

RoboCupRescue 2009 – Rescue Simulation League

Team Description

SBCe_Saviour (Iran)

Eslam Nazemi, Ashkan Radmand
Mohammad Goodarzi, Reza Emami

Department of Electrical and Computer Engineering
Shahid Beheshti University
Tehran, Iran

nazemi@sbu.ac.ir
{radmand, iman.goodarzi, reza.emamii}@gmail.com
<http://www.sbcee.net>

Abstract. This paper describes the implementation of our rescue agents in the RoboCup Rescue 2009 competition. Recently, simulators have improved to the point that the prediction of disasters and making the best decisions has become very difficult. All the participating teams have come to the conclusion that in order to improvise, it is better to implement AI algorithms. In our team the agents use decision making algorithms to select targets and the A* algorithm to find the optimum path to their selected targets. The implementation of our ambulance and fire brigade teams are based on state machines and police forces decide based on Genetic Algorithm and Fuzzy Logic. We use messages in a distributed message management system containing messaging channels that supports center-less environments.

1. Introduction

Our main goal for participating in RoboCup Rescue Simulation [1] competitions is to reach a level of capability for disaster management, and implementing this knowledge in the real world.

Considering the challenges of predicting natural disasters, and its long term impacts on human experience, the use of the new technology such as artificial intelligence can be a beneficial tool, mostly because of its decision making abilities. As a result, we worked on some AI algorithms [6] for achieving better results in multi-agent rescue simulation competitions [2].

This paper consists of several sections. In the following section, we will provide the description of each agent and its functions and the algorithms for path planning and region managing. Afterwards, we will provide some detailed description of World Modeling and Communication System.

2. Agent skills and action selection

In our team, agents decide individually, mainly because of the two main problems that centralized decision making have. The first problem is about delay in sending the instructions from the center to the agents and another problem is the lack of center in some special designed maps. In this case the performance of the whole team will be decreased considerably. Although we do not use the center as the center of decision making but it can be used as a database that can gather the information from its agents and send them to the others. In some cases, we do not have any centre in the map, so we have another solution which will be discussed in section 4.

2.1. Ambulance Agent

The implementation of our ambulance team is based on state machines and automata theory [7]. In an ideal situation, the cyclic process of rescuing and transferring a victim to the refuge involves the following six states in the Fig 1. It is obvious that surveying these steps is dependent on the circumstances that can occur in every step.

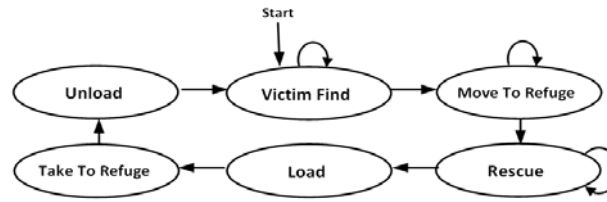


Fig. 1. The ambulance team's state diagram

The first step in the rescuing process is choosing a victim. Considering the time limitation and the cost for rescuing, victim selection must be done in a way to result in the survival of the maximum number of victims. The SBCe strategy that is being used now is as follows: The victims are inserted in lists depending on the number of the ambulances which they need to be rescued before die. Then they are assigned priority values by the parameters containing estimated hp, damage, type of victim, buriedness, estimated time to death and victim's position (for calculating distance of the victim to the refuge and the ambulance agents). This process is shown in Fig 2.

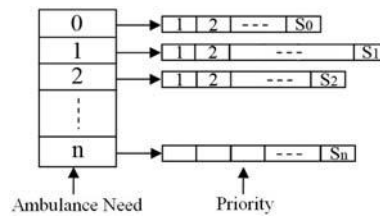


Fig. 2. The victims' priority matrix

Each ambulance agent selects a victim as a target in the list which has the least ambulance need and the most priority value. To prevent interference of other ambulance agents, every ambulance agent considers the target of other ambulances that it has informed by communicating with them.

2.2. Fire Brigade Agents

Generally our fire brigade activity is divided into two intervals. The switch time between these two parts is calculated according to the disaster space's conditions. In the first interval in which the fire is not spread, the agents try to extinguish the nearest fire sites from themselves. The term "fire site" is used to define a set of fiery buildings that are in neighborhood. Fire brigades may work individually at this part. This method controls the fire more efficient as we experienced. If any fire site exists after the first interval, the fire brigades work as a group. Choosing the appropriate fire site is according to some factors including position of the fire sites in the map, number of buried civilians in the fire site's neighborhood, total area of burning buildings and the distance between the fire site centre and fire brigades average position.

The fire site with the highest priority will be chosen as the target. Afterward, the chosen fire site's buildings' priority will be set considering building's area, temperature, fieriness, number of unburned neighbor buildings and number of victims inside.

In this step the agent selects the position to stay and extinguish the burning building with the highest priority. The main factor in choosing this place is that we can finally extinguish more burning buildings from there. The agents also try not to block the roads and make traffic in that region.

If an extinguished building starts to burn again, considering the expansion of the fire, some of the fire brigade agents return to that zone and re-extinguish it. Implementation of the fire brigade agent in SBCe team is based on four states as you can see in Fig 3.

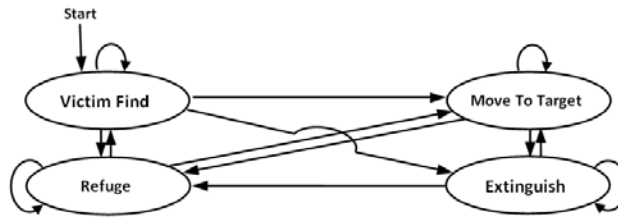


Fig. 3. The fire brigade team's state diagram

2.1. Police Force Agents

After the disaster, some city roads closed by debris. These blocked roads divide the city roads graph into isolated city parts. The major task of police force agents is to

connect the isolated parts of the city that causes the easier transportation of other types of agents. To achieve this goal, the best blocked roads should be chosen to clear. This selection is based on some issues such as number of refuges, stuck agents, burning buildings and victim civilians existing in the mentioned parts.

Therefore, the police force agents should decide which two city parts are more important to get connected first. A linear combination of these factors is essential to determine a priority for each road. To optimize the performance, they should assign good weights to conditions. To achieve this goal, we utilize GA [8,9] approach in some training maps with specific characteristics. We have used some characteristics that are more important in classifying the maps including number of burning buildings, number of victim civilians, number of blocked roads and number of buried agents.

Each of these maps has different values of above characteristics. In fact, training maps are representatives of all maps. Applying the mentioned GA approach to all training maps gives us an array of optimum weights for each map. It is required to have a method to combine these arrays and have a solution for each new map. We have used a method of decision combination using fuzzy logic [10,11].

In this method, each characteristic of map will create a fuzzy set that consists of three membership functions: *Low*, *Medium* and *High*. Fig 4 demonstrates the membership functions of maps characteristics.

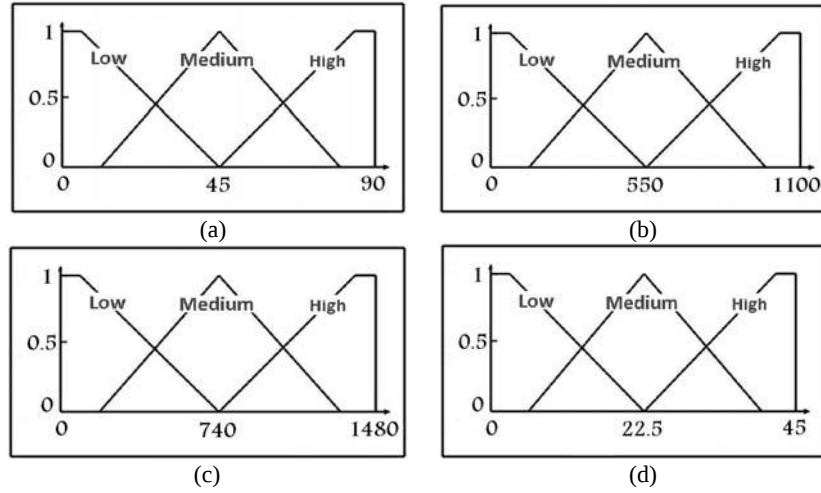


Fig. 4. "(a) Victim Civilians number (b) Burning Buildings number (c) Blocked Roads number (d) Buried Agents number" Membership Functions.

Using defined fuzzy sets, fuzzy if-then rules will be created. These rules are in Mamdani's proposed form [12]. For example, one of the rules is:

IF "number of victims" is High AND "number of burning buildings" is Low AND "number of blocked roads" is Low AND "number of buried agents" is Medium THEN map IS training map #1.

Having these rules, the training part is completed. Given a new map, values of map characteristics will be checked against all fuzzy If-Then rules. In defuzzification step, to achieve final weights for the given map, we should combine "then part" of all rules considering their weights. "Weighted Average" method is used in our proposed method for defuzzification step.

10 test maps were chosen from Robocup2008 China rescue simulation competition maps. The comparison of total scores gained using proposed method and the results gained before the training is shown in Table1.

Table 1. The comparison of results gained before and after training.

Map Name	Old score	New score	Map Name	Old score	New score
Map #01	65.356	66.643	Map #06	90.012	91.103
Map #02	63.165	67.265	Map #07	93.464	95.689
Map #03	76.798	77.013	Map #08	88.645	89.465
Map #04	85.679	88.625	Map #09	87.465	91.465
Map #05	72.946	74.346	Map #10	85.856	87.695

2.2. Path Planning

Our path planner uses the Logical road instead of real road and presumes the city as a graph with logical nodes as its vertices and the logical roads as its edges. **Logical node** is a node with more than two connected roads. And **Logical road** is a set of connected roads and nodes which none of nodes are logical nodes except the first and the last one as it shown in Fig 5.

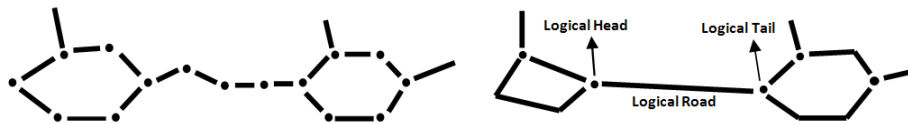


Fig. 5. The Logical Road and Logical Node

In order to find the best path, we use A* algorithm [6]. In this algorithm the cost of each edge (logical road) is equal with the total cost of existing roads in that logical road. Cost of the buildings and the nodes presumed to be zero. Cost of an open road is equal with its length and is infinite if it is blocked for the non-Police agents. A road is blocked whenever there is a blockade on the road or an agent is standing on it. In the first case, cost of the road will be infinite unless the agent receives a "clear report" about that road or senses that the road is open. In the second case, the agent presumes that the road is blocked and its cost is infinite until 5 cycles and then presumes it as an open road and sets its cost with its length.

2.3. Region Manager

One of the most important issues that our team was encountered was how to spread the agents all over the city to explore unseen objects. By having no partition, agents will concentrate in a particular part of the city, which results in traffic and an obstacle for the other agents. So it seems reasonable to partition the city. The method we have been used in our team is explained here.

BFS Division Method: In this method, we partition the city according to the object which is needed, such as building, node or road. We specify the center of the city by the coordinates of all the nodes, then we choose the closest node to center as the central node. Then we use this node as the root for the BFS algorithm and continue until the number of objects (depends on what type of objects we have chosen) in this traverse for a region is less than a specified number. After a region is formed, we choose the closest node which has not been seen yet to that central node and repeat the above operation until all the city is divided into regions. The advantage of this method is that there is no central attraction and the number of the objects in all regions is the same. This method seems to be optimized. The result of partitioning the city into ten regions on Kobe map is shown in Fig 6.

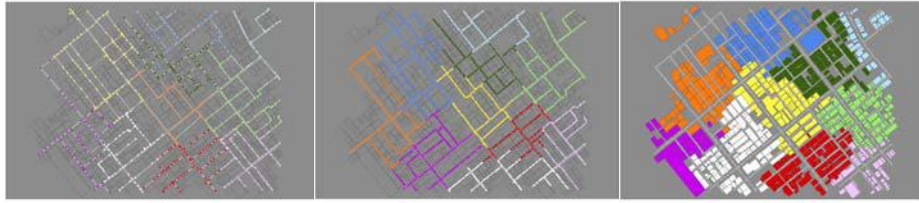


Fig. 6. Partitioning based on Road, Node and Building class

3. World modeling

For the world modeling in our team, we have used the SaviourAPI (which is an improved YabAPI [4]) modeling system as our base. In this system every agent maintains the data from all the objects in the world in a structure called Disaster Space. In this structure, the objects are classified in sets. For instance, we have sets of Buildings, Humanoids, Agents, etc. Each object has some properties which will be updated when the agent observes that or receives its information through the communication system.

4. Agent coordination and communication

For agents to be updated on the disaster space and also their coordination in the rescue operation they need to transfer information. The agents are limited in sending and receiving messages, so we have tried to maximize the use of our messaging space by compressing the messages. Each agent fits all the information it wants to broadcast in one message per cycle and sends it.

We use channels for inter-agent communication. Having centers, agents communicate through four channels. In this method, all agents of one type send their messages to their center using a special channel. Therefore, three channels get occupied. Every center checks the messages which it has received from its agents and deletes the repeated messages, after that, it constructs a new message and sends it on the fourth channel aka the global channel. The global channel is just heard by the agents and only the centers have the privilege of sending information on it. Therefore, every agent listens to three messages on the global channel; also every center receives 'n' messages from its agents in each cycle. Because the centers have no role in the decision making process, they don't listen to the global channel. By implementing this method, each message sent by an agent is received by the other agents after a two cycle delay (See Fig 7).

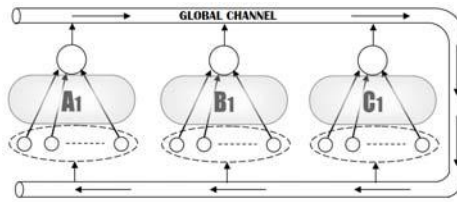


Fig. 7. Channel based communication with all of the centers

If one of the agent types doesn't have a center, by constructing a binary tree of the agents, information transfer will be done and an agent chosen as the root of the tree plays the role of a center (See Fig 8).

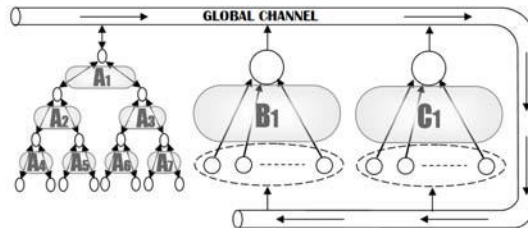


Fig. 8. Channel based communication without one of the centers

As seen above, if only one team of agents will have no center, and in worst case scenario that team has 15 agents, we can build the messaging system using 10 channels. But if two or more teams have no centers, in worst case scenario we will have 15 fire brigade agents, 15 police agents and 8 ambulances. We will need more than

10 channels for full agent connectivity. In this case, not counting the global channel, the remaining 9 channels are divided between the three teams equally (See Fig 9).

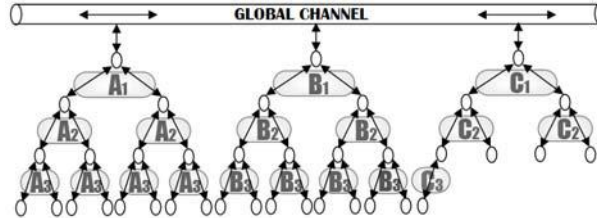


Fig. 9. Channel based communication without any center

In some maps that there is not any radio communication, we can just use the "say" channel. Therefore, we start broadcasting the information by repeating any message which has been heard during recent cycles. In fact, each message has a validity time and will be repeated by the agents until it reaches its expiration time.

To avoid exceeding message size limitation we consider some points; first, just messages with higher importance (e.g. "block message", "victim message" and "fire message") will be sent; second, reporting the observed objects is more important than repeating the heard messages and finally, recent heard messages have more priority over old heard messages to be broadcasted.

5. References

1. RoboCup Rescue Official Website. <http://www.robocuprescue.org>
2. Kitano, H. Tadokoro, S.: RoboCup Rescue: A Grand Challenge for Multi-agent and Intelligent systems, ICMAS-2000, (2000)
3. Morimoto, T.: How to Develop a RoboCupRescue Agent. Rescue Technical Committee.
4. Morimoto, T.: YabAPI: API to develop a RoboCupRescue Agent in Java.
5. SBCE_Saviour TDP and source code, Shahid Beheshti University, Iran (2008)
6. Russell, S. J., Norvig, P.: Artificial Intelligence A Modern Approach. Prentice Hall, Englewood Cliffs, New Jersey (1995)
7. Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata, languages and computation. Pearson education Inc.(2001)
8. Goldberg, David E.: Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA (1989)
9. Banzhaf, W., Nordin, P., Keller, R., Francone, F.: Genetic Programming - An Introduction. Morgan Kaufmann, San Francisco, CA (1998)
10. Kruse, R., Gebhardt, J., and Klawonn, F.: Foundations of Fuzzy Systems. John Wiley & Sons, New York (1994)
11. Dubois, D. and Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
12. Mamdami, E.H., Assilina, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, vol. 7(1), pp. 1--13 (1975)