Swarmathon Technical Report
California State University Channel Islands

Alexandra Collette*
Department of Computer Science
California State University Channel Islands
Camarillo, California 93012
*Complete CSUCI team lisited in the Appendix

***Abstract*—This is a study of the use of physical swarm robots to collect and return resources to a centralized collection region. Multiple algorithms are presented that include deterministic and stochastic search strategies. The goal of these algorithms is to effectively be applied in complex real world environments such as the exploration of the surface of Mars.** *(Abstract)*

I. Introduction

The purpose of this project is to develop and implement an algorithm that allows "swarmie" rovers to autonomously search, collect and return resources to a centralized collection center. This is to simulate a swarm of rovers on Mars collecting resources such as ice and other useful materials to be used by astronauts who will land on Mars in future missions. The rovers being autonomous allows for collection without the delay of a person having to physically control them. With multiple rovers working as a swarm, more area can be covered and in turn, more resources collected.

In order to find the most time efficient manner to collect the resources we developed multiple algorithms using a mixture of deterministic and stochastic search strategies. To accurately execute the algorithms, we implemented a new waypoint navigation system, home location calibration, wall orientation calibration, and obstacle avoidance.

There was the extra challenge of ensuring the algorithms and the changes made would accurately translate outside of the Gazebo simulation environment, and perform on the physical "swarmie" rovers. Waypoint navigation is heavily dependent on the accuracy of the GPS and IMU units working together. Ensuring the calibration of these units is essential.

The paper is organized as follows: Section II reviews related work that we based some of our search algorithms on. Section III describes the different methods that we developed to reach our final result. Section IV goes over how we structured our experiments to test our various methods and hypotheses and Section V reviews the effectiveness of our most efficient algorithm.

II. Related Work

Our deterministic searches were based on M. Schneider-Fontan and M. Mataric [4] and G. Fricke *et al* [1]. The main reasons behind each of the papers are to be able to avoid interference between robots and to cover the entire search area. According to Schneider-Fontan and Mataric interference by other robots with each other negatively affects the efficiency of resource collection. The way they tried to solve the problem, was by dividing the search field by search area [4]. In our research we tried multiple partitioning schemes of our search area. The issue that they ran into was a scalability issue, the more rovers that they included the less effective the algorithm [4].

The research done by G. Fricke *et al* was focused on fully searching the field using a lawn mower spiral search pattern. They found that the spiral pattern allows the robots to optimally collect all resource within the spiral [1]. This led us to developing different lawnmower search patterns within our partitioned search field. The issue that G. Fricke et al's research and our research had in common was the noise that occurs on the physical tests that make traversing the pattern inaccurate.

Stochastic search is less efficient searching the entire field and avoiding interference between robots, but more flexible when it environmental noise is taken into account [1]. Our stochastic search algorithms are based on research by J. Hecker and M. Moses's [3] and W. Moon *et al* [2]. Both papers use different methods to perform a stochastic search but share the ideas of communication between robots reporting location of resources and obstacles. W. Moon *et al* uses marked pheromone nodes when a robot locates a characteristic landmark in the environment and communicates it to the other robots. We use this concept, of communicating with other rovers, the points that have already been searched and obstacles. We do not communicate resource clusters due to the size of the field and the quantity of rovers that we have to utilize. We do not want to cluster the rovers in the same area, limiting the search of other areas. The algorithm itself is based more on the work of J. Hecker and M. Moses. They use an algorithm the intelligently select a new waypoint to simulate the search pattern of ants, using information based on what resources have been found by the robots to influence the search [2]. We use a similar method but include calculations to account for paths that have already been searched and obstacles.

For each search type, the efficiency of the algorithms will vary based on the resource distribution and swarm size [2]. This makes both methods important to understand and test.

III. Methods

A. N-Partitioned Deterministic Search

This deterministic algorithm was based on a partitioned deterministic search with the intention of minimizing potential collisions between rovers. This was achieved by assigning discrete and equal portions of the map to individual rovers. The number of partitions was based on the number of rovers on the field so that each rover would be allocated an equal portion of the field. The goal was to design an algorithm that could easily scale for an arbitrary number of rovers. The preliminary testing and algorithms were based on 3 rovers.

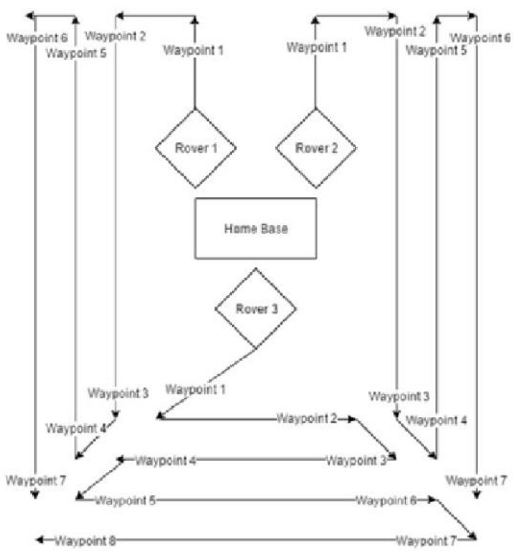


Fig. 1. Search paths for each of the rovers.

We had two primary issues with this algorithm. The first one was developing an algorithm that would accurately allocate the number of partitions to an arbitrary number of rovers. In the implementation we used three static loops to assign the partitions to three rovers, but were unable to develop an algorithm in time to do so for an arbitrary number of rover and partitions. The second issue was the algorithm was not efficient when blocks were in a clustered layout, it is possible for some rovers to receive partitions with a minimal number of cubes. A re-partitioning scheme was planned to be used when a rover had completed a partition, it would then repartition the field based on the remaining unsearched area.

B. Relay Team Hybrid Search

This deterministic algorithm was designed to allow complete coverage of the field while preventing the collision of the rovers during search and drop-off.

The field is divided into three partitions. The leftmost and rightmost sections will be searched using a lawnmower algorithm using calculations based on the number of rovers being used. The search waypoints are calculated based off a starting waypoint, which is used to calculate the remaining waypoints. In the case of three rovers, two will take on the role as search rovers, one rover will be assigned to the right most partition, one to the left most. The third rover will be considered the drop off rover that will be assigned the center drop-off zone partition.

The center section acts as a drop off point for the search rovers, while the pickup rovers will retrieve the resource and return it to the center location. The drop-off rover(s) will perform a random search within the center section, collecting resources located there until a search rover communicates a waypoint where a resource has been dropped off. The waypoint is added to a pickup queue for the drop-off rover to be collected. If a search rover completes its section, the drop off zone is then repartitioned and the search rover takes on the role of a drop-off rover.

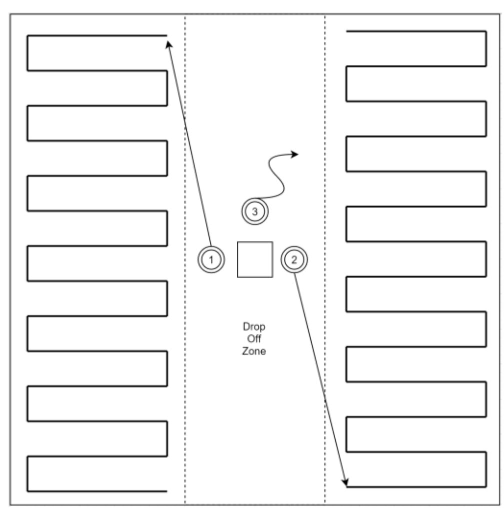


Fig. 2. Search path and partitions for three rovers

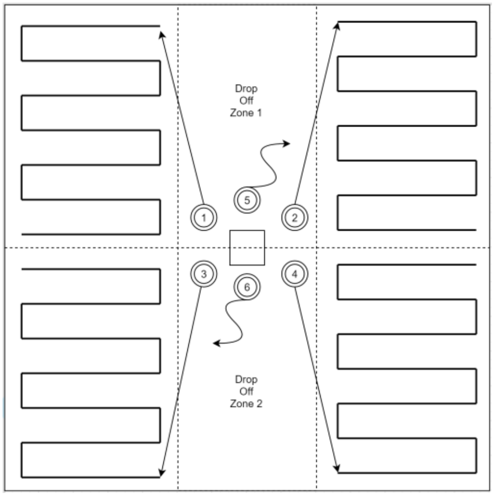


Fig. 3. Search path and partitions for six rovers

We ran into several issues with this algorithm in practice. The first issue was the time it takes to pick up a resource. Having to pickup and drop-off a resource twice, first in search section then in drop-off zone, would cost time. The next issue is where the center drop-off rover would have to collect a large amount of resources in that section, along with what was dropped off by the search rovers. Even with the search rovers eventually becoming drop-off rovers, we return to the first issue of the time efficiency of pickup.

C. Stochastic Anthill Search

This Stochastic search tracks locations that had been previously visited by rovers and tend away from them. The goal of this is a randomized search algorithm without overlapping search paths, allowing the field to be searched quicker.

This search algorithm is a weighted stochastic search designed to expand the boundary of points visited by the rovers. Each new waypoint is calculated using three vectors. The first vector is defined by the sine and cosine of the current heading with a randomized offset uniformly distributed 15° about the x axis and 15° about the y axis. The second vector is defined by the normalized gradient of a virtual potential field defined by a local maxima of 0.6096 meters decreasing at a rate of $1/(x^2 + y^2 + 1)$ at each point within 1.5meters, where x and y are the signed difference between the rover's coordinates and the point in each dimension. This vector is responsible for allowing the rovers to tend away from locations that have already been searched. The third vector is defined by the normalized coordinate vector of the rover scaled by the square root of the number of points within 1.5 meters, divided by the distance of the rovers in relation to the center of the field. The three vectors are summed together, normalized and scaled by 0.6096 meters then added to the rover's current position to generate a new search point 0.6096 meters away.

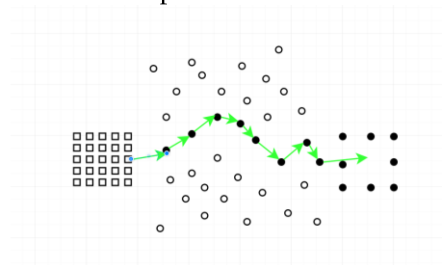


Fig. 4. Stochastic search

The rovers share information over the one-to-many communication system in order to coordinate their localization routines, returns to home and the collection of cube clusters. This also allows the rovers to know if an obstacle is another rover or an environmental obstacle.

In the case of an obstacle being detected, avoidance routines are executed. In this case the rover will broadcast the coordinates it is using as parameters to the search so that the other rovers will exclude those points from their own searches.

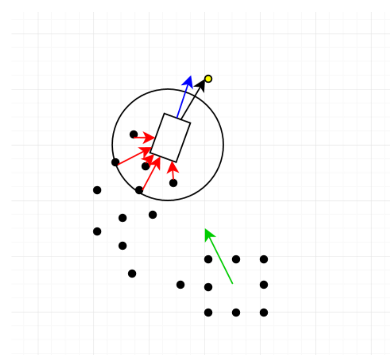


Fig. 5. Avoidance routines

The primary issues have been regarding the time it takes to cover the entire field and the amount of overlap in search points. While they tend to avoid previously visited search points, but they do not avoid them completely. Additionally, due to the inaccuracy of the localization of the physical application, the points broadcasted by each rover may be off by as much as 2 times the error of the GPS.

D. Switching Mode Stochastic Search

For the final submission we decided to fall back to a simplified stochastic algorithm that we will refer to as Switching Mode Stochastic Search (SMSS). The primary motivation for selecting this algorithm was our lack of ability to accurately drive to waypoints and our lack of knowledge about the orientation of the field. Without those two items any strategy that relies on partitioning the search space is vulnerable if not impossible to implement on the physical robots.

The SMSS algorithm is incredibly simple to implement and therefore is a trusted option to overcome some of the uncertainties associated with the physical competition. The premise of SMSS is to randomly drive away from the goal region when the robot is near it, and randomly drive toward the goal region when the robot is far from it. To avoid the chattering that can occur with this type of algorithm we implemented a simple hysteresis one the switching mode controller.

If the robot is in Zone 1, a new waypoint is generated in a random direction that is in the half-plane facing away from the origin that is a uniform random distance from the current location of between 0 and 1.5 meters. If the robot is in Zone 3, a new waypoint is generated in a random direction that is in the half-plane facing toward the origin that is a uniform random distance from the current location of between 0 and 1.5 meters. If the robot is in Zone 2, the direction of the new heading will depend on if the robot had previously been in Zone 1 (away) or Zone 3 (towards). The width of Zone 2 (Outer Radius – Inner Radius) effectively forms the hysteresis region, thus preventing the rover from chattering about a single switching point.

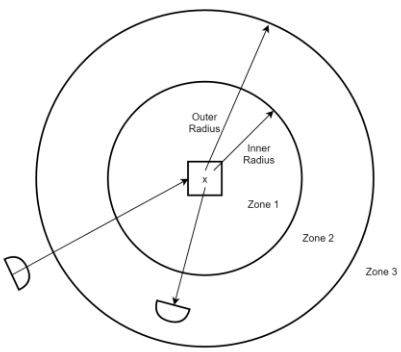


Fig.6 Zoning implementation

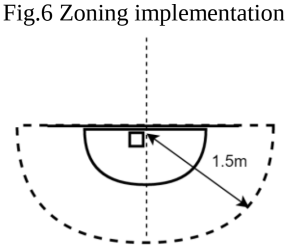


Fig.7 Waypoint selection range

This algorithm did not rely on communication between robots although it could possibly be improved with more time by allowing robots to communicate their locations when dropping off a block or when they have encountered a cluster of blocks. This algorithm did however remember the location of the last place that it picked up a target and returned to this location upon dropping off the block. This was observed to be effective when blocks were clustered although with the physical robots it was often difficult to for the rovers to drive to exact locations.

IV. Experiments

A. *Virtual Simulation Experiments*

Alexander Collin and Crystian Marron running tests on their deterministic algorithm using the Gazebo virtual simulation

While in the beginning stages of development we used the Gazebo simulator, provided by the University of New Mexico, to test the code within a controlled laboratory setting. This allowed us to test the core code for the algorithms to be fine-tuned, provide proof of concept and compare efficiency of different methods. It was very important for us to be able to test the algorithms without having to do a full physical set up to let us test faster and without the noise that the physical rover would experience.

To test the algorithm, simulations were run starting with one rover and no blocks, where we tracked the rover's path to ensure accuracy. ROS topics was used to track the waypoints and states as the simulation executed the tests. As the algorithms evolved, more rovers were added to test the full search scope. Once the rovers were performing the designed search pattern and would not collide, blocks were then added to do a full test.

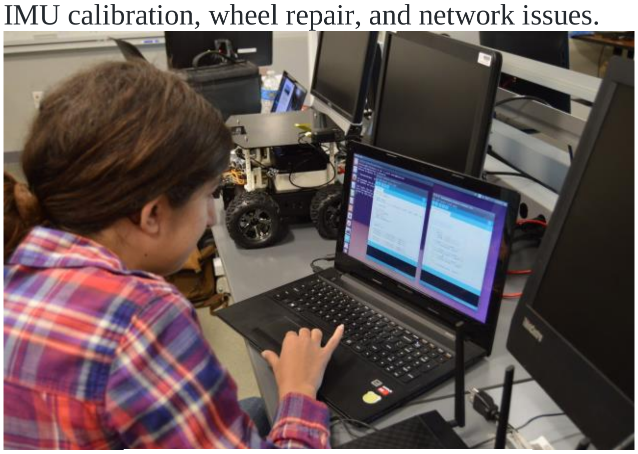
The virtual environment was a very useful tool to build and test our algorithms, but it was necessary to test for outside noise and the accuracy of the GPS and IMU units in a physical test environment.

B. Physical Rover Experiments

Running physical tests

The physical tests were run using the “swarmie” rovers that we build from the kits that NASA provided in order to maintain consistency for the competition.

We had a team of students who were responsible for loading and testing code on the physical rovers. They were also responsible for troubleshooting and maintaining the rovers. This involved tasks such as IMU calibration, wheel repair, and network issues.



Nicole Dubin troubleshooting the Arduino

Like the virtual tests we began with one rover. We then gave the rover specific waypoints, that we had measured and marked on the ground to test the accuracy of the GPS and the IMU to see if it is able to reach the given waypoint. This allowed us to test the drift of the IMU unit and how to better account for it. During these tests we could observe how well the rovers were able to navigate to an assigned waypoint, collect a resource and return it to the collection center. These actions are essential to executing all of the methods that were implemented and test in the virtual simulation. What was functional and accurate in the virtual simulation, did not fully translate to the virtual simulation, did not fully translate to the physical rovers and adjustments were needed.

V. Results

With our methods and testing we found that the efficiency of the algorithms was strongly based on the distribution patterns of the resources. The deterministic search algorithms performed better with a uniform distribution, while the stochastic search algorithm performed better with a clustered distribution. To try and balance the testing we used the power law distribution pattern.

The deterministic search performed better, collecting up to 60 resources in 30 minutes using 3 rovers. The stochastic search was able to collect an average of 40 resources in 30 minute using 3 rovers. Both tests were completed using the virtual simulation environment to prevent outside noise and acquire a baseline statistic to compare to physical testing.

Some of the results may be affected by the efficiency of other operations of the code base such as pickup, drop-off and obstacle avoidance.

VI. Conclusion

Both deterministic and the stochastic search algorithms can be applied to the task of collecting and returning resources in real world applications. Our algorithms do not take into account other obstacles outside of walls or other rovers that they may encounter.

The use of autonomous swarm robotics is the most efficient way to complete the task of collecting resources on Mars due to the number of rover that can be completing the task and the lack of human intervention needed.

References

[1] G. Frick, J. Hecker, A. Griego, L. Tran, and M. Moses, A Distributed Deterministic Spiral Search Algorithm for Swarms. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016

[2] J. Hecker and M. Moses, Beyond Pheromones: evolving error-tolerant, flexible and scalable ant-inspired robot swarms. *Swarm Intell.* 2015

[3] W. Moon *et al.* Virtual pheromone map building and utilization method for a multi-purpose swarm robot system. *International Journal of Control, Automation and Systems 13(6)*, pp. 1446-1453. 2015.

[4] M. Schneider-Fontan and M. Mataric, Territorial multi-robot task division, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1998.

Appendex: Team Members

Leads		
Name	**Email**	**Title**
Alexandra Collette	alexandra.collette133@myci.csuci.edu	Team Lead
Maria Contreras	maria.contreras050@myci.csuci.edu	Outreach Lead
Nicole Dubin	nicole.dubin736@myci.csuci.edu	Technical Lead

Team Members		
Name	**Email**	**Teams**
Claudia Alamillo	claudia.alamillo718@myci.csuci.edu	1, 6, 11
Matthew Ambriz	matthew.ambriz209@myci.csuci.edu	1, 8, 11
Robert Aroutiounian	robert.aroutiounian300@myci.csuci.edu	2, 7, 8
Mathew Atcheson	mathew.atcheson193@myci.csuci.edu	1, 5
Heather Bradfield	heather.bradfield823@myci.csuci.edu	2, 10
Alexandra Collette	alexandra.collette133@myci.csuci.edu	1, 5, 8, 11
Alexander Collins	alexander.collins492@myci.csuci.edu	1, 4, 11
Joseph Contreras	joseph.contreras783@myci.csuci.edu	1, 6, 11
Maria Contreras	maria.contreras050@myci.csuci.edu	1, 6, 11
Steven Coronel	steven.coronel341@myci.csuci.edu	2, 10
Nicole Dubin	nicole.dubin736@myci.csuci.edu	1, 7, 11
Kelsey Geiger	kiel.geiger895@myci.csuci.edu	2, 9
Gevork Gevoian	jeffrey.grammer492@myci.csuci.edu	2, 10
Dylan Hart	dylan.hart758@myci.csuci.edu	2, 10
Timothy Indrieri	timothy.indrieri639@myci.csuci.edu	1, 3
Crystian Marron	crystian.marron173@myci.csuci.edu	1, 5
James Morris	james.morris398@myci.csuci.edu	1, 10
Jeremiah Paltridge	jeremiah.paltridge234@myci.csuci.edu	2, 10
Jessica Perez	jessica.perez@myci.csuci.edu	2, 9
Luis Torres	luis.torres801@myci.csuci.edu	1, 3
Michael Wisener	michael.wisener947@myci.csuci.edu	1, 3
Thomas Yamasaki	thomas.yamasaki509@myci.csuci.edu	2, 10

Mentors		
Name	**Email**	**Title**
Nicholas Stern	nicholas.dolan-stern630@myci.csuci.edu	Graduate student assistant
Jason Isaacs	jason.isaacs@csuci.edu	Professor
Kevin Scrivnor	kevin.scrivnor988@csuci.edu	Co-Instructor

During the development of the algorithms we broke into multiple team projects in order to divide the work and create multiple methods to approach the problem. Members were each involved on multiple projects. The projects were broken down as follows:

(1) Deterministic
(2) Stochastic
(3) Waypoint control
(4) Drop-off
(5) Pickup
(6) Order assignment
(7) Calibrate home
(8) Determine orientation of the field
(9) Communication
(10) Obstacle avoidance
(11) Rover operation and maintenance

Every week each team was required to present their current progress and discuss potential achievements, issues and solutions with the class. Additionally, each member was involved in the outreach program that was completed with the high school.

The final team implementation was a compilation of the most effective methods performed by the teams.