Central Place Foraging: Delivery Lanes, Recruitment and Site Fidelity

Jason T. Isaacs
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
jason.isaacs@csuci.edu

Nicholas Dolan-Stern
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
nicholas.dolan-stern630@csuci.edu

Mark Getzinger
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
mark.getzinger926@myci.csuci.edu

Ethan Warner
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
ethan.warner975@myci.csuci.edu

Alberto Venegas
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
alberto.venegas889@myci.csuci.edu

Adan Sanchez
Department of Computer Science
California St. Univ. Channel Islands
Camarillo, CA USA
adan.sanchez697@csuci.edu

***Abstract*—Central Place Foraging is a behavior in which many agents search an unknown environment for resources and then collect the resources to return them to a central location. We seek to emulate this behavior using a multi-agent system of wheeled mobile robots. Many attributes of this system, such as inter-agent congestion, inter-agent communication, and the use of memory, can impact the efficiency of the system. This study seeks to evaluate the impact that each of these mechanisms play on the rate of resource collection in Central Place Foraging. Three mechanisms were evaluated including the use of delivery lanes around the central depot, site fidelity, and recruitment. Delivery lanes are claimable points around the central collection depot meant to alleviate inter-agent congestion. If an agent has claimed a delivery lane, then other agents avoid this area until the claiming agent has delivered the resource and left the area. Agents utilizing site fidelity use memory to return to the last known resource location before resuming its search after delivering a resource to the collection depot. Lastly, recruitment uses inter-agent communication to alert other agents to assist with resource collection when resources are found. The behaviors were implemented and evaluated using a high-fidelity simulation environment built with ROS and Gazebo. Each behavior included the same randomized search algorithm which is intended to provide uniform coverage of the environment. Simulations were conducted using three robots across twenty-six randomly generated environments while collecting data on the rate of resource collection. The results show that site fidelity has the largest effect on overall performance of the system. Recruitment has a positive but marginal effect while delivery lanes did not show an increase in overall performance.**

***Index Terms*—Swarm Robotics, Foraging, Site Fidelity, Recruitment, Search**

I. Introduction

The field of Swarm Robotics is concerned with using a large number of cheap autonomous mobile robots to accomplish tasks that may otherwise be difficult for a single more expensive robot to achieve. A canonical example of such a task is that of Central Place Foraging (CPF). In CPF agents are expected to disperse throughout an unknown environment in search of resources. Once a resource is located the agent collects the resource and returns it to a centralized collection depot. This type of behavior has been observed in nature among several species such as ants [1], [2], honeybees [3], [4], and starlings [5] which provide inspiration and motivation for robotics researchers. The promise of Swarm Robotics is that well designed swarms will enable advancements in applications such as planetary exploration [6], manufacturing [7], and land mine detection [8] where a large group of autonomous robots could be managed by a single operator.

There are a number of challenges facing engineers designing robotics systems for CPF. First, efficient search algorithms are needed to ensure a sufficient amount of the resources in an environment are discovered quickly. Ideally, such a search algorithm would be decentralized and require little to no communication between agents. Additionally, the search algorithm should be robust to errors in localization and path following. A second challenge is related to inter-robot interference, particularly near clusters of resources and the central depot. As the number of robots in the swarm increases, the odds that multiple robots will need to share the same space goes up drastically. This leads to a natural question concerning recruitment. Should a robot which has found a group of resources be designed to communicate this information to other robots, so they might assist in the collection portion of the CPF task? This study seeks to better understand these questions by presenting some common approaches to these problems and comparing their effectiveness.

A useful search technique in swarm robotics requires minimal communication [9] and complete coverage of the environment meant to be searched. Minimal communication is necessary because keeping all of the agents up to date on the activities of others scattered throughout the environment might not be feasible in a real world scenario. The search problem tends to be approached in two ways: deterministic or random approaches.

A deterministic search pattern is calculated before run-time to lay out a pattern that typically intends to optimally search the environment in the least amount of time with complete utilization of the agents. With deterministic search algorithms, agents typically collect resources as they search in this preset pattern. While these deterministic algorithms theoretically traverse the environment in the least amount of time, the gathering of resources at the home location causes enough inter-robot congestion as to make the algorithm slower than a comparable randomized algorithm when using a large number of agents [10].

A randomized algorithm does not seek to follow a set pattern, but instead follows a pattern generated by a stochastic process [11]. This can cause longer overall search times, longer times between finding individual targets, and the coverage of the same location repeatedly by multiple robots. While this may seem more inefficient, this approach can actually outperform at the overall task of collection, due to spreading the agents further apart, and thereby reducing inter-robot congestion at any location in the environment [10].

Two common mechanisms studied that are tangential to a search algorithm are site fidelity and recruitment. Both mechanisms are biologically inspired by the behaviors of ants and other insects. Ants are able to use site fidelity in order to return to a good site repeatedly, and able to use pheromones to recruit other ants to resource rich locations [12].

Site fidelity is the idea that upon finding a resource, an agent remembers the location where it was found. After taking that resource back to the home location, the agent returns to the remembered site. This mechanism is useful both in situations where an environment has certain resource rich areas, and to keep returning to areas of the environment in which resources are still being discovered.

While site fidelity is an individual activity, recruitment involves others during the collection process. When an agent locates a resource, it signals in some way in order to let the other agents know. Other agents, especially those who are struggling to find a resource, might now gravitate towards that found resource. While recruitment tends to help locate resources, performance worsens when it is used beyond a small number of agents due to inter-robot congestion [11], [13].

Inter-robot congestion seems to be one of the most important issues within the central place foraging domain, and yet very little has been done to systematically study it [14] [15]. Inter-robot congestion can occur all over the environment, but especially at areas rich in resources, and primarily the home location. Several others have introduced various mechanisms in order to mitigate inter-robot congestion, but it remains a largely unsolved issue in this problem domain.

The main contributions of this paper are twofold. First, we provide a simple decentralized search mechanism that efficiently searches the area for resources without the need for centralized planning or inter-robot communication. Second, we present a simulation study to quantify the relative contributions of three additional CPF mechanisms to resource collection efficiency as compared to the basic CPF algorithm. These mechanisms are site fidelity, recruitment, and delivery lanes.

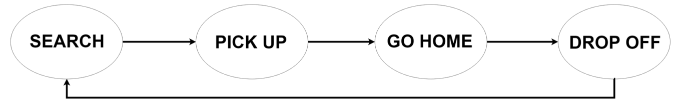


Fig. 1: The block diagram depicts the states associated with basic Central Place Foraging.

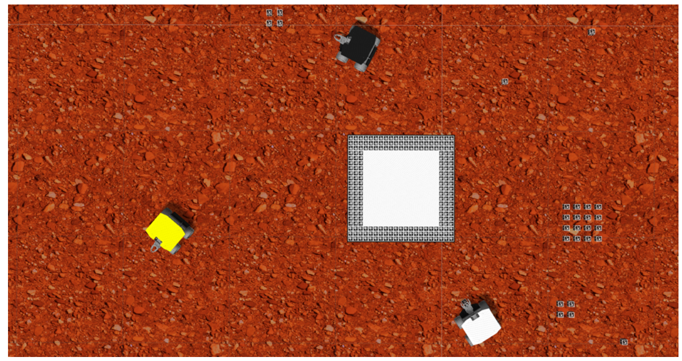


Fig. 2: The robots used in the simulation are shown in addition to AprilTags, representing resources and surrounding the central collection depot.

The remainder of this paper is organized as follows. In Section II, we formally introduce the problem of central place foraging and present the base CPF algorithm. A description of the proposed search algorithm is contained in Section III along with descriptions of the three resource collection efficiency improvement mechanisms that make up the simulation study. The methods of the simulation study are detailed in Section IV, and the results of this study can be found in Section V. Finally, conclusions and directions for future work can be found in Section VI.

II. Central Place Foraging

The basic algorithm for CPF is depicted in Fig. 1. Agents search an area for resources. Upon finding and then picking up a resource, agents will attempt to deliver (DROP OFF) to a central location (HOME). After delivering a resource, agents repeat this process and return to searching. An overhead view of a typical Central Place Foraging simulation can be seen in Fig. 2. In this figure the resources are represented by small cubes covered with AprilTags [16] on all sides. The robots are four wheel skid-steer robots with a front mounted camera to detect resources and a mechanical gripper to grasp resource cubes. Upon collecting a resource the robots return to the home collection location depicted by the white square surrounded by AprilTags. For the sake of the simulation, the AprilTags on resources are of a different identifier than the ones on the collection zone, thus the robots are able to distinguish between the two.

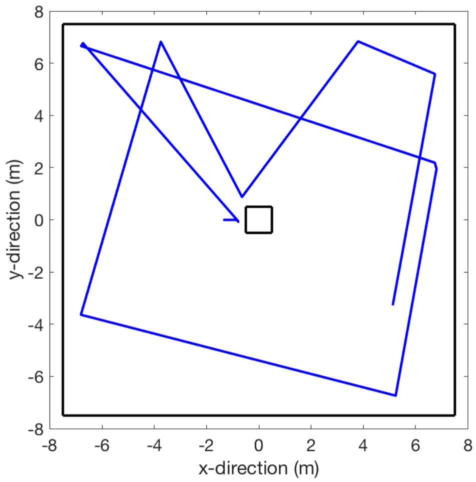


Fig. 3: The trajectory taken by a single robot under the Random Bounce Search Algorithm.

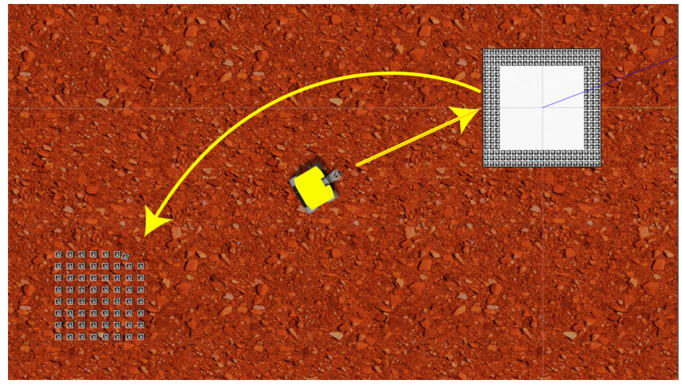


Fig. 4: The foraging agent (yellow) utilizing Site Fidelity returns to the site of the last known resource location. In this example the resources are clustered so the agent would not need to resume a randomized search until the entire cluster of resources was depleted.

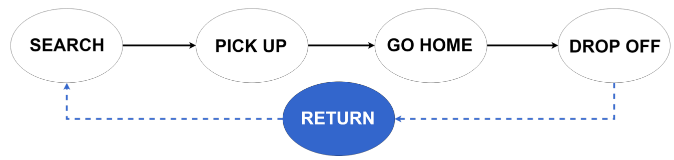


Fig. 5: The block diagram shows the modification of the basic Central Place Foraging algorithm from Fig. 1 to include the (RETURN) state of Site Fidelity.

III. Search Algorithm and Efficiency Mechanisms

A. Random Bounce Search Algorithm

The proposed randomized search algorithm efficiently searches the environment for resources without the need for inter-robot communication or centralized planning. Each robot is assumed to have the ability to sense when they have reached the boundary of the search environment either through some global location information such as Global Positioning System (GPS) or in this case through proximity sensors which can detect the walls of the environment. Additionally, each robot is assumed to have the ability to detect other robots in the environment (using the same proximity sensors) and the central collection depot (in this case using an on-board camera). Each robot simply drives forward at a fixed speed until either the boundary of the environment, another robot, or the central depot is detected. At this time the robot will randomly rotate to a new heading θ_{new} according to the following formula,

$$\theta_{new} = \theta + n \tag{1}$$

where θ is the robot heading at the time of the detection, n is a uniform random variable, and the distribution is decided by which side of the robot encountered the obstacle triggering the rotation. If the robot detects something on the left side $n \sim \mathcal{U}(\frac{\pi}{4}, \frac{3\pi}{4})$, on the right side $n \sim \mathcal{U}(\frac{-3\pi}{4}, \frac{-\pi}{4})$, and in the center $n \sim \mathcal{U}(\frac{3\pi}{4}, \frac{5\pi}{4})$.

B. Site Fidelity

In CPF the act of randomly searching the area can be one of the most time consuming tasks for the agents [4]. The term site fidelity refers to the ability of agents to mark and return to the last known resource location, as can be seen in Fig. 4. This behavior has been observed in various foraging species in nature [4], [17]. By returning to this location, agents have a higher chance of finding more resources. There are two reasons why this may be a good idea. First, in nature resources are rarely distributed uniformly. Typically they exist in groups or clusters. Some examples include water, minerals, and food. By returning to the last known resource location, it is likely that if the resources are clustered, another resource will be detected right way. Second, since resources close to the central depot tend to get discovered first it is likely that if a single resource has been found, then the robot has managed to find an area of the environment that hasn't been cleared yet by other agents. If no new resources are found upon returning to the last known resource location, the robot should still be in an advantageous starting point for the new search as compared to restarting the search from the depot location. The basic Central Place Foraging algorithm can be simply modified, as seen in Fig. 5, to include Site Fidelity, assuming that upon DROP OFF each robot has the ability to return to SEARCH in the direction from which they came.

C. Recruitment

Agents can often spend a large amount of time searching areas of the environment with scarce resources. Agents who have succeeded in finding resources can bring other agents towards them by using recruitment. As seen in Fig. 6, agents broadcast their location when they have found a resource. Those within radio range of that broadcast who are currently searching for a resource modify their search trajectory to move towards the recruitment location. The basic Central Place Foraging algorithm can be modified to include recruitment

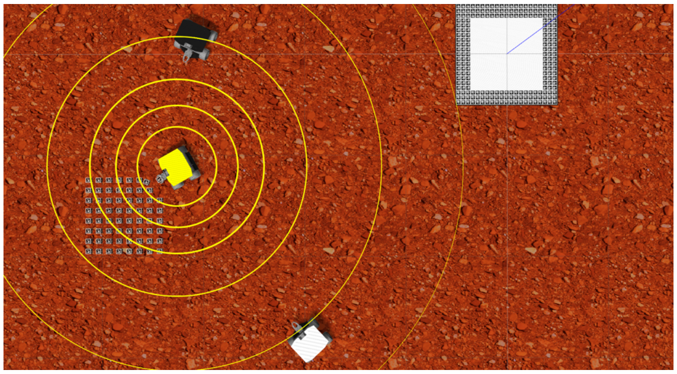


Fig. 6: The recruiting agent (Yellow) broadcasts a message to nearby agents (Black and White) indicating the location of the cluster of resources.

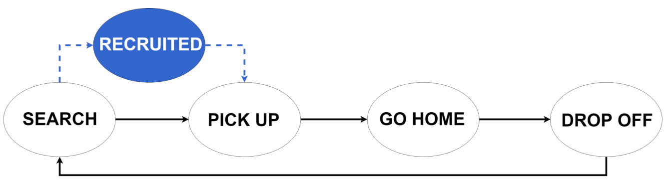


Fig. 7: The block diagram shows the modification of the basic Central Place Foraging algorithm from Fig. 1 to include the (RECRUITED) state where the agent abandons SEARCH in favor of proceeding to the known resource location where PICK UP can occur. If no resource is found the agent returns to SEARCH state (not shown for simplicity).

assuming each robot has a radio transceiver. As seen in Fig. 7, the recruited agent would transition from SEARCH directly to the pickup routine upon arriving at the resource location.

D. Delivery Lanes

Multiple agents delivering resources on the same side of the home location can cause congestion [18]. This problem is often observed in other logistical applications such as airport runways, ticket counters, and trucking warehouses. As can be seen in Fig. 8, delivery lanes allow agents to claim an area in which to go into home, drop off, and back up without colliding with another agent. The number of lanes is able to scale in proportion to the number of agents being used and requires inter-agent communication only near the collection depot location. Figure 9 shows the modifications to the basic Central Place Foraging algorithm to include Delivery Lanes.

IV. Simulation Methods

Two simulation studies were conducted to measure the effectiveness of the proposed approaches. The first study focused on the Random Bounce Search algorithm. In this simulation 3 robots were placed in the simulation environment which included no resources, only boundary walls and the central collection region. The trajectories of each robot were recorded for 1 hour (simulated time) while the Random Bounce Search was executed in order to access the search coverage over time. The

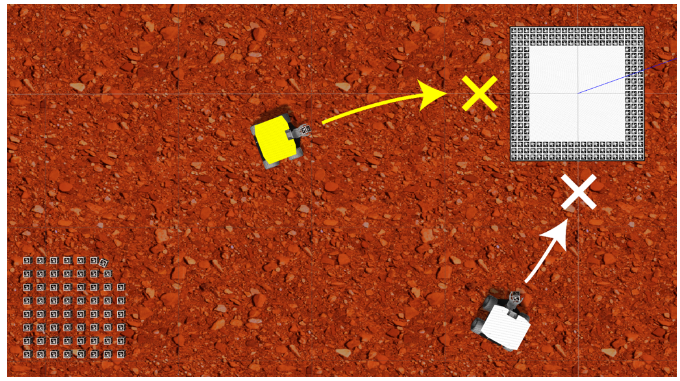


Fig. 8: Both robots are attempting to deliver a resource to the collection zone. The yellow robot has claimed the left delivery lane indicated by the yellow *X*, and the white robot has claimed the bottom delivery lane indicated by the white *X*.

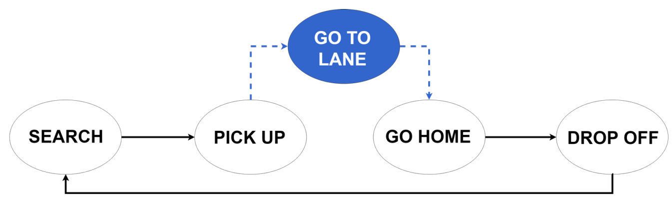
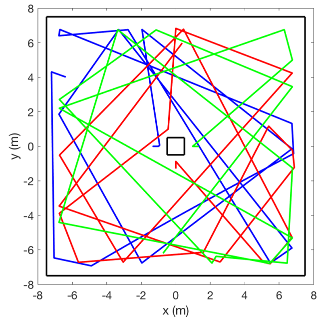


Fig. 9: The block diagram shows the modification of the basic Central Place Foraging algorithm from Fig. 1 to include the additional state of GO TO LANE between picking up a resource (PICK UP) and going to the delivery location (HOME).

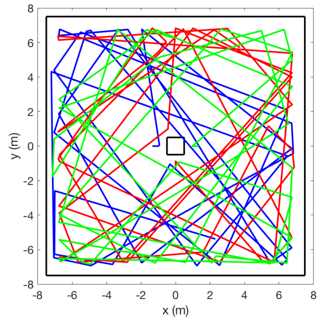
second study evaluated the relative effect on the resource collection rate of the three CPF mechanisms described in Section III, namely the use of site fidelity, recruitment, and delivery lanes. All experiments last 1 hour, and 25 experimental runs were conducted for each configuration of CPF mechanisms. Each experimental run consisted of a new physical distribution of resources, and each resource distribution was tested across all CPF mechanisms.

All algorithms were developed and tested on the NASA Swarmathon 2017 [6] swarm robotics platform. The simulations were implemented in C++ using the Robot Operating System (ROS) framework (Kinetic Kame) [19] and Gazebo plug-ins [20]. The environment was a 15 by 15 meters square with walls at the boundaries, and each robot is equipped with a forward looking camera and three sonar sensors.

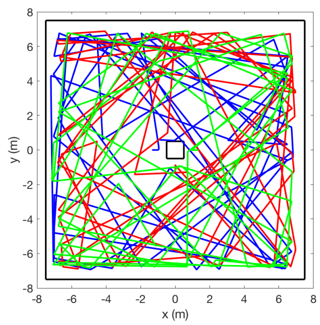
There were 256 resources in this environment that were distributed according to a partially clustered distribution which follows a power law similar to the one described in [10] with 1 cluster of 16, 4 clusters of size 64, 16 clusters of size 4, and 64 single resources. For these simulation experiments $N = 3$ agents were used and all agents were initialized in the search state. In this simulation environment, targets are represented by small cubes with AprilTag patches [16] that



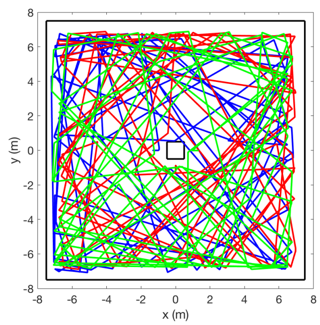
(a) Search Trajectory: 15 min.



(b) Search Trajectory: 30 min.

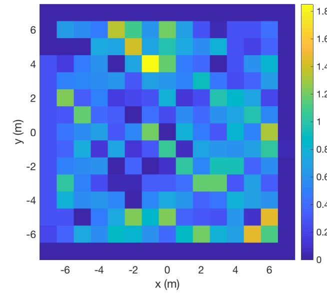


(c) Search Trajectory: 45 min.

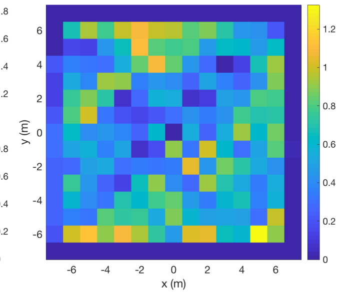


(d) Search Trajectory: 60 min.

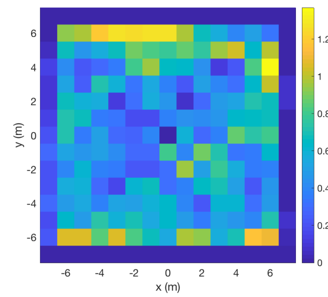
Fig. 10: The trajectories of three robots (Red, Green, and Blue) executing the Random Bounce Search over a one hour period.



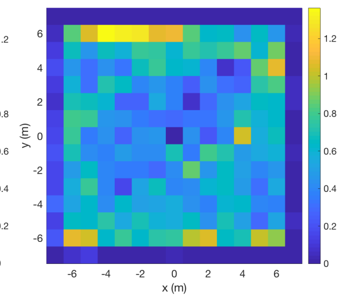
(a) 2D Histogram: 15 min.



(b) 2D Histogram: 30 min.



(c) 2D Histogram: 45 min.



(d) 2D Histogram: 60 min.

Fig. 11: Two dimensional histograms showing percentage of time spent searching each cell in the environment.

are placed on all six sides, and the target detection sensor is a webcam and AprilTag detection software. Targets are picked up with a mechanical gripper when the agent's camera senses an AprilTag on one of the resource cubes.

V. Simulation Results

The search trajectories of the three robots from the first simulation can be seen in Fig. 10. This figure captures the cumulative trajectory at four different points in time at 15, 30, 45, and 60 minutes. As can be seen in the figure, the environment is reasonably searched after 1 hour, however there are areas around the perimeter where there is repeated coverage.

To better illustrate the search coverage, 2D histograms of search coverage are shown in Fig. 11 for the same points in time as Fig. 3. With this view it is clear that more time is spent around the perimeter of the environment, particularly near the top rather than near the center. If uniform search coverage was achieved, we would expect each cell to contain a robot approximately 0.44% of the time. As can be seen in Fig. 11d, after 60 minutes the values for each cell vary between 0% and 1.4%. Some of this discrepancy can be explained by the fact that the robots reflect when they detect the outer boundary or the central collection zone. Therefore, the cells around the outer perimeter and the central cell have very low coverage.

The average number of resources collected versus time for each combination of CPF mechanism is shown in Fig. 12. As seen in the figure, all of the algorithms performed similarly for about 30 minutes into the run. After this point, we begin to see a divergence between the different mechanisms. Algorithms using site fidelity outperformed algorithms without it. In fact, the algorithm using only site fidelity yielded the highest performance. This is because remaining resources tend to be located close together as resources become scarce. This means that returning to a recent pick up location has a higher chance of yielding additional resources than reentering random search. Overall, having site fidelity in a CPF algorithm increases the collection rate of resources as the top four performing mechanisms all included site fidelity. The delivery lane mechanism seems to hinder performance. The added time necessary to align to the lane seems to outweigh its benefits toward reducing congestion. Recruitment mechanisms do not perform as well in the beginning when resources are very dense but their performance increases as resource density decreases. Recruitment starts off slower but its rate of collection remains steady over the duration of the test.

The data presented in Fig. 12 was the average of 25 trials for each CPF mechanism. The confidence intervals were purposely left off the plot to reduce clutter. To better illustrate the benefit of using site fidelity Fig. 13 contains the best performing mechanism (site fidelity only) and the best performing algorithm that didn't use site fidelity (delivery lanes and recruitment) along with their respective 95% confidence intervals. As time goes on site fidelity shows a significant improvement over the combinations of mechanisms that don't include site fidelity.

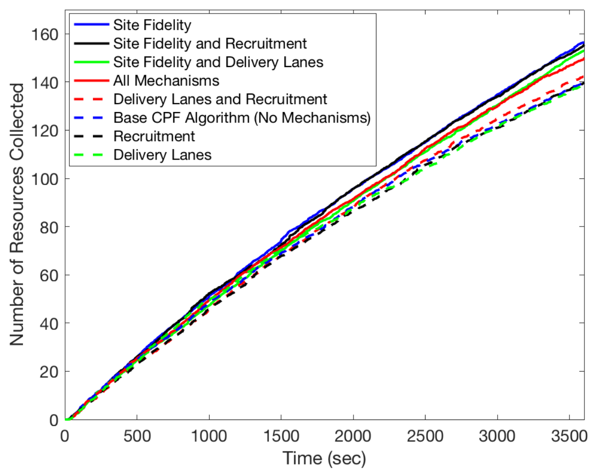


Fig. 12: The number of resources collected is plotted versus time for each combination of mechanisms including base CPF.

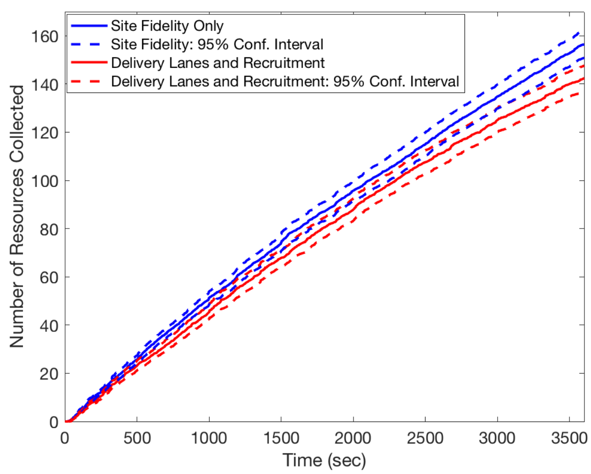


Fig. 13: The number of resources collected is plotted versus time for Site Fidelity only and the combination of Delivery Lanes and Recruitment. The latter being the best performing combination of mechanisms which didn't contain Site Fidelity. The 95% confidence intervals are included as dashed lines.

VI. Conclusions and Future Work

We have proposed a decentralized randomized search algorithm for CPF that while simple to implement provides sufficiently uniform search coverage. Additionally, we performed a simulation study of three mechanisms typically considered to be enhancements: site fidelity, recruitment and delivery lanes. This work could be expanded in several ways. First, these tests could be repeated with varying amounts of robots greater than three. Second, other types of environments, such as those that include obstacles or differing resource configurations could be simulated. Finally, the experiment could be repeated with additional foraging mechanisms or improved algorithms for delivery lanes, recruitment, and site fidelity.

References

[1] R. P. Feynman and R. Leighton, *"Surely you're joking, Mr. Feynman!": adventures of a curious character*. Random House, 1992.

[2] K. Holder and G. Polis, "Optimal and central-place foraging theory applied to a desert harvester ant, pogonomyrmex californicus," *Oecologia*, vol. 72, no. 3, pp. 440–448, 1987.

[3] A. Kacelnik, A. I. Houston, and P. Schmid-Hempel, "Central-place foraging in honey bees: the effect of travel time and nectar flow on crop filling," *Behavioral Ecology and Sociobiology*, vol. 19, no. 1, pp. 19–24, 1986.

[4] T. Mosqueiro, C. Cook, R. Huerta, J. Gadau, B. Smith, and N. Pinter-Wollman, "Task allocation and site fidelity jointly influence foraging regulation in honeybee colonies," *Royal Society Open Science*, vol. 4, no. 8, p. 170344, 2017.

[5] A. Kacelnik, "Central place foraging in starlings (sturnus vulgaris). i. patch residence time," *The Journal of Animal Ecology*, pp. 283–299, 1984.

[6] "Learn More NASA Swarmathon," 2017, Retrieved on 10/19/2017 from http://nasaswarmathon.com/about. [Online]. Available: http://nasaswarmathon.com/about

[7] K. L. Doty and R. E. Van Aken, "Swarm robot materials handling paradigm for a manufacturing workcell," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 778–782.

[8] R. Cassinis, G. Bianco, A. Cavagnini, and P. Ransenigo, "Strategies for navigation of robot swarms to be used in landmines detection," in *Advanced Mobile Robots, 1999.(Eurobot'99) 1999 Third European Workshop on*. IEEE, 1999, pp. 211–218.

[9] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar 2013.

[10] G. M. Fricke, J. P. Hecker, A. D. Griego, L. T. Tran, and M. E. Moses, "A distributed deterministic spiral search algorithm for swarms," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4430–4436.

[11] J. P. Hecker and M. E. Moses, "Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms," *Swarm Intelligence*, vol. 9, no. 1, pp. 43–70, Mar 2015.

[12] J. P. Hecker, J. C. Carmichael, and M. E. Moses, "Exploiting clusters for complete resource collection in biologically-inspired robot swarms," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 434–440.

[13] P. E. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, and M. Gini, "Communication strategies in multi-robot search and retrieval: Experiences with mindart," in *Proc. 7th Int. Symp. Distributed Autonomous Robotic Systems*. Citeseer, 2004.

[14] N. Dolan-Stern, K. Scrivnor, and J. Isaacs, "Multimodal central place foraging," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, Jan 2018, pp. 72–78.

[15] L. S. Marcolino, Y. T. dos Passos, Á. A. F. de Souza, A. dos Santos Rodrigues, and L. Chaimowicz, "Avoiding target congestion on the navigation of robotic swarms," *Autonomous Robots*, vol. 41, no. 6, pp. 1297–1320, 2017.

[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407.

[17] B. D. Beverly, H. McLendon, S. Nacu, S. Holmes, and D. M. Gordon, "How site fidelity leads to individual differences in the foraging activity of harvester ants," *Behavioral Ecology*, vol. 20, no. 3, pp. 633–638, 2009.

[18] K. Letendre and M. E. Moses, "Synergy in ant foraging strategies: memory and communication alone and in combination," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 41–48.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3.