animals. *Proceedings of the National Academy of Sciences*, *107*(36), 15816–15820.
- Banerjee, S., & Moses, M. (2010a). *Modular RADAR: An immune system inspired search and response strategy for distributed systems* (pp. 116–129). Berlin, Heidelberg: Springer.
- Banerjee, S., & Moses, M. (2010b). Scale invariance of immune system response rates and times: Perspectives on immune system architecture and implications for artificial immune systems. *Swarm Intelligence*, *4*(4), 301–318.
- Berman, S., Halász, Á., Hsieh, M. A., & Kumar, V. (2008). Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. In *47th IEEE conference on decision and control, 2008. CDC 2008* (pp. 4376–4381).
- Beverly, B. D., McLendon, H., Nacu, S., Holmes, S., & Gordon, D. M. (2009). How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behavioral Ecology*, *20*(3), 633.
- Bezzo, N., Hecker, J. P., Stolleis, K., Moses, M. E., & Fierro, R. (2015). Exploiting heterogeneous robotic systems in cooperative missions (pp. 1–23). [arXiv:1509.00948](https://arxiv.org/abs/1509.00948).
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, *7*(1), 1–41.
- Brooks, R. A., & Flynn, A. M. (1989). Fast, cheap and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, *42*, 478–485.
- Brown, J. H., Burnside, W. R., Davidson, A. D., DeLong, J. P., Dunn, W. C., Hamilton, M. J., et al. (2011). Energetic limits to economic growth. *BioScience*, *61*(1), 19–26.
- Camazine, S., Franks, N. R., Sneyd, J., Bonabeau, E., Deneubourg, J. L., & Theraula, G. (2001). *Self-organization in biological systems*. Princeton, NJ: Princeton University Press.

⁵ https://www.youtube.com/playlist?list=PLkjRv85y76xILHER0ekXVnVTy_z9TMZD4.

- Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., et al. (2010). Artificial pheromone for path selection by a foraging swarm of robots. *Biological Cybernetics*, *103*(5), 339–352.
- Chapman, C. A., Chapman, L. J., & McLaughlin, R. L. (1989). Multiple central place foraging by spider monkeys: Travel consequences of using many sleeping sites. *Oecologia*, *79*(4), 506–511.
- Couture-Beil, A. & Vaughan, R. T. (2009). Adaptive mobile charging stations for multi-robot systems. In *2009 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1363–1368).
- Crist, T. O., & MacMahon, J. A. (1991). Individual foraging components of harvester ants: Movement patterns and seed patch fidelity. *Insectes Sociaux*, *38*(4), 379–396.
- Fewell, J. H. (1990). Directional fidelity as a foraging constraint in the western harvester ant, *Pogonomyrmex occidentalis*. *Oecologia*, *82*(1), 45–51.
- Fink, W., Dohm, J. M., Tarbell, M. A., Hare, T. M., & Baker, V. R. (2005). Next-generation robotic planetary reconnaissance missions: A paradigm shift. *Planetary and Space Science*, *53*(14–15), 1419–1426.
- Flanagan, T. P., Letendre, K., Burnside, W., Fricke, G. M., & Moses, M. E. (2011). How ants turn information into food. In *2011 IEEE symposium on artificial life (ALIFE)* (pp. 178–185). IEEE.
- Flanagan, T. P., Letendre, K., Burnside, W. R., Fricke, G. M., & Moses, M. E. (2012). Quantifying the effect of colony size and food distribution on harvester ant foraging. *PLOS ONE*, *7*(7), 1–9.
- Flanagan, T. P., Pinter-Wollman, N. M., Moses, M. E., & Gordon, D. M. (2013). Fast and flexible: Argentine ants recruit from nearby trails. *PLOS ONE*, *8*(8), 1–7.
- Fricke, G. M., Hecker, J. P., Griego, A. D., Tran, L. T., & Moses, E. M. (2016). A distributed deterministic spiral search algorithm for swarms. In *IEEE/RSJ international conference on intelligent robots and systems (IROS 2016)*.
- Gazi, V., & Passino, K. M. (2004). Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *34*(1), 539–557.
- Gordon, D. M., & Kulig, A. W. (1996). Founding, foraging, and fighting: Colony size and the spatial distribution of harvester ant nests. *Ecological Society of America*, *77*(8), 2393–2409.
- Halász, Á., Hsieh, M. A., Berman, S., & Kumar, V. (2007). Dynamic redistribution of a swarm of robots among multiple sites. In *2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2320–2325).
- Hecker, J. P., Carmichael, J. C., & Moses, M. E. (2015). Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 434–440).
- Hecker, J. P., & Moses, M. E. (2015). Beyond pheromones: Evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, *9*(1), 43–70.
- Hecker, J. P., Stolleis, K., Swenson, B., Letendre, K., & Moses, M. E. (2013). Evolving error tolerance in biologically inspired robot swarms. In *In ECAL 2013*.
- Hölldobler, B. (1976). Recruitment behavior, home range orientation and territoriality in harvester ants, *Pogonomyrmex*. *Behavioral Ecology and Sociobiology*, *1*(1), 3–44.
- Hou, C., Kaspari, M., Zanden, H. V., & Gillooly, J. (2010). Energetic basis of colonial living in social insects. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(8), 3634–3638.
- Hsieh, M. A., Halász, Á., Berman, S., & Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, *2*(2), 121–141.
- Jackson, D. E., Martin, S. J., Ratnieks, F. L. W., & Holcombe, M. (2007). Spatial and temporal variation in pheromone composition of ant foraging trails. *Behavioral Ecology*, *18*(2), 444–450.
- Kleinberg, J. (2007). Computing: The wireless epidemic. *Nature*, *449*(7160), 287–288.
- Koenig, N., & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems, 2004 (IROS 2004)*. *Proceedings* (Vol. 3, pp. 2149–2154). IEEE.
- Lanan, M. (2014). Spatiotemporal resource distribution and foraging strategies of ants (hymenoptera: Formicidae). *Myrmecological news/Osterreichische Gesellschaft für Entomofaunistik*, *20*, 53–70.
- Landis, G. A. (2004). Robots and humans: Synergy in planetary exploration. *Acta Astronautica*, *55*(12), 985–990.
- Lein, A. & Vaughan, R. T. (2009). Adapting to non-uniform resource distributions in robotic swarm foraging through work-site relocation. In *2009 IEEE/RSJ international conference on intelligent robots and systems* (pp. 601–606).
- Levin, D. F. (2016). *The environment constrains successful search strategies in natural distributed systems*. Ph.D. Thesis, University of New Mexico.
- Liu, W., & Winfield, A. F. T. (2010). Modeling and optimization of adaptive foraging in swarm robotic systems. *International Journal of Robotics Research*, *29*(14), 1743–1760.
- Lu, Q., Hecker, J. P., & Moses, E. M. (2016a). The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. In *IEEE/RSJ international conference on intelligent robots and systems (IROS 2016)*.
- Lu, Q., Moses, M., & Hecker, J. (2016b). A scalable and adaptable multiple-place foraging algorithm for ant-inspired robot swarms. arxiv.org/abs/1612.00480.
- Moses, M. & Banerjee, S. (2011). Biologically inspired design principles for scalable, robust, adaptive, decentralized search and automated response (radar). In *2011 IEEE symposium on artificial life (ALIFE)* (pp. 30–37). IEEE.
- Moses, M., Bezerra, G., Edwards, B., Brown, J., & Forrest, S. (2016). Energy and time determine scaling in biological and computer designs. *Philosophical Transactions of the Royal Society B*, *371*(1701), 20150446.
- Moses, M. E., & Brown, J. H. (2003). Allometry of human fertility and energy use. *Ecology Letters*, *6*(4), 295–300.
- Moyron-Quiroz, J. E., Rangel-Moreno, J., Kusser, K., Hartson, L., Sprague, F., Goodrich, S., et al. (2004). Role of inducible bronchus associated lymphoid tissue (ibalt) in respiratory immunity. *Nature Medicine*, *10*(9), 927–934.
- Nelson, A., Grant, E., & Henderson, T. (2004). Evolution of neural controllers for competitive game playing with teams of mobile robots. *Robotics and Autonomous Systems*, *46*(3), 135–150.
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, *13*(4), 695–711.
- Nurzaman, S. G., Matsumoto, Y., Nakamura, Y., Koizumi, S., & Ishiguro, H. (2009). Biologically inspired adaptive mobile robot search with and without gradient sensing. In *2009 IEEE/RSJ international conference on intelligent robots and systems*.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2012). Argos: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, *6*, 271–295.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., et al. (2009). Ros: An open-source robot operating system. In *ICRA workshop on open source software*.
- Ritchie, M. E. (2009). *Scale, heterogeneity, and the structure and diversity of ecological communities*. Berlin, Boston: Princeton University Press.
- Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Swarm robotics: SAB 2004 international workshop* (Vol. 3342, pp. 10–20).

- Savage, V. M., Deeds, E. J., & Fontana, W. (2008). Sizing up allometric scaling theory. *PLOS Computational Biology*, 4(9), 1–17.
- Schmolke, A. (2009). Benefits of dispersed centralplace foraging: An individualbased model of a polydomous ant colony. *The American Naturalist*, 173(6), 772–778.
- Sebbane, Y. B. (2012). *Lighter than air robots: Guidance and control of autonomous airships* (p. 58). Dordrecht: Springer.
- Secor, P. (2016). *Nasa swarmathon*. <http://www.nasaswarmathon.com/>.
- Singh, M. K., & Parhi, D. R. (2011). Path optimisation of a mobile robot using an artificial neural network controller. *International Journal Systems Science*, 42(1), 107–120.
- Sumpter, D. J., & Beekman, M. (2003). From nonlinearity to optimality: Pheromone trail foraging by ants. *Animal Behaviour*, 66(2), 273–280.
- Tindo, M., Kenne, M., & Dejean, A. (2008). Advantages of multiple foundress colonies in *belonogaster juncea juncea* I.: Greater survival and increased productivity. *Ecological Entomology*, 33(2), 293–297.
- Wall, M. (1996). *GAlib: A C++ library of genetic algorithm components* (Vol. 87). Cambridge: Mechanical Engineering Department, Massachusetts Institute of Technology.
- West, G. B., Brown, J. H., & Enquist, B. J. (1997). A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309), 122–126.
- Winfield, A. F. T. (2009). *Foraging Robots* (pp. 3682–3700). New York, NY: Springer.



Qi Lu is currently a Ph.D. student in the Department of Computer Science at the University of New Mexico. He is doing research in the Moses Lab and his research interests include autonomous swarm robotics, multi-agent systems, autonomous decision-making, self-organized system, and biologically-inspired computation. He focuses on developing a scalable multiple place foraging algorithm for swarm robotics. He received his B.E from Hubei University, China in 2005, and M.S.

from Aalborg University, Denmark in 2011—both in Computer Science.



Joshua P. Hecker is a Computer Science postdoctoral researcher at UNM. He studies the interactions and interdependencies between the behaviors of biologically-inspired robot swarms and the environments in which they are located. His research focuses on developing robust search algorithms, scalable agent-based models, and efficient physical robots that operate autonomously in real-world environments. He also mentors high school and undergraduate students through SFI's Project GUTS, UNM's NSF STEP and Open House events, and the RoboRAVE and VEX robotics competitions.



Melanie E. Moses is currently an Associate Professor in the Department of Computer Science at the University of New Mexico and External Faculty at the Santa Fe Institute. She is also the Principal Investigator for the NASA Swarmathon, a swarm robotics competition that challenges to develop new swarm robotic algorithms to revolutionize space exploration. She was the co-director of the NIH funded UNM Program in Interdisciplinary Biological and Biomedical Sciences 2013–2015.

Her interdisciplinary research exists at the boundaries of Computer Science and Biology with over 50 peer reviewed publications in computational and mathematical biology and biologically-inspired swarm robotics. She earned a B.S. from Stanford University in Symbolic Systems, an interdisciplinary program in cognition and computation, and a Ph.D. in Biology from the University of New Mexico in 2005.