Implementing the Distributed Deterministic Spiral Search Algorithm in the Gazebo Robot Simulator

T. Ogunyale*, M. Moses†, M. Fricke†, A.Griego†, J. Jones†
*Georgia State University: Atlanta, Georgia
†University of New Mexico: Albuquerque, New Mexico

***Index Terms*—DDSA, CPFA, AprilTag Signatures, Gazebo, ARGos**

I. Introduction

Swarm robotics allows for the extraction of various simple behaviors that are observed in ecological systems to be used to solve complex task [1]. One task that lends it self nicely to the solutions that swarm robotics provides, is the surface exploration and harvesting of planetary resources. With the current state of planetary exploration making human exploration infeasible, sending small, low-cost and low-maintenance rovers to traverse the planet is currently the best option[2].The harvesting of planetary resources to a single collection point or center location is viewed as a central foraging problem[3].

The Distributed Deterministic Spiral Search Algorithm (DDSA) is a search implementation that creates a singular spiral that can be traversed by a number of rovers. Each spiral that is traversed by a rover is interlocking with the others. This allows for complete coverage of the spiral. It also ensures the closet targets will be collected first.

We implemented the DDSA using the Gazebo simulator. We observed the efficiency of rovers collection of targets. The physical robots that were used in the simulator were known as Swarmies. We compared the results the results to the central-placing foraging algorithm(CPFA). Our goal is to extend the works of using DDSA as baseline comparison to the CPFA.

II. Related Works

The CPFA is an ant inspired algorithm. While searching their environments, rovers, by some probability, will place waypoints at locations with high concentration of targets. Rovers are influenced to search these waypoints in order to search locations of high concentrations. Otherwise, rovers will perform a random search.Parameters for the CPFA, such as the probability to set waypoints, are optimized using a generic algorithm [4], [5].

III. Methods

A. Robot Simulation

Our Implementation of DDSA was simulated using the robot simulator Gazebo. With Gazebo being a 3D simulator, rovers in simulation were able to be exact replicas of their physical counterparts.The dimensions of each rover was 36cm wide and 45cm long and the detection range for the rovers was about 6cm in front of them.The only slight differences between physical and simulation had to do with there noise sensitivity in their respective sensors. The resources that the rovers were tasked with collecting are in the shape of small cubes that contain AprilTag signatures on all sides. The AprilTag signatures on the cubes make them detectable by the rovers on-board camera.

In order to incorporate the DDSA into the simulation, each rover was aware of the size of the swarm, its index within the swarm, how far into its respective spiral, and its range of detection[1]. When an AprilTag cube was detected by a rover the rover would internally place a checkpoint at the exact location that the rover successfully picked up the AprilTag cube. After dropping off the cube, the rover would then return back to a modified version of the previously set checkpoint location. The checkpoint would be modified to A) be a translation of the current checkpoint to the current path in the spiral B) and be 1.0 meter back from the current checkpoint. If on returning to the checkpoint more cubes were observed, the rover would repeat the process, else the rover would continue on with its spiral.

The Obstacle avoidance method was simple. If the rover came in contact with an obstacle, it would take a slight turn left and would drive 0.5m forward then continue on with its target location. The only Obstacles that were possibly present during the runs were, other rovers, the collection plate, and the walls that set the boundary.

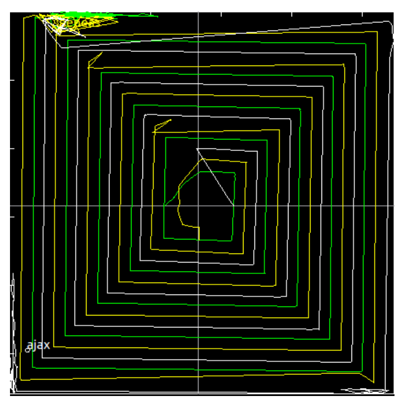


Fig. 1. Representation of three rovers implementing one interlocking spiral, each different color spiral in the graph represent an individual rover.

B. Experimental Setup

The experimental runs consisted of three rovers that were initiated around a collection point that had dimensions that were 1m by 1m. 256 cubes were placed in a 7.5m by 7.5m environment. The placement of the cubes around the environment were placed in a Power Law distribution. AprilTag cubes were only counted as collected if they were located within the collection plate at the end of each run. Each run lasted 30 min, in simulation time. All figures are accounting for 15 experimental runs. If during a run a rover died, we restarted the entire simulation, unless the rover died after the 25min mark.

IV. RESULTS

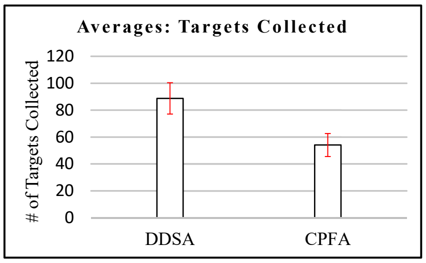
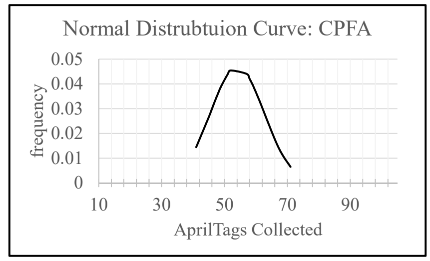


Fig. 2. The average AprilTags collected for both the CPFA and the DDSA. The graph also depicts the standard deviation for each search algorithm.



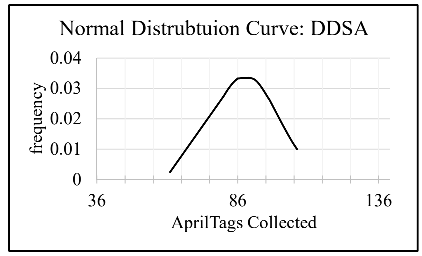


Fig. 3. The Normal Distribution curve for the CPFA and the DDSA

The results show that, on average, DDSA out performs CPFA (figure 2). DDSA also had a slightly higher standard deviation (σ) equaling 11.659 compared to CPFA's σ only equating to 8.531. Both the DDSA and CPFA benefit from the trait of returning to to large clusters, but DDSA's implementation will always send it back towards the location of which it picked up an target, while CPFA's implementation allows it the option to search a new location after dropping off a target instead of returning to its previous pickup location.

The distribution for both DDSA and CPFA both fell under a nearly mesokurtic distribution. We calculated the kurtois by using the sample excess kurtosis formula, and we got a kurtosis of 0.092 and -0.706 for DDSA and CPFA respectfully. In terms of the symmetry of each of distribution, both search algorithms were nearly symmetrical. DDSA had a slightly negative skewness, while CPFA had a slightly positive skewness.

V. DISCUSSION

Our results showed that DDSA out performed CPFA, which is similar to the results found by Fricke. Both of our results saw DDSA outperform CPFA, but we saw a greater margin between the performances in CPFA and DDSA. We believe that since Gazebo took in to account more factors than the ArGos simulator, such as the rovers sensitivity to noise, the complexities of obstacle avoidance and the parameters dealing with a physical robot, it affected how efficiently the CPFA could work be implemented in a 3D environment.

Our goal going forward would to continue to debug our code for and bugs that come up when the simulation is running, also when that is complete, to run more experimental runs to gather more data. Our results accounted for only 15 runs per each search algorithm, so having more data will paint a bigger picture for both the DDSA and CPFA.

Our biggest goal would to test and compare both methods on the live physical rovers in a real world environment. With the rovers seen in simulation being drastically different from there physical counterpart, it would interesting to see how each algorithm fairs in the real world.

VI. CONCLUSION

With DDSA's ability to take advantage of its deterministic characteristics of gathering the nearest targets first and nearly collecting all targets that lay within the coverage of the spiral, it was able to outperform the CPFA search method[1]. Gazebo's role in this truth, stemmed from how far more complex it was from ArGos. It will be interesting to see the comparison of these two algorithms when the bugs and tunning are fixed and the algorithms get tested on physical robots.

ACKNOWLEDGMENT

Acknowledgment goes to Moses Biological Computation Laboratory, at the University of New Mexico. My Thanks goes to the head of the lab, Dr. Melanie E. Moses, as well as Dr. G. Matthew Fricke, Antonio D. Griego and Jarret Jones. I also Thank DREU and CRA for funding my research.

REFERENCES

1 Fricke, G., Hecker, J., Griego, A., Tran, L., and Moses, M., "A distributed deterministic spiral search algorithm for swarms." IEEE, Oct. 2016.

2 Elfes, A., Dolan, J. M., Podnar, G., Mau, S., and Bergerman, M., “Safe and efficient robotic space exploration with tele-supervised autonomous robots.” in *AAAI spring symposium: to boldly go where no human-robot team has gone before*, 2006, pp. 104–113.

3 Wineld, A. F. T., “Foraging robots,” in *Encyclopedia of complexity and systems science*, 2009, pp. 3682–3700.

4 Hecker, J. P. and Moses, M. E., “Beyond pheromones: evolving errortolerant, exible, and scalable ant-inspired robot swarms,” in *Swarm Intelligence*, no. 1, 2015, pp. 43–70.

5 J. P. Hecker, J. C. C. and Moses, M. E., “Exploiting clusters for complete resource collection in biologically-inspired robot swarms,” 2015.