Conference on Autonomous Robot Systems and Competitions

Volume No. | Issue No.

2021-04

Congestion Strategies for Clustered Central Place Foraging

Jason Isaacs
California State University Channel Islands

Congestion Strategies for Clustered Central Place Foraging

Apurva Bharaswadkar
Department of Computer Science
California State University Channel Islands
Camarillo, USA
apurvagopal.bharaswadkar420@myci.csuci.edu

Vida Vakilian
Department of Computer Science
California State University Channel Islands
Camarillo, USA
vida.vakilian@csuci.edu

Brian Thoms
Department of Computer Science
California State University Channel Islands
Camarillo, USA
brian.thoms@csuci.edu

Jason T. Isaacs
Department of Computer Science
California State University Channel Islands
Camarillo, USA
jason.isaacs@csuci.edu

***Abstract*—In central place foraging tasks, multiple robots search for and gather resources in an environment. The robots then proceed to deposit the collected resources in a single central location. The performance of central place foraging approaches is reduced due to congestion around the central collection point. The congestion problem is made worse in the case where resources are distributed in clusters whereby several robots collect resources in the same area and deliver those resources to the collection point along the same path. The approach proposed here seeks to alleviate this congestion problem through simple path planning strategies that reduce the number of inter-robot collisions. Path Planning And Collision Avoidance Algorithm For Clustered Central Place Foraging (PPCA-CCPFA) addresses congestion by detecting possible inter-robot collisions and finding alternate collision free paths for each robot. We compare our approach to the Distributed Deterministic Spiral Search Algorithm (DDSA). This approach provides a notable increase in the performance of DDSA in cases where resources are distributed in a single cluster. A simulation study was conducted using the swarm robotics simulation tool ARGoS to measure the effectiveness of the proposed approach as measured by the number of resources collected per unit time and by the number of inter-robot collisions per unit time.**

***Index Terms*—swarms, cooperating robots, planning, scheduling and coordination, collision avoidance, path planning for multiple mobile robots or agents**

I. Introduction

The research area of swarm robotics seeks to create robust and complex group behaviors through the use of many simple autonomous individual agents combined in large groups called swarms [1], [2]. The promise of swarm robotics is that a large swarm of inexpensive robots offer advantages over a single expensive robotic in terms of performance, robustness, flexibility, and scalability [3]. Central Place Foraging, a subproblem of Swarm Robotics, is gaining attention due to applications such as planetary exploration [4], mining, and land mine detection [5]. In particular, the National Aeronautics and Space Administration (NASA) is interested in using swarms of rovers to explore other planets such as Mars in search of ice for *in situ* resource utilization [4].

Spiral search patterns for foraging are studied extensively and have been found to provide desirable performance [6]–[8]. They guarantee collection of nearest targets first, and they have complete coverage of the area with minimum sampling [7]. The Distributed Deterministic Spiral Search Algorithm (DDSA) generalizes a single robot square spiral to any number of robots [7]. Ryan and Hedrick proposed a square search pattern for fixed-wing unmanned aerial vehicle (UAV) for searching water targets which is similar to DDSA [9]. Other similar approaches use parallel searching in the plane with a fixed number of robots [10], use a distributed spiral search algorithm for the odor localization problem as observed in ants [11], [12], or use a search pattern consisting of system of loops of ever increasing size centered about the origin with path integration as observed in Cataglyphis ants [13]. Others perform a spiral search by equally partitioning the environment among multiple robots [14], use a circular distributed spiral search for multiple robots whose movements are coordinated using shared data structure [15], or use a deterministic interlocking spiral starting from a common point for multiple agents searching targets in coordination [16].

In Central Place Foraging, robots search for resources in the environment. When a new resource is discovered the robot returns to the nest to deposit the found resource and resumes the search pattern once the collected resource is deposited. When robots leave their search pattern and go towards the nest or vice versa, there may be cases where multiple robot paths intersect or are collinear. Typical approaches to central place foraging utilize reactive inter robot collision avoidance wherein robots slightly turn off their course to avoid each other and retry to get on their original course. This has been found to increase the time required to collect resources by the robot swarm [7]. In both the DDSA [7] and CPFA [17], [18] papers, targets were collected faster in the uniform resource distribution than in the clustered resource distribution. This can be due to the unequal allocation of targets to the robots and increased collisions between the robots. This observation inspires the focus of this work on collision avoidance for clustered resource distributions.

While designing distributed foraging algorithms for multiple robot systems, interference can be considered a pragmatic tool for evaluating the performance of these algorithms. The mathematical formulation for multi-robot task allocation with deadlines considering the effect of interference is formulated in [19]. The research models interference as a linear function and studies how interference affects the performance of task allocation in multiple robots. The optimal solution is obtained by solving the linear integration function. Similarly, [20] presents a mathematical model of homogeneous foraging robots with the goal of understanding the effects of inter-robot collision on their performance. The paper studies two foraging cases. The first case, where homogeneous robots only collect objects, and the second case where the homogeneous robots find and deposit the object at a predefined "home" location. It is observed that in the first case, the foraging performance improves with the swarm size. In the second case, it is found that the performance is maximized for an optimal swarm size. Above the optimal swarm size inter-robot collision causes the individual robot's performance to be a monotonically decreasing function of swarm size.

The most similar approach to the proposed approach is in [21]. In this paper a "holding pattern" is used for depositing the detected target to the "nest" location. This is similar to the idea used at airports to avoid congestion and collision of the airplanes. If the robots collect the resource from the same cluster and are close to each other, they take turns to go to the "home" location instead of all going together. The robots pick up a closest of four points around the "home" location forming larger triangular paths resulting in lesser collisions.

In this work, path planning and collision avoidance is integrated with an existing central place foraging algorithm, DDSA, to improve its performance. The main contributions of this paper are two-fold:

1) Develop a path planning and collision avoidance technique for a multi-robot system. This helps reduce the physical interference of robots with coincident paths using spatial delay and intersecting paths by adding time delay.
2) Evaluate the performance of the multi-robot foraging system in terms of target collection rate and average collision rate using the ARGoS swarm simulator.

The remainder of this paper is organized as follows. In Section 2, the problem of congestion in central place foraging is described. The method for multi-robot path planning and collision avoidance for clustered resource distribution problem is presented in Section 3. Section 4 describes the simulation tests used to compare the results of the proposed algorithm with that of DDSA. The results of these tests are discussed in Section 5, which is followed by some final conclusions and directions for future work in Section 6.

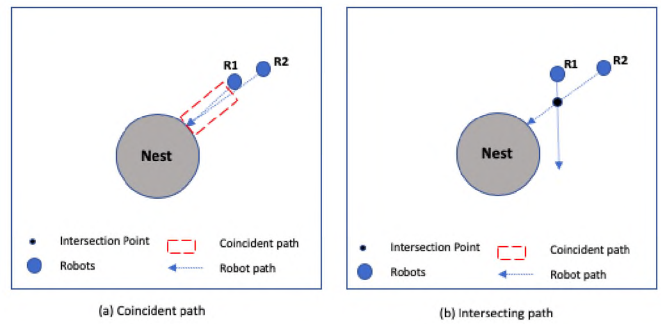

Fig. 1. Inter robot collision cases

II. Problem Statement

The central place foraging task consists of multiple agents searching for desired targets in an unexplored environment and depositing the found targets at a central "home" location. Analysis, understanding, and evaluation of foraging tasks can be efficiently carried out by focusing on a single behavior or combination of behaviors. The motivation of this work is to focus on path planning and collision avoidance for the multi-robot central place foraging task.

Resources can be distributed spatially and temporally [22]. Resource distribution can be uniform, clustered, or partially clustered. Usually the resources found are clumped in varying sizes. The distribution of the resources affect the performance of multi-robot central place foraging tasks. When the resources are uniformly distributed, all the robots benefit. However, if the resources are distributed in clumped patches, robots spend a great amount of time avoiding physical interference near the central "home" location and the clumped resource. The effect of interference is proportional to the swarm size. Thus, there is a need of path planning and collision avoidance technique to reduce the inter-robot collision and lower the time expense on the collision avoidance. This paper focuses on developing a path planning and collision avoidance method for a robot swarm collecting resources from a cluster and depositing them at the central depot as resources are usually found in clumps.

Given a swarm size R, the robots have to search the space for resources and deposit the collected resources at a central depot. There are chances of collisions when more than one robot travels from the nest location to the search position or vice versa. The colliding robots can be in the same, different or opposite direction. The collisions can be avoided by adjusting the robot speeds, adding a time delay or choosing an alternate path. The possibility of collision exists when robot paths intersect at a particular point called intersection point or when robot paths are coincident and there are multiple points of collision. Figure 1 depicts the above mentioned cases.

There can be more than two robots having coincident paths (robot paths very close to each other or overlapping) when they simultaneously collect the resources from cluster and also when they return to their respective search positions from the "nest". In case (a) of Figure 1, collision is avoided by adding

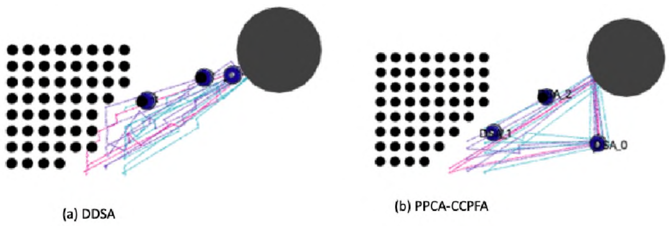

Fig. 2. Comparison of Coincident Case in PPCA-CCPFA and DDSA

spatial delay so that there is a free passage for robots to travel. In case (b) of Figure 1, the path of the robot searching and the robot going to or coming from the “nest” may intersect at a particular point. The collision in this case is avoided by adding time delay.

III. Path Planning And Collision Avoidance for Clustered Central Place Foraging: PPCA-CCPFA

The PPCA-CCPFA approach is presented in Algorithm 1. For a robot swarm of size R, every robot that collects or deposits the target exchanges messages with its neighbors N within certain radius to determine possible collisions. Then, the robot computes a collision adjacency matrix for its neighbors indicating the values of *No Collision*, *Coincident Paths*, or *Intersecting Paths*. Based on the values of this matrix different actions are taken.

For the case of *Coincident Paths*, a waypoint at a certain angle and distance from the nest is added in order to add space between the paths of robots and avoid the collisions near the “nest”. The waypoint added forms a triangular path along which the robot travels and helps add safety margin between the incoming robots to the “nest” and outgoing robots from the “nest”. Assuming that the center of the nest is given by the origin the new waypoint coordinates can be calculated as $W(x_{wi}, y_{wi})$ given by:

$$x_{wi} = d\cos\theta_i \quad (1)$$

$$y_{wi} = d\sin\theta_i \quad (2)$$

Figure 2(a) displays the reactive collision avoidance approach for coincident paths in DDSA. Figure 2(b) shows the coincident case for PPCA-CCPFA. The robot travels a triangular path before returning back to its search position.

For the case of *Intersecting Paths* a time delay is added for the robot farthest from the predicted intersection point. The time interval Δt added to stop the robot farther from the collision point is given by:

$$\Delta t = \min(t_i, t_j) \quad (3)$$

where t_i and t_j represent the predicted time it takes for robot i and robot j to reach the intersection point respectively.

The base algorithm for Algorithm 1 is the DDSA described in [7]. The DDSA broadens a square spiral from one robot to any number of robots. The generated spiral path of the robot preserves the determinism and guarantees optimality [7]. The integration of PPCA-CCPFA with DDSA is shown in Algorithm 2. The blue pseudo-code indicates the integration of PPCA-CCPFA into the DDSA algorithm showing in black pseudo code.

Algorithm 1 PPCA-CCPFA

▷ Distributed across robots
for all Robots $i \leftarrow 0$ *to* R **do**
  if Path Planning and Collision Avoidance Activated **then**
    if Target Collected or Target Deposited **then**
      $FindNeighbors() \leftarrow N$
      $M \leftarrow ComputeCollisionMatrix()$;
      for all Robots $i \leftarrow 0$ *to* N **do**
        for all Robots $j \leftarrow (i+1)$ *to* $(N-1)$ **do**
          if $M_{i,j} == COINCIDENT$ **then**
            $SetWayPoint() \leftarrow W$
            $W \leftarrow CalculateWayPoint()$;
            if Robot going away from the nest **then**
              Go to waypoint W
          else if $M_{i,j} == INTERSECTING$ **then**
            if Robot is farther from Collision Point **then**
              $\Delta t \leftarrow CalculateStopTime()$;
              StopRobot();

Algorithm 2 Integration of PPCA-CCPFA with DDSA

▷ Distributed across robots
for all Robots $i \leftarrow 0$ *to* R **do**
  ▷ Create a spiral pattern to follow and store it
  for $c \leftarrow 0$ *to* N *Circuits* **do**
    Q.enqueue ($\langle 0,\ {}_gD_N(\ i, c, R)\ \rangle$)
    Q.enqueue ($\langle {}_gD_E(\ i, c, R)\ , 0 \rangle$)
    Q.enqueue ($\langle 0,\ -{}_gD_S(\ i, c, R)\ \rangle$)
    Q.enqueue ($\langle -{}_gD_W(\ i, c, R)\ , 0 \rangle$)
  ▷ Start at collection point and perform spiral
  while $\neg Q.empty()$ **do**
    if Check if First circuit completed **then**
      Activate PPCA-CCPFA
    $w \leftarrow s + Q.dequeue()$
    Move toward w
    if target found at current location s **then**
      if PPCA-CCPFA Activated **then**
        PPCA-CCPFA Check
      Return to collection point with target
      if at collection point **then**
        Deposit target
        if PPCA-CCPFA Activated **then**
          PPCA-CCPFA Check
        Return to location s

IV. Simulation Experiment Setup

The problem domain of the central place foraging task should work efficiently over different resource distributions in the workspace. Also the collisions around the “nest” are more as multiple robots collect resources from the cluster simultaneously. To evaluate the proposed PPCA-CCPFA approach, a cluster distribution with single cluster of size 8 x 8 is used. The performance of this approach is evaluated by measuring the target collection rate and average collision rate for multiple random locations of the cluster in the workspace.

The simulator used for this study is the ARGoS simulator [23]. ARGoS simulator is a multi physics robot simulator. The simulator provides high accuracy (close similarity to real environments), high flexibility (supports heterogeneous robots), and high efficiency (optimized computational resources to provide shortest simulation run time possible). ARGoS can simulate complex environments with a large heterogeneous robot swarm.

In the experiments, 64 resources in the form of a single 8 x 8 cluster are randomly placed in the square arena space of 100 m^2. All experiments are run for 30 minutes. The performance of DDSA [7] and the proposed approach are compared on 10 random locations of clusters in the arena. An update cycle of 480 per second for the 2D physics solver in the ARGoS simulator is used for the experiments. The robots simulated have parameters similar to the physical iAnt robots [7], [17]. The simulation setup is similar to the DDSA [7] so that it is easier to compare its performance with that of PPCA-CCPFA. To simulate the robot hardware, the robot has an 8 cm radius with a camera facing downward to detect the resources. The resources have a radius of 5 cm. The gap between the spirals is 13 cm [7]. The robot has a forward speed of 8 cm/s and a rotation rate of 10 cm/s approximately equal to 1.25 rad/s. The "nest" radius is 4 cm, and it is assumed that the "nest" is represented by a beacon [7], [17] . The robots move 8 cm/s towards their goal between reorientations [7], [24]. There are no static obstacles in the arena. The parameters for the PPCA-CCPFA approach: location of the cluster, the distance of waypoint and angle of the waypoint along with the environment and robot parameters mentioned above can be configured in the ARGoS simulator. The performance of each approach is evaluated at 10 different locations of the cluster for swarm size of 3 to 15 robots.

V. Simulation Experiment Results

The performance evaluation is measured using target collection rate and average collision rate for swarm sizes of 3 to 15 robots. Each experiment was performed for 16 combinations of waypoint distance taking values d from the set $D = \{0.2\ m, 0.3\ m, 0.5\ m, 0.6\ m\}$, and waypoint angle taking values θ from the set $\Theta = \{30°, 40°, 70°, 80°\}$. The consistent performing values of waypoint distance and waypoint angle are compared with the performance of DDSA. The combination values depend on the robot radius and a gap distance that need to be between consecutive robots.

The performance of PPCA-CCPFA is evaluated based on the distance d and angle θ of the waypoint. The best performing set is chosen by analyzing the graphs and selecting the most consistent performing set with respect to the DDSA. For each angle, the best performing distance is selected. Later

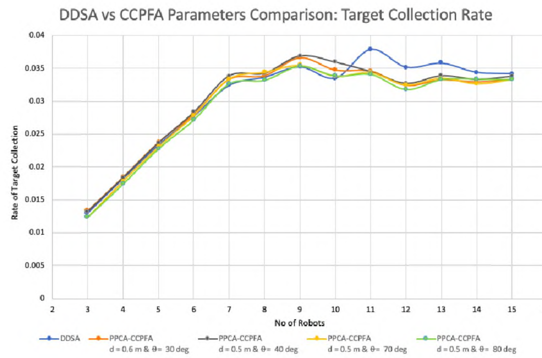

Fig. 3. PPCA-CCPFA vs DDSA: Consistent performing sets for Rate of Target Collection

the four sets of angle and distance are compared to simplify the consistent performing set. Two parameters considered for performance evaluation: rate of target collection and rate of average collision. The rate of target collection specifies the number of targets collected in 30 minutes for each swarm size. The rate of average collision specifies the average number of collisions encountered by all the robots in 30 minutes altogether for a swarm size.

The consistent performing sets are selected and compared to the DDSA. Figures 3 and 4 show the comparison of consistent performing sets of PPCA-CCPFA. The selected sets have better target collection rate as well as lower average collision rate than the DDSA for swarm size of 3 to 10. From the figures, it can be observed that PPCA-CCPFA algorithm with parameters waypoint distance $d = 0.5\ m$ and waypoint angle $\theta = 40°$ perform better than the DDSA for swarm size range of 3 to 10. The performance of PPCA-CCPFA algorithm however, decreases with the increase in the swarm size. Figure 4 shows the average collision rate for the swarm for DDSA and PPCA-CCPFA. The average reduction in average collision rate for the swarm size 3 to 15 is approximately 33% for the PPCA-CCPFA approach in comparison to the DDSA. Figure 3 shows the average target collection rate per robot for a particular swarm size for both the DDSA and PPCA-CCPFA. It is observed in foraging tasks that an increase in the swarm size reduces the performance of each robot in the swarm due to interference between the robots and competition for finite resources. It can be observed that the reduction of inter-robot collisions in the PPCA-CCPFA approach helps the per robot target collection rate by approximately 3% for swarm size 3 to 10.

VI. Conclusion and Future Work

The focus of the work was to pursue performance improvements in Central Place Foraging by addressing the congestion problem with path planning and collision avoidance methods for clustered resource distributions. This included adding spatial delay for coincident paths and time delay for intersecting paths for the robots collecting resources from a cluster. The robots having coincident paths travel a triangular path by going

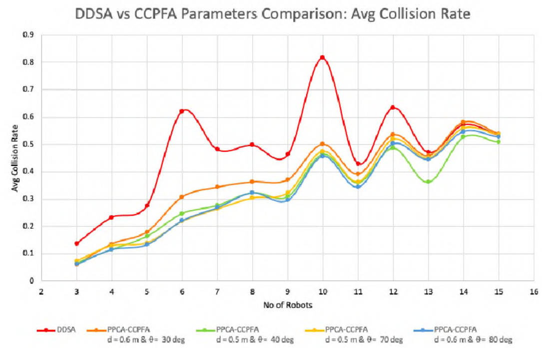

Fig. 4. PPCA-CCPFA vs DDSA: Consistent performing sets for Average Collision Rate

to a waypoint calculated based on the parameters distance and angle. Different combinations of waypoint distance and angle were used to observe the performance of the proposed approach. For robots having intersecting paths a time delay was calculated based on motion kinematics. The concentration was to avoid the congestion observed near the cluster and central depot location when multiple robots collect resources affecting overall performance of the central place foraging algorithm. The proposed approach then was compared to the popular DDSA for performance evaluation with a single 8 x 8 cluster, swarm sizes of 3 to 15 and ten random cluster locations. The results showed that there was a reduction of average number of collisions among the robots and increment of target collection rate for the swarm size from 3 to 10. However, the target collection rate decreased with the increase in the swarm size. It was also observed that the average number of collisions are reduced significantly and are lower than that observed for the DDSA even when the swarm size increases.

This work represents preliminary findings for using path planning techniques for central place foraging to improve multi-robot foraging performance. The proposed approach can be improved with integration of recruitment and techniques with more intelligent understanding of the environment and communication between the robots [25]. It is worth investigating the causes for performance degradation when swarm size increases above 10. Also, it is worth investigating the performance of the proposed approach on different resource distributions and exploiting the clusters using an optimal number of robots relative to cluster size.

References

[1] 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.

[2] A. Reina, T. Bose, V. Trianni, and J. A. Marshall, "Effects of spatiality on value-sensitive decisions made by robot swarms," in *Proceedings of 13th International Symposium on Distributed Autonomous Robotic Systems (DARS 2016).*, 2016.

[3] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics*. Springer, 2004, pp. 10–20.

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

[5] 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.

[6] S. Burlington and G. Dudek, "Spiral Search As An Efficient Mobile Robotic Search Technique," Tech. Rep., 1999. [Online]. Available: https://www.cim.mcgill.ca/ mrl/pubs/scottyb/burl-aaai99.pdf

[7] 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.

[8] E. Langetepe, "On the optimality of spiral search," in *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2010, pp. 1–12.

[9] A. Ryan and J. K. Hedrick, "A mode-switching path planner for uav-assisted search and rescue," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 1471–1476.

[10] R. Baeza-Yates and R. Schott, "Parallel searching in the plane," *Comput. Geom. Theory Appl.*, vol. 5, no. 3, pp. 143–154, Oct. 1995.

[11] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, Oct 2001, pp. 1073–1078 vol.2.

[12] O. Feinerman, A. Korman, Z. Lotker, and J.-S. Sereni, "Collaborative search on the plane without communication," in *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing*, ser. PODC '12. New York, NY, USA: ACM, 2012, pp. 77–86.

[13] M. Müller and R. Wehner, "The hidden spiral: systematic search and path integration in desert ants, cataglyphis fortis," *Journal of Comparative Physiology A*, vol. 175, no. 5, pp. 525–530, Nov 1994.

[14] Z. B. Hao, N. Sang, and H. Lei, "Cooperative coverage by multiple robots with contact sensors," in *2008 IEEE Conference on Robotics, Automation and Mechatronics*, Sep. 2008, pp. 543–548.

[15] H. Skubch, *Evaluation*. Wiesbaden: Springer Fachmedien Wiesbaden, 2013, pp. 195–233.

[16] A. López-Ortiz and D. Maftuleac, "Optimal distributed searching in the plane withand without uncertainty," in *WALCOM: Algorithms and Computation*, M. Kaykobad and R. Petreschi, Eds. Cham: Springer International Publishing, 2016, pp. 68–79.

[17] 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.

[18] 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.

[19] J. Guerrero, G. Oliver, and O. Valero, "Multi-robot coalitions formation with deadlines: Complexity analysis and solutions," *PLOS ONE*, vol. 12, no. 1, pp. 1–26, Jan. 2017.

[20] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, Sep 2002.

[21] S. Bhattacharya and R. Agrawal, "Development of robot swarm algorithms on an extensible framework," in *SoutheastCon 2017*, March 2017, pp. 1–6.

[22] P. Monaghan and N. B. Metcalfe, "Group foraging in wild brown hares: effects of resource distribution and social status," *Animal Behaviour*, vol. 33, no. 3, pp. 993 – 999, 1985.

[23] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "Argos: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, Dec 2012.

[24] BCLab-UNM, "DDSA-ARGoS github code repository," https://github.com/BCLab-UNM/DDSA-ARGoS, 2016. [Online]. Available: https://github.com/BCLab-UNM/DDSA-ARGoS

[25] 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.