Visualizing and Designing Multi Agent Search Algorithms

David Bushta, Chris Till
Rutgers University, Camden

Introduction

The purpose of our research is to study distributed search-and-escape algorithms. These algorithms involve mobile agents (or bots) searching in geometric domains, such as a closed disk or a convex polygon. By working together and communicating with one another, the mobile agents search for an exit hidden on the perimeter. The goal of our research is to create and study exit strategies that terminate as quickly as possible.

Main Problem(Proposition)

The purpose of our research is to propose an algorithm for two distinguished and a helper bot to find an unknown exit on a disk.

mypics/temp.JPG

Figure 1. Temp 1

Definitions

An **exit** is a point unknown to the agents, that is located on an perimeter on the domain. The agents must find the exit for the algorithm to finish. A **priority** agent is one that must reach the exit for the algorithm to terminate. A **helper** agent is one that simply assists the priority agent(s) in finding the exit. The exit is **found** if an agent's coordinates match the exit location. The algorithm **terminates** if a specified subset of agents shares a position with the exit.

Main Topic

Figure 2. Temp 2

Figure 3. Temp 3

Algorithm Description

In our algorithm, we use two **priority** agents and one **helper** agent. The termination condition in our algorithm is reached when either one of the two priority agents reaches the exit. We send one priority and one helper to some angle α on the perimeter of the shape, in the third quadrant. The priority agent travels counter-clockwise, and the helper travels clockwise. The other priority agent travels to 0 radians and travels counter-clockwise.

Future Work

Later on we will study further distributed algorithms of search and escape, such as lines or triangles.

Our Conjectures and Analysis

This algorithm does not have the best lower bound.	*True*
If the game is linear with n squares then its graph is a path with n vertices.	*True*
If a game is non-linear its graph must have a vertex degree 2 or more.	*True*
If a game is non-linear its graph must have at least one vertex with degree three.	*False*
There is a non-linear game whose graph is almost a path.	*True*
Any game with 3 or less squares is linear.	*True*

Conclusions

This algorithm has an upper bound of blank, being faster than an algorithm of one distinguished and two helpers with an upper bound of 3.83.

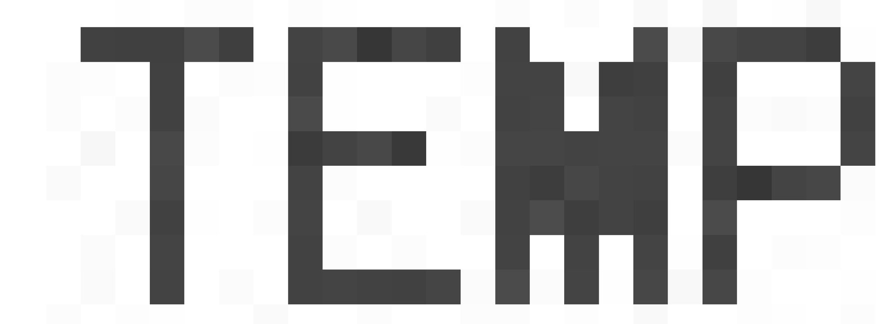

Figure 4. Temp 4

Acknowledgements

This research is supported by the National Science Foundation under grant # CCf-AF 1813940 (RUI: Search, Evacuation and Reconfiguration with Coordinated Mobile Agents).

References

[1] Jurek Czyzowicz, et. al., Evacuating Robots via Unknown Exit in a Disk, Springer-Verlag Berlin Heidelberg; 2014.
[2] Jurek Czyzowicz, et. al., Fun with Algorithms, arXiv:1804.06011v1 [cs.MA] 17 Apr 2018.

Figure 5. Temp 5