Design of a Swarm Search Algorithm: DustySWARM Reverse-Twister Code for NASA Swarmathon

Roger Hernandez, Raquel Yanez, Jorge Gonzalez Jr., Hector Moreno, Valeria Escobar, Dr. Tariq Tashtoush
Texas A&M International University, School of Engineering

***Abstract*—The development of search algorithms is essential to optimize space exploration. The use of swarms search robots can optimize the exploration techniques. The DustySWARM team at Texas A&M International University developed a reverse twister code search algorithm which can be used with a swarm of autonomous robots. The final version of the Twister code indicated an increase in the volume of resources collected by the swarm of robots.**

Keywords: *Swarm Robotics; Searching Algorithm; Autonomous Robot Swarm*

I. Introduction

In order to revolutionize space exploration techniques the Texas A&M International University DustySwarm team took the challenge to improve current search algorithms. The focus of the research was to create an efficient search algorithm applicable to future NASA space exploration missions. The use of autonoumous search rovers which can react to their environment facilitates to exploration of unknown territories. Furthermore, swarms of autonmous robots reduce the data collecting time frame.

In nature, swarms of animals and insects have developed systems to collect resources. Within the groups, the ability to achieve their goal of resource collection is obtained by systems of comunication and reactions to the environment. Similarly, robot swarms can collabortate to explore an unknown terrain. Previous search algorithms have been developed, such as the iAnt code developed by the University of New Mexico, which implements the use of sensor feedback and programmed decision logic to search for and collect resources.

The Reverse-Twister code was developed by DustySWARM team compete in the first NASA Swarmathon. The R-T code created for the competition implements a modified Archemidean Spiral. The purpose of the competion is to expand student knowledge in robotics and improve technolgy in space exploration missions. The results of the project indicated the possibility of more efficient space explorations in the future.

II. Related Work

A. Multi-Robot Search Algorithm

The purpose of using multiple search robots is to efficiently locate and collect any material found in a way that will optimize the ground covered. Most articles use algorithms to effectively and efficiently disperse their multiple search robots. The testing of algorithms or any other type of search related coding is done through a simulation software in which a known environment is home to the resources a robot is seeking. The robots that are programmed to follow a specific type of algorithm tend to follow a random direction path while systematically covering the complete area of the known environment. These multiple robots have also been programmedto detect any obstacles in their paths and avoid collisions, making the probability of finding an object much higher.

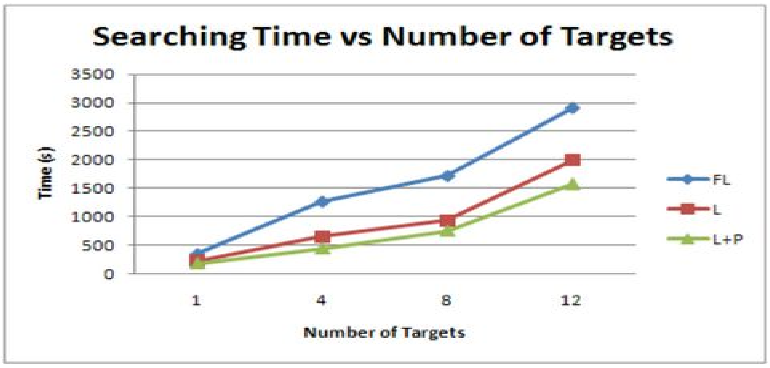

Figure 1.[1]

Figure 1, shows the number of targets found compare to the time taken by three different algorithm approaches, which are the levy random-walk and potential field (green), levy random walk (red), and the fixed-length random (blue). The levy algorithm generates the length of movement, while the artifical potential field improves the dispersion of the robots by generating a repulsion force.[1] The table above shows that the mixture of levy random-walk and the potential field gather the most targets in the least amount of time, meaning that the algorithm used to determine the length of movement and the dispersion amongst the robots results in finding the targets much quicker.

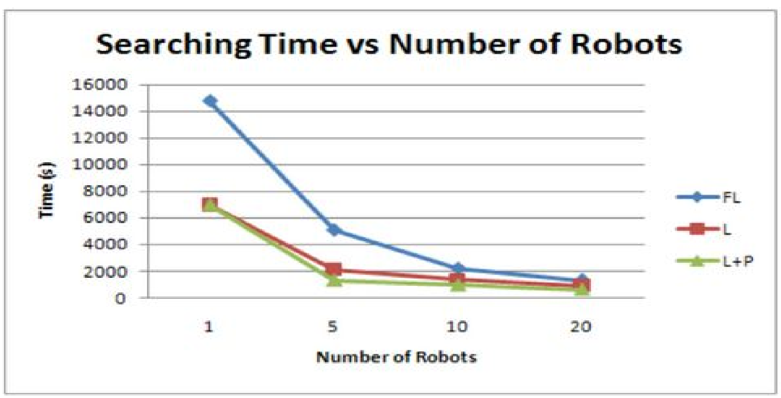

Figure 2.[1]

Figure 2, displays the comparison between the three algorithms to show how the number of robots used will affect the time to seek and find a set number of targets. The results show that the levy random-walk and potential field program will take the least time, but the other algorithms show they are really close. The reason for this is that the potential field can have a negative effect when many robots are constantly changing paths, since they are programmed to repulse against each other. DustySwarm has used the concept of both Levy random-walk and potential field, while traveling in a reversed twister form to cover all necessary ground without missing any objects near the base location. The team has also programmed the robots to return and search nearby where an object has last been found. This is effective in scenarios where the objects are either clustered or semi-clustered.

III. Methods

In order to incorporte the most adequete approach the team brainstormed on different search patterns that would allow the robots to cover more area within the allotted time constraint. The objective of the competition is to collect the maximum amount of targets in a given area and return them to the base location. As knowledge and familiarity with the Robot Operating System and the programming techniques increased, the code was modified to optimise target collection.

A. Initial designs

prior to programming

B. Spiral Equation

The archemidian Spiral

C. Modifications

thoughout the durtation of the

D. Final Design

The reverse twister code

IV. Experiments

In order to verify the improvement of the search algorithm, the DustySWARM team observed the reaction of the rovers to the iant code in progress by collection data after each simulation. First the team, focused the reaction of the rovers to the code. The team also collected

A. Initial Code Collection

The code developed

B. Progress Tables

the

TABLE I. Table Type Styles

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

a. Sample of a Table footnote. (Table footnote)

Figure 1. Example of a figure caption. *(figure caption)*

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

Results

iant		
Round Type	**Target Distribution**	**Targets collected**
Preliminary	Clustered	15
Preliminary	Power Law	100
Preliminary	Uniform	124

When the program was first run, without any modifications in the coding, it was shows that within the Preliminary test, the swarm robots collected 15 targets for clustered distribution, 100 targets for power law distribution, and 124 targets for unifrom distribution.

As shown in the first five tables below, when our team began trying different modifications and tests were run the values we were getting for power law and clustered distribution were very low compared to the original code. Therefore because of the low numbers no more trails were made with that coding.

Twister Code Velocity = 0.35		
Round Type	**Target Distribution**	**Targets collected**
Preliminary	Power Law	20

Velocity = 0.35, updated target detected handler		
Preliminary	Power Law	87

Twister Code Return		
Preliminary	Clustered	60
Preliminary	Power Law	76
Returns to Previous Mobility Count		
Preliminary	Clustered	45

Second Iteration		
Preliminary	Clustered	63

twister code return		
Preliminary	Power Law	80

Updated Twister Scaling Factor .25, v=.9		
Preliminary	Clustered	73
Preliminary	Power Law	103
Preliminary	Uniform	114

Updated Twister Scaling Factor rng .5-1.5m , v=.8 rot=.3			
Preliminary	Clustered	70	68
Preliminary	Power Law	121	
Preliminary	Uniform	146	

Updated Twister Scaling Factor rng .5-1.5m , v=.8 rot=.3			
Preliminary	Clustered	70	68
Preliminary	Power Law	121	
Preliminary	Uniform	146	

Updated Twister Scaling Factor rng .5-1.5m , v=.85 rot=.3		
Preliminary	Clustered	80
Preliminary	Power Law	101
Preliminary	Uniform	130

Updated Twister Scaling Factor rng .5-1.5m , v=1 rot=.3		
Preliminary	Clustered	62
Preliminary	Power Law	104
Preliminary	Uniform	138

Updated Twister Scaling Factor rng .5-1.5m , v=.75		
Preliminary	Clustered	77
Preliminary	Power Law	102
Preliminary	Uniform	122
Final	Uniform	247

Updated Twister Scaling Factor rng .5-1.5m , v=1.5 rot= .2			
Preliminary	Clustered	69	73

Return Logic		
Preliminary	Clustered	33
Preliminary	Power Law	95
Preliminary	Uniform	110

Return Logic v=1.0		
Preliminary	Clustered	58
Preliminary	Power Law	111
Preliminary	Uniform	118

Return Logic v=2.0		
Preliminary	Clustered	
Preliminary	Power Law	90
Preliminary	Uniform	

Roger		
Preliminary	Clustered	57
Preliminary	Power Law	71
Preliminary	Uniform	83

Jorge , v=.8 sf=1 rot=.2		
Preliminary	Clustered	58
Preliminary	Power Law	113
Preliminary	Uniform	122

Updated Twister Scaling Factor rng .2-.75m , v=.8 rot=.2		
Preliminary	Clustered	78
Preliminary	Power Law	110
Preliminary	Uniform	117

Returned Logic Spin Test Factor rng .5-1.5m , v=.8 rot=.3				
Preliminary	Clustered	76	78	
Preliminary	Power Law	118		
Preliminary	Uniform	162	135	141

Returned Logic Spin Test Factor rng .5-1.5m , v=.8 rot=.3				
Preliminary	Clustered	97	85	72
Preliminary	Power Law	135	128	131
Preliminary	Uniform	157	152	157

Returned Logic Spin Test Factor rng .5-1.5m , v=.8 rot=.3 limit parameter		
Preliminary	Clustered	
Preliminary	Power Law	
Preliminary	Uniform	
Final	Clustered	134
Final	Power Law	195

Conclusion

The twister code developed by DustySwarm robotics team demonstrated to be superior to the default iAnt searching algorithm. The team was proud of the work done. However, there is more room for improvement.

Future work

The team plans to continue their work with the NASA Swarmathon competition and hope to develop and even better search algorithm for nex year. Next year the team hopes to incorporate new fresh ideas from incoming team members

References

[1] Donny K. Sutantyo, Serge Kernbach, Valentin A. Nepomnyashchikh and Paul Levi: Multi-Robot Searching Algorithm Using Levy Flight and Artificial Potential Field, Mountain View, CA: University of Stuttgart, Universitatstrasse 38, Stuttgart, Germany , August 2011

[2] W. E. Shots, The Linux Commmand Line, 2nd ed.,San Francisco, CA: Creative Commons, 2013.

[3] J. M. O'Kane, A Gentle Introduction to ROS, Columbia, SC: University of South Carolina, 2014.