A Genetic Algorithms Approach to Learning Communication and Coordination in Simulated Robots

Chris Sotzing
Computer Science Department
Colby College
5830 Mayflower Hill
Waterville, ME 04901
`ccsotzin@colby.edu`

Abstract. This project is motivated by an existing robot system for mapping unknown environments and attempts to improve its effectiveness through the use of genetic algorithms. Using the Webots simulator, the mapping system is created using simulated Khepera robots and a simulated environment. The robots are controlled by a supervisor agent that makes the high-level decisions about tasks for individual robots to complete to accomplish the mapping effort. This research investigates the ability of adding a GA learning component to the supervisor to improve its ability to coordinate the robotic agents.

1 Introduction

The goal of our research is to develop an efficient controller for multi-robot mapping of unknown environments. We will take an existing mapping algorithm for multiple robots and add a genetic algorithm (GA) learning component in order to improve upon the algorithm and improve the efficiency of the mapping process.

2 Background

Systems for robot exploration and mapping of unknown environments are becoming more and more common as science and technology push us to places that humans cannot safely go. Robots are the perfect way to explore harsh environments like war zones, hazardous waste spills, and most recently, Mars. There is much research in this field using single-robot strategies and recently this has been widened to include multiple robots [1]. In order for a robot mapping system to be efficient, the exploration strategy must be so as well. In particular, robots need to determine what areas are most worthwhile to explore and in the case of multiple robots, good coordination with each other is essential. Using multiple robots is also beneficial because in theory the more robots, the less time necessary to map the environment. A multiple robot system is also a favored approach when a situation makes losing a robot a possibility.

The system can still perform its mapping function despite the loss of a search member.

This notion was focused on by Simmons et al.[1] in their study on multi-robot mapping. In their paper, Simmons et al. used multiple robots to map an unknown environment. The robots communicated via a central controller that would combine all the information sent from the robots into a central map; the controller also made decisions about where each robot should go next. Robots made “bids” to the central controller on potential areas to explore. These areas focused on “frontier cells”, or areas that were unexplored but that bordered an explored area. Each bid was given a weight determined by a number of factors, including the robot’s proximity to the bid location and the amount of data expected to be gained from exploring the location; the controller evaluated each bid based on this weight. By using this system, the robots were able to map out the unknown area in a coordinated fashion.

3 Materials & Methods

In order to create the robot mapping system, Webots, a robot simulation program developed by Cyberbotics[2], is used in conjunction with simulated Khepera robots developed by K-Team[3]. The decision to use simulated robots vs. actual robots was made so as to eliminate as much “noise” as possible and focus on the central controller algorithm as well as to allow for large numbers of test runs without the necessity of human supervision.

The environment for the robots to explore was created in Webots and was essentially a large grid set up with the possibility of each square either having a block in it or none at all. This, in addition to the fact that robots were limited to moving one square at a time, helps simplify the problem and also helps to lessen the possibility of noise in the data. The map was represented by a 2-dimensional array where each square could be one of three types: unknown, clear or obstacle. See Figure 1 for a screen shot of the robot in the grid environment.

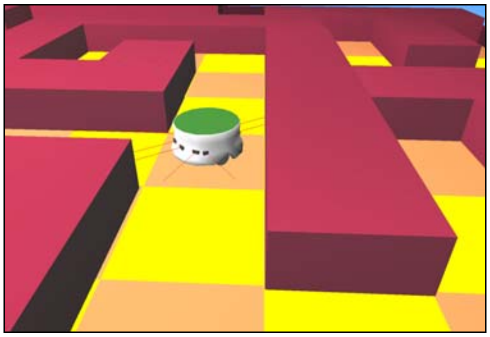

Figure 1: The simulated Khepera robot in the grid world

Each robot is relatively simple, with two wheels for locomotion and 8 infrared proximity sensors, 6 on the front and 2 on the back. With this set up the robot can detect one cell out, objects in front, behind, to the left, to the right, and both forward diagonal directions.

As in the Simmons et al. paper, the robots were coordinated through a central controller. The bidding structure is similar to the version of the one used in the Simmons et al. paper except that it is simplified because of the simplified sensing. Bids are made on "frontier" cells which are squares in the map that are unexplored, but border explored areas. The controller in this study differs from that of Simmons et al. in that the decision-making process is subject to learning.

4 Genetic Algorithms System Design

To hasten the development of the system, Genesis [4] was used as the GA component. The GA starts with a population of randomly generated strings. Each string is evaluated by calling the Webots simulator.

In this multi-robot mapping system, the GA string contains information telling the controller how to react to the bids sent in by the robots. It is a fixed-length string containing rules that describe what the behavior should be for any situation. Each GA string consists of 12 bits. The first 6 bits encode for two variables, a & b, that control the weight of both the cost and the information gain of each bid. Each of the two variables is represented by a 3-bit section and can code for a weight of 0-7 used to construct a bid using the following formula: $bid = a * informationGain + b * cost$ where information gained is a number from 1 to 5 cells mapped from the target frontier cell and cost is the Manhattan distance of traveling to that cell. The next 3 bits represent how close simultaneous targets of different robots can be, in terms of the number of squares. The range is 1-8 squares using Manhattan distances; the effect of this rule is to eliminate some bids from consideration. The third rule is represented by 1 bit and contains the information on whether robots should favor targets that are near or distant to their current location; the effect of this rule is to mitigate ties. The fourth rule of the string contains information about how far the robot should move towards the target before it must stop its current task and re-bid; 1, 2 or 3 steps towards the target and then re-bid, or go all the way.

For each string evaluation, the group of robots starts off in the same place in the map and immediately makes bids and sends their local map to the supervisor. The supervisor then rates those bids and then makes decisions based on the GA string as to what each robot should do. Until a robot has an accepted bid, it remains idle which is a setback that the system should learn to overcome. The fitness of the run is the number of cells mapped during the time allotted. In order to evolve a mapping strategy that isn't specific to one map, more than one map is tested. In addition, the coordination abilities of the system will be evaluated by using different numbers of robots. In this system, the code for the individual robots is essentially very simple. There is no decision-making process that takes place. Rather, data is sent to the supervisor that contains the local map created by the robot and a number of bids for where to go next. The supervisor first takes the map and combines it with the global map. Then after analyzing the bids, sends robots instructions on their next task, after which the process begins again until the number of time steps is up.

5 Expected Results

The goal of this research is to compare a non-learning controller algorithm for robot mapping with a learning version. We expect that the learning will enable the system to be more efficient. In particular the ability of the controller to interrupt robots en route to a task may be a strength of this approach. Furthermore, the ability of the controller to assign different weights to the information gain and costs may give the system additional flexibility.

6 Future Work

It would be extremely interesting to expand out of the simulated world and see how the mapping works with real robots in a real environment. In order for this to be useful, another part of the project would have to be updated which is the limitations of the grid environment. The simplicity provided by this type of world lends itself well to the simulation but not as well to real life situations. To remove this limitation would undoubtedly lead to a far more sophisticated and better mapping system.

Acknowledgements

I would like to thank Katelyn Mann and my advisor Clare Bates Congdon for all their help with this project.

References

1. Reid Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, Håkan Younes. Coordination for Multi-Robot Exploration and Mapping. *American Association for Artificial Intelligence.* 2000.
2. Cyberbotics: www.cyberbotics.com
3. K-Team: www.k-team.com
4. J.J. Grefenstette. A user's guide to GENESIS. Technical report, Navy Center for Applied research in AI, Washington, DC, 1987. Source code updated 1990.