

Comparing the Performance of Depth First Search and Breadth First Search for a Robot with Energy Constraints Exploring an Unknown Environment

Shyam Rajendren

N01185608

1 Problem Statement and Motivation

This paper strives to compare different methods of searching an unknown environment, namely Depth First Search (DFS) and Breadth First Search (BFS). The robot which will be searching the environment will have an energy constraint which requires it to return to a charging station to recharge its battery before it continues its exploration. Comparing and creating new algorithms to search an unknown environment is not new but hardly has it been done with a robot's battery capacity kept in mind. However, in a real world environment, a robot would not have an infinite energy supply so it is prudent for us to consider this constraint when comparing algorithms. This paper looks to compare the two widely know methods (DFS and BFS) to help start as a base of comparison for other algorithms. Researching this problem would be of great benefit in many fields, for example, they could help robots in deep sea floor mapping as well as help robots searching for missing people after a disaster.

2 Introduction and Literature Review

There has been some research into this topic by other authors. Strimel and Veloso used boustrophedon decomposition to cover the environment [1]. Using this method, the robot returns to the charging station when its energy level is too low to continue the coverage. Mishra *et al.* designed a coverage planning algorithm consisting of multiple robots [2]. In order to continuously cover the environment, the robots are divided into two groups, workers and helpers. When a worker needs to recharge its battery, an associated helper will continue the worker's coverage. Neither of these studies formally analyze their algorithms and instead only prove that their methods correctly cover every point in the environment. This paper hopes to set a precedent of comparing algorithms

based on the same parameters. This will help decide which algorithms are superior so that they can be used practically.

3 Methodology

The input for this project will be a grid environment with a random distribution of obstacles which will be traversed using DFS and BFS. The environment can be traversed in all four cardinal directions and each node in the tree has a maximum of two parents and two children. Depth First Search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root of the tree and traverses as far as possible down the tree's branch before backtracking. The time complexity for this algorithm is $O(b^d)$ where b is the branching factor, in this case two, and d is the depth of the tree. Breath First Search (BFS) is similar to DFS except it traverses all the nodes at the present depth of the tree before going to the next depth level. The time complexity for this algorithm is also $O(b^d)$.

4 Evaluation

A simulation approach will be taken to compare the DFS and BFS methods. Three environments will be created: a 100x100 environment, a 200x200 environment, and a 300x300 environment. Each environment will have a random obstacle distribution where the obstacle percentage can be either 10%, 20%, or 30%. The environments will be created using Graph Stream, an open source Java graph handling package available online [3]. The two unique data sets that will be collected are time taken to search the environment, and the length of the path taken by the robot to search the environment.

5 Results

The data collected will be visually represented using a line graph for each environment and obstacle percentage. This will help in the comparison of the two algorithms and help us determine which one performs better.

6 Description of List of Required Resources

The open source Java graph handling package, Graph Stream, will be required for this project [3].

7 Completion Timeline

Week 6: Come up with a project idea.

Week 7: Finish brainstorming the specifics of the project.

Week 8: Submit the Research Project Plan.

Week 9: Start creating the environment that will compare the DFS and BFS algorithms.

Week 10-12: Continue to work on creating the environment that will compare the DFS and BFS algorithms.

Week 13: Complete work on the environment that will compare the DFS and BFS algorithms.

Week 14: Collect data from the various simulation trials.

Week 15: Tabulate and compare collected data.

Week 16: Compile results and submit final project report..

References

[1] Grant P. Strimel and Manuela M. Veloso. 2014. Coverage planning with finite resources. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), 2950–2956

[2]Saurabh Mishra, Samuel Rodríguez, Marco Morales, and Nancy M. Amato. 2016. Battery-constrained coverage. In *CASE*. IEEE, 695–700.

[3]GraphStream - A Dynamic Graph Library.[online] Available at: <https://graphstream-project.org/>