An Approach To Genetic Algorithm Path Planning Used In RobocupRescue

Mona Behnam Moradi And Eslam Nazemi

Department Of Computer Engineering
Shahid Beheshti University (SBU) , Tehran , Iran

nazemi@cc.sbu.ac.ir behnam_m@ecef.sbu.ac.ir
http://www.sbcee.net

Abstract. SBUCRescue, as an another group to develop rescue simulation project, have tried to put Another step forward in this field. In our simulated model, we will encounter with a real-time, multi-agent System. There, six kinds of agent cooperate with each other to come the disaster off like reality, using some methods of learning and artificial intelligence. This paper describes our strategies used in the RoboCup Rescue Simulation contest. We first introduce our policies in the communication system which reduces the loss of cycles for unvalued messages. Also some features of Genetic Algorithms have been used to find the best possible way in the disaster space.

1 Introduction

After doing some researches in various methods, we found out that two dominant methods exist .One is "Agent Development Kit" [3] which has classified the project to some specific objects, and the other is the kernel word model. In this project we have concentrated our attention on the latter.

There are two major phase in the rescue system which we are going to discuss here. Gathering information about the disaster space, saving these information and communication problems are discussed In section 2 as "Information System". Choosing the target and path findings are mentioned in section 3 as "Decision Making". Section 4 discusses future works and concludes this paper.

2 Information system

2-1 Communication

Agents operate independently with a limited communication, so if some of them become inactive the rest can continue to operate. These Limitations evolve that an ambulance team for example can not directly communicate with fire brigade and the message should pass ambulance center and fire station as well. The network diagram of rescue agent is shown below.

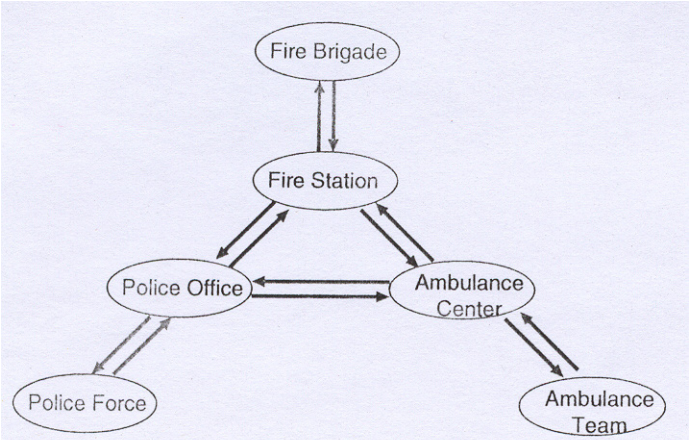

Fig. 1. Network Diagram of Rescue agent

It takes a long time for the message to go through this way, therefore optimizing the system to transmit more necessary messages seems inevitable. We can evaluate these messages by marking them as High/Low priority. In emergency cases mentioned below we send a High/Low priority message to a specific agent. So if this agent receives more than one message it only responds to those with high priority. If there be no message with high priority the system puts its consideration on those with low priority. At the end, if the system receives message with no priority it simply discard that message.

From	To	Priority	Situation
AmbulanceTeam	Firebrigade	High	When the refuge or ambulance center is burning.
AmbulanceTeam	Police force	High	When the target is a building with fieryness >3
AmbulanceTeam	Police force	Low	When the target is a building with Fieryness=2
Firebrigade	AmbulanceTeam	High	When a firebrigade is injured
Firebrigade	AmbulanceTeam	Low	When a Police force or AmbulanceTeam is injured
Firebrigade	Police force	High	When there are more than 5 buildings in the target's neighborhood , the fieryness of the target is more than 3 or the firebrigade is located in a deadend.

Firebrigade	Police force	Low	When there are 3 to 5 buildings in the target's neighborhood , the fieryness of the target is 2.
Police force	AmbulanceTeam	High	When a Police force is injured
Police force	AmbulanceTeam	Low	When a firebrigade or an AmbulanceTeam is injured
Police force	Firebrigade	High	When police office is burning.

2-2 Memory

Gathered information needs to be saved in a memory for further usage. In order to provide this, We Have used kernel's class, **Object Pool**, as memory in the system.

3 Decision Making

The first step in decision making is to find the target. As we know there might be more than one target for a specific agent. KA_SENSE body consists of Id, Time, Self and Map. The map contains information of objects within the radius of 10m and all fires [2]. Thus To optimize the work of fire brigades some policies have been used.

At first each fire brigade checks for fires within the radius of 30m and extinguishes them if there be any. If there were no fires there, to find the best target we considered some parameters such as the fieryness of the building, the number of is neighbors, the type of the building (to see if it is a station or refuge) and the distance to the building. We managed a function of these parameters to find the best target having the highest value.

Path finding for reaching the target is the most important part of the job. In order to find the Best way we examined some search algorithms .We found out that classic algorithms like Dijkstra's Shortest path algorithm can not work well. Because they need a lot of time and accurate information, Two things that our system is in lack of.

Then we reviewed kernel's model of moving for civilians finding out that in each step they try to Approach to the nearest possible target but because of shortage of time if the steps to reach the target Becomes more than fifty they start to move randomly, and might not reach the target. So a way to teach them how to treat, is by learning methods.

Another way of path finding which we are working on, is **Genetic Algorithm** [5]. The different paths from agent to target are our chromosomes. Some characteristics of these paths such as the length of the way, the number of buildings and nodes in the way, the brokenness of road etc are the genes. We are trying to develop these chromosomes by using a fitness function and produce better ones, so we can produce our best solution.

4 Conclusion and Future Work

In this project we are facing a Real-Time Multi-agent system in which valid and valuable information is one of the most important things to be considered. To gain this we managed some policies to make the communication system work better. The better this communication system is the more valuable the gathered information will be. So what we are working on now is to make the best possible communication system.

References

1. Robocup official web page,
   http://www.robocup.org

2. "Robocup-Rescue Simulor Manual" and "How to develop a Robocup Rescue Agent" Robocup Rescue official web page,
   http://robomec.cs.kobe-u.ac.jp/robocup-rescue

3. M.Bowing, Robocup Rescue, Agent Development Kit, Version 0.4, Computer Science Department, Carnegie Mellon University, Dec. 2000.

4. T.Morimoto, K.Kono, I.Takeuchi, "YabAI the first Rescue SimulationLeague Champion",
   http://ne.cs.uec.ac.jp/~morimoto

5. Alexander Benjamin Doyle ,"Algorithms and Computational Techniques for Robot Path Planning"