RoboCup 2017 Rescue Simulation League Team Description A.T.F (Iran)

Borna Barazandeh, Danial Ghiasvand, Parsa Haghighat, Shahdad Khakpoor, , Reza Vazirinejad

Andisheh Mehr Institute, Ekhlas, St Daneshgah Square, North Sattari High Way, Tehran, Iran

{borna.barazandeh,danialgheiasvand2002,shahdadkh1353,parsahaghighat19991999 ,vaziri.79.reza}@gmail.com

Abstract

This paper describes the codes and solutions of A.T.F rescue simulation team which is going to take part in the RoboCup 2017. In this paper, we will introduce our codes, planning algorithm of agents, and cooperation between agents. In 2017 our focus was on improving efficiency and applying some changes to improve bandwidth utilization. This year we did a substantial innovation changes on messaging communication between police forces and ambulance team. We also focused on utilizing modules and more apprehensible message types which runs between agent. Our team developed a useful route searching algorithm, estimating the spread of the fire and each agent's clustering. Moreover, our team this year is using "K-means" clustering. "K-means" is one of the simplest algorithms that solves the clustering problem. "K-means" chooses a particular center for clustering. After that zones are created. We have partitioned the map with the Hungarian algorithm to assign agents to partitions. These methods granted proportional agents distribution around the map without passing long paths. In order to search, we have used of maximal covering location problem. We also apply remarkable changes on ambulance decision making to have better estimation in the case of death time and number of needed ambulances to rescue a civilian from the base code.

1 Introduction

A.T.F, began to take part in Iran-Open 2017. We have achieved the second place in Iran-Open 2017. This is our first time that we have joined in Iran-Open and RoboCup . We have tried to improve the performance and we have been working on problems and weaknesses of our code. First, we have made the priority between our agents to save more civilians in less time. We have created zones for each agent so figuring out the best way for saving get easier. In each section we are going to explain our codes. This year we have used some new methods for target selecting, time to death, calculating HP, calculating distance between agents and civilians, and also selecting the best damaged civilian to survive.

2 Modules

Clustering and path planning are two modules with the most changes, and target selector module was also slightly changed.
Since the main focus of our team was on cooperation between agents, we added some new methods besides a new clustering method known as Hungarian clustering.
The path planning module will leverage messaging system to have a better knowledge of the map's state, so agents can plan their own path more dynamically.

2.1 Clustering

We have used the ADF default K-Means algorithm. For having a better function, we also implemented the Hungarian algorithm. The Hungarian algorithm consists of the four steps below. The first two steps are executed once, while Steps 3 and 4 are repeated until an optimal assignment is found. The input of the algorithm is an n*n square matrix with some non-negative elements.

Step 1: Subtract row minima
For each row, find the lowest element and subtract it from each element in that row.

Step 2: Subtract column minima
Similarly, for each column, find the lowest element and subtract it from each element in that column.

Step 3: Cover all zeros with a minimum number of lines.

We have used Hungarian Algorithm to cluster the map and build a cooperation between all agents. This method is a "combinatorial Optimization Algorithm" that solves assignment problems. It will select a particular center and it will divide the map to some zones. (Fig. 1)

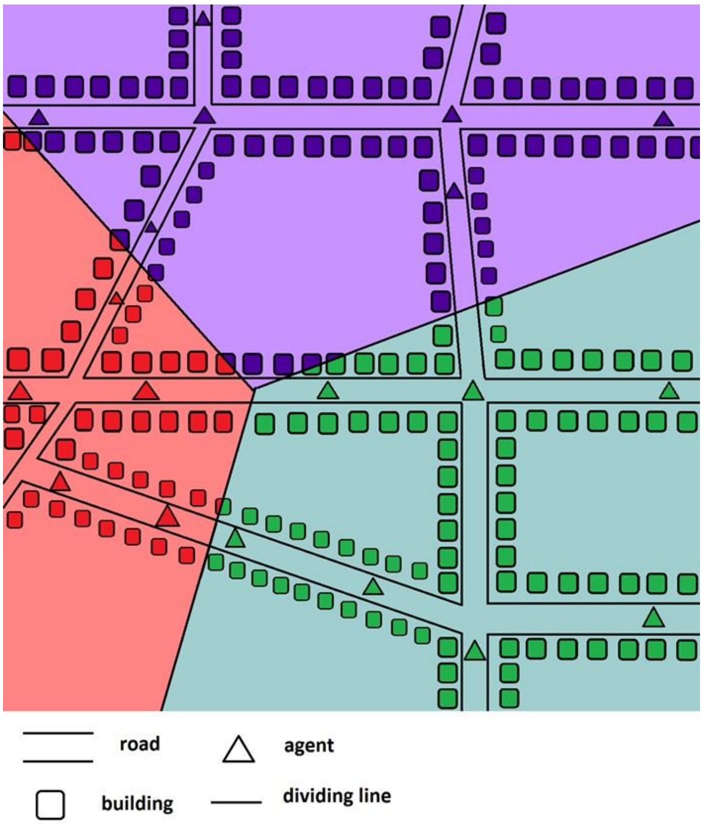

Fig.1 *Roads and buildings Hungarian clustering.*

2.2 Path Planning

Path planning traces an invisible line around a specified target. Path planning is also used to set priorities (for the nearest agent or civilians) between various entities. Sometimes, agent plans on a target and apply some changes to the scenario. Because of that, agents have been obligated to trace another path around a specified target but that strategy will make more time to execute and it's also possible to have massaging problem between itself and other agents. (Fig. 2)

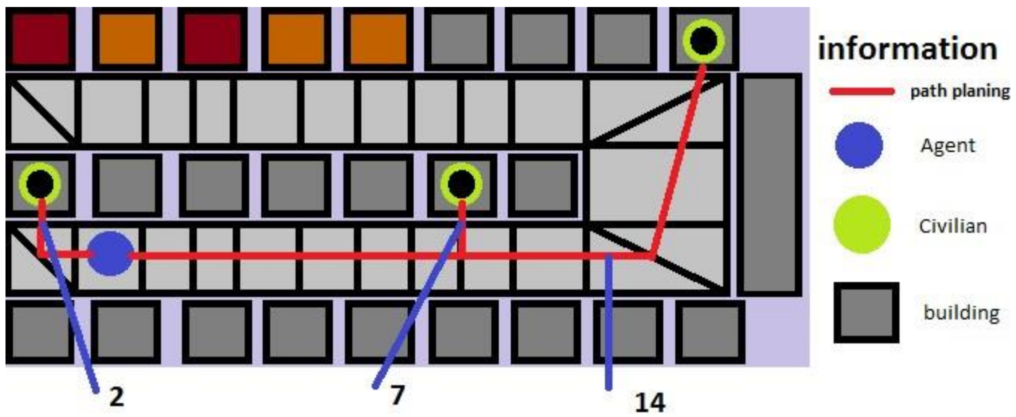

Fig.2 *The schema of path planning.*

2.3 Target Allocators

Target selection is one of the most sensitive and important parts of the decision making process of an agent. For that high importance, the main attention of our team was putting on related modules of target selection.

Since each agent has its own special task and operational sector, it needs its own special target selection algorithms. These algorithms can be principally divided into three sub-category based on the agent type that use them. be very useful on estimating the probability of buildings ignition and help the fire brigades to forecast the fire buildings and take actions even before the actual fire.

2.3.1 Agents

Each agent has its own plan for saving civilians but there is coordination between all agents for saving civilians in less time. We most worked on Fire Brigades. We cogitate that having a good code for Fire Brigades will be helpful. For having a good coordination between agents, each mission that agents do will be informed to all agents, so agents will use that information if needed.

2.3.1.1 Police Force

After Fire Brigades, the next helpful agent is Police. Time Consuming is important for all agent, and having time consuming will save civilians, turning off the fires, and healing each agents as fast as it can.

At the first of the game, some roads will be blocked. Ambulances and Fire Brigades will need to access all the roads for saving life of Civilians and turn off fires of the buildings. Clearing the blocks is time consuming. Police will open the blocks on the road. For time consuming police just open a part of blocks for agents to cross the road. The police work same as Fire brigade and also ambulance search. For example, police agents may be near each other, for that each agent must select a specific road and they mustn't follow each other in the whole game. (Fig. 3)

Sometimes there are situations that police agents need to select a road that was selected by another agent to access other roads. For example, we have 6 agents in a cross road (Fig.4). According to the mentioned points above each agent will select one of the roads connected to the cross road.
There are four roads connected to the cross road. After choosing a road by agents, there are two agents who haven't chose any roads because the roads were selected by other agents. Because of that the two agents will follow another agent until they approach to another cross road.(Fig.5) You may see our results in conclusion Fig.5 before and after using our method.

Fig.3 *Police force optimized clearing.*

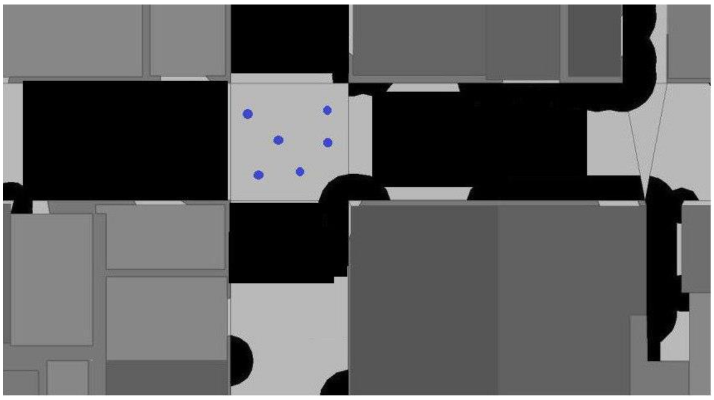

***Fig.4** six police in a cross road*

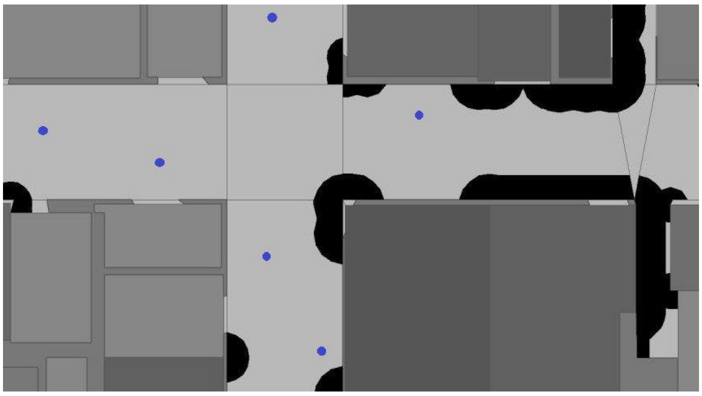

***Fig.5** After choosing roads by Police*

2.3.1.2 Fire Brigade

The year, we have focused on Fire Brigades. By using a well algorithm for fire brigades and having a well zone classification will help us getting the best result in the competition.

Fire search is designed to satisfy following requirements:

- Updating fire zones shapes and size.
- Finding new fire zones

Fire search is done according to our fire estimator and other parameters such as distance to fire zones and etc. After finding a new fire zones, fire searchers try to update the shape of the fire zone by visiting randomly, probable and far building near the new born fire zone. In some situations, fire brigades need help to control the fire in this case, fire brigades in the nearest zones to the fire will help to control the fire.

As all knows, having a good algorithm and cooperation between fire brigades and control the fire will help to get more point. Having a good cooperation depends on map, clustering priority, and spreading them in the map will be helpful. For getting a good result in cooperation we must have a good search. Mostly at the beginning of the game.
We found out that, when there is no fire, Fire Brigade should fill up buildings with water. By filling up the buildings, fire risk will decrease and if there is any fire near to the building it will help to prevent the fire.
The other important situation is searching for fire. If it found civilian or a blocked way, it will inform other agents, and if needed other agents will come there.
You may see our results in conclusion Fig.5 before and after using our method.

2.3.1.3 Ambulance Team

The important job of ambulance is saving civilians. Any team that wants to implement a good Ambulance Team agent should meet some prerequisites such as a reliable communication system and a good updated world model, an effective searching method, an optimized path finding method and finally reliable death time estimator.

Priorities help ambulance to save time and victims life.

- Each buried agent should be rescue first.
- Each civilian should be rescued by only one agent, excluded some critical situations.
- Rescuing less emergency civilians is a waste of time that is avoided if there are targets that are more important.

Based on these assumptions the total strategy of our ambulance team agent would be to:

1. Collect an updated list of injured agents and civilians that should be rescued.

2. Determine nearest reachable refuge for each injured civilian.

3. Remove the civilians that are impossible to be rescued from the mentioned list.

4. Assign priority to the remaining civilians in the list.

5. Simulate rescuing operation and change the priorities if needed.

6. Rescue civilians based on simulating result.

When the game starts ambulance should search the map for finding civilians and if needed inform Police and Fire Brigade for help. (Fig. 4)

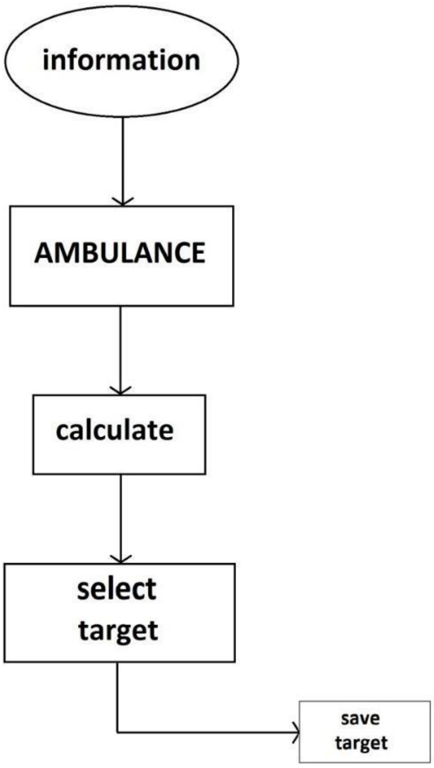

Fig.6 *Ambulance team decision making workflow.*

In some situations, polices and fire brigades will find civilians sooner than ambulances so if needed polices and fire brigades will send the information about HP, the place where the civilian is. When the ambulance achieves the information it will calculate and estimate the time to the civilian. If there is more than one civilian, it will give points to each of these subjects:

- Distance

When ambulance is choosing his target, one of the main thing that must be used, is distance. If the distance to the ambulance was low, it will get a higher score, if not it will get a lower score.

- Distance calculation

Each one will get scored from 1to 5. After that it will calculate the Sum. Last, the civilian that has the most score between all civilians will be saved by the ambulance. After reaching to the Civilian; Ambulance should take the civilian to the refuge.

3 Conclusion

One of the most effective approaches to improve the overall performance of the team is to leverage the co-operation of the agents. For this purpose, there is a need for custom and dynamic clustering algorithms and a path planning system which can set its state on-the-fly and change the final path if needed. By implementing the new algorithms and methods which have been mentioned in this paper, an overhaul in team's performance was witnessed.

You may see our results before using our method and after using it. We have made about 14 percent progress in our results. Our results may be temporary. It might change after fixing some bugs and problems. (Fig. 4)

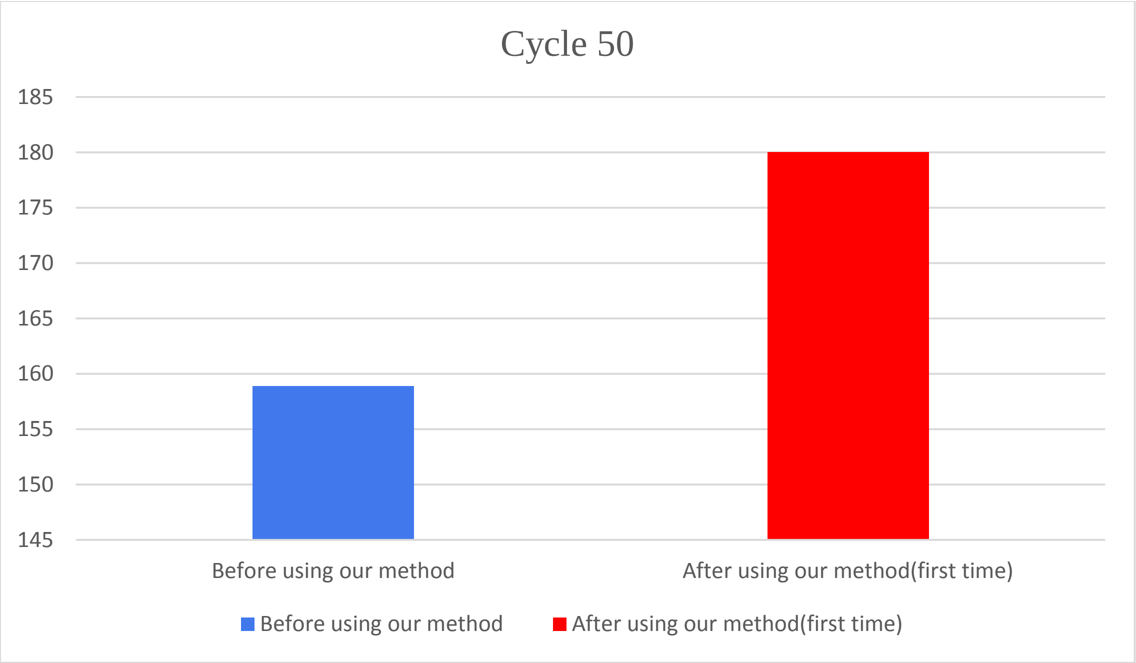

Fig.7 *Result Chart*

References

[1] Russell, Stuart J. – Norving, Peter. Artificial Intelligence: A Modern Approach. Third Edition. Upper Saddle River: Prentince Hall. 2010

[2] M. A khavan Khaleghi, M., Sarjamee, E., Foroutannezhad, A.:

[3] POSEIDON Team Description IranOpen 2012

[4] ESKILAS Team Description IranOpen 2012

[5] http://www.hungarianalgorithm.com/ examplehungarianalgorithm.php